

مسیریابی بهینه در شبکه‌ی حمل و نقل درون شهری با ادغام الگوریتم‌های فراابتکاری ژنتیک (GA) و جستجوی ممنوع (TS)

ابوذر شهمرادی^۱، سعید بهزادی^{۲*}

^۱ دانشجوی کارشناسی ارشد سیستم‌های اطلاعات مکانی - دانشکده مهندسی عمران - دانشگاه تربیت دبیر شهید رجایی
a.shahmoradi73@gmail.com

^۲ استادیار گروه مهندسی نقشه‌برداری - دانشکده مهندسی عمران - دانشگاه تربیت دبیر شهید رجایی
behzadi.saeed@gmail.com

(تاریخ دریافت خرداد ۱۳۹۸، تاریخ تصویب آذر ۱۳۹۸)

چکیده

با توجه به گسترش شهرها و پیچیده‌تر شدن شبکه‌ی راه‌های درون شهری به ویژه در شهرهای بزرگ، مسئله‌ی مسیریابی و در واقع یافتن کوتاه‌ترین مسیر تبدیل به یکی از دغدغه‌های افراد در هنگام تصمیم‌گیری برای انتخاب مسیر در جابجایی از مبدا حرکت به یک مقصد مشخص، شده است. روشی که در این پژوهش برای حل مسئله‌ی کوتاه‌ترین مسیر پیشنهاد می‌شود، استفاده از ترکیب الگوریتم‌های فراابتکاری ژنتیک (GA) و جستجوی ممنوع (TS) می‌باشد. بدین منظور پس از اعمال یک سری پیش‌پردازش‌های هندسی بر روی شبکه‌ی مورد نظر برای سرعت بخشیدن به روند جستجوی الگوریتم از یک محدوده‌ی جستجو حول نود مبدا و مقصد استفاده می‌شود. در الگوریتم پیشنهادی، تابع هزینه به صورت یک عدد مختلط تعریف می‌شود که قسمت حقیقی آن نشان‌دهنده‌ی مجموع وزن یال‌های واقعی و قسمت موهومی آن نشان‌دهنده‌ی تعداد یال‌های مجازی و در واقع تعداد عدم اتصالات بین نودها در کروموزوم‌های الگوریتم ژنتیک می‌باشد. همچنین در بحث اعمال جهش بر روی کروموزوم‌های الگوریتم ژنتیک، از الگوریتم جستجوی ممنوع استفاده می‌گردد. علت پیشنهاد این روش، جدید بودن و نیز زمان‌بر بودن روش‌های قطعی مثل الگوریتم دایجسترا و نیز جواب نامناسب الگوریتم ژنتیک خالص (غیر ترکیبی) از لحاظ وزن نهایی مسیر در حل مسئله‌ی مسیریابی در شبکه‌های واقعی بخصوص شبکه‌های بزرگ می‌باشد. به منظور ارزیابی کارایی الگوریتم پیشنهادی، الگوریتم بر روی یک شبکه‌ی واقعی جهت‌دار شامل ۷۳۹ نود و ۱۱۶۰ یال که بخشی از شبکه‌ی راه‌های شهر تهران می‌باشد، پیاده‌سازی شد. نتایج نشان می‌دهد که در الگوریتم پیشنهادی، طول مسیر تا حد ممکن به جواب حاصل از الگوریتم قطعی دایجسترا نزدیک است. این الگوریتم طول نهایی مسیر را ۵ درصد بیشتر پیش‌بینی می‌کند. اما از لحاظ سرعت اجرا به طور متوسط ۵/۱۲ برابر نسبت به الگوریتم دایجسترا سریع‌تر است. در مقایسه با الگوریتم ژنتیک خالص نیز الگوریتم پیشنهادی از نظر طول مسیر به طور متوسط ۹ درصد کوتاه‌تر می‌باشد و از نظر زمان اجرا سرعت الگوریتم پیشنهادی با الگوریتم ژنتیک خالص تقریباً برابر است. همچنین به لحاظ قابلیت تکرارپذیری نیز الگوریتم پیشنهادی ۲۵/۳۶ درصد، تکرارپذیری را نشان می‌دهد.

