نشریه تصمیم‌گیری و تحقیق در عملیات

Journal of Decisions and Operational Research

دوره ۴، شماره (۳)، پاییز ۱۳۹۸

ارائه‌ی روشی برای انتخاب کوتاه‌ترین مسیر مقید با استفاده از برش‌های منطقی

سجاد مرادی[*] ، غلامرضا کرمعلی

دانشکده علوم پایه، گروه ریاضی کاربردی، دانشگاه علوم و فنون هوایی شهید ستاری، مهرآباد جنوبی، تهران، ایران.

چکیده

مسئله‌ی کوتاه‌ترین مسیر یکی از مسائل کلاسیک و پرکاربرد بهینه‌سازی است که الگوریتم‌های کارآمدی برای آن ارائه شده است. در این مسئله شبکه‌ای شامل مجموعه‌ای از نقاط و کمان‌های بین آن‌ها درنظر گرفته شده و به هر کمان پارامتری مانند طول، هزینه یا زمان طی مسیر نسبت داده می‌شود. هدف اصلی مسئله، یافتن کوتاه‌ترین یا کم‌هزینه‌ترین مسیر بین دو نقطه‌ی مشخص است. با درنظر گرفتن پارامتر دیگری برای هریک از کمان‌ها و اضافه‌کردن یک محدودیت دیگر، به‌صورت قید ظرفیت، مسئله به شرایط واقعی نزدیک‌تر خواهد شد. این مسئله توسعه داده‌شده به مسئله‌ی کوتاه‌ترین مسیر مقید معروف است که پیچیدگی بالاتری دارد و برای حل آن به الگوریتم‌های کارآمدی نیاز است. در این مطالعه، یک روش حل برای این مسئله ارائه شده است که قادر است در مدت زمان کوتاهی به جواب بهین برسد. در این روش از یک الگوی تکراری حل مدل آزادشده و اضافه‌کردن برش‌های منطقی در هر تکرار استفاده می‌شود. نتایج پیاده‌سازی الگوریتم ارائه‌شده بر روی شبکه‌های مختلف، کارایی آن را به‌خوبی نشان می‌دهد.

واژه‌های کلیدی: شبکه، مسیریابی مقید، مدل آزادشده، الگوریتم حل.

دریافت: ۱۳۹۸/۰۳/۱۸ اصلاح: ۱۳۹۸/۰۵/۰۷ پذیرش: ۱۳۹۸/۰۷/۱۵

۱- مقدمه

مسئله‌ی پیداکردن کوتاه‌ترین مسیر یک مسئله مهم و کاربردی در بهینه‌سازی است که کاربردهای زیادی در زمینه‌های مختلفی از قبیل انتخاب مسیر حرکت وسایل نقلیه، جریان‌های الکتریکی، کنترل پروژه و عملیات‌های نظامی دارد (آهوجا و همکاران، ۱۹۹۵). در این مسئله شبکه‌ای از نقاط و راه‌های مواصلاتی بین آن‌ها وجود دارد که هرکدام، هزینه یا طول مسیر مشخصی دارند و یافتن کوتاه‌ترین یا کم‌هزینه‌ترین مسیر بین دو نقطه‌ی مشخص، مطلوب مسئله است. روش‌های حل ترکیباتی زیادی، با پیچیدگی چندجمله‌ای، برای این مسئله ارائه شده است که به برخی از آن‌ها در کتاب جریان‌های شبکه (آهوجا و همکاران، ۱۹۹۵) اشاره شده است. معروف‌ترین روش‌های حل ارائه‌شده برای مسئله‌ی کوتاه‌ترین مسیر، الگوریتم‌های دایجسترا، فلوید–وارشال و بلمن–فورد می‌باشد که از مرتبه‌ی چندجمله‌ای هستند. در این روش‌ها، با استفاده از یک روند اصلاح مسیرهای پیموده شده، کوتاه‌ترین مسیر از مبدا به همه‌ی نقاط شبکه، در یک مدت زمان کوتاه به‌دست می‌آید. چرکاسکی و همکاران (۱۹۹۶) و هم‌چنین ژان و نون (۱۹۹۸) در مطالعات جداگانه‌ای پیاده‌سازی نمونه‌های مختلف

[*] نویسنده مسئول مکاتبات: sajadmoradi@aut.ac.ir

شناسه دیجیتال: 10.22105/dmor.2019.193389.1126

الگوریتم‌های موجود برای مسئله‌ی کوتاه‌ترین مسیر را بر روی شبکه‌های واقعی مورد بررسی قرار داده‌اند. علاوه‌بر روش‌های ترکیباتی استفاده از روش سیمپلکس هم مورد توجه بوده است. برای مثال سدونو-نودا و گنزالس-مارتین (۲۰۱۰) یک روش سیمپلکس اصلاح‌شده‌ی کارآمد با مرتبه‌ی چندجمله‌ای برای مسئله‌ی کوتاه‌ترین مسیر را ارائه دادند. هم‌چنین لوزانو و مداگلیا (۲۰۱۳) با استفاده از روش تولید ستون، یک روش حل دقیق برای این مسئله ارائه داده و کاربرد آن را بر روی چند مسئله‌ی واقعی مانند زمانبندی مورد بررسی قرار دادند.

کاربرد این مسئله در بسیاری از مسائل دنیای واقعی تنها یافتن مسیری که کم‌ترین هزینه یا مسافت ممکن را دارد نیست، بلکه علاوه‌بر آن، مسیر بهینه باید در برخی محدودیت‌ها و قیود اضافی نیز صدق کند. پژوهش‌گران توسعه و گسترش‌های مختلفی برای این مسئله ارائه کرده‌اند تا آن را به دنیای واقعی نزدیک‌تر کنند. برای مثال، مسیر موردنظر حتما از برخی از نقاط عبور کند یا ممنوعیت عبور از برخی از نقاط وجود داشته باشد، یا تعداد کمان‌های مسیر انتخاب‌شده در بازه‌ی مشخصی باشد. نمونه‌ی دیگری که می‌توان به آن اشاره کرد مسیریابی درون‌شهری است که در آن پارامترهای مختلفی مانند مسافت مسیر، مدت‌زمان ماندن در ترافیک، هزینه‌ی عبور از معابر ممنوعه مانند مناطق تحت پوشش طرح ترافیک، برای انتخاب مسیر وجود دارد. یک توسعه‌ی پرکاربرد مسئله‌ی کوتاه‌ترین مسیر به این صورت است که به هر کمان دو پارامتر مختلف مانند مسافت و زمان اختصاص داده شده و محدودیت زمان طی مسیر نیز اضافه شود. اضافه‌کردن محدودیت جدید به مسئله‌ی کوتاه‌ترین مسیر، آن‌را به مسئله‌ی کوتاه‌ترین مسیر مقید توسعه می‌دهد و باعث می‌شود که پیچیدگی مسئله مقید بالا رفته و به کلاس مسائل چند جمله‌ای زمان سخت نامعین[1] (ان‌پی-هارد) تعلق گیرد (گری، ۱۹۷۹). انتخاب بهترین مسیر برای حرکت نیروها در صحنه‌ی نبرد یک کاربرد جالب از این مسئله است که در این زمینه می‌توان به کارهای تاراپاتا (۲۰۰۳) و مورا و همکاران (۲۰۰۹) اشاره کرد. تاراپاتا (۲۰۳) با استفاده از نظریه‌ی گراف‌ها، پیچیدگی کاربرد نظامی مسئله‌ی کوتاه‌ترین مسیر و روش‌های حل آن را مطالعه کرده است. مورا و همکاران (۲۰۰۹) مسئله‌ی مسیریابی نیروهای نظامی را به‌صورت یک مسئله‌ی چندهدفه و با درنظر گرفتن فاکتورهای سرعت و امنیت مسیر مورد بررسی قرار دادند و با کمک الگوریتم لانه‌ی مورچگان یک روش ابتکاری برای دستیابی سریع به مسیرهای شدنی مسئله (نه لزوما بهینه) ارائه دادند. با این‌که استفاده از روش‌های ابتکاری و فراابتکاری کاربرد زیادی در حل مسائل ان‌پی-هارد دارد اما کیفیت چنین جواب‌هایی، به‌خصوص در حل مدل‌های با اندازه‌ی بزرگ، تضمین‌شده نیست (میرحسنی و همکاران، ۲۰۱۱). آولا و همکاران (۲۰۰٤) با درنظر گرفتن محدودیت برخی از منابع، مسئله‌ی مدیریت ناوگان حمل‌ونقل را در قالب کوتاه‌ترین مسیر مقید مدل‌سازی و برای حل آن از روش شاخه و برش استفاده کرد. روش حل ابتدایی که می‌توان برای این مسئله ارائه داد استفاده از الگوریتم پیدا کردن kامین مسیر کوتاه است که توسط اپستین ارائه شده است (اپستین، ۱۹۹۸). اگر همه‌ی مسیرهای موجود از مبدا به مقصد را بر حسب مقدار تابع هدف به‌صورت صعودی مرتب کنیم مسیری که در رتبه‌ی kام قرار می‌گیرد را kامین مسیر کوتاه می‌گویند؛ اگر اولین مسیر کوتاه (کوتاه-ترین مسیر) در شرط اضافه‌ی قید ظرفیت صدق نکند دومین مسیر کوتاه بررسی می‌شود و اگر این مسیر نیز در شرط اضافه صدق نکند به سراغ سومین مسیر کوتاه می‌رویم و این روند را تا جایی ادامه می‌دهیم که به جواب بهین برسیم. پوگلیس و گریرو (۲۰۱۳) در مطالعه‌ای مروری به بررسی روش‌ها و الگوریتم‌های دقیق ارائه‌شده برای حل مسئله مسیریابی مقید پرداختند. در این مطالعات، سه راه‌کار برای دستیابی به مسیر بهین ارائه شد که در هر مورد از یک یا چند تا از این راه‌کارها به‌صورت همزمان استفاده شده است. این راه‌کارها عبارتند از: الف) کوچک سازی شبکه‌ی موردنظر با حذف کمان‌هایی که با اطمینان می‌توان گفت بر روی مسیر بهینه قرار ندارند ب) استفاده از روش آزاد سازی لاگرانژ که در آن قید ظرفیت با درنظر گرفتن جریمه به تابع هدف منتقل شده و مسئله به مسئله‌ی کوتاه‌ترین مسیر تبدیل می‌شود و در آن ضرایب تابع هدف ترکیبی از همه پارامترهای مسئله است ج) استفاده از روش‌های مختلف برای ازبین‌بردن گپ

