# زمان‌بندی حرکت قطارها با استفاده از الگوریتم شاخه و حد و الگوریتم ابتکاری جستجوی پرتو- مطالعه موردی: مسیرهای دوخطه ریلی ایران

محمود صفارزاده (مسئول مکاتبات) استاد، دانشکده مهندسی عمران و محیط زیست، دانشگاه تربیت مدرس، تهران، ایران و رییس پژوهشگاه حمل و نقل پارسه، تهران، ایران

محمد تمنایی، دانشجوی دکتری، دانشکده مهندسی عمران و محیط زیست، دانشگاه تربیت مدرس، تهران، ایران

امین جمیلی، استادیار، دانشکده مهندسی صنایع، دانشگاه تهران، تهران، ایران

سید احسان سیدابریشمی، استادیار، دانشکده مهندسی عمران و محیط زیست، دانشگاه تربیت مدرس، تهران، ایران

E-mail:saffar_m@modares.ac.ir

دریافت: ۱۳۹۲/۰۷/۲۳ پذیرش: ۱۳۹۳/۰۲/۲۲

**چکیده:**

هدف از این پژوهش، ارایه روشهای دقیق و تقریبی جهت حل مسئله زمان‌بندی حرکت قطارها در مسیرهای دوخطه ریلی است. به منظور حل دقیق و یافتن جواب بهینه مسئله زمان‌بندی حرکت قطارها، از بسته نرم‌افزاری CPLEX11 و نیز الگوریتم شاخه وحد زمان‌بندی حرکت قطارها استفاده می‌شود. مدل ریاضی زمان‌بندی حرکت قطارها در مسیر دوخطه و نیز الگوریتم شاخه و حد پیشنهادی در نرم‌افزار جاوا پیاده‌سازی شدند و مسائلی با ابعاد مختلف در مسیرهای ریلی بافق-سیرجان و تهران-مشهد مورد آزمایش قرار گرفتند. مقایسه نتایج حاصل، نشان از برتری عملکرد الگوریتم شاخه و حد پیشنهادی نسبت به CPLEX بویژه در مواجهه با مسائل زمان‌بندی با ابعاد بزرگ دارد. نتایج نشان می‌دهند که CPLEX ، مسائل بزرگ را در زمانهای بسیار طولانی حل می‌کند؛ در حالی که الگوریتم شاخه و حد می تواند جواب بهینه این‌گونه مسائل را در زمانهایی منطقی و بسیار کمتر از زمان حل CPLEX به دست آورد. همچنین در این پژوهش، از پنج روش تقریبی حل مبتنی بر الگوریتم جستجوی پرتو استفاده شد. نتایج بررسی روشهای تقریبی برای مسائل با ابعاد مختلف در مسیر تهران-مشهد حاکی از عملکرد مناسب‌تر روشهای ابداع‌شده پژوهش حاضر، در مقایسه با روشهای قبلی، به ویژه در مواجهه با مسائل با ابعاد بزرگ است؛ به طوری که بسته به میزان اهمیت پارامتر دقت و سرعت در شرایط مختلف، روشهای جستجوی پرتو جدید می توانند جداول زمان‌بندی نزدیک به بهینه را در زمانهای منطقی و با اشغال کنترل‌شده فضای حافظه ارایه کنند. با استفاده از نتایج این پژوهش، امکان تهیه جداول زمانی حرکت قطارها در زمانهای مناسب برای مسیرهای دوخطه به ویژه مسیرهای طولانی با تعداد زیاد قطار فراهم می‌شود.

**واژه‌های کلیدی:** زمان‌بندی حرکت قطارها، روشهای ابتکاری، الگوریتم شاخه و حد، الگوریتم جستجوی پرتو، جواب بهینه.

## ۱. مقدمه

یکی از ابعاد مهم برنامه‌ریزی حمل و نقل ریلی، موضوع زمان‌بندی حرکت قطارهاست. زمان‌بندی حرکت قطار، تعیین زمانهای ورود و خروج هر قطار در هر ایستگاه با توجه به محدودیتهای زمان سفر و محدودیتهای ایمنی حرکت قطارهاست. هدف از پژوهش حاضر، بررسی روشهای دقیق و تقریبی جهت حل مسئله زمان‌بندی حرکت قطارها و ارایه روشهایی جهت یافتن جواب بهینه و یا نزدیک به بهینه برای مسائل با ابعاد مختلف است. در این پژوهش، مدل ریاضی زمان‌بندی حرکت قطارها در مسیرهای دوخطه[1]ریلی استفاده شده‌است. این مدل توسط هیگینز و همکاران جهت زمان‌بندی حرکت قطارها در مسیرهای ریلی پیشنهاد شده‌است [Higgins, et al. 1996] و تاکنون مبنای مطالعات زیادی قرار گرفته‌است. در پژوهش حاضر، مدل مذکور با استفاده از الگوریتم‌های مختلف حل دقیق و تقریبی، جهت حل مسائلی با ابعاد مختلف استفاده شده‌است.

در بخش دوم این مقاله، مروری بر ادبیات موضوع زمان‌بندی حرکت قطارها انجام شده‌است. در بخش سوم، مدل ریاضی زمان‌بندی اولیه حرکت قطارها ارایه شده‌است. در بخش چهارم، جهت حل سریع‌تر و بهتر مسئله زمان‌بندی حرکت قطارها، یک الگوریتم شاخه و حد[2] پیشنهاد شده؛ مفاهیم و ساختار این الگوریتم توضیح داده‌شده‌است. الگوریتم مذکور می تواند جواب بهینه مسئله زمان‌بندی حرکت قطارها در مسیر دوخطه را ارایه کند. همچنین به منظور کاهش فضای جواب و رسیدن به جوابهای نزدیک به بهینه در زمان کمتر از زمان حل الگوریتم شاخه و حد، پنج روش تقریبی حل مبتنی بر الگوریتم جستجوی پرتو[3] استفاده شدند که سه مورد از این روشهای ابتکاری، به عنوان ابداعات و نوآوریهای این پژوهش مطرح هستند. در بخش پنجم، مسیرهای ریلی دوخطه بافق- سیرجان و تهران- مشهد مورد بررسی قرار گرفتند و نتایج حل مسائل با ابعاد مختلف توسط الگوریتم‌های پیشنهادی حل دقیق و تقریبی محاسبه شدند. در بخش ششم، جمع‌بندی و نتیجه‌گیری انجام شده‌است.

## ۲. مروری بر ادبیات موضوع

مسئله زمان‌بندی حرکت قطارها یکی از مهم ترین مسائل در حیطه برنامه ریزی راه آهن محسوب می‌شود. اولین بار اسپیگل در سال ۱۹۷۳ این مسئله را با هدف کاهش مجموع تأخیرات مورد بررسی قرار داد[Szpigel, 1973]. از آن تاریخ تاکنون پژوهشگران، مدلهای ریاضی و روشهای حل مختلفی را در این زمینه ارایه کرده‌اند. در زیر به اهم مطالعات مرتبط با پژوهش حاضر اشاره می‌شود.

هیگینز و همکاران مدلی جهت زمان‌بندی حرکت قطارها ارایه کرده‌اند. این مدل مبنای بسیاری از مطالعات در سالهای اخیر قرار داشته و در مسیرهای تک‌خطه و دوخطه ریلی کاربرد دارد[Higgins, et al. 1996]. یوسف شفاهی، زمان‌بندی حرکت قطارها را با هدف بیشینه کردن قابلیت اطمینان برنامه مورد توجه قرار داده است. در این مطالعه ابزار تحلیل روش شبیه سازی و مطالعه موردی مسیر تهران- مشهد انتخاب شده است[Shafahi and Sadeghi, 2004] . قصیری و همکاران در مقاله‌ای، مسئله زمان‌بندی قطارها در یک شبکه راه آهن را به کمک یک مدل ریاضی با دو هدف کمینه کردن مصرف سوخت و کمینه کردن تأخیر قطارها ارایه کردند در این مقاله، چند مثال عددی جهت اعتباردهی به مدل ارایه شده مورد بررسی قرار گرفته است[Ghoseiri, et al. 2004]. کاپرارا و همکاران مسئله زمان‌بندی قطارها در یک مسیر دوخطه تک‌جهته در ایتالیا را مورد بررسی قرار داده اند[Caprara, et al. 2006]. تورن کوئیست و پیرسون مسئله زمان‌بندی قطارها را در شبکه راه آهن سوئد مدنظر قرار داده‌اند. در این مقاله روشهایی جهت حل مسئله ارایه شده و از لحاظ دستیابی به جواب بهتر و سرعت در رسیدن به جواب

مقایسه شده‌اند[Tornquist, 2007]. جمیلی و کیانفر، مدل ریاضی زمان‌بندی حرکت قطارها با هدف حداقل کردن تأخیرات قطارها را توسعه داده‌اند. در این مطالعه، الگوریتم حل مسئله به کمک روش عملیات حرارتی شبیه‌سازی شده ارایه شده‌است [Jami-li and Kianfar, 2009]. خادم ثامنی یک مدل ریاضی برای زمان‌بندی حرکت قطارها در خطوط دوخطه ارایه کرده است. در این مدل، محدودیت اقامه نماز در نظر گرفته شده‌است[Khadem Sameni, 2007] . یقینی و محمدزاده، مدلی با هدف کمینه کردن مجموع تأخیرات قطارها با در نظر گرفتن زمانهای توقف اجباری برای ادای فریضه نماز ارایه کرده‌اند. در این پژوهش، چند مسئله آزمایشی جهت ارزیابی مدل بررسی شده‌اند[Yaghini and Mo-hammadzadeh, 2011] . کورمن و همکاران مدلی جهت حل مسئله زمان‌بندی قطارها ارایه کرده‌اند و به منظور حل آن از روش فوق ابتکاری جستجوی ممنوعه بهره برده اند. استفاده از این روش، به تولید جوابهای بهتر و در مدت زمان کوتاه‌تر کمک کرده‌است[Corman et al. 2009]. بوردت و کوزن جهت حل مسئله زمان‌بندی حرکت قطارها یک روش جدید تلفیقی بر پایه زمان‌بندی کار کارگاهی ارایه کرده‌اند. در این مدل مسئله توسط یک گراف منفصل نمایش داده شده‌است. همچنین مثالهایی جهت ارزیابی مدل معرفی شده‌است[Burdett and Kozan, 2010]. مو و همکاران مسئله بهینه سازی را با رویکرد زمان‌بندی حرکت قطارهای باری توسعه دادند. در این مطالعه جهت روش حل از الگوریتم تجزیه استفاده شده است[Mu et al. 2011].