واژگان کلیدی: یافتن کوتاه‌ترین مسیر، الگوریتم ژنتیک، الگوریتم جستجوی ممنوع، پیش‌پردازش هندسی، محدوده‌ی جستجو

* نویسنده رابط

۱- مقدمه

حمل و نقل درون شهری امروزه یکی از مسائل مهم زندگی شهری خصوصا در شهرهای بزرگ می باشد. توسعه‌های شهری و به دنبال آن افزایش مسیرها و ارتباطات، نقش جابه‌جایی را پررنگ‌تر می‌کند. مسئله کوتاه ترین مسیر در یک شبکه از جمله اساسی ترین مسائل تحلیل شبکه است. در واقع برای انجام تحلیل های سطوح بالاتر، یافتن پاسخ این مسئله از ضروریات است [۱].

از طرف دیگر و به طور کلی، روش‌های حل مسائل با استفاده از الگوریتم های بهینه سازی، به دو دسته الگوریتم های دقیق و تخمینی تقسیم بندی می‌شوند. در الگوریتم های دقیق، دستیابی به جواب بهینه، نیازمند صرف زمان و در نتیجه هزینه بیشتری است. اما در طرف مقابل، الگوریتم های تخمینی قرار دارند که در یک زمان اندک به جواب می‌رسند. در این میان، الگوریتم های فراابتکاری، از جمله الگوریتم های تخمینی هستند که توانایی یافتن جواب های بهینه و یا نزدیک به بهینه را در زمان معقول دارا می‌باشند [۲]. در این راستا و تاکنون، روش‌های بهینه‌سازی و فراابتکاری برای حل مسائل مسیریابی، زیاد مورد استفاده قرار گرفته اند که در ادامه به برخی از آنها اشاره شده است.

گلنارکار و همکاران براساس تئوری فازی به حل مسئله FSPP یا همان مسئله کوتاه ترین مسیر فازی که هدف آن مسیریابی در شبکه های چند ساختی در شبکه حمل و نقل شهری می باشد، پرداخته اند. در این تحقیق وزن یال ها بر اساس تابع دیود فازی شده است و بر همین اساس به حل مسئله مسیریابی در یک شبکه چند ساختی شبیه سازی شده با تعداد محدود یال و راس می پردازد. هم چنین در این تحقیق از تکنیک افزایش سرعت الگوریتم Contraction استفاده شده است [۳].

آمبروزینو و اسموخن یک الگوریتم جدید برای کوتاه ترین مسیر روی شبکه های چند ساختی ارائه دادند که تابع هدف از اجزای مختلفی از قبیل: هزینه(پول)، زمان و ناراحتی که مسافران در هنگام تغییر ساخت متحمل می شوند، تشکیل شده است. ویژگی کلیدی الگوریتم پیشنهاد شده این است که بر روی نود های تغییر ساخت(مود) تمرکز می کند و مسیر یابی ها را تا آنجا که ممکن است مجبور به عبور از آن نود ها می کند. آنها نتایج حاصل از

یک آزمایش محاسباتی با هدف اعتبارسنجی الگوریتم پیشنهاد شده بر روی اندازه های مختلف شبکه های چندساختی به همراه یک مطالعه موردی مربوط به شهر جنوا ایتالیا ارائه دادند [۴].