[1] Non-Deterministic Polynomial-Time Hard (NP-hard)

موجود بین جواب حاصل از مدل آزادسازی شده و جواب بهین. مهم‌ترین مطالعات موجود که در مطالعه‌ی آن‌ها به آن اشاره شده است عبارتند از مطالعات هندلر و ژانگ (۱۹۸۰)، سانتوس و همکاران (۲۰۰۷) و کارلایل و همکاران (۲۰۰۸). هندلر و ژانگ (۱۹۸۰) با استفاده از دوگان مسئله و روش آزادسازی لاگرانژ ابتدا یک جواب شدنی برای مسئله یافته و سپس با به‌کارگیری الگوریتم kامین مسیر کوتاه، برای حذف گپ موجود، جواب بهین مسئله را یافتند. سانتوس و همکاران (۲۰۰۷) به این نتیجه رسیدند که در روش آزادسازی لاگرانژ انتخاب مقدار مناسب برای جریمه در تابع هدف تاثیر زیادی در سرعت رسیدن به جواب بهینه دارد و این مقدار به تراکم مسیرهای شدنی مسئله‌ی مقید نسبت به مسیرهای نشدنی وابسته است و در شبکه‌های مختلف فرق می‌کند. آن‌ها با درنظر گرفتن مقادیر مختلف برای جریمه نشان دادند که می‌توان با تغییر جهت‌های جستجو، سریع‌تر به جواب بهین دست یافت. آن‌ها برای حذف جواب‌های غیربهینه از روش kامین مسیر کوتاه استفاده کردند. کارلایل و همکاران (۲۰۰۸) نیز با استفاده از روش آزادسازی لاگرانژ و درنظر گرفتن یک مقدار اولیه برای جریمه، یک جواب اولیه به‌دست آورده و سپس با مقایسه‌ی مسیرهای مختلف و هم‌چنین یک روش پیش پردازش، به مسیر بهینه دست یافتند. در این‌گونه روش‌ها انتخاب مناسب مقدار جریمه در تابع هدف مدل آزادسازی شده از اهمیت بالایی برخوردار است. هم‌چنین برای دستیابی به جواب بهین، در هر تکرار فقط یک مسیر غیربهینه حذف می‌شود و امکان دارد که جواب بهینه در مرتبه بالایی از kامین مسیر کوتاه قرار داشته باشد و نیاز به تعداد زیادی تکرار باشد که کارایی و سرعت الگوریتم را کاهش می‌دهد. تفاوت الگوریتم پیشنهادی در این مقاله با روش‌های اشاره شده این است که از آزاد سازی لاگرانژ استفاده نشده است و نیازی به تخمین مناسب برای مقدار جریمه در تابع هدف نمی‌باشد، بلکه از نسخه آزادشده‌ی مدل مسئله کوتاه‌ترین مسیر مقید، با حذف شرط صحیح‌بودن متغیرها، استفاده می‌شود. در این‌صورت ابتدا با استفاده از متغیرهای غیرصفر به‌دست آمده می‌توان یک جواب شدنی برای مسئله به‌دست آورد. سپس با به‌کارگیری یک روش صفحه–برش کارآمد در هر مرحله، مسیرهای دیده‌شده‌ی قبلی و تعدادی مسیر غیربهینه دیگر را حذف کرده و پس از تعداد متناهی تکرار به جواب بهینه می‌رسیم.

یکی از روش‌های برخورد با مسائل برنامه‌ریزی عدد صحیح که اخیرا موردتوجه بسیاری از محققین قرار گرفته است، استفاده از برش‌های منطقی است که در این زمینه می‌توان به کارهای ورستیکل و همکاران (۲۰۱۵) و جندرون و همکاران (۲۰۱۳) اشاره کرد. ورستیکل و همکاران (۲۰۱۵) برای مدیریت کانتینرها و زمانبندی ورود و خروج در بنادر در مسئله زمانبندی و تخصیص محموله‌ها به کشتی‌ها یک مدل عدد صحیح آمیخته‌ی پیچیده ارائه داده و برای حل آن از برش‌های منطقی کمک گرفتند. جندرون و همکاران (۲۰۱۳) نیز برای مسئله‌ی مکان‌یابی در شبکه‌های بی‌سیم از این روش بهره جستند. در این روش با استفاده از شرط صفرویک بودن متغیرهای دودویی و نیز شرایط منطقی حاکم بر مسئله، قیود جدیدی به مدل اضافه می‌شوند که بدون حذف جواب بهین مسئله، یک یا چند جواب غیربهینه را از فضای مسئله حذف کرده و باعث می‌شود که دستیابی به جواب بهین سریع‌تر شود. در این مقاله الگوریتمی برای حل مسئله کوتاه‌ترین مسیرمقید ارائه شده است که در آن با توجه به شرایط حاکم بر مسئله، در هر تکرار با اضافه‌کردن یک قید منطقی به مسئله و استفاده از فرم آزاد شده‌ی مدل عدد صحیح در هر تکرار، مسیرهای دیده شده قبلی و برخی از جواب‌های غیر بهینه را حذف کرده و یک گام به جواب بهین نزدیک‌تر می‌شویم.

در ادامه، در بخش ۲ پس‌از تعریف مسئله به معرفی پارامترها، متغیرها و مدل ریاضی مسئله می‌پردازیم. در بخش بعدی به ارائه‌ی الگوریتم حل مسئله و اثبات اعتبار آن می‌پردازیم. نتایج حاصل از پیاده‌سازی الگوریتم پیشنهادی را در بخش چهارم ارائه می‌کنیم و در بخش آخر به جمع‌بندی و نتیجه‌گیری خواهیم پرداخت.

