چکیده

نام خانوادگی: شریفی **نام:** لیلا
عنوان پایان‌نامه: یافتن کوتاه‌ترین مسیر در شبکه‌های با پارامترهای فازی
استاد راهنما: دکتر منصور سراج **استاد مشاور:** دکتر هادی بصیرزاده
درجه تحصیلی: کارشناسی ارشد **رشته:** ریاضی کاربردی **گرایش:** تحقیق در عملیات **محل تحصیل:** دانشگاه شهید چمران اهواز **دانشکده:** علوم ریاضی و کامپیوتر
تاریخ فارغ‌التحصیلی:۱۳۸۹/۱۱/۳
تعداد صفحات:۹۸
واژه‌های کلیدی: مسائل کوتاه‌ترین مسیر،الگوریتم فورد-مور-بلمن، الگوریتم ین، برنامه‌ریزی ریاضی فازی،اعداد فازی، رتبه‌بندی اعداد فازی

چکیده: مسئله‌ی کوتاه‌ترین مسیر با هزینه‌های غیر قطعی یکی از مهم‌ترین مسائل مورد مطالعه در حوزه‌ی مجموعه‌های فازی است. الگوریتم‌های زیادی در مقالات معرفی شدند که بسیاری از آن ها هزینه‌ها را بدون پیدا کردن مسیر مربوطه، می‌یابند. در واقع آن‌ها مجموعه‌ی جواب فازی را بدون اینکه تصمیم گیرنده را به سمت انتخاب بهترین مسیر هدایت کنند، به دست می‌آورند. هم‌چنین آن‌ها تنها برای شبکه‌هایی با یال‌های نامنفی کاربرد دارند.

در این پایان‌نامه، الگوریتم تکراری را معرفی می‌کنیم که شاخص رتبه‌بندی به کار رفته در آن، عمومی است. به عبارت دیگر، هر زمان تصمیم‌گیرنده بخواهد مسئله‌ای را حل کند، می‌تواند شاخصی را که برای آن مسئله، بهترین است را انتخاب کند. این الگوریتم که تمام ایرادات بالا را برطرف می‌کند، براساس الگوریتم فورد-مور-بلمن برای شبکه‌های کلاسیک است و نیز می‌تواند برای شبکه‌هایی با پارامترهای منفی نیز به کار رود. هم‌چنین قادر است که دور منفی شبکه را تشخیص دهد.

علاوه بر این، الگوریتم جدیدی که تمام موارد فوق را داراست و تعداد تکرارهایش کمتر از تکرارهای الگوریتم فورد- مور-بلمن می‌باشد، پیشنهاد می‌کنیم.

فهرست مطالب

پیشگفتار

مسئله‌ی یافتن کوتاهترین مسیر از گره مبدأ مشخص به دیگر گره‌ها از اساسی‌ترین مسائلی است که در بسیاری از کاربردهای عملی از قبیل حمل و نقل، مسیر یابی، ارتباطات و... ظاهر می‌شود. در شبکه‌های عملی، طول یال‌ها معمولاً نشان دهنده‌ی زمان حمل و نقل یا هزینه‌ی جابه‌جایی می‌باشند. از آن‌جایی که زمان یا هزینه، با توجه به شرایط ترافیکی، در آمد وغیره، در حال نوسان است، نمی‌توان طول یال‌ها را با اعداد قطعی نشان داد و لذا استفاده از اعداد فازی در این مسائل، عملی‌تر است. هم‌چنین برای یافتن کوتاه‌ترین مسیر در شبکه‌هایی که طول یال‌های آن اعداد فازی باشند، نیاز به رتبه‌بندی این اعداد خواهیم داشت. بدین منظور، روش‌های رتبه‌بندی بسیاری در مقالات، ارائه گردیده است که در این پایان‌نامه به برخی از آن‌ها اشاره می کنیم. هم‌چنین مقالات بسیاری در زمینه‌ی مسئله‌ی کوتاه‌ترین مسیر فازی منتشر شده است[۲۹، ۳۰، ۳۳، ۱۷، ۱۰].

این مسئله، نخستین بار توسط دابویس و پراد[۱] در سال ۱۹۸۰ مورد مطالعه قرار گرفته است[۱۲]. یکی از الگوریتم‌هایی که برای حل این گونه مسائل مطرح شده است، الگوریتم دایکسترای تعمیم‌یافته است[۲۸]. در این الگوریتم، وزن یال‌ها، اعداد بازه‌ای در نظر گرفته می‌شود. اوکادا[۲][۲۶] نیز مفهوم درجه‌ی امکان وجود یال روی کوتاه‌ترین مسیر را تعریف می‌کند. لین و چن[۳][۲۵] تابع عضویت کوتاه‌ترین طول را به دست آورده و الگوریتمی برای یافتن بیشترین یال‌های اساسی در شبکه را پیشنهاد می‌کند. اوکادا و سوپر[۴][۳۰] مفهوم مسیر غیر مغلوب را تعریف کرده و الگوریتمی را که بر اساس روش برچسبی چندگانه برای مسائل کوتاه‌ترین مسیر چندگانه است را معرفی می‌کند.