بهبودی و همکاران به مسیر یابی بهینه درون شهری به کمک الگوریتم ژنتیک در فضای رستر پرداختند. در این تحقیق تعداد کروموزوم ها برابر تعداد سلول های موجود بین مبدأ-مقصد در نظر گرفته شد و عملگر تقاطع به شکل تقاطع چند نقطه ای (کمتر از تعداد کل کروموزوم ها) اعمال شد. جهش نیز در بخشی از کل تعداد کروموزوم ها بدین صورت رخ داد که نقطه مورد نظر برای جهش، کمتر از دو سلول با سلول جهش یافته قبلی، فاصله داشته باشد [۵].

زک زوک و همکاران از الگوریتم کلونی مورچه برای حل مسئله کوتاه ترین مسیر در حالتی که هزینه یال ها عدد فازی مثلثی است، استفاده کردند. آنها الگوریتم خود را در هشت مرحله با تعداد ۶۰ مورچه مورد آزمایش قرار دادند. نتایج نشان داد که با افزایش تعداد مورچه ها نتیجه ای بهتر در زمان مناسب تر تولید می شود [۶].

عباسپور و صمدزادگان از ارائه الگوریتم تکاملی ژنتیک برای حل مسئله کوتاه ترین مسیر در شبکه چندساختی حمل و نقل در کلان شهرها استفاده نمودند، این تحقیق برای شبکه دو ساختی، اتوبوس و مترو، به صورت تک هدفه برای قسمتی از شبکه حمل و نقل عمومی شهر تهران پیاده شد که نتایج مثبتی در کاربرد این الگوریتم نشان می دهد [۷].

دیب و همکاران یک رویکرد ابتکاری توسعه یافته که به موجب آن یک الگوریتم ژنتیک (GA) با یک جستجوی همسایگی متغیر(VNS) ترکیب می شود را برای حل مسئله کوتاه ترین مسیر چند معیاره (MSPP) در شبکه های چند منظوره پیشنهاد دادند. نتایج نشان دادند که ترکیب پیشنهادی GA-VNS بهترین روش از نظر زمان محاسباتی و کیفیت راه حل برای یک سیستم مسیریابی در جهان واقعی را نمایش می دهد [۸].

آتیراتانا و همکاران به تحلیل زمان تداوم الگوریتم بهینه سازی کلونی مورچگان، برای یافتن کوتاه ترین مسیر پرداخته اند و نتایج و دستاوردهای خود را در این زمینه ارائه نموده اند [۹]. قصیری و همکاران نیز، از الگوریتم بهینه سازی کلونی مورچگان برای حل مسئله کوتاه ترین مسیر دو هدفه استفاده نمودند. این الگوریتم در مقایسه با

الگوریتم انجام می‌شود. در بخش چهارم نیز نتایج حاصل از اجرای الگوریتم بر روی شبکه‌ی مورد نظر با نتیجه‌ی حاصل از الگوریتم قطعی دایجسترا و همچنین نتیجه‌ی حاصل از الگوریتم ژنتیک غیر ترکیبی مقایسه می‌شود و همچنین نتیجه حاصل از چند بار تکرار الگوریتم پیشنهادی بر روی شبکه بررسی می‌شود. در بخش انتهایی نیز، نتایج اجرای الگوریتم روی شبکه مورد بحث و بررسی قرار گرفته و نتیجه‌گیری تحقیق ارائه خواهد شد.

۲- مبانی نظری

۲-۱- مسئله یافتن کوتاه‌ترین مسیر

مسئله کوتاه‌ترین مسیر^۱، بیشتر از ۴۰ سال است که در زمینه‌های گوناگون مانند علوم کامپیوتر و حمل‌ونقل مورد مطالعه قرار گرفته است. مسئله عام کوتاه‌ترین مسیر، نیاز به یک شبکه از پیش تعریف شده دارد. مسئله اصلی، تعیین یک یا تعداد بیشتری از کوتاه‌ترین مسیرهای بین یک مبدا و مقصد با یک سری خطوط داده شده می‌باشد. تعیین کوتاه‌ترین مسیر، می‌تواند به صورت رابطه (۱)، به عنوان یک مسئله برنامه ریزی خطی مشخص شود.