همچنین پژوهشهای متعددی جهت حل مسئله زمان‌بندی حرکت قطارها توسط الگوریتم‌های مبتنی بر شاخه و حد انجام شده‌است. ژو و ژانگ یک مدل دوهدفه جهت مسئله زمان‌بندی حرکت قطارها در مسیر دوخطه ارایه کردند. مدل با دو هدف: کمینه کردن زمان انتظار قطارهای سریع السیر و حداقل کردن کل زمان سفر همه قطارها بود. یک الگوریتم شاخه و حد جهت تولید جوابهای پارتو بسط داده شد. همچنین یک الگوریتم جستجوی پرتو جهت یافتن جوابهای نزدیک به بهینه در زمان کوتاه ابداع شد. طول ۶۱۵ کیلومتر از مسیر بیجینگ- شانگهای با تعداد ۱۷ بلاک جهت مقایسه عملکرد الگوریتم ها انتخاب شد. [Zhou and 2005].

واکر و همکاران مسئله زمان‌بندی حرکت قطارها و خدمه قطار را در یک مسیر تک‌خطه به صورت همزمان مورد بررسی قرار داده‌اند. هدف مدل علاوه بر کاهش تأخیر قطارها، حداقل اختلاف نسبت به برنامه کاری خدمه است. روش پیشنهادی حل بر مبنای الگوریتم شاخه و حد است و جهت مطالعه موردی، مسیر کوتاهی از راه‌آهن نیوزیلند مورد بررسی قرار گرفته‌است. [Walker, et al. 2005].

دی‌آریانو و همکاران، در مقاله خود، مسئله زمان‌بندی مجدد قطارها در یک شبکه راه آهن را بر مبنای زمان‌بندی کار کارگاهی مورد مطالعه قرار داده‌اند. روش حل پیشنهادی ایشان بر مبنای ارایه یک الگوریتم حل به کمک روش شاخه و حد است. همچنین یک مطالعه موردی مربوط به منطقه‌ای از شبکه راه آهن آلمان در این مقاله مورد بررسی قرار گرفته است[D'Ariano, et al. 2007].

ژو و ژانگ مسئله زمان‌بندی قطارها در یک مسیر تک خطه دو جهته را مورد بررسی قرار داده اند. روش حل پیشنهادی مبتنی بر الگوریتم شاخه و حد و با بکارگیری ۳ شیوه جهت کاهش اندازه فضای جواب است: بکارگیری قاعده حد پایین بر اساس شیوه آزادسازی لاگرانژ، بکارگیری یک قاعده حد پایین دقیق و بکارگیری یک روش ابتکاری جهت تعیین حد بالایی نزدیک به جواب بهینه. مطالعه موردی مسیری تک خطه ۱۳۸ کیلومتری شامل ۱۸ ایستگاه از لایژو به شائو در ایالت فوجیان چین بوده است[Zhou, 2007].

کاچیانی و همکاران، الگوریتمهای ابتکاری و دقیق برای مسئله زمان‌بندی حرکت قطارها در حالات دوره ای و غیر دوره ای ارایه کرده‌اند. در این پژوهش، مطالعه موردی مربوط به برخی از

خطوط راه آهن ایتالیا مورد بررسی قرار گرفته است[Cacchiani et al. 2008].

شفیعا و همکاران مدل جدیدی جهت حل مسئله زمان‌بندی استوار حرکت قطارها در مسیرهای تک خطه ریلی برای مسائل با ابعاد بزرگ پیشنهاد دادند و جهت حل آن، یک الگوریتم بر شاخه و حد به همراه یک الگوریتم جدید جستجوی پرتو توسعه دادند. مسائل با ابعاد مختلف از ۶ تا ۲۴ بلاک برای ۶ تا ۲۰ قطار حل شد. نتایج حکایت از برتری عملکرد الگوریتم شاخه و حد نسبت به بسته های نرم افزاری متداول همچون لینگو دارد. همچنین الگوریتم جستجوی پرتو توانسته جوابهای نزدیک به بهینه را در زمانهای منطقی بیابد [Shafia, et al. 2012].

## ۳. مدل ریاضی زمان‌بندی اولیه حرکت قطارها

در این بخش، مدل ریاضی مورد استفاده در این پژوهش توضیح داده می‌شود. این مدل ریاضی توسط هیگینز و همکاران جهت زمان‌بندی حرکت قطارها در مسیرهای ریلی ارایه شده‌است [Higgins, et al. 1996] و تاکنون مبنای مطالعات زیادی قرار گرفته‌است. متغیرهای تصمیم در این مدل ریاضی، به صورت پیوسته هستند و شامل زمانهای ورود و خروج قطار به/از ایستگاههای ریلی هستند. خروجی مدل، جواب بهینه زمان‌بندی حرکت قطارهاست. به این معنا که متغیرهای تصمیمی که در جواب خروجی مقداردهی می‌شوند، قطعاً مناسب‌ترین مقادیر برای نیل به تابع هدف مدل (کمینه سازی تأخیرات کل سیستم) هستند. لازم به ذکر است در صورتی که زمان‌بندی حرکت قطارها برای مسیرهای دوخطه انجام شود، مسئله متناظر با مسئله زمان‌بندی جریان کارگاهی[۴] است، ولی اگر این مدل برای مسیر تک‌خطه استفاده شود، مسئله متناظر با مسئله کار کارگاهی[۵] است. تابع هدف مدل، حداقل کردن زمان تأخیر قطارهاست. به منظور تفکیک قطارهای گذرنده از جهات رفت و برگشت در مسیرهای دوخطه، قطارهای مسیر رفت با عنوان «قطارهای شمالی» و قطارهای مسیر برگشت با عنوان «قطارهای جنوبی» معرفی شده‌اند.

**جدول ۱. نمادهای مورد استفاده در مدل ریاضی**

| نماد | | توضیحات |
|---|---|---|
| مجموعه ها | $B$ | مجموعه بلاک‌های مسیر |
| | $N$ | مجموعه قطارهای شمالی |
| پارامترها | $bo_i$ | اولین بلاکی که توسط قطار $i$ پیموده می‌شود |
| | $bd_i$ | آخرین بلاکی که توسط قطار $i$ پیموده می‌شود. |
| | $EO_{bo_i}$ | زودترین زمان مجاز برای خروج قطار $i$ از مبدأش |
| | $G_i$ | ضریب اولویت وزنی قطار $i$ |
| | $H_{ijb}$ | حداقل سرفاصله قطارهای $i$ و $j$ در ایستگاه انتهای بلاک $b$ |
| | $STmin_{i,b}$ | حداقل مدت زمان توقف پیش‌بینی‌شده قطار $i$ در ایستگاه انتهای بلاک $b$ |
| | $tt_{i,b}$ | زمان طی بلاک $b$ توسط قطار $i$ |
| متغیرهای تصمیم | $A_{ijb}$ | (باینری) برابر ۱ اگر قطار $i$ زودتر از قطار $j$ وارد بلاک $b$ شود و برابر صفر اگر برعکس |
| | $X_{i,b}$ | (پیوسته) زمان رسیدن قطار $i$ به انتهای بلاک $b$ |
| | $Y_{i,b}$ | (پیوسته) زمان شروع به حرکت قطار $i$ از ابتدای بلاک $b$ |

### ۳-۱ نمادها

نمادهای مورد استفاده در تابع هدف و محدودیتهای مدل در جدول ۱ نامگذاری شده‌اند. از آنجا که در مسیرهای دوخطه، حرکت قطارهای شمالی و جنوبی مستقل از یکدیگر انجام می‌شود، در این مقاله تنها روابط مربوط به حرکت قطارهای شمالی ارایه می‌شود. بدیهی است که روابط مربوط به حرکت قطارهای جنوبی نیز به طریق مشابه است.

### ۳-۲ تابع هدف و محدودیتها

مدل ریاضی مورد استفاده جهت زمان‌بندی حرکت قطارها در مسیر دوخطه به صورت زیر است:

$$Min\, z = \sum_{i \in N} \left( G_i \times \left( X_{i,bd_i} - Y_{i,bo_i} \right) \right) \qquad (۱)$$

*s.t.*

$$EO_{bo_i} \le Y_{i,bo_i} \qquad \forall i \in N \qquad (۲)$$

$$STmin_{i,b} \le Y_{i,(b+1)} - X_{i,b} \qquad \forall i \in N, \forall b \in B - \{bd_i\} \qquad (۳)$$

$$\frac{d_b}{V_{max_{i,b}}} \le X_{i,b} - Y_{i,b} \le \frac{d_b}{V_{min_{i,b}}} \qquad \forall i \in N, \forall b \in B \qquad (۴)$$

$$Y_{i,b} \ge X_{i,(b-1)} \qquad \forall i \in N, \forall b \in B, b \ge bo_i + 1 \qquad (۵)$$

$$Y_{j,b} + M \times \left(1 - A_{ijb}\right) \ge X_{i,b} + H_{ijb} \qquad \forall i, j \in N, \forall b \in B, i \ne j \qquad (۶)$$

$$Y_{i,b} + M \times A_{ijb} \ge X_{j,b} + H_{ijb} \qquad \forall i, j \in N, \forall b \in B, i \ne j \qquad (۷)$$

تابع هدف مدل- رابطه (۱)- به صورت کمینه کردن مجموع وزنی زمان تأخیر کل قطارها تعریف می‌شود.

ضریب اولویت وزنی قطارها – که در تابع هدف مدل در نظر گرفته شده‌است- تأثیر اختلاف در اهمیت قطارهای مختلف را در مدل ریاضی در نظر می گیرد و سبب می‌شود که مدل به سمت یافتن جوابهایی متمایل شود که در آنها قطارهای با اهمیت بیشتر (ضریب اولویت وزنی بیشتر) با تأخیر کمتری مواجه شوند. ضریب اولویت وزنی قطارها برای قطارهای مشابه یکسان در نظر گرفته می‌شود.

**محدودیتها:**

(الف) پنجره زمانی حرکت از مبدأ: دسته محدودیت (۲) بیانگر آن است که زمان حرکت هریک از قطارها از مبدأ خود، باید بزرگ‌تر از زودترین زمان تعیین‌شده حرکت آن قطار باشد. زودترین و دیرترین زمان تعیین‌شده حرکت قطارها به عنوان ورودیهای مدل هستند.

(ب) حداقل زمان توقف قطارهای مسافری: دسته محدودیت (۳) بیانگر حداقل زمان توقف پیش‌بینی شده قطارها است که به عنوان ورودی مدل در نظر گرفته می‌شوند. توقف قطارهای مسافری در ایستگاهها به دلایل مختلف همچون پیاده و سوار نمودن مسافرین انجام می‌شود.

(پ) زمان طی بلاک: دسته محدودیت (۴) عنوان می‌کند که زمان طی بلاک توسط هریک از قطارها، بایستی بین بیشترین و کمترین زمان مجاز باشد. طول بلاکها و نیز کمینه و بیشینه سرعت مجاز برای هر قطار مسافری یا باری در بین هر دو ایستگاه متوالی، به عنوان ورودیهای مدل هستند.

(ت) حفظ پیوستگی در حرکت قطارها: دسته محدودیت (۵) بیانگر آن است که زمان شروع به حرکت قطار $i$ از ابتدای بلاک $b$ باید همواره بزرگ تر یا مساوی زمان رسیدن قطار $i$ به انتهای بلاک قبلی $(b-1)$ باشد. محدودیت تضمین می‌کند که قطارها بلاکها را یکی پس از دیگری طی کنند.