---

[۱] Dubois and prad
[۲] okada
[۳] lin and chen
[۴] okada and super

در این پایان‌نامه، به معرفی الگوریتمی برای یافتن کوتاه‌ترین مسیر فازی، می‌پردازیم. این الگوریتم بر اساس الگوریتم فورد-مور-بلمن برای شبکه‌های کلاسیک پایه‌ریزی شده است. روش‌های رتبه‌بندی در این الگوریتم، عمومی است، به عبارت دیگر، تصمیم گیرنده می‌تواند هر روشی را که برای آن مسئله مناسب است، انتخاب کند. این الگوریتم علاوه بر اینکه در شبکه‌های با پارامترهای فازی منفی کاربرد دارد، قادر است در صورت وجود دور منفی در شبکه، آن را تشخیص دهد. در فصل اول، ابتدا مقدماتی از گراف را بیان کرده سپس مسئله‌ی کوتاه‌ترین مسیر را معرفی و برخی از روش‌های حل این مسئله را با ذکر مثال، مطرح می‌کنیم. در فصل دوم مفاهیمی از منطق فازی، اعداد فازی و حساب آن‌ها را بیان کرده و سپس به معرفی برخی از روش‌های رتبه‌بندی اعداد فازی می‌پردازیم. در فصل سوم، ضمن معرفی مسئله‌ی کوتاه‌ترین مسیر فازی، الگوریتم مورد نظر را ارائه داده و در پایان با حل چند مثال، صحت کار الگوریتم را بررسی می‌کنیم. در فصل چهارم نیز، الگوریتم جدیدی را که بر اساس الگوریتم ین برای شبکه‌های کلاسیک می‌باشد و تقریباً در نصف تعداد تکرارهای الگوریتم معرفی شده در فصل سوم به جواب می‌رسد را پیشنهاد می‌کنیم. حل مثال‌های فصل سوم با این الگوریتم، پایان‌بخش این فصل خواهد بود.

فصل ۱

مسئله‌ی کوتاه‌ترین مسیر

۱.۱ مقدمه‌ای بر گراف

تعریف ۱.۱.۱. یک گراف بدون جهت $G = (N, E)$ از یک مجموعه‌ی N از گره‌ها و یک مجموعه‌ی E از یال‌های بدون جهت تشکیل می‌شود،که یک یال e یک زوج نامرتب از گره‌های متمایز می‌باشد، یعنی یک زیرمجموعه‌ی $\{i, j\}$ از N. توجه داریم که یک یال بدون جهت مانند $\{i, j\}$ همان یال بدون جهت $\{j, i\}$ می‌باشد. به علاوه طوقه‌هایی مانند $\{i, i\}$ قابل قبول نمی‌باشند.

تعریف ۲.۱.۱. درجه‌ی یک گره در یک گراف بدون جهت برابر با تعداد یال‌های واقع بر آن گره است.

تعریف ۳.۱.۱. رأسی با درجه‌ی ورودی صفر را ریشه می‌نامند.

تعریف ۴.۱.۱. یک گشت از گره $i_۱$ به گره i_t در یک گراف بدون جهت، دنباله‌ای متناهی از گره‌های $i_۱, i_۲, ..., i_t$است به طوری که $\{i_k, i_{k+۱}\} \in E$ و $i = ۱, ۲, ..., t - ۱$.

تعریف ۵.۱.۱. یک گشت را یک مسیر می‌گوییم هرگاه گره تکراری نداشته باشد.

تعریف ۶.۱.۱. یک دور، گشتی مانند $i_۱, i_۲, ..., i_t$ است که گره‌های $i_۱$ و $i_۲$و... و $i_{t-۱}$ متمایز باشند (وبنابراین یک مسیر تشکیل دهند) و $i_۱ = i_t$. به‌علاوه لازم است که تعداد $t - ۱$ گره

متمایز، حداقل برابر ۳ باشد.این به خاطر گشت‌هایی به صورت i, j, i است که در واقع همان یال $\{i, j\}$ می‌باشد که به صورت پیشرو و پسرو پیموده شده است.

تعریف ۷.۱.۱. یک گراف بدون جهت را همبند می‌گوییم هر گاه بین هر دو گره متمایز $i, j \in N$ مسیری از i به j وجود داشته باشد.

برای گراف‌های بدون جهت، اغلب تعداد گره‌ها را با $|N|$ یا n، و تعداد یال‌ها را با $|E|$ یا m نشان می‌دهیم.

تعریف ۸.۱.۱. یک گراف جهت‌دار $G = (N, E)$ شامل یک مجموعه‌ی N از گره‌ها ویک مجموعه‌ی A از یال‌ها (های جهت‌دار)است، که یک یال جهت‌دار، زوج مرتب (i, j) از گره‌های متمایز می‌باشد.

بنابر تعریف فوق، ممکن است که هر دو یال (i, j) و (j, i) متعلق به مجموعه‌ی یال‌های A باشند ولی یال‌هایی مانند (i, i) امکان‌پذیر نمی‌باشند.