۲- تعریف مسئله و مدل‌سازی

شبکه $N(V, A)$ متشکل از مجموعه نقاط V (گره‌ها) و مجموعه کمان‌های متصل به گره‌ها (A) را در نظر بگیرید، به‌طوری که برای هر کمان دو پارامتر مانند مسافت و مدت زمان طی مسیر تعریف شده و مقدار آن‌ها مشخص است. هدف پیداکردن کوتاه‌ترین مسیر بین دو نقطه‌ی مشخص مبدا و مقصد است به‌طوری که مدت زمان طی مسیر از سقف تعیین‌شده λ بیشتر نشود. از اندیس‌های i و j برای نشان‌دادن گره‌ها استفاده می‌شود و اگر کمانی از گره i به j وجود داشته باشد، آن را با زوج مرتب (i,j) نشان می‌دهیم. پارامترهای مسئله که به‌صورت مسافت و مدت زمان حرکت بین دو گره مجاور i و j می‌باشند را به‌ترتیب با d_{ij} و t_{ij} نشان می‌دهیم. هم‌چنین برای هر دو گره مجاور i و j متغیر دودویی (صفر یا یک) x_{ij} را به این صورت تعریف می‌کنیم که اگر کمان (i,j) روی مسیر انتخابی قرار دارد مقدار ۱ بگیرد و در غیر این صورت مقدار صفر بگیرد.

محدودیت‌های حاکم بر این مسئله که باید به مدل‌سازی ریاضی آن ها بپردازیم به‌شرح زیر است:

- *نقطه‌ی ابتدایی مسیر ($i=1$) همان مبدا موردنظر است و حتما باید یک کمان خروجی از آن به سمت یکی از گره‌های مجاورش انتخاب شود.*
- *نقطه‌ی انتهایی مسیر ($i=n$) مقصد موردنظر است و حتما باید یک کمان ورودی به آن انتخاب شود.*
- *اگر i گره‌ای غیر از مبدا و مقصد باشد و یکی از کمان‌های ورودی آن انتخاب شود، برای برقراری جریان حتما باید یکی از کمان‌های خروجی آن نیز انتخاب شود.*
- *مدت زمان طی مسیر نباید از مقدار مشخص شده λ تجاوز کند.*

هدف مسئله یافتن کوتاه‌ترین مسیر بین مبدا و مقصد مشخص شده است.

با درنظر گرفتن شبکه $N=(V,A)$ و نمادهای تعریف‌شده و مفروضات مسئله، مدل ریاضی مسئله به فرم زیر خواهد بود.

$$CSP: min \sum_{i} \sum_{j|(i,j)\in A} d_{ij}\, x_{ij}, \tag{۱}$$

$subject\ to:$

$$\sum_{j|(1,j)\in A} x_{1j} - \sum_{j|(j,1)\in A} x_{j1} = 1, \tag{۲}$$

$$\sum_{j|(j,n)\in A} x_{jn} - \sum_{j|(n,j)\in A} x_{nj} = 1, \tag{۳}$$

$$\sum_{i|(i,j)\in A} x_{ij} = \sum_{k|(j,k)\in A} x_{jk}, \quad \forall j \in V - \{1,n\}, \tag{٤}$$

$$\sum_{i} \sum_{j|(i,j)\in A} t_{ij}\, x_{ij} \le \lambda, \tag{۵}$$

$$x_{ij} \in \{0,1\}, \qquad \forall (i,j) \in A. \tag{٦}$$

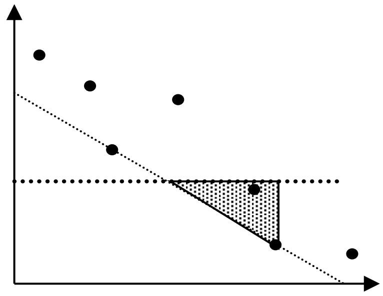
اگر در مدل مسئله، کوتاه‌ترین مسیر مقید (CSP) قید ظرفیت (۵) نادیده گرفته شود مدل مسئله کوتاه‌ترین مسیر کلاسیک به‌دست می‌آید. ساختار منحصربه‌فرد این مدل، که در آن همه‌ی ضرایب در محدودیت ها ۰، ۱ یا ۱- هستند، باعث می‌شود که خروجی جواب مدل آزادشده نیز عدد صحیح باشد. اما با اضافه‌کردن قید ظرفیت به مسئله، خروجی مدل آزادشده لزوما عدد صحیح نمی‌باشد. اگر جواب مدل آزادشده مسئله‌ی کوتاه‌ترین مسیر مقید ($RCSP$) غیرصحیح باشد، باتوجه به قیود ۲ تا ٤، خروجی مدل آزادشده به‌گونه‌ای است که یک واحد جریان از مبدا خارج شده و با تقسیم روی دو یا چند مسیر مختلف به مقصد می‌رسد. به عبارت دیگر خروجی مدل آزادشده، چند مسیر را مشخص می‌کند که روی هرکدام جریانی به‌اندازه $0 < \alpha_l < 1$ عبور می‌کند. اگر P مسیری از مبدا به مقصد باشد مسافت آن را با $D(P)$ و

نشریه تصمیم‌گیری و تحقیق در عملیات، دوره ٤، شماره (۳)، پاییز ۱۳۹۸، صفحه: ۲۲۰-۲۰۹

مدت زمان حرکت روی آن‌را با $T(P)$ نشان می‌دهیم. مسیر P یک مسیر شدنی برای مسئله‌ی کوتاه‌ترین مسیر مقید است اگر $T(P) = \sum_{(i.j)\in P} t_{ij} \leq \lambda$.

۳- ارائه‌ی الگوریتمی برای حل مسئله

در الگوریتم‌های مهم ارائه‌شده برای حل مسئله‌ی کوتاه‌ترین مسیر مقید ابتدا یک جواب اولیه شدنی یا نشدنی پیدا می‌شود و سپس با استفاده از روش‌های مختلف جواب‌های نامطلوب را حذف کرده و به سمت جواب بهین حرکت می‌کنند. در روش ترکیباتی kامین مسیر کوتاه، ابتدا کوتاه‌ترین مسیر به‌دست آمده و درصورتی که این مسیر در قید ظرفیت صدق نکند به‌سراغ دومین مسیرکوتاه می‌روند و این روند را تا رسیدن به اولین جواب شدنی ادامه می‌دهند. برای مثال اگر شبکه‌ای شامل چند مسیر از مبدا به مقصد، با مشخصات مسافت و زمان، مشابه آن‌چه که در **شکل ۱** آمده است وجود داشته باشد و روش kامین مسیر کوتاه روی آن پیاده‌سازی شود، در تکرار پنجم، مسیر بهینه‌ی *P5* به‌دست می‌آید. هم‌چنین در روش‌های مبتنی‌بر روش آزادسازی لاگرانژ، که در آن قید ظرفیت با درنظر گرفتن جریمه‌ای به تابع هدف منتقل می‌شود، ابتدا یک جواب شدنی به‌دست می‌آید و سپس با استفاده از الگوریتم kامین مسیر کوتاه یا دیگر روش‌های مشابه، برای ازبین بردن گپ موجود به حذف جواب‌های غیربهینه می‌پردازند. اگر این الگوریتم برای شبکه‌ی متناظر **شکل ۱** به‌کار گرفته شود، ابتدا جواب *P6* به‌عنوان کاندید جواب بهین مشخص شده و سپس با حذف مسیرهای *P1*، *P2*، *P3*، *P6* و *P7* جواب بهین به‌دست می‌آید. در این روش‌ها سرعت همگرایی به مقدار پارامتر جریمه و تراکم جواب‌های شدنی و نشدنی وابسته است (سانتوس و همکاران، ۲۰۰۷).



شکل ۱- مثالی از نمودار مسافت-زمان مسیرهای مختلف از مبدا تا مقصد.