(ث) عدم برخورد قطارها در بلاک: اگر مقدار $A_{ijb}$ برابر با یک شد (قطار $i$ زودتر از قطار $j$ وارد بلاک $b$ شود)، محدودیت شماره (۶) فعال می‌شود. یعنی زمان شروع به حرکت قطار $j$ (قطار دوم) از ایستگاه ابتدایی بلاک $b$ بایستی برابر یا بیش از زمان رسیدن قطار $i$ (قطار اول) به انتهای بلاک $b$ به علاوه سرفاصله بین دو قطار $i$ و $j$ در بلاک $b$ باشد. در غیر این‌صورت، محدودیت شماره (۷) فعال می‌شود.

در این پژوهش، امکان سبقت قطارهای مشابه از یکدیگر وجود ندارد. در نتیجه، اگر قطار $i$ زودتر از قطار مشابه $j$ حرکت از مبدأ خود را آغاز کند، مدل بایستی مقدار $A_{ijb}$ $(b$: هریک از بلاکهای مشترک) را برابر یک قرار دهد.

## ۴. الگوریتم شاخه و حد زمان‌بندی حرکت قطارها

در پژوهش حاضر به منظور حل مسئله زمان‌بندی حرکت قطارها در زمان مناسب، از الگوریتم شاخه و حد زمان‌بندی حرکت قطارها استفاده شده‌است. الگوریتم مذکور مبتنی بر روش شاخه‌زنی تولید زمان‌بندی فعال است که توسط پیندو [Pinedo, 2002] برای حل مسئله زمان‌بندی کار کارگاهی مورد استفاده قرار گرفته‌است. در این روش، در ابتدا یک ماشین انتخاب می‌شود و در مرحله بعد، به ازای هر یک از عملیاتهایی که بر روی آن ماشین می‌تواند انجام شود یک گره تولید می‌شود، به گونه‌ای که زمان‌بندی جزئی حاصله فعال باقی بماند. مفاهیم «ماشین و عملیات» در مسئله کار کارگاهی به ترتیب معادل «بلاک و قطار» در مسئله زمان‌بندی حرکت قطارها هستند. در پژوهشهای مختلف [Zhou, 2005; Walker, 2005; Shafia, 2012] الگوریتم شاخه و حد جهت حل مسئله زمان‌بندی حرکت قطارها استفاده شده‌است. جهت انجام پژوهش حاضر، الگوریتم شاخه و حد زمان‌بندی حرکت قطارها با استفاده از مفهوم لیستهای پیوندی[۶]، طراحی و در نرم‌افزار جاوا پیاده‌سازی شد.

### ۴-۱ نمادهای مورد استفاده در الگوریتم

در جدول ۲، نمادهای مورد استفاده در الگوریتم شاخه و حد زمان‌بندی حرکت قطارها مشخص شده‌اند.

### ۴-۲ ساختار الگوریتم شاخه و حد زمان‌بندی حرکت قطارها

ساختار الگوریتم شاخه و حد زمان‌بندی حرکت قطارها به نحوی طراحی شده که تمامی جوابها را با بکارگیری یک طرح مشخص شاخه‌زنی جهت تولید گره‌ها، یک به یک بررسی و نهایتا، زمان‌بندی بهینه را ارایه کند. تابع هدف (حد پایینی گره‌ها)،

**جدول ۲. نمادهای مورد استفاده در الگوریتم شاخه و حد زمان‌بندی حرکت قطارها**

| نماد | توضیحات |
|---|---|
| $ANL$ | لیست پیوندی حاوی گره‌های فعال |
| $NL^{\lambda}$ | لیست پیوندی شاخه‌های تولیدشده ذیل گره $\lambda$ |
| $\lambda$ | اندیس گره |
| $(i,b)$ | دوتایی عبور که بیانگر عبور قطار $i$ از بلاک $b$ است. |
| $\hat{b_i^{\lambda}}$ | بلاک فعال قطار $i$ در گره $\lambda$ که باید توسط این قطار پیموده شود. |
| $X_{i,b}^{\lambda}$ | زمان رسیدن قطار $i$ به انتهای بلاک $b$ در گره $\lambda$ |
| $Y_{i,b}^{\lambda}$ | زمان شروع به حرکت قطار $i$ از ابتدای بلاک $b$ در گره $\lambda$ |
| $H_{\min}$ | حداقل سرفاصله بین دو قطار |
| $LB^{\lambda}$ | حد پایینی در گره $\lambda$ |
| $\Delta^{\lambda}$ | مجموعه قطارهایی در گره $\lambda$ که هنوز به بلاک مقصد خود نرسیده‌اند |
| $LA_b^{\lambda}$ | زمان رسیدن آخرین قطاری که بلاک $b$ را در گره $\lambda$ طی کرده است. |
| $MA^{\lambda}$ | حداقل زمان رسیدن به انتهای بلاک یافت‌شده در مرحله شاخه‌زنی گره $\lambda$ |
| $TB^{\lambda}$ | شمارنده تعداد قطار-بلاک برنامه‌ریزی شده در گره $\lambda$ |
| $PASS_{i,b}$ | زمان عبور قطار i از بلاک $b$ |
| $UB$ | حد بالایی (بهترین جواب تاکنون) |

حداقل کردن جمع وزنی تأخیرهای قطارها است.

هر گره $\lambda$ متناظر با یک زمان‌بندی جزئی است. با زمان‌بندی هر قطار- بلاک در گره $\lambda$ به مقدار شمارنده $TB^{\lambda}$ یک واحد اضافه می‌شود. هرگاه $TB^{\lambda}$ به مقدار مجموع قطار-بلاکهای مسئله برسد، آنگاه می‌توان نتیجه گرفت که گره $\lambda$ در انتهای درخت قرار دارد و این حالت بیانگر یک زمان‌بندی کامل خواهد بود و طبیعتاً در این گره امکان شاخه زدن وجود ندارد. به علاوه در هر مرحله از شاخه‌زنی، لیست گره‌های فعال ($ANL$) شامل گره‌هایی است که امکان شاخه زدن را دارند. گره‌های موجود در $ANL$ همواره بر مبنای مقدار حد پایینی، مرتب هستند. در هر مرحله از الگوریتم این لیست بهنگام می‌شود. زمانی که این لیست تهی شد، الگوریتم شاخه و حد اتمام خواهد یافت. مراحل مختلف الگوریتم به شرح زیر است:

**گام ۱- یافتن جواب امکان‌پذیر اولیه**

کلیه قطار-بلاکهای مسئله به صورت تک تک زمان‌بندی شوند (نحوه زمان‌بندی هریک از قطار-بلاکها مشابه روش ارایه‌شده در گام ۳ است). مقدار حد پایین به دست‌آمده این جواب به عنوان $UB$ در نظر گرفته‌شود.

**گام ۲- انتخاب گره از لیست گره‌های فعال ($ANL$)**

گره $\lambda$ که دارای اولویت جهت ورود به مسئله است، از $ANL$ انتخاب شود. اگر لیست تهی است، الگوریتم پایان یافته‌است.

**گام ۳- انتخاب قطار**

۳-الف) برای هریک از قطارهای متعلق به مجموعه $\Delta^{\lambda}$ (مجموعه قطارهایی در گره $\lambda$ که هنوز به بلاک مقصد خود نرسیده‌اند)، زمان رسیدن قطار به انتهای بلاک فعال آن قطار ($X^{\lambda}_{i,b}$) محاسبه شود. این کار با افزودن زمان عبور قطار i از بلاک $\hat{b}_i^{\lambda}$ به زمان شروع به حرکت این قطار از ابتدای این بلاک انجام می‌شود (رابطه ۸).

۳-ب) در بین کلیه قطارهای متعلق به مجموعه $\Delta^{\lambda}$، مقدار $MA^{\lambda}$ محاسبه شود (رابطه ۹). بلاک مرتبط با $MA^{\lambda}$ با نماد $b^*$ نمایش داده می‌شود.

۳-پ) برای هر یک از قطارهای متعلق به مجموعه $\Delta^{\lambda}$، زمان رسیدن قطار به انتهای بلاک فعال آن قطار ($X^{\lambda,1}_{i,b}$) برابر با صفر قرار داده شود (رابطه ۱۰).

$$X^{\lambda}_{i,\hat{b}_i^{\lambda}} \leftarrow Y^{\lambda}_{i,\hat{b}_i^{\lambda}} + PASS^{\lambda}_{i,\hat{b}_i^{\lambda}} \qquad \forall i \in \Delta^{\lambda} \qquad (۸)$$

$$MA^{\lambda} \leftarrow Min\left\{X^{\lambda}_{i,\hat{b}_i^{\lambda}} \mid i \in \Delta^{\lambda}\right\} \qquad (۹)$$

$$X^{\lambda,1}_{i,\hat{b}_i^{\lambda}} \leftarrow 0 \qquad \forall i \in \Delta^{\lambda} \qquad (۱۰)$$

**گام ۴- شاخه‌زنی در گره $\lambda$**

از مجموعه $\Delta^{\lambda}$، قطارهایی که زمان شروع به حرکت آنها در بلاک $b^*$ کمتر از $MA^{\lambda}$ است انتخاب و به عنوان قطارهای شاخه‌زنی گره $\lambda$ ($Ch^{\lambda}$) نامگذاری شوند (رابطه ۱۱). در نتیجه، برای هر قطار $i^*$ ($i^* \in Ch^{\lambda}$) بلاک فعال آن قطار در گره $\lambda$ ($\hat{b}_i^{\lambda}$) برابر با $b^*$ خواهد بود. برای هر قطار $i^*$ ($i^* \in Ch^{\lambda}$) عملیات زیر انجام شود:

۴-الف- کلیه اطلاعات از گره $\lambda$ به گره $\lambda'$ انتقال یابد.

۴-ب- زمان رسیدن قطار $i^*$ به انتهای بلاک $b^*$ در گره $\lambda'$ محاسبه شود (رابطه ۱۲).

۴-پ- $LA^{\lambda'}_{b^*}$ بهنگام شود (رابطه ۱۳).

۴-ت-حد پایینی گره $\lambda'$ بهنگام شود (رابطه ۱۴).

حد پایینی گره $\lambda'$ با افزودن تأخیر وزنی قطار $i^*$، بهروز می‌شود. در صورتی که قطار $i^*$، در ابتدای شروع حرکت از بلاک مبدأ خود قرار داشته‌باشد، میزان تأخیر وزنی این قطار برابر با اختلاف زمان شروع به حرکت از مبدأ با زودترین زمان مجاز جهت اعزام ($ETO_{i^*}$) است. در غیر اینصورت، میزان تأخیر وزنی قطار $i^*$، برابر خواهد بود با اختلاف زمان شروع به حرکت در بلاک بعدی ($Y^{\lambda'}_{i^*,b^*}$) و مجموع زمان رسیدن در بلاک قبلی و حداقل زمان توقف در ایستگاه واقع در انتهای بلاک قبلی

$(X^{\lambda'}_{i^*,b^*-1} + STmin_{i^*,b^*-1})$.