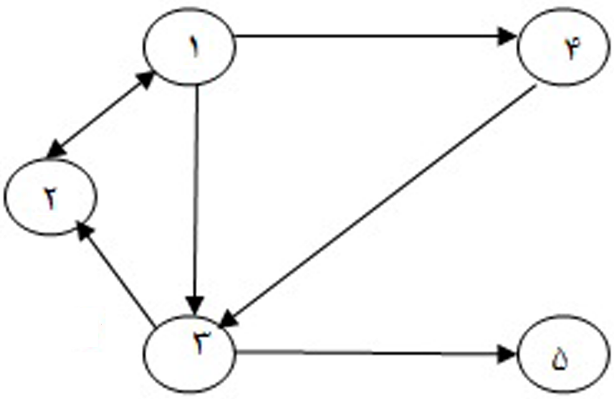
تعریف ۹.۱.۱. یک گراف جهت‌دار را همبند می‌گوییم هر گاه گراف بدون جهت وابسته به آن همبند باشد.

حال به تعریف یک گشت در گراف های جهت‌دار می‌پردازیم؛ توجه به این نکته مهم است که بنابه این تعریف می توانیم یک یال را در هر جهتی طی کنیم. به طور دقیق‌تر، یک گشت دنباله‌ای مانند $i_1, i_2, ..., i_t$ از گره‌ها،همراه با دنباله‌ی $a_1, a_2, ..., a_{t-1}$ از یال‌ها است که به ازای $k = 1, 2, ..., t-1$، یا $a_k = (i_k, i_{k+1})$ (در این حالت می‌گوییم a_k یک یال پیشرو است)یا $a_k = (i_{k+1}, i_k)$ (در این حالت a_kرا یک یال پسرو می‌نامیم).

تعریف ۱۰.۱.۱. یک گشت در یک گراف جهت‌دار را یک مسیر می‌گوییم هر گاه گره‌های $i_1, i_2, ..., i_t$ آن متمایز باشند.

تعریف ۱۱.۱.۱. یک دور در یک گراف جهت‌دار گشتی مانند $i_1, i_2, ..., i_t$ است که گره‌های $i_1, i_2, ..., i_{t-1}$ متمایز باشند و $i_1 = i_t$.

توجه می‌کنیم که یک دور می‌تواند شامل فقط دو گره متمایز باشد. (بر خلاف تعریف متناظر برای گراف‌های بدون جهت) بنابراین یک دنباله مانند $i, (i, j), j, (j, i), i$ یک دور است.
در پایان یک گشت، مسیر یا دور را جهتدار می‌گوییم هرگاه فقط شامل یال‌های همسو باشد.



شکل ۱

برای گراف نشان داده شده در شکل ۱ دنباله‌ی $۱, (۱,۳), ۳, (۳,۲), ۲, (۱,۲), ۱, (۱,۴), ۴$، یک گشت است ولی یک گشت جهت‌دار نیست. چون $(۱,۲)$ یک یال پسرو می‌باشد.
دنباله‌ی $۱, (۲,۱), ۲, (۳,۲), ۳, (۱,۳), ۱$ یک دور جهت‌دار است.
دنباله‌ی $۱, (۱,۲), ۲, (۲,۱), ۱$ نیز یک دور جهت‌دار است.
دنباله‌ی $۴, (۴,۳), ۳, (۱,۳), ۱, (۱,۲), ۲$ یک مسیر است، ولی یک مسیر جهت‌دار نیست، چون $(۱,۳)$ یک یال پسرو می‌باشد.

تعریف ۱۲.۱.۱. دوری که مجموع وزن یال‌های آن عددی منفی باشد دور منفی[۱] می‌گویند.

تعریف ۱۳.۱.۱. برای یک گره v در یک گراف جهت‌دار، تعداد یال‌هایی که جهت آن‌ها از رأس‌های دیگر به طرف v است، درجه‌ی ورودی گره v نامیده می‌شوند.

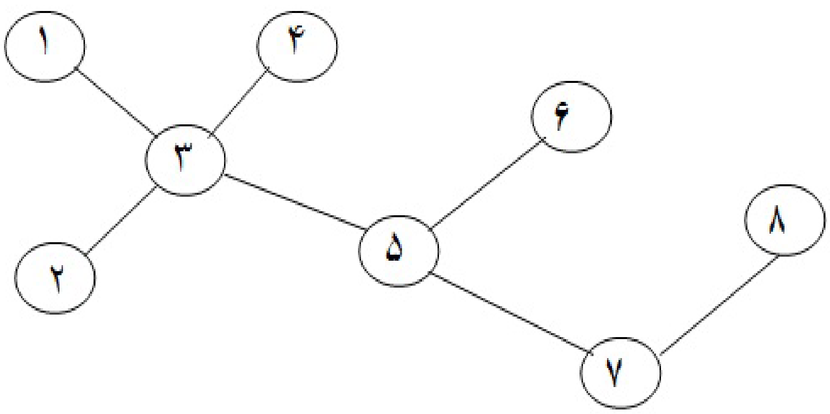
[۱] negative circuite

تعریف ۱.۱.۱۴. برای یک گره v در یک گراف جهت‌دار، تعداد یال‌هایی که جهت آن‌ها از v به رأس‌های دیگر است، درجه‌ی خروجی گره v نامیده می‌شود.