در روش حل جدید پیشنهادی این مقاله به‌جای استفاده از آزادسازی لاگرانژ از مدل عدد صحیح آزادشده مسئله و برش‌های منطقی استفاده شده است. در ابتدا یک جواب شدنی برای مسئله پیدا کرده و سپس با استفاده از برش‌هایی که به مدل اضافه می‌شود مسیرهای دیده‌شده را حذف کرده تا به مسیرهای بهتری برسیم. به این‌منظور، ابتدا مدل *RCPS* را حل می‌کنیم؛ اگر جواب به‌دست آمده صحیح باشد که به جواب بهین رسیده‌ایم، در غیر این‌صورت برخی از متغیرها مقدار غیرصفر از بازه‌ی *[*۰و*1]* اختیار می‌کنند و بقیه‌ی متغیرها مقدار صفر. هر متغیر نماینده‌ی یکی از کمان‌های شبکه است و با دنبال‌کردن کمان‌هایی که مقدار غیرصفر گرفته‌اند می‌توان مسیرهای مختلفی از مبدا تا مقصد را شناسایی کرد (کمان‌هایی که مقدار ۱ می‌گیرند روی مسیرهای به‌دست آمده مشترک هستند). پس خروجی این مدل دو یا چند مسیر خواهد بود که روی هرکدام جریانی بین صفر و یک می‌گذرد و مقدار تابع هدف کران پایین جواب مسئله *CSP* خواهد بود. فرض کنید یک واحد جریان از مبدا خارج شده و روی m مسیر $P_1, P_2, \dots, P_m$ به مقصد می‌رسد. اگر روی مسیر P_l جریانی به اندازه‌ی $0 < \alpha_l < 1$ عبور کند آن‌گاه

$$T(P_l) = \sum_{(i,j)\in P_l} t_{ij}\, x_{ij} = \sum_{(i,j)\in P_l} \alpha_l t_{ij}, \quad (۷)$$

$$\sum_{l=1}^{m} T(P_l) = \sum_{l=1}^{m} \Big(\sum_{(i,j)\in P_l} \alpha_l t_{ij} \Big) \le \lambda, \quad (۸)$$

$$\sum_{l=1}^{m} \alpha_l = 1. \quad (۹)$$

باتوجه به نامعادله‌ی (۸) حداقل یکی از مسیرهای به‌دست آمده شدنی خواهد بود و مسافت آن یک کران بالا برای جواب مسئله *CSP* خواهد بود. این مسیر را به‌عنوان کاندید جواب بهین ذخیره می‌کنیم تا زمانی که به جواب شدنی کوتاه‌تری برسیم. در ادامه با اضافه‌کردن یک برش منطقی، می‌توان جواب فعلی را که معرف چند مسیر مختلف است حذف کرده و به مسیر یا مسیرهای جدیدی رسید که ممکن است یکی از آن‌ها مسیر شدنی کوتاه‌تری باشد و جایگزین کاندید جواب بهین شود. باید توجه کرد که با اضافه‌کردن برش منطقی به مدل، جواب فعلی نشدنی خواهد بود و می‌توان با استفاده از روش سیمپلکس دوگان جواب جدیدی به‌دست آورد. چون فضای شدنی مسئله هر بار به‌دلیل برش اضافه‌شده محدودتر می‌شود، مقدار تابع هدف مدل آزادشده افزایش می‌یابد تا این‌که به کران بالای ذخیره‌شده برسد. در این حالت جواب بهین مسئله *CSP* همان کران بالای ذخیره‌شده می‌باشد. اگر این الگوریتم برای شبکه‌ی متناظر **شکل ۱** به‌کار گرفته شود، ابتدا جوابی به‌صورت ترکیبی از مسیرهای *P3* و *P6* به‌دست می‌آید. چون مسیر *P6* شدنی است به‌عنوان کاندید جواب بهین ذخیره می‌شود. سپس با اضافه‌کردن قیدی به شکل $\sum_{i\in P3\cup P6}\sum_{j|(i,j)\notin P3\cup P6} x_{ij} \ge 1$، مسیرهای جدیدی به‌دست می‌آید. در تکرارهای بعدی، ترکیبی از مسیر *P5* با مسیرهای دیگر به‌دست می‌آید و چون مسافت این مسیر از *P6* کم‌تر است به عنوان کاندید جواب بهین جایگزین *P6* می‌شود. در هرمرحله چون فضای شدنی مسئله محدودتر می‌شود جواب بهین مسئله‌ی آزادشده به‌سمت جواب بهین مسئله‌ی عدد صحیح حرکت می‌کند و زمانی که جواب بهین مدل آزادشده بزرگ‌تر یا مساوی کران بالای ذخیره شده یعنی *D(P5)* شد، توقف می‌کنیم.

فرم کلی برش‌های منطقی در مدل‌های دودویی به این‌صورت است که یک جواب غیربهینه و متناظر با آن یک دنباله‌ی صفر و یک برای متغیرهای دودویی در دست داریم و می‌خواهیم چنین جوابی را از فضای جواب شدنی حذف کنیم تا به جواب دیگری برسیم و لذا دنباله‌ی متغیرهای دودویی مسئله باید چنان مقدار بگیرند که با دنباله‌ی موجود تفاوت داشته باشد. درنتیجه حداقل یکی از متغیرهایی که در جواب فعلی مقدار صفر دارد باید مقدار یک اختیار کند یا یکی از متغیرهایی که مقدار یک داشته باید صفر شود. بنابراین اگر فرض کنیم در مدل *CSP* متغیرها مقدار $x_{i,j}^{o}$ گرفته‌اند فرم کلی این برش به‌صورت زیر خواهد بود:

$$\sum_{(i,j)|x_{i,j}^{o}=0} x_{ij} + \sum_{(i,j)|x_{i,j}^{o}=1} (1 - x_{ij}) \ge 1 \quad (۱۰)$$

چنین برشی فقط یک جواب را از فضای شدنی مسئله حذف می‌کند اما همان‌طور که در کارهای (ورستیکل و همکاران، ۲۰۱۵؛ جندرون و همکاران، ۲۰۱۳) اشاره شده است، اگر بتوان به‌صورت منطقی مجموعه اندیس‌هایی که در این قید به‌کار رفته است را کوچک‌تر کرد، برش قوی‌تری حاصل می‌شود و می‌تواند تعداد بیش‌تری جواب غیر بهین را حذف کند. هم‌چنین باتوجه به شرایط مسئله ممکن است نیازی به نوشتن هر دو جمله‌ی نامعادله (۱۰) نباشد. اما این کار باید به‌گونه‌ای انجام شود که قید منطقی اضافه‌شده علاوه‌بر این‌که باعث می‌شود جواب فعلی مسئله دوباره حاصل نشود و برخی از جواب‌های غیربهین دیگر نیز تولید نشود اما موجب حذف و نادیده گرفتن جواب بهین مسئله نشود. برای ایجاد برش منطقی پیشنهادی در این مقاله به نکاتی که در ادامه می‌آید باید توجه کرد.

نشریه تصمیم‌گیری و تحقیق در عملیات، دوره ٤، شماره (۳)، پاییز ۱۳۹۸، صفحه: ۲۲۰-۲۰۹

فرض کنید با حل مدل $RCSP$ جواب غیر صحیحی به‌دست بیاید که معرف مسیرهای $P_1, P_2, \dots, P_m$ باشد (به این مسیرها، مسیرهای دیده‌شده می‌گوییم) و بخواهیم با حذف آن و تولید جواب جدیدی به مسیرهای دیگری برسیم. ابتدا همه‌ی گره‌ها و کمان‌های روی این m مسیر را به‌ترتیب در دو مجموعه‌ی O و Q ذخیره کرده و مشخصات همه‌ی مسیرهای دیده‌شده شامل مسافت و مدت زمان آن‌ها را بررسی کرده و پس از به‌روز رسانی کران بالا، با اضافه‌کردن یک برش منطقی جواب فعلی را حذف کرده و به جواب جدیدی می‌رسیم. در جواب جدید، حداقل باید یک کمان دیده‌نشده مقدار غیرصفر بگیرد تا بتوان به مسیر جدیدی دست یافت. هر کمان جدید روی برخی از مسیرهای قبلی انشعاب درست می‌کند که با دنبال‌کردن آن‌ها یک یا چند مسیر جدید از مبدا تا مقصد به‌دست می‌آید. اگر پس‌از به‌روز رسانی، کران بالا از کران پایین به‌دست آمده بزرگ‌تر باشد شرط توقف برقرار نشده است و باید این روند را ادامه داد. در تکرارهای بعد با اضافه‌کردن کمان‌های جدید به‌دست آمده به کمان‌های دیده‌شده‌ی قبلی، مجموعه‌های O و Q را به‌روز رسانی کرده و براساس آن برش جدیدی اضافه می‌کنیم. در زمان به‌روز رسانی مجموعه‌های O و Q اعضای قبلی حفظ و اعضای جدید دیده‌شده نیز اضافه می‌شود و لذا با هر تکرار، مجموعه‌ی Q بزرگ‌تر می‌شود اما مجموعه‌ی O ممکن است ثابت بماند یا بزرگ‌تر شود. برش منطقی پیشنهادی به‌صورت زیر است:

$$\sum_{i\in O}\sum_{j|(i,j)\notin Q} x_{ij} \geq 1. \qquad (۱۱)$$

این برش به این معنا است که از یکی از گره‌هایی که قبلا در مسیرهای به‌دست آمده مشاهده شده است باید یک کمان جدید خارج شود که روی هیچ‌یک از مسیرهای دیده‌شده‌ی قبلی نبوده است (متغیر x_{ij} متناظر آن در همه‌ی تکرارهای قبلی صفر بوده است و گره i دیده‌شده اما کمان (i,j) دیده نشده است).