۴- ث- بلاک فعال قطار $i^*$ در گره $\lambda'$ بهنگام شود.

در صورتی که قطار $i^*$ جزو قطارهای شمالی باشد، بلاک فعال آن با افزودن یک واحد به مقدار قبلی بهنگام می‌شود. در حالی که بلاک فعال قطارهای جنوبی با کاهش یک واحد از مقدار قبلی بهروز می‌شود.

۴- ج- زمان شروع به حرکت هر قطار از بلاک فعال جدید خود در گره $\lambda'$ بهنگام شود.

برای قطار $i^*$: (رابطه ۱۵). اگر قطار $i^*$ به ایستگاه مقصد خود رسیده‌باشد، از لیست قطارهای $\Delta^\lambda$ حذف شود.

برای سایر قطارها: (رابطه ۱۶).

۴- چ - بستن گره: در صورتی که حد پایینی گره $\lambda'$ کمتر از $UB$ باشد، گره $\lambda'$ به لیست شاخه‌های تولیدشده ذیل گره $\lambda$ ($NL^\lambda$) اضافه شود. در غیر این‌صورت گره $\lambda'$ حذف شود. $NL^\lambda$ بهنگام شود.

۴- ح - در صورتی که هیچ گرهای در $NL^\lambda$ موجود نباشد به گام ۶ بروید.

$$Ch^\lambda \leftarrow \left\{(i \in \Delta^\lambda \,|Y^{\lambda}_{i,b^*} < MA^\lambda\right\} \quad (۱۱)$$

$$X^{\lambda'}_{i^*,b^*} \leftarrow Y^{\lambda'}_{i^*,b^*} + PASS^{\lambda'}_{i^*,b^*} \quad (۱۲)$$

$$LA^{\lambda'}_{b^*} \leftarrow X^{\lambda'}_{i^*,b^*} + H_{\min} \quad (۱۳)$$

$$\begin{cases} if\ b^* = bo_{i^*} \Rightarrow LB^{\lambda'} \leftarrow LB^\lambda + G_{i^*} \times \left\{Y^{\lambda'}_{i^*,b^*} - ETO_{i^*}\right\} \\ if\ b^* \neq bo_{i^*} \Rightarrow LB^{\lambda'} \leftarrow LB^\lambda + G_{i^*} \times \left\{Y^{\lambda'}_{i^*,b^*} - X^{\lambda'}_{i^*,b^*-1} - STmin_{i^*,b^*-1}\right\} \end{cases} \quad (۱۴)$$

$$Y^{\lambda'}_{i^*,\hat{b}^{\lambda'}_{i^*}} \leftarrow X^{\lambda'}_{i^*,b^*} + STmin_{i^*,b^*} \quad (۱۵)$$

$$if\ (\hat{b}^{\lambda'}_i \neq bd_i \ \ \&\ \ \hat{b}^{\lambda'}_i = b^*) \Rightarrow Y^{\lambda'}_{i,\hat{b}^{\lambda'}_i} \leftarrow LA^{\lambda'}_{b^*} \quad (۱۶)$$

### گام ۵- انتخاب گره از $NL^\lambda$

اولین گره در $NL^\lambda$ جهت ادامه عملیات انتخاب شود. سایر گره‌های باقیمانده در $NL^\lambda$ (در صورت وجود)، با توجه به مقدار حد پایینیشان، طی عملیات مرتب‌سازی به روش جستجوی دوتایی[۷] به $ANL$ افزوده شوند.

### گام ۶- معیار توقف

دو شرط زیر بررسی شوند:

الف) یک برنامه زمان‌بندی کامل به دست آمده‌باشد. یعنی الگوریتم به انتهای درخت رسیده‌باشد.

ب) در انتهای گام ۲ هیچ گرهای در لیست $NL^\lambda$ موجود نباشد.

اگر یکی از دو شرط زیر برقرار باشد، $UB$ بهنگام شود. به گام ۲ بروید.

اگر هیچیک از دو شرط برقرار نباشد، به گام ۳ بروید.

الگوریتم شاخه و حد زمان‌بندی حرکت قطارها براساس مفهوم لیست پیوندی، در محیط نرم‌افزار برنامه‌نویسی جاوا پیاده‌سازی شد.

### ۴-۳ الگوریتم‌های ابتکاری جستجوی پرتو

خروجی الگوریتم شاخه و حد زمان‌بندی حرکت قطارها جواب بهینه است. برای مسائل بزرگ زمان حل طولانی می‌شود. الگوریتم جستجوی پرتو با استفاده از روشهای ابتکاری، در زمانهای کمتر به جوابهای امکان‌پذیر مناسب دست می‌یابد.این الگوریتم، به عنوان نوع ویرایش‌یافته از الگوریتم شاخه و حد در مطالعات مختلف استفاده شده‌است. الگوریتم جستجوی پرتو عملکردی مشابه با الگوریتم شاخه و حد دارد؛ با این تفاوت که جستجوی پرتو همه گره‌های حاصله طی عملیات شاخه و حد را در حافظه نگه‌نمیدارد. بلکه با اعمال شروطی، توقف در ادامه شاخه‌زنی برخی گره‌ها را تسریع می کند و در نتیجه یک جواب امکان‌پذیر مناسب و احتمالاً غیربهینه در زمان کمتر از زمان حل الگوریتم شاخه و حد به دست می‌آید. جهت انجام این پژوهش، پنج روش ابتکاری جستجوی پرتو استفاده شده‌اند و در محیط نرم‌افزار برنامه‌نویسی جاوا پیاده‌سازی شدند. دو مورد از این روشها (جستجوی پرتو ۱ و ۲) در مطالعه انجام‌شده توسط شفیعا و همکاران ارایه شده‌است [Shafia, 2012]. در حالی که سه روش دیگر (جستجوی پرتو ۳ و ۴ و ۵) جزو نوآوریهای مربوط به پژوهش حاضر هستند. در ادامه، هریک از این روشها

توضیح داده می‌شوند:

**جستجوی پرتو ۱:** تعداد شاخه‌های تولید شده در هر گره به یک آستانه مشخص محدود شود. هر چقدر این آستانه بزرگ تر باشد، فضای جواب بزرگ تر و زمان حل طولانی تر خواهد بود. در این مقاله آستانه مذکور برابر با ۲ در نظر گرفته شده‌است.

**جستجوی پرتو ۲:** اگر حد پایین گرهی بزرگ تر از درصدی از حد بالایی مسئله بود، گره بسته شود. به عبارت دیگر، در صورت برقراری شرط زیر، گره $\lambda$ از ادامه شاخه‌زنی متوقف شود:

$$LB^{\lambda} > (1-f_2 \times \frac{TB - TB^{\lambda}}{TB}) \times UB$$

که در آن $TB^{\lambda}$ تعداد قطار- بلاک زمان‌بندی‌شده در گره $\lambda$ و $TB$ تعداد کل قطار- بلاکهای مسئله است. مقدار $f_2$ متناسب با مطلوبیت موردنظر از جواب و نیز زمان مورد نظر جهت حل تعیین می‌شود ($0 \le f_2 \le 1$). اگر $f_2$ برابر صفر باشد آنگاه الگوریتم تمامی گره‌ها را بررسی می‌کند و جواب بهینه حاصل خواهدشد.

**جستجوی پرتو ۳:** اندازه لیست گره‌های فعال ($ANL$) به یک آستانه محدود شود. به نحوی که امکان بزرگ تر شدن لیست وجود نداشته‌باشد و جوابهای مناسب برای مسئله در زمان مناسب به دست آیند. آستانه مذکور تابعی از تعداد بلاکها و تعداد قطارها است. پس از بررسی مسائل با ابعاد مختلف، آستانه مذکور با فرمول ابتکاری زیر پیشنهاد شد:

(۱۷)

$$List\ Size = f_3 \times NumBlock \times \frac{(NumTn \times NumTs)}{(NumTn + NumTs)}$$

که در آن $List\ Size$ آستانه اندازه لیست گره‌های فعال و $NumBlock$، $NumTn$ و $NumTs$ به ترتیب تعداد بلاکها، تعداد قطارهای شمالی و تعداد قطارهای جنوبی هستند. در این مقاله، مقادیر $f_3$ برابر با ۴۰ تا ۲۰۰ قرار داده شد. لازم به ذکر است که افزایش آستانه اندازه لیست قطعاً زمان حل را افزایش خواهد داد، ولی لزوماً برنامه زمان‌بندی حاصله بهبود نخواهد یافت، زیرا با افزایش آستانهف ممکن است برخی از گره‌ها با زمان‌بندی جزئی[۸] مناسب نتوانند به لیست گره‌های فعال وارد شوند و در نتیجه زمان‌بندی کامل حاصله، بدتر از زمان‌بندی گره حذف شده باشد. با این حال، نتایج حاصله در این پژوهش نشان دهنده آن است که بزرگ شدن آستانه اندازه لیست گره‌های فعال در اکثر حالات منجر به بهبود جواب شده‌است.

**جستجوی پرتو ۴:** تعداد جوابهای کامل به دست‌آمده مسئله به یک آستانه محدود شود. در این صورت، عملیات شاخه‌زنی متوقف و بهترین جواب به دست‌آمده به عنوان حد بالایی مسئله در نظر گرفته می‌شود. آستانه مذکور تابعی از تعداد بلاکها و تعداد قطارها است. پس از بررسی مسائل با ابعاد مختلف، آستانه مذکور با فرمول ابتکاری زیر پیشنهاد شد:

(۱۸)

$$NumInc = f_4 \times \sqrt{NumBlock \times \frac{(NumTe \times NumTw)}{(NumTe + NumTw)}}$$

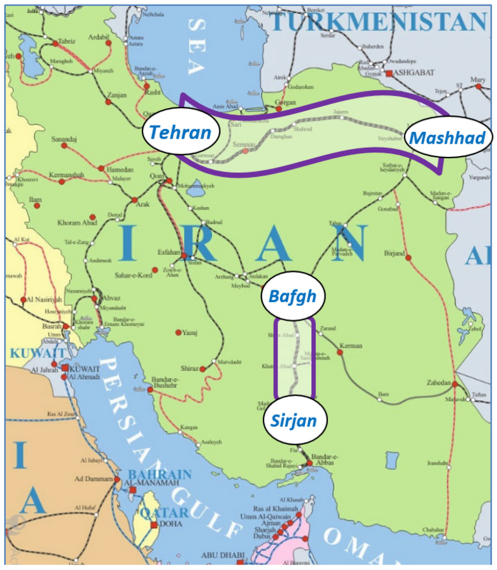
که در آن $NumInc$ آستانه تعداد جوابهای کامل است. به عنوان مثال، در مسئلهای با ۴۹ بلاک، ۱۰ قطار رفت و ۱۰ قطار برگشت و $f_4$ برابر با ۲، $NumInc$ برابر با ۱۹ است. در این صورت در زمان اجرای برنامه، اگر تعداد جوابهای کامل به دست‌آمده به ۱۹ عدد برسد، عملیات شاخه‌زنی متوقف و بهترین جواب به دست‌آمده به عنوان حد بالایی مسئله ارایه می‌شود. در این مقاله، مقادیر $f_4$ برابر با ۰/۵ تا ۲/۵ قرار داده شد. با افزایش این آستانه، مقدار تابع هدف کاهش و زمان حل افزایش می‌یابد.