تعریف ۱.۱.۱۵. یک گراف جهت‌دار را در نظر می‌گیریم، برای گره‌ای چون i، هر گره j، به طوری که یالی بین i و j وجود داشته باشد که جهت آن از j به i باشد را گره مقدم گویند و مجموعه‌ی این گره‌ها را مجموعه‌ی مقدم می‌نامند.

تعریف ۱.۱.۱۶. یک گراف بدون جهت $G = (N, E)$ را یک درخت می‌گوییم هر گاه همبند و بدون دور باشد.

تعریف ۱.۱.۱۷. اگر درجه‌ی گره‌ای از یک درخت مساوی با یک باشد، آن‌گاه آن گره را یک برگ می‌نامیم. برای مثال گره ۱،۲،۴،۶،۸ در شکل ۲ یک برگ می باشند .



شکل ۲

تعریف ۱.۱.۱۸. اگر به ازای $i \neq n$ مسیر از گره i به گره n کوتاه‌ترین مسیر باشد، آن‌گاه آن درخت را درخت کوتاه‌ترین مسیرها می‌گوییم.

خواص مهم درخت‌ها

هر درخت با بیش از یک گره دارای حداقل یک برگ است.
یک گراف بدون جهت یک درخت است اگر و تنها اگر همبند و دارای $|N|-۱$ یال باشد.
به ازای هر دو گره متمایز i و j در یک درخت، یک مسیر یکتا از i به j وجود دارد.
اگر به یک درخت یک یال جدید اضافه کنیم، آن‌گاه گراف حاصل دقیقاً یک دور خواهد داشت.

تعریف ۱.۱.۱۹. یک گراف بدون جهت و همبند $G=(N,E)$ را در نظر بگیرید، فرض کنید $E_۱$ یک زیرمجموعه‌ی E باشد به‌طوری‌که $T=(N,E_۱)$ یک درخت باشد، به چنین درختی درخت فراگیر می‌گویند.

تعریف ۱.۱.۲۰. درخت فراگیر مینیمال درخت فراگیری است که مجموع هزینه‌ی یال‌های آن مینیمم باشد.

مسئله‌ی درخت فراگیر مینیمال به طور طبیعی در اکثر کاربردها ظاهر می‌شود، مثلاً اگر یال‌ها متناظر با جاده‌های ارتباطی باشند، آن‌گاه یک درخت فراگیر، مجموعه‌ای از جاده‌هاست که یک گره را به گره دیگر وصل می‌کند، بنابراین یک درخت فراگیر مینیمال یک شبکه‌ی ارتباطی است که این ارتباطات را میسر می‌سازد و هزینه‌ی آن کم‌ترین مقدار است.

۲.۱ مسائل جریان در شبکه

مسائل جریان در شبکه (که به مسائل نقل و انتقال نیز معروفند) از جمله مسائل برنامه‌ریزی خطی هستند که کراراً در کاربردهای عملی رخ می‌دهند. حالت‌های خاص آن‌ها شامل مسائل تخصیص، حمل‌ونقل، ماکزیمم جریان و کوتاه‌ترین مسیر است و به طور طبیعی در تحلیل و طراحی شبکه‌های ارتباطی، حمل‌ونقل و لجستیکی، علاوه بر خیلی زمینه‌های دیگر بروز می‌کند.

۱.۲.۱ مدل‌بندی مسئله‌ی جریان در شبکه

یک شبکه، یک گراف جهت‌دار $G = (N, A)$ همراه با بعضی اطلاعات عددی اضافی می‌باشد، مانند اعداد b_i که نشان‌دهنده‌ی عرضه‌ی گره‌های $i \in N$ هستند، اعداد نامنفی (مثبت یا صفر) u_{ij} که نشان‌دهنده‌ی ظرفیت یال‌های $(i, j) \in A$ می‌باشند و اعداد c_{ij} که نشان‌دهنده‌ی هزینه‌ی یک واحد جریان بر یال‌های (i, j) هستند.
پیش‌فرضمان از یک شبکه آن است که بعضی از کالاها بر یال‌های شبکه جریان دارند. از x_{ij} برای نشان‌دادن اندازه‌ی جریان در یال (i, j) استفاده می‌کنیم. عرضه‌ی b_i رابه عنوان اندازه‌ی جریانی که به گره i وارد می‌شود، تعبیر می‌کنیم. به‌ویژه گره i را یک مبدأ می‌گوییم هرگاه $b_i \geq ۰$ و مقصد می‌نامیم هرگاه $b_i \leq ۰$ باشد.
حال شرایط زیر را بر متغیرهای جریان x_{ij}، $(i, j) \in A$ تحمیل می‌کنیم:

$$b_i + \sum_j x_{ji} = \sum_j x_{ij} \qquad \forall i \in N \tag{۱}$$

$$۰ \leq x_{ij} \leq u_{ij} \qquad \forall (i, j) \in A \tag{۲}$$

رابطه‌ی (۱) یک قانون بقای انرژی است و اظهار می‌دارد که اندازه‌ی جریان ورودی به گره i باید برابر با کل جریان خروجی از آن باشد. رابطه‌ی (۲) الزام می‌دارد که جریان در یک یال باید نامنفی باشد و نمی‌تواند بیشتر از ظرفیت آن یال باشد. یک بردار مؤلفه‌های x_{ij}،$(i, j) \in A$ را یک جریان می‌نامیم.
یک جریان صادق در (۲)-(۱) را یک جریان شدنی می‌گوییم.
با جمع‌بندی هر دو طرف رابطه‌ی (۱) بر همه‌ی گره‌های $i \in N$، خواهیم داشت:

$$\sum_{i \in N} b_i = ۰ \tag{۳}$$

که به معنی آن است که کل جریان ورودی به شبکه (در مبدأها) باید برابر با کل جریان خروجی از شبکه (در مقصدها) باشد. از این پس، همواره فرض می‌کنیم که شرط $\sum_{i \in N} b_i = ۰$ برقرار است، چون در غیر این صورت هیچ بردار جریانی نمی‌تواند در شرایط بقاء جریان صدق کند و یک

مسئله‌ی نشدنی خواهیم داشت.
مسئله‌ی عمومی جریان در شبکه با هزینه‌ی مینیمم عبارت است از مینیمم‌سازی یک تابع هزینه‌ی خطی به شکل

$$\sum_{(i,j)\in A} c_{ij}x_{ij}$$

بر همه جریان‌های شدنی. ملاحظه می‌کنیم که مسئله‌ی فوق یک مسئله‌ی برنامه‌ریزی خطی است.
اگر $u_{ij} = \infty$، به ازای هر $(i, j) \in A$، آن‌گاه مسئله را بدون ظرفیت می‌گوییم. در غیر این صورت آن را ظرفیت‌دار می‌گوییم.
یکی از حالت‌های مهم مسئله‌ی جریان در شبکه، مسئله‌ی کوتاه‌ترین مسیر است که در ادامه به آن می‌پردازیم.

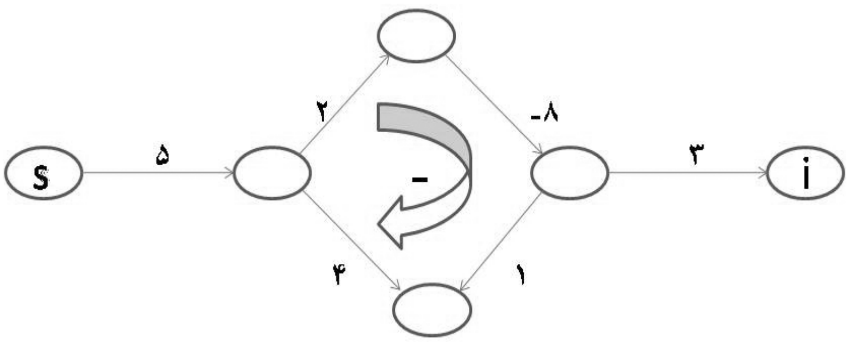
۳.۱ مسئله‌ی کوتاه‌ترین مسیر

مسئله‌ی یافتن کوتاه‌ترین مسیر یکی از مسائل اساسی در نظریه‌ی گراف است که مطالعات بسیاری در مورد آن صورت گرفته است. این مسئله در کاربردهای عملی بسیاری از قبیل حمل ونقل، مسیریابی، ارتباطات، مدیریت زنجیره موجودی و... ظاهر شده است.
گراف جهت‌دار $G = (V, E)$ که در آن V، مجموعه‌ی گره‌ها و E، مجموعه‌ی یال‌های آن است را در نظر می‌گیریم. به ازای هر یال $(i, j) \in E$ یک هزینه یا طول c_{ij} وجود دارد که در حالت کلی می تواند منفی، مثبت یا صفر باشد. طول یک گشت، مسیر یا دور را مجموع طول یال‌های آن تعریف می‌کنیم. مسیری از یک گره معین به گره دیگری را کوتاه‌ترین مسیر می‌گوییم هرگاه طول آن از طول هر مسیر دیگری بین آن دو گره کوچک‌تر باشد.
مسئله‌ی کوتاه‌ترین مسیر را می‌توان در چند شکل مختلف مطرح کرد:

۱. کوتاه‌ترین مسیر از یک گره مبدأ مفروض به گره‌های دیگر

۲. کوتاه ترین مسیر بین هر دو جفت گره دلخواه

در اینجا مسئله‌ی کوتاه‌ترین مسیر حالت اول را در نظر می‌گیریم. برای راحتی فرض می‌کنیم که s گره مبدأ باشد، هم‌چنین فرض می‌کنیم که حداقل یک مسیر از s به گره دلخواه $s \neq i$ و $i \in v - \{s\}$ وجود دارد، به عبارت دیگر فرض می‌کنیم مسئله‌ی کوتاه‌ترین مسیر مورد نظر، شدنی باشد. البته وزن یال‌ها ممکن است منفی باشد که این یال‌های منفی کار را پیچیده می‌کنند، چون در صورت وجود دور منفی در شبکه، کوتاه‌ترین مسیر بدست نخواهد آمد.
در حالت کلی، کوتاه‌ترین مسیر از s به $s \neq i$ وجود دارد اگر و تنها اگر حداقل یک مسیر از s به $i \neq s$ که شامل دور منفی نباشد وجود داشته باشد زیرا در صورت وجود دور منفی برای به‌دست آوردن کوتاه‌ترین مسیر در دور نامتناهی خواهیم افتاد و هیچ‌گاه به نقطه‌ی مقصد نخواهیم رسید. شکل ۳ را ببینید.