برای اعتبارسنجی یک برش منطقی باید داشتن دو مشخصه را برای آن بررسی کرد: اول این‌که با اضافه‌کردن آن، جواب فعلی دیگر تولید نشود دوم این‌که جواب بهین مسئله نادیده گرفته نشود. با اضافه‌کردن قید (۱۱)، باتوجه به تعریف مجموعه‌های O و Q، حداقل یک متغیری که در تمام تکرارهای قبلی مقدار صفر داشته است مقدار غیرصفر می‌گیرد پس جواب فعلی مطمئنا تولید نمی‌شود و جواب جدیدی حاصل می‌شود. حال به اثبات مشخصه‌ی دوم می‌پردازیم. در تکرار اول، مدل $RCSP$ بدون برش منطقی اجرا شده و m مسیر مختلف از مبدا به مقصد مشخص می‌شود. اگر یکی از این مسیرها جواب بهین مسئله CSP باشد مشخصات آن به‌عنوان کاندید جواب بهین ذخیره می‌شود در غیر این‌صورت جواب بهین مسیری غیر از این m مسیر دیده‌شده است و حداقل یکی از کمان‌های مسیر بهین مقدار صفر گرفته است و عضو مجموعه‌ی Q نیست و باید در تکرارهای بعد مقدار بگیرد و اضافه‌کردن قید (۱۱) نیز باعث می‌شود که حداقل یک کمان که عضو مجموعه‌ی Q نیست مقدار بگیرد. فرض کنید کمان‌های مسیر بهینه به‌صورت $(1,i_1),(i_1,i_2),\dots,(i_k,n)$ از مبدا آغاز و به مقصد می‌رسند. هم‌چنین فرض کنید با دنبال‌کردن کمان‌های مسیر بهینه از مبدا، کمان (i_l,i_{l+1}) اولین کمانی از این مسیر باشد که عضو مجموعه‌ی Q نیست پس کمان قبلی مسیر عضو Q و گره l عضو مجموعه O می‌باشد (اگر l=1، کمان قبلی معنا ندارد اما گره ۱ که همان گره مبدا است حتما عضو مجموعه‌ی O هست چون روی همه مسیرهای دیده‌شده وجود دارد) و لذا با اضافه کردن قید (۱۱) می‌توان این کمان را به‌عنوان کمان جدید در تکرارهای بعد مشاهده کرد ($l \in O, (i_l,i_{l+1}) \notin Q$). با تکرار این روند همه‌ی کمان‌های مسیر بهینه عضو مجموعه‌ی Q شده و این مسیر به‌عنوان کاندید جواب بهین ذخیره خواهد شد. پس می‌توان نتیجه گرفت که اضافه‌کردن قید (۱۱) باعث حذف جواب بهین مسئله نمی‌شود.

حال الگوریتم جدید را برای حل مسئله‌ی کوتاه‌ترین مسیر مقید به‌شرح زیر ارائه می‌دهیم:

گام اول: مقدار کران بالا را برابر با $UB = \infty$ و قرار بده $O = Q = \emptyset$. مدل *RCSP* را حل کن. اگر مدل آزادشده نشدنی است مسئله‌ی اصلی نیز نشدنی است. اگر جواب به‌دست آمده صحیح است آن را به‌عنوان جواب بهین معرفی کن و توقف کن؛ در غیر این‌صورت به گام دوم برو.

گام دوم: با دنبال‌کردن کمان‌های جدید و باتوجه به انشعاب‌های به‌وجود آمده روی مسیرهای مختلف، همه مسیرهای جدید را استخراج کرده و مقادیر *D(P)* و *T(P)* را برای هریک از آن‌ها محاسبه کن. اگر $(T(P) \leq \lambda \,\&\, D(P) < UB)$ مسیر P را به‌عنوان کاندید جواب بهین جایگزین کن و قرار بده $UB = D(P)$. مجموعه‌های O و Q را به‌ترتیب با مجموعه‌های $O \cup \{i \in N \mid \sum_{j|(i,j)\in A} x_{ij} > 0\}$ و $Q \cup \{(i,j) \in A | x_{ij} > 0\}$ جایگزین کن و به گام سوم برو.

گام سوم: برش منطقی (۱۱) را به مدل *RCSP* اضافه کن و مدل جدید را با استفاده از سیمپلکس دوگان حل کن. اگر مسئله نشدنی است کاندید جواب بهین ذخیره‌شده را به‌عنوان جواب بهین گزارش کن و الگوریتم را متوقف کن. اگر جواب به‌دست آمده صحیح است آن‌را به‌عنوان جواب بهین معرفی کن و توقف کن. اگر $Z_{LP}^{*} \geq UB$ کاندیدای جواب بهین را به‌عنوان جواب بهین معرفی کن و توقف کن؛ در غیر این صورت به گام دوم برو.

اگر مسئله‌ی *CSP* نشدنی باشد در همان تکرار اول مشخص می‌شود و اگر جواب داشته باشد در تکرار اول یک کران بالا و یک کران پایین برای آن به‌دست می‌آید. چون در هر تکرار، حداقل یک متغیر جدید مقدار می‌گیرد، پس این الگوریتم در دور قرار نمی‌گیرد. از طرفی در هر تکرار، فضای شدنی مسئله محدودتر شده و لذا کران پایین به‌سمت بالا حرکت می‌کند تا به آن برسد و لذا این الگوریتم در تعداد متناهی تکرار به‌اتمام می‌رسد. تا زمانی که همه‌ی کمان‌های مسیر بهینه‌ی *CSP* عضو Q نشوند اضافه‌کردن برش (۱۱) موجب نشدنی‌شدن مسئله‌ی آزاد شده نمی‌شود (چون کمان‌های دیده‌نشده مسیر بهین در این قید صدق می‌کنند). در مدل آزادشده هر جوابی که به‌دست می‌آید معرف ترکیبی است از مسیرهای شدنی مسئله‌ی اصلی (با مسافت بیش‌تر) و نشدنی (با مسافت کم‌تر) و مقدار تابع هدف به‌دست آمده ترکیب خطی مسافت این مسیرهاست. در برخی موارد ممکن است حالتی پیش بیاید که همه کمان‌های مسیر بهینه دیده شده‌اند اما کران پایین مسئله به کران بالا نرسیده است ولی در تکرار بعد با به‌روز رسانی برش (۱۱)، که فضای شدنی را محدودتر می‌سازد، مسئله نشدنی شود در چنین حالتی، همه‌ی مسیرهای شدنی مسئله ی اصلی در تکرارهای قبل دیده شده است و با برش منطقی حذف شده‌است و دیگر ترکیبی از مسیرهای شدنی و نشدنی برای مسئله‌ی آزادشده وجود ندارد.

شبکه‌ی نمایش داده شده در **شکل ۲** شامل ٦ گره و ۹ کمان است و پارامترهای مسافت و مدت‌زمان برای هر کمان به‌صورت $[d_{ij}, t_{ij}]$ نشان داده‌شده است. در مجموع، ۷ مسیر از گره ۱ به گره ٦ وجود دارد. با درنظر گرفتن $\lambda = 40$ و پیاده‌سازی الگوریتم، در مرحله‌ی اول دو مسیر *P1: 1-2-4-5-6* و *P2: 1-3-4-5-6* به‌دست می‌آید که به‌ترتیب ۰٫۷۵ و ۰٫۲۵ واحد جریان روی آن‌ها تقسیم می‌شود. چون $T(P2) = 11 \leq \lambda$ پس مسیر *P2* شدنی است و به‌عنوان کاندید جواب بهین و کران بالا به‌صورت $UB = D(P2) = 55$ ذخیره می‌شود. گره‌ها و کمان‌های مشاهده‌شده به‌ترتیب در مجموعه‌های O و Q ذخیره می‌شوند ($O = \{1,2,3,4,5\}$ و $Q = \{(1,2),(1,3),(2,4),(3,4),(4,5),(5,6)\}$) و قید (۱۱) به‌صورت $x_{32} + x_{35} + x_{46} \geq 1$ به مدل *RCSP* اضافه می‌شود.