**جستجوی پرتو ۵:** درصد قطار- بلاکهای زمان‌بندی‌شده در هر گره ($f_5$) به یک آستانه مشخص محدود شود. در صورت برقراری این شرط در زمان اجرای برنامه، زمان‌بندی قطار- بلاکها برای هر گره مطابق با الگوریتم شاخه و حد ادامه می‌یابد تا جایی که تعداد قطار- بلاک زمان‌بندی‌شده گره به آستانه مشخص (مثلا ۲٪ کل قطار-بلاکهای مسئله) برسد. از آن مرحله به بعد، عملیات شاخه‌زنی برای آن گره متوقف می‌شود و قطار- بلاکهای باقیمانده

گره، یکی پس از دیگری زمان‌بندی می‌شوند تا زمان‌بندی کامل آن گره حاصل شود. نحوه زمان‌بندی هریک از قطار- بلاکهای باقیمانده، دقیقاً مشابه زمان‌بندی قطار- بلاکها در الگوریتم شاخه و حد است. سپس حد بالایی مسئله بهنگام می‌شود و عملیات شاخه و حد برای سایر گره‌های موجود با توجه به شرط «درصد قطار-بلاکهای زمان‌بندی‌شده هر گره»، ادامه می‌یابد. در این مقاله، مقادیر $f_5$ برابر با ۲٪ تا ۱۰٪ قرار داده شد. با افزایش این آستانه، مقدار تابع هدف کاهش و زمان حل افزایش می‌یابد.

## ۵. نتایج و تحلیل آنها

به منظور ارزیابی روشهای ارایهشده جهت حل مدل برنامه‌ریزی ریاضی، مسیرهای ریلی دوخطه بافق-سیرجان و تهران-مشهد مورد بررسی قرار گرفتند. مسیر بافق-سیرجان یکی از مسیرهای مهم شبکه ریلی کشور است که دارای طول ۲۵۳ کیلومتر و شامل ۱۰ بلاک است. مسیر ریلی تهران- مشهد طولانی‌ترین مسیر دوخطه ریلی کشور به طول ۹۲۴ کیلومتر و ویژه قطارهای مسافری است و شامل ۴۹ بلاک است. در شکل ۱، موقعیت این دو مسیر در شبکه ریلی ایران نمایش داده شده‌است.




**شکل ۱. موقعیت مسیرهای ریلی بافق- سیرجان و تهران- مشهد در شبکه ریلی ایران**

در جداول پ۱ و پ۲ مربوط به بخش پیوست، ایستگاههای موجود واقع در مسیرهای به ترتیب بافق-سیرجان و تهران-مشهد و فاصله ایستگاههای آنها از یکدیگر ارایه شده‌اند. مسائل مختلف حلشده در این پژوهش با استفاده از رایانه‌ای با مشخصات Core 2 CPU at 2.66 GHz و 4 GBs of Ram اجرا شده‌اند.

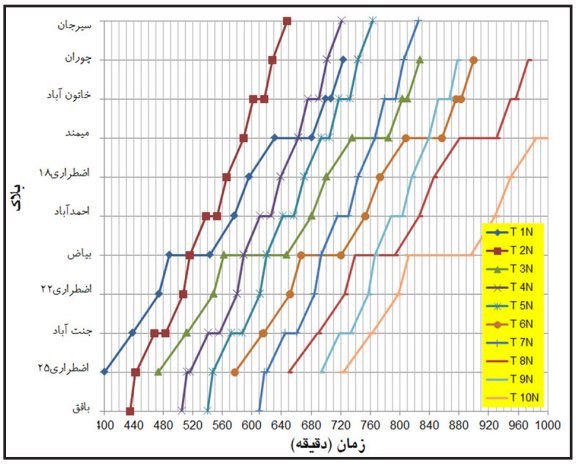
### ۵-۱ مقایسه عملکرد بسته نرم‌افزاری CPLEX و الگوریتم شاخه حد

به منظور ارزیابی عملکرد الگوریتم شاخه و حد زمان‌بندی حرکت قطارها، مسیر ریلی دوخطه بافق-سیرجان (به عنوان مسئله‌ای با ابعاد نسبتاً کوچک) و نیز مسیر ریلی دوخطه تهران-مشهد (به عنوان مسئلهای با ابعاد نسبتاً بزرگ) مورد آزمایش قرار گرفت. به این منظور، در قالب مسائل مختلف، تعداد ۶ تا ۱۱ قطار در هر جهت از مسیر بافق-سیرجان و تعداد ۵ تا ۹ قطار در هر جهت از مسیر تهران-مشهد در نظر گرفته‌شدند و هر مسئله، یک بار توسط الگوریتم پیشنهادی شاخه و حد و بار دیگر توسط بسته نرم‌افزاری CPLEX 11 حل شد. کلیه ورودیهای زوج مسئله‌های حل‌شده به دو روش مذکور با یکدیگر برابر بوده‌اند. قطارها در مسیر بافق-سیرجان به دو نوع مسافری و باری و در مسیر تهران مشهد به چهار دسته با سرعتها و اولویتهای مختلف تقسیم شده‌اند. مبدأ و مقصد قطارها لزوماً همان ایستگاه ابتدایی و انتهایی مسیر نیست؛ بلکه شروع و انتهای حرکت برخی قطارها، ایستگاههای میانی واقع در مسیر است. به‌علاوه، سرعت حرکت، مبدأ و مقصد، زودترین و دیرترین زمان مجاز عزیمت از مبدأ و زمان توقف اجباری هر قطار در هر ایستگاه میانی به عنوان ورودیهای مسئله در نظر گرفته شده‌اند. در جدول ۳، نتایج مشاهده می‌شوند.

برای هریک از مسائل حل‌شده گراف زمان‌بندی حرکت قطارها حاصل شده‌است. در شکل ۲ گراف زمان‌بندی تولیدشده توسط هر دو روش برای ۱۰ قطار جهت رفت در مسیر بافق- سیرجان نمایش داده شده‌است.

همان گونه که در جدول ۳ ملاحظه می‌شود، با افزایش تعداد قطارها، زمان حل و بیشینه اندازه لیست گره‌ها به صورت نمائی افزایش یافته‌اند که این امر به دلیل ماهیت NP-Hard مسئله است. همچنین مشاهده می‌شود که جوابهای تولیدشده توسط دو روش (استفاده از CPLEX و الگوریتم شاخه و حد) برای کلیه زوج مسئله‌ها یکسان و برابر با جواب بهینه مسئله است. با این حال، پس از مقایسه زمان حل مسائل مختلف ارایه‌شده در این جدول برای مسیر بافق–سیرجان و نیز مسیر تهران– مشهد مشخص می‌شود که در همه مسائل، الگوریتم شاخه و حد توانسته‌است در زمان کوتاه‌تری نسبت به نرم‌افزار CPLEX به جواب بهینه دست یابد. با افزایش ابعاد مسائل، اختلاف زمان حل دو روش افزایش معناداری نشان می‌دهد.

نتایج نشان می‌دهد که CPLEX مسائل بزرگ را در زمانهای خیلی طولانی حل می‌کند؛ در حالی که الگوریتم شاخه و حد می تواند جواب بهینه اینگونه مسائل را در زمانهایی منطقی و بسیار کمتر از زمان حل CPLEX به دست آورد. این امر نشان دهنده عملکرد مناسب‌تر الگوریتم شاخه و حد نسبت به CPLEX است. با افزایش تعداد قطارها به ۱۰ قطار در هر جهت از مسیر تهران–مشهد، ابعاد مسئله به حدی بزرگ می‌شود که هر دو روش، با کمبود فضای حافظه رایانه مواجه میگردند و قادر به حل مسئله نیستند.



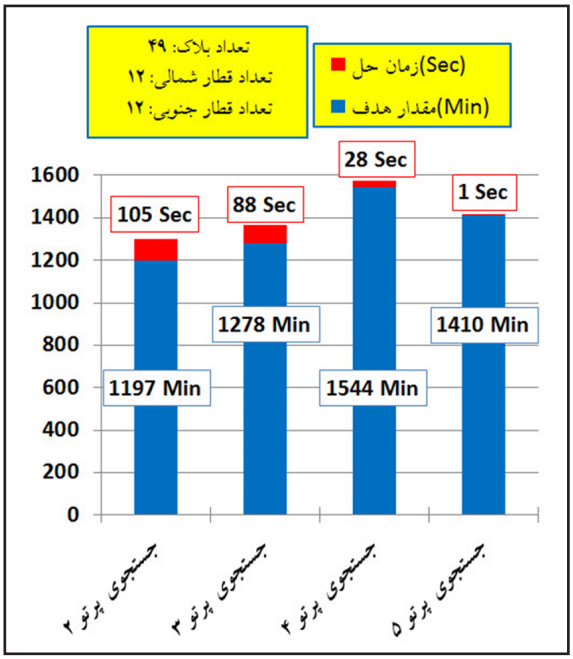

**شکل ۲. گراف زمان‌بندی تولیدشده توسط الگوریتم شاخه و حد– مسیر بافق–سیرجان**

**جدول ۳. مقایسه نتایج حل مسائل با ابعاد مختلف توسط CPLEX و الگوریتم شاخه حد**

| مسیر | | بافق–سیرجان | | | | | | تهران–مشهد | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| تعداد بلاک ها | | ۱۰ | ۱۰ | ۱۰ | ۱۰ | ۱۰ | ۱۰ | ۴۹ | ۴۹ | ۴۹ | ۴۹ | ۴۹ | ۴۹ |
| تعداد قطارهای شمالی | | ۶ | ۷ | ۸ | ۹ | ۱۰ | ۱۰ | ۵ | ۶ | ۷ | ۸ | ۹ | ۱۰ |
| تعداد قطارهای جنوبی | | ۶ | ۷ | ۸ | ۹ | ۱۰ | ۱۰ | ۵ | ۶ | ۷ | ۸ | ۹ | ۱۰ |
| CPLEX | جمع وزنی تأخیرات (دقیقه) | ۳۶۵ | ۵۰۹ | ۵۴۱ | ۶۹۷ | ۷۳۵ | ۹۰۳ | ۲۱۰ | ۳۰۸ | ۳۳۴ | ۳۸۵ | ۵۱۱ | کمبود فضای حافظه |
| | زمان حل (ثانیه) | ۸,۳۶ | ۲۵,۳ | ۱۷,۴ | ۱۰۸ | ۴۴۶,۵ | ۱۰۵۹ | ۳,۳ | ۶,۷ | ۱۰,۸ | ۱۵,۸ | ۳۴۸,۷ | |
| شاخه و حد | جمع وزنی تأخیرات (دقیقه) | ۳۶۵ | ۵۰۹ | ۵۴۱ | ۶۹۷ | ۷۳۵ | ۹۰۳ | ۲۱۰ | ۳۰۸ | ۳۳۴ | ۳۸۵ | ۵۱۱ | کمبود فضای حافظه |
| | زمان حل (ثانیه) | ۰,۰۱ | ۰,۰۵ | ۰,۰۶ | ۰,۳۹ | ۰,۸۵ | ۱۹,۹ | ۰,۰۶ | ۰,۱۴ | ۰,۲۰ | ۰,۴۲ | ۷۳,۷ | |
| | ماکزیمم اندازه لیست گره‌ها | ۵۹ | ۲۷۶ | ۴۴۶ | ۲۱۷۸ | ۳۶۸۶ | ۱۵۸۳۹ | ۳۶۴ | ۴۳۶ | ۶۸۳ | ۱۰۷۴ | ۲۵۰۴۲ | |
| | تعداد جوابهای کامل تولیدشده | ۵ | ۷ | ۷ | ۱۱ | ۱۲ | ۱۲ | ۱۴ | ۱۲ | ۱۴ | ۱۴ | ۲۳ | |