شکل ۳: دور منفی در شبکه

۱.۳.۱ فرمول‌بندی ریاضی مسئله‌ی کوتاه‌ترین مسیر

می توان مسئله‌ی کوتاه‌ترین مسیر در شبکه را همان مسئله‌ی جریان در شبکه در نظر گرفت اگر بخواهیم در آن یک واحد جریان را از گره s (مبدأ) به گره n با کم‌ترین هزینه ارسال کنیم. بنابراین $b_i = ۰, b_۱ = ۱, b_n = -۱$ برای $i \neq ۱$ یا n. بنابراین شکل ریاضی این مسئله به‌صورت زیر خواهد بود:

$$
\begin{aligned}
&min \quad f(x)=\sum_{i=۱}^{n}\sum_{j=۱}^{n}c_{ij}x_{ij}\\
&s.t \quad \sum_{j=۱}^{n}x_{ij}-\sum_{k=۱}^{n}x_{ki}=\begin{cases} ۱ & \text{if} \quad i=۱\\ ۰ & \text{if} \quad i\neq ۱ \quad \text{یا} \quad n\\ -۱ & \text{if} \quad i=n\end{cases} \qquad (۴)\\
&x_{ij}=۰ \text{یا} ۱ \qquad i,j=۱,۲,...,n
\end{aligned}
$$

محدودیت‌های ۱یا۰ $= x_{ij}$ نشان‌دهنده‌ی این است که یال (i,j) روی مسیر قرار دارد یا نه.

۲.۳.۱ دوگان مسئله‌ی کوتاه‌ترین مسیر

$$
\begin{aligned}
&max \qquad w_۱-w_m\\
&s.t \qquad w_i-w_j\leq c_{ij} \qquad i,j=۱,۲,...,m\\
&\qquad\qquad w_i \quad \text{نامقید} \qquad i=۱,۲,...,m
\end{aligned}
$$

برای راحتی بیشتر از جای‌گزینی $w'_i=-w_i$ استفاده می‌کنیم. در بهینگی $w'_i-w'_۱$ نشان دهنده‌ی طول کوتاه‌ترین مسیراز گره ۱ به گره i است.

با توجه به نوع شبکه که هزینه‌ی یال‌های آن مثبت، منفی یا دلخواه باشند الگوریتم‌های مختلفی برای حل این مسئله وجود دارد که در زیر به برخی از آن‌ها اشاره می‌کنیم:

۴.۱ الگوریتم‌های حل مسئله‌ی کوتاه‌ترین مسیر

۱.۴.۱ الگوریتم دایکسترا

الگوریتم دایکسترا[۲] یکی از مؤثرترین الگوریتم‌ها برای یافتن کوتاه‌ترین مسیر جهت‌دار از گره مشخص s به دیگر گره‌های شبکه $G = (V, E, l)$ با طول یال‌های نامنفی $l_{ij} \geq ۰$ برای تمام $(i, j) \in E$، است که اولین بار توسط دایکسترا در سال ۱۹۵۹ مطرح شد[۱۷].
این الگوریتم بر اساس برچسب‌گذاری[۳] گره‌ها می‌باشد که این برچسب‌ها مرتباً به وسیله‌ی روش تکراری به‌هنگام می‌شوند.
روند الگوریتم بدین صورت است که گره i از G که با عدد نامنفی $l(i)$ برچسب گذاری شده باشد را در نظر می‌گیریم. ابتدا تمام برچسب‌ها، موقت در نظر گرفته می‌شوند سپس در هر تکرار، یکی از برچسب‌های موقت مثلاً $l(j)$ را به دائم تبدیل کرده و آن را با l^* نشان می‌دهیم. توقف الگوریتم نیز زمانی صورت می‌گیرد که هیچ برچسب موقتی باقی نمانده باشد. اعداد روی برچسب‌ها نیز طول کوتاه‌ترین مسیر و مسیر مربوط به آن را نشان می‌دهند.
روند الگوریتم به‌صورت زیر است:

گام ۱: : قرار دهید

$$\begin{aligned} &l^*(s) = ۰ \\ &l(i) = l(s, i) \qquad i \in V, i \neq s \\ &l(s, i) = \infty \qquad i \neq s, (s, i) \notin E \end{aligned}$$

و گره i را با $[s, l(i)]$ برچسب‌گذاری کنید.

گام ۲: :از میان تمام برچسب‌های موقت، برچسب زیر را انتخاب کنید.

$$l(k) = minl(i)$$

[۲] Dijkstra algorithm
[۳] labeling

و $l(k)$ را به $l^*(k)$ تغییر دهید.

گام ۳: تمام برچسب‌های موقت گره i (برای تمام iها) که $(k,i) \in E$ را به صورت زیر به هنگام کنید.

$$l(i) = min[l(i), l^*(i) + l(k,i)]$$

و در صورت کاهش $l(i)$ ، برچسب گره i یعنی $[k, l(i)]$ را تغییر دهید در غیر این صورت آن را نگه داشته و به گام ۲ بروید.