$$\begin{aligned}
 Z &= \sum \sum c \\
 Z &= \sum \sum C_{ij} * X_{ij} \\
 &\text{subject to} \\
 \sum X_{ji} - \sum X_{ik} &= m, \text{ where} \\
 m &= 0 \text{ for } i \neq s \\
 m &= 1 \text{ for } i = t \\
 m &= -1 \text{ for } i = s \\
 X_{ij} &\in \{0,1\}
 \end{aligned} \quad (1)$$

s به عنوان نقطه مبدا، t به عنوان نقطه مقصد و $C_{ij} > 0$ هزینه یا فاصله ارتباط کمان یا یال (i,j) می‌باشد. هدف جستجوی حداقل مقدار Z است [۱۴].

۲-۲- الگوریتم ژنتیک

الگوریتم ژنتیک^۲، الهامی از علم ژنتیک و نظریه‌ی تکامل داروین است و بر اساس بقای برترین‌ها یا انتخاب طبیعی استوار است [۱۵]. یک کاربرد متداول الگوریتم ژنتیک، استفاده از آن به عنوان تابع بهینه‌کننده است. در

الگوریتم تصحیح برچسب، توانست جواب‌هایی با کیفیت بالاتر و در زمان سریع‌تری را ارائه دهد [۱۰].

لینزنگ لیو و همکاران از یک درخت پوششی جهت‌دار (OST) مبتنی بر الگوریتم ژنتیک، برای حل مسائل یافتن کوتاه‌ترین مسیر چندمعیاری (MCSPP) استفاده کردند [۱۱]. در پژوهشی دیگر، کیوری و همکاران، برای مسیریابی از الگوریتم فراابتکاری جستجوی ممنوع استفاده نمودند [۱۲]. موحد و همکاران نیز، در مقاله خود به حل مسئله یافتن کوتاه‌ترین مسیر در شبکه با استفاده از الگوریتم بهینه‌سازی جمعی ذرات پرداخته‌اند [۱۳].

در این تحقیق هدف توسعه‌ی یک مدل تصمیم‌گیری برای تعیین مسیر بهینه در محیط یک شبکه حمل و نقل شهری با در نظر گرفتن معیار فاصله‌ی هندسی به عنوان وزن یال‌های شبکه با استفاده از ترکیب و ادغام الگوریتم‌های بهینه‌سازی فراابتکاری ژنتیک و جستجوی ممنوع می‌باشد به این صورت که برای اعمال جهش بر روی کروموزوم‌ها در الگوریتم ژنتیک از الگوریتم جستجوی ممنوع استفاده خواهد شد. همچنین نوآوری دیگر این پژوهش تعریف تابع هدف به صورت یک عدد مختلط می‌باشد. با توجه به اینکه در مسئله‌ی ما یک سری اتصالات واقعی و یک سری اتصالات مجازی بین نودهای شبکه وجود دارد، تنها حالتی که بتوان دو عدد را به صورت یک عدد نشان داد و حالت برداری هم نداشته باشد، استفاده از عدد مختلط می‌باشد. اهداف دیگر این پژوهش بررسی سرعت همگرایی، زمان اجرای الگوریتم پیشنهادی نسبت به یک الگوریتم قطعی مسیریابی (الگوریتم دایجسترا) و همچنین نسبت به یک الگوریتم ژنتیک خالص (غیر ترکیبی) و در نهایت بررسی قابلیت تکرارپذیری الگوریتم پیشنهادی در تکرارهای مختلف الگوریتم بر روی نقاط مبدا و مقصد یکسان در یک شبکه‌ی حمل و نقل واقعی شهری می‌باشد. بدین منظور و در ادامه پس از تشریح مسئله‌ی یافتن کوتاه‌ترین مسیر و الگوریتم‌های فراابتکاری ژنتیک و جستجوی ممنوع در بخش دوم، در بخش سوم و به منظور پیاده‌سازی الگوریتم پیشنهادی بر روی شبکه ابتدا یک پیش‌پردازش بر روی نقاط شبکه اعمال می‌شود و سپس پارامترها و متغیرهای مسئله‌ی مسیریابی به پارامترهای الگوریتم ژنتیک پیشنهادی نسبت داده می‌شود و با اعمال عملگرهای الگوریتم ژنتیک بر روی مسئله و استفاده از الگوریتم جستجوی ممنوع کار پیاده‌سازی