### ۵-۲ مقایسه روشهای ابتکاری جستجوی پرتو- مطالعه موردی: مسیر ریلی تهران- مشهد

جهت ارزیابی و مقایسه عملکرد روشهای جستجوی پرتو ارایه‌شده، مسائل با ابعاد مختلف توسط هریک از این روشها مورد بررسی قرار گرفتند. به این منظور، مسیر ریلی تهران- مشهد جهت مطالعه موردی انتخاب شد و مسائلی با ۹، ۱۰، ۱۲، ۱۵، ۲۰ و ۲۵ قطار در هریک از مسیرهای رفت و برگشت مورد ارزیابی واقع شدند. مقدار تابع هدف -جمع وزنی تأخیرات- (بر حسب دقیقه)، زمان حل (بر حسب ثانیه) و بزرگ ترین اندازه لیست گره‌ها برای هر مسئله استخراج شد. در جدول ۴ نتایج ارایه شده‌اند. در شکل ۳ نتایج حل برای مسئله ای با ابعاد متوسط (۴۹ بلاک و ۱۲ قطار در هر جهت) ارایه شده‌است. با توجه به این شکل و نتایج ارایه‌شده در جدول ۴ می‌توان گفت برای مسائل با ابعاد متوسط، دو روش جستجوی پرتو ۲ و ۳ عملکرد تقریباً مشابهی دارند. همچنین با استفاده از روش جستجوی پرتو ۵ امکان دستیابی بسیار سریع به یک برنامه زمان‌بندی فراهم می‌شود.




**شکل ۳. مقایسه نتایج حل برای مسئله ای با ابعاد متوسط**

در جدول ۴ در مواردی که با علامت «---» نشان داده شده‌اند، امکان حل مسئله به دلیل اشغال کامل فضای حافظه هیپ[۹] رایانه وجود نداشته‌است. جدول ۴ نشان می‌دهد که با افزایش تعداد قطارها به بیش از ۱۲ قطار در هر جهت، هیچ یک از روشهای جستجوی پرتو ۱ و ۲ در هیچ سطحی قادر به کاهش فضای جواب و کنترل فضای اشغال‌شده حافظه و یافتن جوابی برای مسئله نبوده‌اند. در حالی که هر سه روش ابتکاری ارایه شده در این پژوهش (جستجوی پرتو ۳ و ۴ و ۵) در مواجهه با مسائل با ابعاد بزرگ، عملکرد مناسب‌تری از روشهای پرتو جستجوی پرتو ۱ و ۲ داشته‌اند. روش ابتکاری جستجوی پرتو ۳ (تحدید سایز لیست گره‌ها) برای مسائل با ابعاد مختلف (کوچک، متوسط و بزرگ) دارای عملکرد مطلوب‌تری نسبت به سایرین بوده‌است و توانسته جوابهای نزدیک به بهینه را در زمانهای نسبتاً قابل قبول به دست آورد. البته در روش جستجوی پرتو ۳ زمان حل مسائل با ابعاد بزرگ و مقادیر $f_3$ برابر با ۸۰ و بیشتر، طولانی است. این امر به دلیل اندازه بسیار بزرگ لیست گره‌های فعال است که طبیعتاً نیاز به زمان حل زیادی جهت بررسی گره‌های مختلف وجود دارد. نتایج ارایه‌شده در جدول ۴ برای روش جستجوی پرتو ۳، نشان می‌دهند که جهت یافتن یک جواب مناسب برای مسائل با ابعاد مختلف در زمانی منطقی، مقدار $f_3$ برابر با ۴۰ مقدار مطلوبی است. در جدول ۵ مزایا و معایب هریک از روشهای حل بررسی شده در این پژوهش به اجمال اشاره شده‌اند.

در نهایت می‌توان نتیجه‌گیری کرد که برای حل مسائل با ابعاد کوچک، استفاده از روش دقیق شاخه و حد مناسب است. برای حل مسائل با ابعاد متوسط، استفاده از روشهای جستجوی پرتو ۱ و ۲ کافی به نظر می‌رسند. در این نوع مسائل، روشهای ابتکاری جستجوی پرتو ۳ و ۴ نیز جوابهای مناسب تولید می‌کنند. برای مسائل با ابعاد بزرگ، روشهای جستجوی پرتو ۱ و ۲ کارا نیستند. در مواجهه با چنین مسائلی، در صورتی که دقت بالا مطلوب

جدول ۴. مقایسه نتایج حل مسائل با ابعاد مختلف توسط الگوریتم‌های شاخه حد و جستجوی پرتو (مسیر تهران- مشهد)

| | تعداد بلاک ها | ۴۹ | | ۴۹ | | ۴۹ | | ۴۹ | | ۴۹ | |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | تعداد قطارهای شمالی | ۹ | | ۱۰ | | ۱۲ | | ۱۵ | | ۲۰ | |
| | تعداد قطارهای جنوبی | ۹ | | ۱۰ | | ۱۲ | | ۱۵ | | ۲۰ | |
| | روش حل | مقدار هدف (Min) | زمان حل (Sec) | مقدار هدف | زمان حل | مقدار هدف | زمان حل | مقدار هدف | زمان حل | مقدار هدف | زمان حل |
| | شاخه و حد | ۵۱۱ | ۷۳,۷ | --- | --- | --- | --- | --- | --- | --- | --- |
| | جستجوی پرتو ۱ | ۵۱۱ | ۲,۹ | --- | --- | --- | --- | --- | --- | --- | --- |
| جستجوی پرتو ۲ | $f_p$=۱,۰ | ۶۷۷ | ۰,۱ | ۱۲۹۱ | ۰,۶ | ۱۵۹۰ | ۰,۹ | --- | --- | --- | --- |
| | $f_p$=۰,۹ | ۱۰۱۸ | ۰,۲ | ۱۰۷۰ | ۱۸,۹ | ۱۱۹۷ | ۱۰۵,۵ | --- | --- | --- | --- |
| | $f_p$=۰,۸ | ۵۱۹ | ۰,۶ | ۱۰۵۹ | ۲۴۳,۴ | --- | --- | --- | --- | --- | --- |
| | $f_p$=۰,۷ | ۶۱۱ | ۰,۸ | ۱۰۵۹ | ۲۸۳۹ | --- | --- | --- | --- | --- | --- |
| | $f_p$=۰,۶ | ۶۱۶ | ۱,۶ | --- | --- | --- | --- | --- | --- | --- | --- |
| | بهترین جواب حاصله | ۵۱۹ | ۰,۶ | ۱۰۵۹ | ۲۴۳,۴ | ۱۱۹۷ | ۱۰۵,۵ | --- | --- | --- | --- |
| جستجوی پرتو ۳ | $f_p$=۴۰ | ۵۱۱ | ۱۴,۲ | ۱۱۸۹ | ۴۷,۹ | ۱۲۷۸ | ۸۸,۱ | ۲۳۳۹ | ۲۸۰,۷ | ۵۲۵۵ | ۱۱۳۵ |
| | $f_p$=۸۰ | ۵۱۱ | ۴۰,۸ | ۱۱۸۹ | ۱۴۳,۵ | ۱۳۶۰ | ۳۴۶,۰ | ۲۴۱۸ | ۹۶۳,۹ | ۵۳۸۰ | ۳۹۳۲ |
| | $f_p$=۱۲۰ | ۵۱۱ | ۷۲,۷ | ۱۱۱۲ | ۳۹۶,۷ | ۱۳۴۴ | ۶۱۵,۲ | ۲۴۱۸ | ۲۳۰۵ | ۴۹۷۷ | ۹۳۳۸ |
| | $f_p$=۱۶۰ | ۵۱۱ | ۷۲,۸ | ۱۰۵۹ | ۶۴۷,۷ | ۱۳۳۸ | ۱۱۷۷ | ۲۳۳۹ | ۱۲۱۹۳ | ۵۱۶۷ | ۲۶۷۷۴ |
| | $f_p$=۲۰۰ | ۵۱۱ | ۷۳,۳ | ۱۱۱۲ | ۱۳۵۷ | ۱۲۷۸ | ۲۵۳۳ | زمان حل طولانی | | زمان حل طولانی | |
| | بهترین جواب حاصله | ۵۱۱ | ۱۴,۲ | ۱۰۵۹ | ۶۴۷,۷ | ۱۲۷۸ | ۸۸,۱ | ۲۳۳۹ | ۲۸۰,۷ | ۴۹۷۷ | ۹۳۳۸ |
| جستجوی پرتو ۴ | $f_p$=۰,۵ | ۶۱۷ | ۰,۳ | ۲۳۶۰ | ۱,۳ | ۲۴۷۰ | ۳,۸ | ۵۲۵۴ | ۱۱۷,۰ | --- | --- |
| | $f_p$=۱,۰ | ۵۶۹ | ۴,۴ | ۱۸۲۳ | ۲,۶ | ۱۹۲۹ | ۹,۳ | --- | --- | --- | --- |
| | $f_p$=۱,۵ | ۵۱۹ | ۴۰,۱ | ۱۵۵۳ | ۲,۹ | ۱۹۲۹ | ۱۰,۸ | --- | --- | --- | --- |
| | $f_p$=۲,۰ | ۵۱۱ | ۷۱,۷ | ۱۵۵۳ | ۲۰,۳ | ۱۵۴۴ | ۲۸,۰ | --- | --- | --- | --- |
| | $f_p$=۲,۵ | ۵۱۱ | ۷۴,۵ | ۱۵۵۳ | ۲۶,۳ | ۱۵۴۴ | ۳۲,۶ | --- | --- | --- | --- |
| | بهترین جواب حاصله | ۵۱۱ | ۷۱,۷ | ۱۵۵۳ | ۲,۹ | ۱۵۴۴ | ۲۸,۰ | ۵۲۵۴ | ۱۱۷,۰ | --- | --- |
| جستجوی پرتو ۵ | $f_p$= ۲٪ | ۹۷۹ | ۰,۰ | ۱۶۷۲ | ۰,۰ | ۱۸۴۹ | ۰,۰ | ۳۳۴۱ | ۰,۱ | ۶۸۴۰ | ۰,۵ |
| | $f_p$= ۴٪ | ۹۷۹ | ۰,۰ | ۱۶۷۲ | ۰,۱ | ۱۸۳۸ | ۰,۲ | ۲۹۰۵ | ۱,۵ | ۶۱۶۸ | ۵۴,۲ |
| | $f_p$= ۶٪ | ۸۱۹ | ۰,۲ | ۱۴۱۰ | ۰,۵ | ۱۵۸۲ | ۲,۲ | ۲۹۰۵ | ۱۰۶,۳ | --- | --- |
| | $f_p$= ۸٪ | ۷۵۹ | ۱,۰ | ۱۴۱۰ | ۱۵,۱ | ۱۵۸۲ | ۹۵,۵ | --- | --- | --- | --- |
| | $f_p$= ۱۰٪ | ۷۵۹ | ۱۱,۴ | ۱۴۱۰ | ۴۶۵,۶ | --- | --- | --- | --- | --- | --- |
| | بهترین جواب حاصله | ۷۵۹ | ۱,۰ | ۱۴۱۰ | ۰,۵ | ۱۵۸۲ | ۲,۲ | ۲۹۰۵ | ۱,۵ | ۶۱۶۸ | ۵۴,۲ |