برای توضیح پیچیدگی الگوریتم ابتدا مفهومی به نام اوی بزرگ را تعریف می کنیم.

تعریف ۱.۴.۱. فرض کنید $f(n)$ و $g(n)$ توابعی از اعداد صحیح مثبت به اعداد حقیقی مثبت باشند. می نویسیم $f(n) = O(g(n))$، اگر عدد مثبت $c > ۰$ وجود داشته باشد به طوری که برای هر n به اندازه ی کافی بزرگ، داشته باشیم $f(n) \leq cg(n)$.

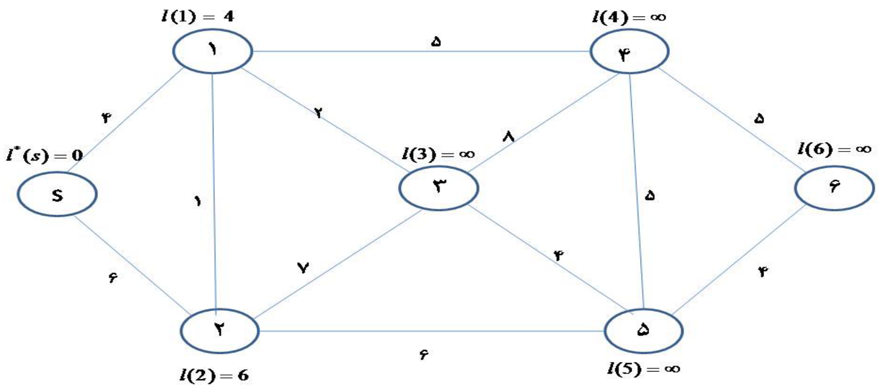
در واقع در یک الگوریتم، پیچیدگی الگوریتم متناظر است با اندازه ی ورودی الگوریتم. بنابراین اگر فرض کنیم اندازه ی الگوریتم برابر با تعداد گره ها باشد که با n نشان داده می شود. لذا پیچیدگی الگوریتم تابعی از اندازه ی ورودی خواهد بود.

پیچیدگی الگوریتم

این الگوریتم شامل عمگرهای مقایسه و جمع است که می توان تعداد آن‌ها را شمرد. در گام ۲، ابتدا $n - ۱$ عمل مقایسه صورت می پذیرد. در تکرار بعدی با برچسب دائم گرفتن یکی از گره‌ها، در این گام $n - ۲$ مقایسه انجام می‌شود. به همین ترتیب مجموع تمام مقایسات انجام شده در گام ۲ در تمام تکرارها به صورت زیر است.

$$(n - ۱) + (n - ۲) + ... + ۲ + ۱ = \frac{۱}{۲}n(n - ۱)$$

به طریق مشابه، در گام ۳ نیز، $n - ۱$ مقایسه و به همان تعداد عمل جمع انجام می گیرد. در تکرار بعدی، $n - ۲$ مقایسه و جمع انجام صورت می پذیرد. به همین ترتیب تعداد اعمال این گام در



شکل ۴

تمام تکرارها به قرار زیر است .

$$(n-۱)+(n-۲)+...+۲+۱=\frac{۱}{۲}n(n-۱)$$

بنابراین،پیچیدگی محاسباتی این الگوریتم از $O(n^۲)$ خواهد بود. برای درک بیشتر الگوریتم به مثال زیر توجه کنید.

مثال: شبکه‌ی $G=(N,E)$ در شکل ۴ را در نظر می‌گیریم که در آن هر یال بی‌جهت یک جفت یال با طول یکسان را نشان می‌دهد که یکی از i به j و دیگری از j به i می‌باشد. در اینجا قصد داریم الگوریتم دایکسترا را برای تعیین کوتاه‌ترین مسیر از s به تمام گره‌های ۱, ۲, ۳, ۴, ۵, ۶ به کار ببریم.

گام ۱: گره s، برچسب دائم $l^*(s)=۰$ و تمام گره‌های دیگر برچسب موقتی را که در شکل بالا نشان داده شده است، می‌گیرند.

تکرار اول :

گام ۲: از بین تمام برچسب‌های موقت، برچسب زیر انتخاب می‌شود.

$$l(۱) = min[l(۱), l(۲), l(۳), l(۴), l(۵), l(۶)]$$
$$= min[۴, ۶, \infty, \infty, \infty, \infty] = ۴$$

بنابراین $k = ۱$. برچسب $l(۱)$ را به $l^*(۱) = ۴$ تغییر می‌دهیم.