در تکرار دوم، سه مسیر *P3: 1-2-4-6* و *P4: 1-3-5-6* و *P5: 1-3-4-6* به‌دست می‌آید که در بین آن‌ها مسیر *P5* یک مسیر شدنی با طول ٤٥ واحد است و چون از مسیر *P2* کوتاه‌تر است به‌عنوان کاندید جواب بهین ذخیره شده و کران بالا به‌صورت $UB = D(P5) = 45$ به‌روز رسانی می‌شود. چون مقدار تابع هدف برابر است با $Z_{LP}^{*} = 41.64$ و کم‌تر از کران بالاست، شرط توقف برقرار نیست. گره‌ها و کمان‌های جدید به‌ترتیب به مجموعه‌های O و Q افزوده می‌شود ($O = \{1,2,3,4,5\}$ و $Q = \{(1,2),(1,3),(2,4),(3,4),(3,5),(4,5),(4,6),(5,6)\}$) و قید (۱۱) به‌صورت $x_{32} \geq 1$ به‌روز رسانی

نشریه تصمیم‌گیری و تحقیق در عملیات، دوره ٤، شماره (۳)، پاییز ۱۳۹۸، صفحه: ۲۲۰-۲۰۹

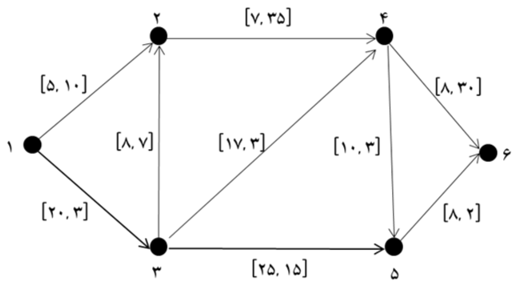
می‌شود. در تکرار سوم، مسئله نشدنی است و لذا کاندید جواب بهین ذخیره‌شده یعنی *P5*، به‌عنوان جواب بهین معرفی می‌شود.

۴- پیاده‌سازی الگوریتم

برای بررسی عملکرد الگوریتم ارائه‌شده، آن را بر روی تعدادی شبکه با ساختارهای متفاوت و با نمونه داده‌های مختلف که به‌صورت تصادفی تولید می‌شوند پیاده‌سازی می‌کنیم. دو ساختار برای شبکه‌ها درنظر می‌گیریم ساختار مشبک (شطرنجی) و ساختار تصادفی. در ساختار مشبک فرض می‌کنیم که هر گره با هشت گره دیگر و در جهت‌های اصلی و فرعی جغرافیایی مرتبط است. در ساختار تصادفی برای هر گره مجموعه‌ای از چند گره درنظر گرفته می‌شود و به‌صورت تصادفی تعیین می‌شود که بین هر دو گره مجزا کمانی وجود دارد یا خیر. برای هر شبکه مقدار هرکدام از پارامترهای مسافت و مدت‌زمان نیز به‌صورت تصادفی و یکنواخت تعیین می‌شود. همه‌ی محاسبات بر روی سیستمی با مشخصات پردازشگر *Intel 2-GHz*، *2-GB RAM* و *solver CPLEX 12.4.* و با استفاده از نسخه‌ی ۳٫۱۲ نرافزار *AIMMS* پیاده‌سازی می‌شود. معیار خاتمه نیز دو ساعت پردازش یا رسیدن به جواب بهین می‌باشد.

۴-۲- پیاده‌سازی بر روی شبکه‌ی مشبک یا شطرنجی

فرض کنید تعداد *n* گره به‌صورت چند سطر و ستون چیده شده‌اند و از هر گره در جهت‌های اصلی و فرعی جعرافیایی به هشت گره مجاور دیگر کمان وجود دارد. گره شماره ۱ را گره مبدا و گره شماره *n* را گره مقصد درنظر می‌گیریم. ابتدا بر روی یک مورد شبکه‌ی معرفی‌شده جزییات اجرای الگوریتم توضیح داده می‌شود و سپس با درنظر گرفتن شبکه‌هایی با اندازه‌های مختلف و تولید نمونه داده‌هایی به‌صورت تصادفی، الگوریتم پیشنهادی را برای هر نمونه پیاده‌سازی کرده و میانگین تعداد تکرارها و مدت زمان رسیدن به جواب بهین مسئله مورد بررسی قرار می‌گیرد.



شکل۲- مثالی از یک شبکه با دو پارامتر مسافت و مدت‌زمان طی مسیر.

ابتدا یک شبکه مشبک ۲۰×۲۰ با ۴۰۰ گره درنظر گرفته شده و مقدار پارامترهای مسافت و مدت‌زمان برای هر کمان به‌صورت تصادفی به‌ترتیب از بازه‌های *[۱,۲۰]* و *[۰,۱۰]* با توزیع یکنواخت تولید واستخراج می‌شود. سقف قید ظرفیت برابر با $\lambda = 100$ قرار داده شده و الگوریتم پیاده‌سازی می‌شود. نتایج حاصل از به‌کارگیری الگوریتم پیشنهادی، در **جدول ۱** آمده است. در ستون دوم این جدول مقدار تابع هدف مدل خطی حاصل از اضافه‌کردن برش منطقی *RCSP*، به‌دست آمده است که کران پایین جواب بهین مدل *CSP* است. در ستون سوم مسافت کوتاه‌ترین مسیر شدنی به‌دست آمده به‌عنوان کران بالای جواب بهین نشان داده‌شده است. همان‌طور که در این جدول نشان داده‌شده است، در تکرار اول، مدل *RCSP* حل شده که جواب آن به‌صورت غیرصحیح و مقدار تابع هدف به‌دست آمده برابر است با ۱۰۶٫۹۳ که این مقدار کران پایین جواب بهین مدل *CSP* می‌باشد. در تکرار اول، دو مسیر مجزا مشخص شده که یکی از آن‌ها یک مسیر شدنی با مسافت ۱۲۰ واحد و مدت‌زمان ۸۶ واحد می‌باشد. این مسیر به‌عنوان کاندید جواب بهین و مسافت آن به‌عنوان کران بالای جواب بهین ذخیره می‌شود. مجموعه‌گره‌ها و کمان‌های دیده‌شده در این تکرار به‌ترتیب دارای ۲۹ و ۳۰ عضو است. در تکرار دوم برش منطقی (۱۱) به مدل *RCSP* اضافه می‌شود. در این تکرار نیز جواب به‌صورت غیرصحیح

است و دو مسیر جدید را معرفی می‌کند که یکی از آن‌ها یک مسیر شدنی با مسافت ۱۱۳ واحد و مدت‌زمان ۹٤ واحد می‌باشد؛ چون مسافت این مسیر شدنی از کران بالای موجود کم‌تر است پس کران بالا به‌صورت $UB = 113$ تغییر می‌کند. تعداد گره‌ها و کمان‌های جدید دیده‌شده در این تکرار به‌ترتیب ۱۰ و ۱۲ است و با اضافه‌کردن آن‌ها مجموعه‌های O و Q و براساس آن‌ها برش (۱۱) به‌روز می‌شود. در تکرار سوم یک مسیر شدنی به‌دست می‌آید که مسافت آن از کران بالای ذخیره‌شده بیش‌تر است و جایگزین کاندید جواب بهین نمی‌شود. در تکرار بعدی، مسیری شدنی با مسافت ۱۱۲ واحد به‌دست می‌آید و جایگزین کاندید جواب بهین می‌شود. با ادامه‌ی این روند و پس‌از ۸ تکرار، مقدار تابع هدف مسئله به کران بالای ذخیره‌شده می‌رسد و جواب بهین مسئله که مسیری با مسافت ۱۱۰ واحد و مدت‌زمان ۱۰۰به‌دست می‌آید. تمام این فرآیند در مدت‌زمان ۰/۰۸ ثانیه به‌انجام می‌رسد. همان‌طور که در ستون‌های ۲-٤ **جدول ۱** مشاهده می‌شود، در هر تکرار مقدار گپ بین کران بالا و پایین کم‌تر می‌شود، زیرا از طرفی در هر تکرار فضای شدنی مدل آزادشده محدودتر می‌شود و مقدار تابع هدف افزایش می‌یابد و از طرف دیگر در برخی از تکرارها جواب شدنی بهتری به‌دست می‌آید و کران بالا کوچکتر می‌شود. نکته‌ی دیگر این است که تعداد محدودیت‌ها و متغیرها در هر تکرار بدون تغییر می‌ماند؛ آن‌چه که باعث کوچک‌تر شدن فضای جواب شدنی می‌شود تغییر مجموعه اندیس‌های برش منطقی اضافه شده است (در هر تکرار اندازه‌ی مجموعه‌ی Q بزرگ‌تر می‌شود و امکان انتخاب کمان جدید محدودتر می‌شود).