| | تعداد بلاک ها | ۴۹ | |
|---|---|---|---|
| | تعداد قطارهای شمالی | ۲۵ | |
| | تعداد قطارهای جنوبی | ۲۵ | |
| | روش حل | مقدار هدف | زمان حل |
| | شاخه و حد | --- | --- |
| | جستجوی پرتو ۱ | --- | --- |
| جستجوی پرتو ۲ | $f_p$=۱,۰ | --- | --- |
| | $f_p$=۰,۹ | --- | --- |
| | $f_p$=۰,۸ | --- | --- |
| | $f_p$=۰,۷ | --- | --- |
| | $f_p$=۰,۶ | --- | --- |
| | بهترین جواب حاصله | --- | --- |
| جستجوی پرتو ۳ | $f_p$=۴۰ | ۶۹۸۱ | ۳۰۸۳ |
| | $f_p$=۸۰ | ۷۰۶۹ | ۱۳۰۹۲ |
| | $f_p$=۱۲۰ | ۷۰۶۷ | ۴۷۲۳۸ |
| | $f_p$=۱۶۰ | زمان حل طولانی | |
| | $f_p$=۲۰۰ | زمان حل طولانی | |
| | بهترین جواب حاصله | ۶۹۸۱ | ۳۰۸۳ |
| جستجوی پرتو ۴ | $f_p$=۰,۵ | --- | --- |
| | $f_p$=۱,۰ | --- | --- |
| | $f_p$=۱,۵ | --- | --- |
| | $f_p$=۲,۰ | --- | --- |
| | $f_p$=۲,۵ | --- | --- |
| | بهترین جواب حاصله | --- | --- |
| جستجوی پرتو ۵ | $f_p$= ۲٪ | ۸۵۳۹ | ۱,۵ |
| | $f_p$= ۴٪ | ۷۸۷۴ | ۱۹۷۵ |
| | $f_p$= ۶٪ | --- | --- |
| | $f_p$= ۸٪ | --- | --- |
| | $f_p$= ۱۰٪ | --- | --- |
| | بهترین جواب حاصله | ۷۸۷۴ | ۱۹۷۵ |

باشد، روش ابتکاری جستجوی پرتو ۳ و در صورتی که زمان حل سریع مدنظر باشد، روش ابتکاری جستجوی پرتو ۵ کارآیی مناسبی دارند. در شکل ۴ نمونهای از گراف زمانبندی تولیدشده توسط روش جستجوی پرتو ۳ برای ۲۵ قطار در هر جهت از مسیر ریلی تهران- مشهد نمایش دادهشده است.

## ۶. جمع بندی و نتیجه گیری

در این پژوهش، روشهای مختلف حل مسئله زمانبندی حرکت قطارها استفاده شدهاند و عملکرد هریک در مواجهه با مسائل با ابعاد متفاوت بررسی شدهاست. این روشها به دو دسته دقیق و تقریبی تقسیمبندی میشوند.

به منظور حل دقیق، از بسته نرمافزاری CPLEX و نیز الگوریتم شاخه و حد زمانبندی حرکت قطارها استفاده شد. به این جهت، مدل ریاضی زمانبندی حرکت قطارها در مسیر دوخطه و نیز الگوریتم شاخه و حد پیشنهادی در نرمافزار جاوا پیادهسازی شدند و مسائلی با ابعاد مختلف در مسیرهای ریلی بافق- سیرجان (به طول ۲۵۳ کیلومتر و شامل ۱۱ ایستگاه) و تهران- مشهد (به طول ۹۲۴ کیلومتر و شامل ۵۰ ایستگاه) مورد آزمایش قرار گرفتند. به منظور پیادهسازی الگوریتم شاخه و حد پیشنهادی از لیستهای پیوندی جهت ذخیره گرههای تولیدشده استفاده شد. خروجی هر دو روش دقیق، یکسان و برابر با جواب بهینه بود. ولی زمان حل مسائل توسط الگوریتم شاخه و حد بسیار کمتر از زمان حل مسائل مشابه توسط CPLEX بود. با بزرگ شدن ابعاد مسائل، اختلاف زمان حل دو روش افزایش معناداری نشان میدهد. نتایج نشان میدهد که CPLEX مسائل بزرگ را در زمانهای خیلی طولانی حل میکند؛ در حالی که الگوریتم شاخه و حد پیشنهادی می تواند جواب بهینه اینگونه مسائل را در زمانهایی منطقی و بسیار کمتر از زمان حل CPLEX به دست آورد. این امر نشان از برتری الگوریتم شاخه و حد پیشنهادی نسبت به نرمافزار CPLEX به ویژه در مواجهه با مسائل زمانبندی با ابعاد بزرگ دارد. با افزایش ابعاد مسائل، امکان حل مسئله توسط هیچ یک از دو روش دقیق فوقالذکر میسر نبود. دلیل این امر، اشغال شدن کامل فضای حافظه تخصیصیافته جهت حل مسئله بود. به همین جهت، از روشهای تقریبی حل استفاده شد. هدف از استفاده از

**جدول ۵. مزایا و معایب نسبی روشهای حل بررسی شده در پژوهش حاضر**

| روش حل | | توضیحات |
|---|---|---|
| الگوریتم شاخه و حد | مزیت | تولید برنامه زمانبندی بهینه |
| | ضعف | اشغال فضای زیادی از حافظه / عدم حل مسائل با ابعاد متوسط در زمان کوتاه |
| جستجوی پرتو ۱ | مزیت | تولید برنامه زمانبندی مناسب برای مسائل با ابعاد کوچک در زمان کوتاه |
| | ضعف | اشغال فضای زیادی از حافظه / عدم حل مسائل با ابعاد بزرگ در زمان کوتاه |
| جستجوی پرتو ۲ | مزیت | تولید برنامه زمانبندی نزدیک به بهینه برای مسائل با ابعاد متوسط |
| | ضعف | اشغال فضای زیادی از حافظه / عدم حل مسائل با ابعاد بزرگ در زمان کوتاه |
| جستجوی پرتو ۳ | مزیت | کنترل فضای جواب/ تولید برنامه زمانبندی مناسب برای مسائل با ابعاد بزرگ |
| | ضعف | ---- |
| جستجوی پرتو ۴ | مزیت | تولید برنامه زمانبندی مناسب برای مسائل با ابعاد متوسط در زمان کوتاه |
| | ضعف | درصد خطای نسبتاً زیاد/ عدم حل مسائل با ابعاد بزرگ در زمان کوتاه |
| جستجوی پرتو ۵ | مزیت | تولید برنامه زمانبندی برای مسائل با ابعاد بزرگ در زمان کوتاه |
| | ضعف | درصد خطای نسبتاً زیاد |

این روشها، ایجاد تغییراتی در الگوریتم شاخه و حد پیشنهادی به منظور توقف سریع‌تر عملیات شاخه‌زنی، کاهش حافظه اشغالی و نیز کاهش زمان حل مسائل بود. الگوریتم تغییریافته شاخه‌و حد، به الگوریتم جستجوی پرتو موسوم است. در این پژوهش، پنج روش جستجوی پرتو بررسی شد. این روشها عبارتند از:

جستجوی پرتو ۱ (تحدید تعداد شاخه تولیدشده در هر گره به ۲ شاخه)

جستجوی پرتو ۲ (مقایسه حد پایین هر گره با درصدی از حد بالایی مسئله جهت توقف شاخه‌زنی در گره)

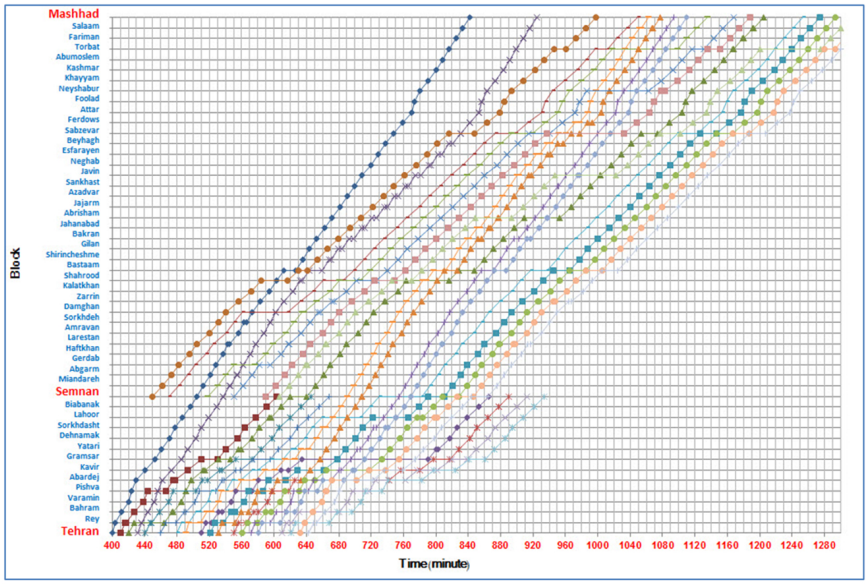
جستجوی پرتو ۳ (تحدید اندازه لیست پیوندی حاوی گره‌های فعال)

جستجوی پرتو ۴ (تحدید تعداد جوابهای کامل تولیدشده توسط الگوریتم شاخه و حد)

جستجوی پرتو ۵ (تحدید درصد قطار-بلاکهای هر گره که توسط الگوریتم شاخه و حد زمان‌بندی شده‌اند)

روشهای جستجوی پرتو ۱ و ۲ در مطالعات قبلی استفاده شده‌بودند. در حالی که روشهای ابتکاری جستجوی پرتو ۳ و ۴ و ۵ برای اولین بار در پژوهش حاضر بررسی شدند. نتایج بررسی روشهای تقریبی برای مسائل با ابعاد مختلف در مسیر تهران-مشهد حاکی از برتری نسبی روش جستجوی پرتو ۳ نسبت به سایر روشها دارد. این روش، می تواند با کنترل فضای جواب، برنامه زمان‌بندی مناسب در زمانی منطقی ارایه کند.