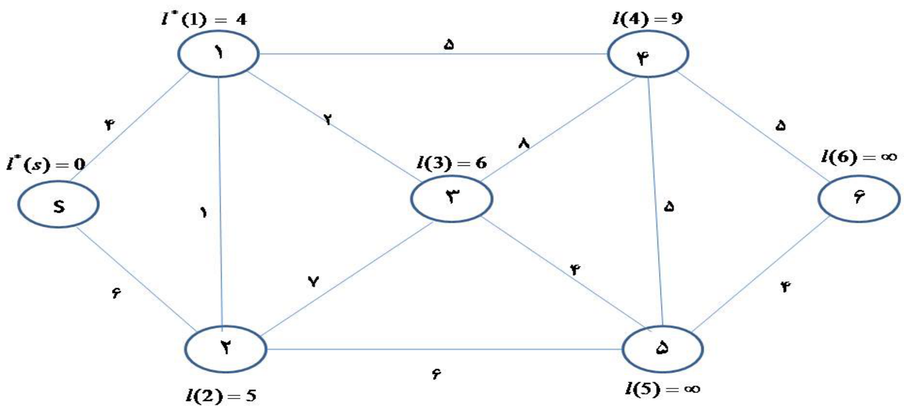
گام ۳:

$$l(i) = min[l(i), l^*(۱) + l(۱, i)] \qquad i = ۱, ۲, ۳, ۴, ۵, ۶$$

پس داریم:

$$l(۲) = min[l(۲), l^*(۱) + l(۱, ۲)] = min[۶, ۴ + ۱] = ۵$$
$$l(۳) = min[l(۳), l^*(۱) + l(۱, ۳)] = min[\infty, ۴ + ۲] = ۶$$
$$l(۴) = min[l(۴), l^*(۱) + l(۱, ۴)] = min[\infty, ۴ + ۵] = ۹$$

نتایج این مرحله در شکل زیر نشان داده شده است.



شکل ۵

تکرار دوم:

گام ۲: از بین تمام برچسب‌های موقت در تکرار قبل، برچسب با کم‌ترین طول را انتخاب می‌کنیم.

$$l(۲) = min[l(۲), l(۳), l(۴), l(۵), l(۶)]$$
$$= min[۵, ۶, ۹, \infty, \infty] = ۵$$

پس $k = ۲$. برچسب موقت $l(۲)$ را به برچسب دائم $l^*(۲) = ۵$ تغییر می‌دهیم.

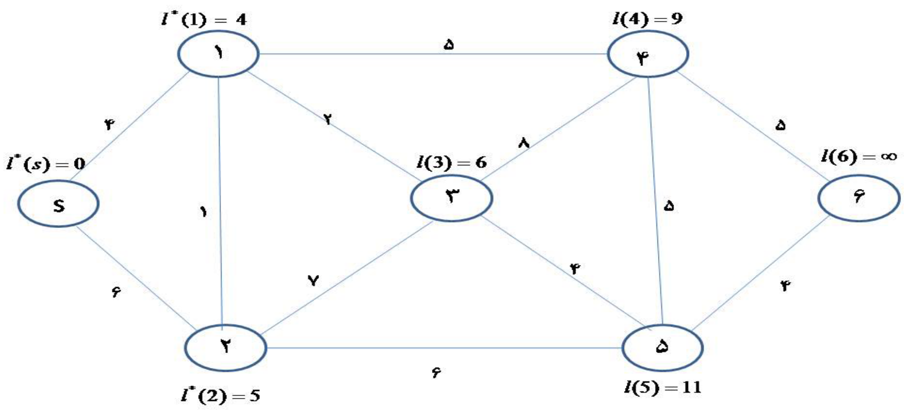
گام ۳:

$$l(i) = min[l(i), l^*(۲) + l(۲, i)] \qquad i = ۳, ۴, ۵, ۶$$

پس داریم

$$l(۳) = min[l(۳), l^*(۲) + l(۲, ۳)] = min[۶, ۵ + ۷] = ۶$$
$$l(۵) = min[l(۵), l^*(۲) + l(۲, ۵)] = min[\infty, ۵ + ۶] = ۱۱$$

نتایج این مرحله نیز در شکل زیر نشان داده شده است.



شکل ۶

تکرار۳:

گام۲: از بین تمام برچسب‌های موقتی که در تکرار دوم به دست آوردیم، برچسب با کم‌ترین طول را انتخاب می‌کنیم.

$$l(۳) = min[l(۳), l(۴), l(۵), l(۶)]$$
$$=min[۶, ۹, ۱۱, \infty] = ۶$$

پس $k = ۳$. برچسب موقت $l(۳)$ را به برچسب دائم $l^*(۳)$ تغییر می‌دهیم.

گام۳:

$$l(i) = min[l(i), l^*(۳) + l(۳, i)] \qquad i = ۴, ۵, ۶$$

پس داریم

$$l(۴) = min[l(۴), l^*(۳) + l(۳, ۴)] = min[۹, ۶ + ۸] = ۹$$
$$l(۵) = min[l(۵), l^*(۳) + l(۳, ۵)] = min[۱۱, ۶ + ۴] = ۱۰$$

تکرار چهارم:

گام۲:

$$l(۴) = min[l(۴), l(۵), l(۶)] = min[۹, ۱۰, \infty] = ۹$$

پس $k = ۴$. برچسب موقت $l(۴)$ را به برچسب دائم $l^*(۴)$ تغییر می‌دهیم.

گام۳:

$$l(i) = min[l(i), l^*(۴) + l(۴, i)] \qquad i = ۵, ۶$$

پس داریم

$$l(۵) = min[l(۵), l^*(۴) + l(۴, ۵)] = min[۱۰, ۹ + ۵] = ۱۰$$
$$l(۶) = min[l(۶) + l^*(۴) + l(۴, ۶)] = min[\infty, ۹ + ۵] = ۱۴$$