در ادامه شبکه‌های مشبکی با ۴۰۰،۹۰۰، ۱۶۰۰، ۲۵۰۰ و ۳۶۰۰ گره درنظر گرفته شده و آن‌ها را به‌ترتیب با *GN1-GN5* نمایش می‌دهیم و برای هر کدام ۱۰۰ نمونه‌داده‌ی تصادفی تولید کرده و به‌صورت جداگانه، الگوریتم برای هر کدام پیاده‌سازی می‌شود. نتایج به‌دست آمده شامل میانگین تعداد تکرارها و میانگین و ماکزیمم مدت‌زمان اجرا، برای هر مورد در جدول ۲ گزارش شده است. همان‌طور که مشاهده می‌شود با بزرگ‌ترشدن اندازه‌ی شبکه‌ی میانگین تعداد تکرارها و مدت زمان حل افزایش محسوسی ندارد. توجیهی که برای این قضیه می‌توان داشت این است که با بزرگ‌ترشدن مسئله ممکن است فاصله‌ی کران پایین و بالا زیاد شود و نیاز به تکرارهای بیش‌تری باشد اما چون تعداد مسیرهای ممکن هم زیاد می‌شود احتمال این‌که جواب بهین به کران پایین نزدیک باشد نیز وجود دارد.

جدول ۱- نتایج حاصل از اجرای الگوریتم پیشنهادی بر روی یک شبکه‌ی مشبک با ۴۰۰ گره.

تکرار	کران پایین	کران بالا	گپ (%)	تعداد متغیرها	تعداد قیود	$\|O\|$	$\|Q\|$
۱	۱۰۶/۹۳	۱۲۰	۱۲/۲	۲۹۶۴	۴۰۱	۲۹	۳۰
۲	۱۰۷/۸۶	۱۱۳	۴/۸	۲۹۶۴	۴۰۲	۳۹	۴۲
۳	۱۰۸/۱۲	۱۱۳	۴/۵	۲۹۶۴	۴۰۲	۵۲	۵۹
۴	۱۰۸/۵۷	۱۱۲	۳/۱	۲۹۶۴	۴۰۲	۵۲	۶۲
۵	۱۰۸/۸۶	۱۱۲	۲/۸	۲۹۶۴	۴۰۲	۵۵	۶۷
۶	۱۰۹/۱	۱۱۲	۲/۶	۲۹۶۴	۴۰۲	۶۱	۷۸
۷	۱۰۹/۵	۱۱۰	۰/۴	۲۹۶۴	۴۰۲	۶۳	۸۲
۸	۱۱۰/۱۵	۱۱۰	-	۲۹۶۴	۴۰۲	-	-

٤-۲- پیاده‌سازی بر روی شبکه‌ای با ساختار تصادفی

در این بخش مشابه بخش قبلی تعداد مشخصی گره درنظر گرفته می‌شود اما ساختار شبکه به‌صورت تصادفی تعیین می‌شود؛ برای هر گره چند گره همسایه درنظر گرفته شده و به‌صورت تصادفی تعیین می‌شود که کمانی از این گره به هر کدام از گره‌های همسایه وجود دارد یا خیر. ابتدا برای هر گره ١٦ گره همسایه درنظر گرفته شده و سپس به‌صورت تصادفی وجود کمان بین گره‌های همسایه تعیین می‌شود. تعداد کمان‌ها، هم‌چنین مقادیر پارامترها مشابه قسمت قبلی به‌صورت تصادفی تعیین شده و برای هر شبکه ۱۰۰ نمونه‌داده‌ی تصادفی تولید می‌شود و سپس الگوریتم به‌صورت جداگانه برای هر نمونه پیاده‌سازی می‌شود. نتایج پیاده‌سازی الگوریتم بر روی ۱۰۰ نمونه‌داده برای شبکه‌های مختلف با

نشریه تصمیم‌گیری و تحقیق در عملیات، دوره ٤، شماره (۳)، پاییز ۱۳۹۸، صفحه: ۲۲۰-۲۰۹

تعداد گره ٤٠٠، ٩٠٠، ١٦٠٠، ٢٥٠٠ و ٣٦٠٠ و با ساختار تصادفی که آن‌ها را به‌ترتیب با *RN1-RN5* نشان می‌دهیم که در **جدول ۳** گزارش شده است. باتوجه به داده‌های این جدول مشاهده می‌شود که عملکرد الگوریتم شامل میانگین تعداد تکرار و میانگین حداکثر مدت‌زمان اجرا عملکرد الگوریتم برای شبکه‌های مختلف مشابه بخش قبلی است و تعداد تکرارها و مدت‌زمان اجرای الگوریتم با بزرگ‌ترشدن اندازه‌ی شبکه‌ها تغییر چشم‌گیری نمی‌کند.

برای مقایسه‌ی عملکرد این الگوریتم با روش‌های موجود توجه شما را به این نکته جلب می‌کنم که مدت‌زمان اجرای الگوریتم‌های موجود بر پایه‌ی آزادسازی لاگرانژ علاوه‌بر تعداد کمان‌های شبکه به مقدار درنظر گرفته شده برای جریمه‌ی تابع هدف نیز وابسته است به همین دلیل در مطالعه‌ی انجام‌گرفته توسط سانتوس و همکاران (۲۰۰۷) که سه الگوریتم مهم در این زمینه را موردبررسی قرارداده است آن‌ها را بر روی شبکه‌های مختلف با اندازه‌های مختلف و به‌ازای مقادیر مختلف جریمه در تابع هدف پیاده‌سازی کرده است. در این مقایسات مشاهده می‌شود که تعداد تکرارهای الگوریتم و میانگین زمان اجرای الگوریتمی واحد بر روی شبکه‌هایی با اندازه‌ی یکسان به‌ازای ۵ مقدار مختلفی که برای جریمه تابع هدف درنظر گرفته‌شده است فرق می‌کند. به همین دلیل مقایسه‌ی مورد به مورد اجرای الگوریتم پیشنهادی ما که وابسته به مقدار جریمه تابع هدف نیست منطقی به‌نظر نمی‌رسد. با این وجود با مشاهده‌ی زمان اجرای الگوریتم‌های پیشنهادی بر روی شبکه‌هایی با اندازه‌ی ۲۰ تا ۳۰ هزار کمان نشان می‌دهد که میانگین زمان اجرای این الگوریتم‌ها از ۰/٦ تا ۱/۷ ثانیه و بیش‌ترین زمان اجرای آن‌ها از ۰/۷ تا ۷/٦ ثانیه تفاوت می‌کند و عملکرد الگوریتم پیشنهادی ما نیز مشابه همین مدت زمان‌های اجراست.

جدول ۲- نتایج حاصل از اجرای الگوریتم پیشنهادی بر روی نمونه‌های مختلف شبکه‌ی مشبک.

شبکه	$\|V\|$	$\|A\|$	میانگین تعداد تکرار	میانگین مدت زمان اجرا (ثانیه)	بیشترین زمان اجرا (ثانیه)
GN1	۴۰۰	۲۹۶۴	۱۴	۰/۰۷	۰/۱۳
GN2	۹۰۰	۶۸۴۴	۱۶	۰/۱	۰/۲۳
GN3	۱۶۰۰	۱۲۳۲۴	۲۵	۰/۳	۰/۷
GN4	۲۵۰۰	۱۹۴۰۴	۴۹	۰/۷	۱
GN5	۳۶۰۰	۲۸۰۸۴	۷۸	۰/۹	۱/۶

جدول ۳- نتایج حاصل از اجرای الگوریتم پیشنهادی بر روی نمونه‌های مختلف شبکه‌ی تصادفی.