برای مسائل با ابعاد کوچک و متوسط، به ترتیب استفاده از روشهای جستجوی پرتو ۱ و ۲ کافی به نظر می‌رسند. البته در این نوع مسائل، روشهای جستجوی پرتو ۳ و ۴ نیز جوابهای مناسب تولید می‌کنند. ولی برای مسائل با ابعاد بزرگ روشهای جستجوی پرتو ۱ و ۲ کارا نیستند. در مواجهه با چنین مسائلی، در صورتی که دقت بالا مطلوب باشد، روش جستجوی پرتو ۳ و در صورتی که زمان حل سریع مدنظر باشد، روش جستجوی پرتو ۵ کارآیی مناسبی دارند. در نتیجه می‌توان گفت که بسته به میزان اهمیت پارامتر دقت و سرعت در شرایط مختلف، روشهای جستجوی پرتو ۳ و ۵ قادر به ارایه برنامه زمان‌بندی برای مسائل با ابعاد کوچک، متوسط و بزرگ هستند. با استفاده از نتایج این پژوهش، امکان تهیه جداول زمانی حرکت قطارها در زمانهای مناسب برای مسیرهای دوخطه به ویژه مسیرهای طولانی با تعداد زیاد قطار فراهم می‌شود.




**شکل ۴. نمونه‌ای از گراف زمان‌بندی تولیدشده توسط روش جستجوی پرتو ۳- مسیر تهران مشهد**

## ۷. پانویسها

1.Double-track corridors

2.Branch and bound algorithm

3.Beam search algorithm

4.Flow shop scheduling problem

5.Job shop scheduling problem

6.Linked list

7.Binary search

8.Partial schedule

9.Heap space

## ۸. مراجع


-جمیلی، امین و کیانفر، فریدون (۱۳۸۸) "زمان‌بندی حرکت قطارها به کمک روش فوق ابتکاری عملیات حرارتی شبیه سازی شده"، پژوهشنامه حمل و نقل، سال ششم، شماره اول.

-خادم ثامنی، ملودی (۱۳۸۶) "زمان‌بندی حرکت قطارها در مسیرهای دوخطه"، رساله کارشناسی ارشد، دانشکده مهندسی صنایع، دانشگاه تربیت مدرس، تهران.

-شفاهی، یوسف و صادقی، نازنین (۱۳۸۳) «کاربرد یک مدل شبیه سازی برای تعیین اثر سیاستهای عملیاتی مختلف بر روی قابلیت اطمینان برنامه زمان‌بندی حرکت قطارها»، اولین کنگره ملی مهندسی عمران، دانشگاه شریف.

-یقینی، مسعود و انجمن علمی دانشکده مهندسی راه آهن (۱۳۸۹) "زمان‌بندی و سیر و حرکت در راه آهن"، ترجمه مسعود یقینی، تهران، انتشارات پیشرو فنآوری قائد.

-یقینی، مسعود و محمدزاده، علی (۱۳۹۰) "یک مدل زمان‌بندی حرکت قطارها با در نظر گرفتن زمانهای توقف برای نماز"، نشریه تخصصی مهندسی صنایع، دوره ۴۵، شماره ۱، فروردین ماه ۱۳۹۰، از صفحه ۱۰۳ تا ۱۱۶.

-یقینی، مسعود و لسان، جواد (۱۳۸۹) " برنامه ریزی عملیات حمل و نقل ریلی"، ایران: انتشارات دانشگاه علم و صنعت ایران.

- Burdett, R. L. and Kozan, E. (2010) “A disjunctive graph model and framework for constructing new train schedules”, European Journal of Operational Research, Vol. 200, pp.85-98.

- Cacchiani, V., Caprara, A. and Toth, P. (2008) “A column generation approach to train timetabling on a corridor,” 4OR, vol. 6, no. 2, pp. 125–142.

- Caprara, A., Monaci, M., Toth, P. and Guida, P. L. (2006) “A Lagrangian heuristic algorithm for a real-world train timetabling problem,” Discrete Appl. Math., vol. 154, no. 5, pp. 738–753.

- Corman, F., D’Ariano, A., Pacciarelli, D. and Pranzo, M. (2009) “A tabu search algorithm for rerouting trains during rail operations”, Transportation Re-

search Part B, 44 (1), pp.175–192.

-D'Ariano, A., Pacciarelli, D. and Pranzo, M. (2007) "A branch and bound algorithm for scheduling trains in a railway network", European Journal of Operational Research, 183 (2), 643–657.

-Ghoseiri, K.; Szidarovszky, F. and Asgharpour, M. J. (2004) "A multi-objective train scheduling model and solution", Transportation Research Part B, vol. 38, no. 10, pp. 927-952.

-Higgins, A., Kozan, E. and Ferreira, L. (1996) "Optimal scheduling of trains on a single line track", Transportation Research Part B: Methodological, vol.30, PP. 147-161.

-Mu, Shi and Dessouky, Maged (2011) "Scheduling freight trains traveling on complex networks", Transportation Research Part B 45, pp.1103–1123.

-Pinedo, M. (2002) "Scheduling theory, algorithms and systems", Prentice Hall, Upper Saddle River, NJ.2002.

- Shafia, Mohammad Ali, Pourseyedaghaee, Mohsen, Sadjadi, Seyed Jafar and Jamili, Amin (2012) "Robust train timetabling problem: Mathematical model and branch and bound algorithm", IEEE Transactions On Intelligent Transportation Systems, Vol. 13, No. 1.

-Szpigel, B. (1973) "Optimal train scheduling on a single track railway", Operational Research 72, M. Ross (ed.), North-Holland, Amsterdam, pp. 343-352.

-Tornquist, J. and Persson, J. A. (2007) "N-tracked railway traffic re-scheduling during disturbances", Transportation Research Part B, Volume 41, Issue 3, pp. 342–362.

-Walker, C. G., Snowdon, J. N. and Ryan, D. M. (2005) "Simultaneous disturbance recovery of a train timetable and crew roster in real time," Computers & Operations Research, Vol. 32(8), pp. 2077-2094.

-Zhou, X., Zhong, M. (2005) "Bi-criteria train scheduling for high-speed passenger railroad planning applications," Eur. J. Oper. Res., Vol. 167, no. 3, pp. 752–771.

-Zhou, X. and Zhong, M. (2007) "Single-track train timetabling with guaranteed optimality: Branch-and-bound algorithms with enhanced lower bounds," Transportation Research - Part B, Vol. 41, pp. 320–341.

## ۹. پیوست ها

**جدول پ۱- ایستگاههای موجود در مسیر ریلی بافق-سیرجان و فواصل آنها**

| ایستگاه | فاصله (km) | ایستگاه | فاصله (km) |
|---|---|---|---|
| ۱-بافق | -- | ۷-اضطراری۱۸ | ۲۰ |
| ۲-اضطراری۲۵ | ۱۱ | ۸-میمند | ۳۵ |
| ۳-جنت آباد | ۳۸ | ۹-خاتون آباد | ۱۹ |
| ۴-اضطراری۲۲ | ۳۶ | ۱۰-چوران | ۱۷ |
| ۵-بیاض | ۱۴ | ۱۱-سیرجان | ۳۰ |
| ۶-احمدآباد | ۳۳ | مجموع | ۲۵۳ |

**جدول پ۲- ایستگاههای موجود در مسیر ریلی تهران-مشهد و فواصل آنها**

| ایستگاه | فاصله (km) | ایستگاه | فاصله (km) | ایستگاه | فاصله (km) | ایستگاه | فاصله (km) | ایستگاه | فاصله (km) |
|---|---|---|---|---|---|---|---|---|---|
| ۱-تهران | -- | ۱۱-سرخدشت | ۱۴,۱ | ۲۱-سرخده | ۱۵,۹ | ۳۱-ابریشم | ۱۹,۵ | ۴۱-عطار | ۲۳,۶ |
| ۲-ری | ۹,۹ | ۱۲-لاهور | ۲۰,۵ | ۲۲-دامغان | ۱۴,۳ | ۳۲-جاجرم | ۲۲,۲ | ۴۲-فولاد | ۸ |
| ۳-بهرام | ۱۶,۵ | ۱۳-بیابانک | ۲۱,۲ | ۲۳-زرین | ۲۲,۸ | ۳۳-آزادور | ۲۰,۲ | ۴۳-نیشابور | ۱۴ |
| ۴-ورامین | ۱۷,۴ | ۱۴-سمنان | ۱۸ | ۲۴-کلاتخوان | ۲۳,۸ | ۳۴-سنخواست | ۱۹,۵ | ۴۴-خیام | ۲۲,۲ |
| ۵-پیشوا | ۸,۸ | ۱۵-میاندره | ۱۸,۵ | ۲۵-شاهرود | ۱۸,۹ | ۳۵-جوین | ۱۹,۲ | ۴۵-کاشمر | ۲۲,۸ |
| ۶-ابردژ | ۱۲ | ۱۶-آبگرم | ۱۶,۳ | ۲۶-بسطام | ۲۰,۶ | ۳۶-نقاب | ۲۳,۳ | ۴۶-ابومسلم | ۱۷,۴ |
| ۷-کویر | ۲۳,۴ | ۱۷-گرداب | ۱۴ | ۲۷-شیرینچشمه | ۲۰,۳ | ۳۷-اسفراین | ۲۰,۵ | ۴۷-تربت | ۱۸,۶ |
| ۸-گرمسار | ۲۶ | ۱۸-هفتخوان | ۲۱ | ۲۸-گیلان | ۱۸,۳ | ۳۸-بیهق | ۲۰,۴ | ۴۸-فریمان | ۱۸,۸ |
| ۹-یاتاری | ۱۷ | ۱۹-لارستان | ۱۳ | ۲۹-بکران | ۲۰,۶ | ۳۹-سبزوار | ۲۲ | ۴۹-سلام | ۱۸,۵ |
| ۱۰-ده نمک | ۲۱,۹ | ۲۰-امروان | ۲۳ | ۳۰-جهان آباد | ۲۲,۲ | ۴۰-فردوس | ۲۳,۶ | ۵۰-مشهد | ۱۹,۹ |