شبکه	$\|V\|$	$\|A\|$	میانگین تعداد تکرار	میانگین مدت زمان اجرا (ثانیه)	بیش‌ترین زمان اجرا (ثانیه)
RN1	۴۰۰	۳۷۶۱	۱۴	۰/۰۶	۰/۱۹
RN2	۹۰۰	۶۳۲۸	۱۸	۰/۱۲	۰/۲۱
RN3	۱۶۰۰	۱۳۱۲۹	۲۴	۰/۶۸	۰/۹
RN4	۲۵۰۰	۲۱۶۶۱	۵۱	۰/۹۸	۱/۳
RN5	۳۶۰۰	۳۱۵۹۳	۸۳	۱/۲	۱/۸

۵- نتیجه‌گیری

در این مقاله مسئله‌ی کوتاه‌ترین مسیر مقید مورد بررسی قرار گرفت و برای حل آن یک الگوریتم حل تکراری براساس مدل آزادشده و اضافه‌کردن برش‌های منطقی ارائه شد. در این روش با حل مسئله‌ی آزادسازی شده یک جواب غیرصحیح به‌دست می‌آید که درواقع ترکیبی از دو یا چند مسیر شدنی و نشدنی می‌باشد. پس‌از بررسی مشخصات این مسیرها، کوتاه‌ترین جواب شدنی که به‌دست می‌آید به‌عنوان کاندید جواب بهین ذخیره شده و در تکرارهای بعدی سعی می‌شود با اضافه‌کردن یک برش منطقی، مسیرهای جدیدی تولید کنیم. در تولید برش‌های منطقی، که کاربرد زیادی در حل مسائل برنامه‌ریزی عدد صحیح دارند، براساس ساختار مسئله و منطق حاکم بر آن، قیدی به مسئله افزوده می‌شود که جواب فعلی و برخی از جواب‌های غیربهین دیگر را حذف می‌کند اما جواب بهین حذف نمی‌شود. در این مقاله با

استفاده از منطق حاکم بر مسئله، یک برش منطقی قوی پیشنهاد شد که با افزودن آن به مدل ریاضی مسئله می‌توان تعداد زیادی جواب غیربهینه را از فضای شدنی مسئله حذف کرد. سپس برای اثبات اعتبار آن دو مشخصه برش‌های منطقی، یعنی حذف جواب فعلی و نادیده نگرفتن جواب بهین، بررسی و اثبات شد. در هر تکرار فضای شدنی محدودتر شده و پس‌از چند تکرار کران بالا و پایین، مقدار تابع هدف به هم می‌رسند و الگوریتم در تعداد متناهی تکرار به جواب می‌رسد. این الگوریتم بر روی دو نوع شبکه با ساختار مشبک و تصادفی و در اندازه‌های مختلف پیاده‌سازی شد. نتایج حاصل از پیاده سازی الگوریتم بر روی نمونه‌های مختلف نشان داد که این الگوریتم قادر است مشابه روش‌های موجود در مدت‌زمان مناسبی به جواب بهینه مسئله دست یابد. نکته‌ی مثبت چنین روشی این است که برای پیاده‌سازی آن از روش‌های حل مدل‌های خطی موجود کمک گرفته می‌شود و با گذشت زمان و به‌روزر سانی نرم‌افزارهای حل مسائل بهینه‌سازی و ارتقا روش‌های حل مدل‌های خطی، کارآمدی این روش نیز بیش‌تر می‌شود. نکته‌ی دیگر این است که در هر تکرار فقط یک قید برای برش منطقی به مدل *RCSP* اضافه می‌شود (در هر تکرار فقط به‌روز رسانی می‌شود) و اندازه‌ی مدل تغییری نمی‌کند. این قید جدید جواب بهین تکرار قبلی را نشدنی می‌سازد اما با استفاده از الگوریتم سیمپلکس دوگان به‌راحتی به جواب بهین جدید می‌رسد و همان‌طور که در پیاده‌سازی‌های مختلف نشان‌داده شد با افزایش اندازه‌ی شبکه‌ها تغییر محسوسی در زمان اجرای الگوریتم رخ نمی‌دهد. نکته دیگر این است که می‌توان از اطلاعات مسیرهای به‌دست آمده در مراحل تکرار الگوریتم برای زمانی استفاده کرد که سقف قید ظرفیت تغییر کند، به این‌ترتیب که پس‌از تغییر ظرفیت، مشخصات همه‌ی این مسیرها بررسی و کران بالای مناسب انتخاب شده و سپس همه‌ی آن‌ها را با همان برش منطقی حذف کرد. در مطالعات مشابه آینده پیشنهاد می‌شود این مسئله در حالت عدم قطعیت داده‌ها مورد رررسی قرار گیرد و هم‌چنین می‌توان بر روی گسترش‌های دیگر مسئله و ارائه برش منطقی قوی‌تر کار کرد.

منابع

Ahuja, R. K., Magnanti, T. L., Orlin, J. B., & Weihe, K. (1995). Network flows: theory, algorithms and applications. *ZOR-methods and models of operations research, 41*(3), 252-254.

Cherkassky, B. V., Goldberg, A. V., & Radzik, T. (1996). Shortest paths algorithms: Theory and experimental evaluation. *Mathematical programming, 73*(2), 129-174.

Zhan, F. B., & Noon, C. E. (1998). Shortest path algorithms: an evaluation using real road networks. *Transportation science, 32*(1), 65-73.

Sedeño-Noda, A., & González-Martín, C. (2010). Shortest path simplex algorithm with a multiple pivot rule: a comparative study. *Asia-pacific journal of operational research, 27*(06), 677-691.

Lozano, L., & Medaglia, A. L. (2013). On an exact method for the constrained shortest path problem. *Computers & operations research, 40*(1), 378-384.

Garey, M. R. (1979). Computers and intractability: A guide to the theory of np-completeness. *Revista Da Escola De Enfermagem Da USP, 44*(2), 340.

Tarapata, Z. (2003). Military route planning in battlefield simulation: effectiveness problems and potential solutions. *Journal of telecommunications and information technology,* 47-56.

Mora, A. M., Merelo, J. J., Laredo, J. L. J., Millán, C., & Torrecillas, J. (2009). CHAC, A MOACO algorithm for computation of bi-criteria military unit path in the battlefield: Presentation and first results. *International journal of intelligent systems, 24*(7), 818-843.

MirHassani, S. A., Moradi, S., & Taghinezhad, N. (2011). Algorithm for long-term scheduling of multiproduct pipelines. *Industrial & engineering chemistry research, 50*(24), 13899-13910.

Avella, P., Boccia, M., & Sforza, A. (2004). Resource constrained shortest path problems in path planning for fleet management. *Journal of mathematical modelling and algorithms, 3*(1), 1-17.

Eppstein, D. (1998). Finding the k shortest paths. *SIAM journal on computing, 28*(2), 652-673.

Pugliese, L. D. P., & Guerriero, F. (2013). A survey of resource constrained shortest path problems: Exact solution approaches. *Networks, 62*(3), 183-200.

Handler, G. Y., & Zang, I. (1980). A dual algorithm for the constrained shortest path problem. *Networks, 10*(4), 293-309.

Santos, L., Coutinho-Rodrigues, J., & Current, J. R. (2007). An improved solution algorithm for the constrained shortest path problem. *Transportation research part B: methodological, 41*(7), 756-771.

Carlyle, W. M., Royset, J. O., & Kevin Wood, R. (2008). Lagrangian relaxation and enumeration for solving constrained shortest-path problems. *Networks: an international journal, 52*(4), 256-270.

Verstichel, J., Kinable, J., De Causmaecker, P., & Berghe, G. V. (2015). A Combinatorial Benders′ decomposition for the lock scheduling problem. *Computers & operations research, 54*, 117-128.

Gendron, B., Garroppo, R. G., Nencioni, G., Scutellà, M. G., & Tavanti, L. (2013). Benders decomposition for a location-design problem in green wireless local area networks. *Electronic notes in discrete mathematics, 41*, 367-374.