^۱ Shortest Path Problem

^۲ Genetic Algorithm

الگوریتم‌های ژنتیکی، نحوه تکامل ژنتیکی موجودات زنده شبیه‌سازی می‌شود [۱۶]. این الگوریتم‌ها با الهام از روند تکاملی طبیعت مسائل را حل می‌نمایند. یعنی مانند طبیعت یک جمعیت از موجودات را تشکیل می‌دهند و با اعمالی بر روی این مجموعه به یک مجموعه بهینه و یا موجود بهینه دست می‌یابند [۱۷]. برای اینکه بتوان یک مسئله را بوسیله الگوریتم‌های ژنتیک حل کرد، بایستی آن را به فرم مخصوص مورد نیاز این الگوریتم‌ها تبدیل کرد. در این روند بایستی راه حل مورد نیاز مسئله را به گونه‌ای تعریف کرد که قابل نمایش بوسیله یک کروموزوم باشد [۱۸].

تابع شایستگی^۱: برای اینکه بتوان موجودات بهتر را درون جمعیت شناسایی کرد، بایستی معیاری را تعریف کرد که بر اساس آن بتوان موجودات بهتر را مشخص کرد. این معیار تابع شایستگی (برازش) نامیده می‌شود. بسته به نوع مسئله بیشینه کردن و یا کمینه کردن تابع شایستگی مورد نظر است. به عنوان مثال در مورد تابع هزینه و تابع خطا کمینه کردن تابع شایستگی مطرح است و در مورد تابع برازندگی و تابع سود بیشینه‌سازی مورد نظر است [۱۹].

الگوریتم ژنتیک با استفاده از یک سری عملگرها مسئله بهینه‌سازی را حل می‌کند. این عملگرها عبارتند از: ۱) عملگر انتخاب^۲ (انتخاب والدین): برای سوق دادن جستجو به بخش‌هایی از فضا که امکان یافتن جواب‌های با کیفیت بالاتر وجود دارد از عملگر انتخاب استفاده می‌شود. این عملگر نسل جدیدی از راه حل‌ها را با انتخاب والدینی که بالاترین شایستگی را دارند تولید می‌کند [۲۰]. روشهای مختلفی برای انتخاب والدین وجود دارد. چند مورد از این روش‌ها قابل ذکر است:

- I انتخاب تصادفی^۳: بصورت تصادفی تعدادی از موجودات جمعیت به عنوان والدین انتخاب می‌شوند، این انتخاب می‌تواند با جایگذاری و یا بدون جایگذاری باشد.
- II انتخاب چرخ رولت^۴: یک روش انتخاب است که در آن عنصری که عدد برازش (شایستگی) بیشتری داشته باشد، انتخاب می‌شود.

III. انتخاب رقابتی^۵: یک زیر مجموعه از صفات یک جامعه انتخاب می‌شوند و اعضای آن مجموعه با هم رقابت می‌کنند و سرانجام فقط یک صفت از هر زیرگروه برای تولید انتخاب می‌شوند [۲۱].

۲) عملگر تقاطع یا ترکیب مجدد^۶: این عملگر امکان ترکیب جواب‌های جزئی^۷ یافت شده و در نتیجه بدست آوردن جواب‌هایی با کیفیت بالاتر را فراهم می‌آورد. در جریان عمل بازترکیبی به صورت اتفاقی بخش‌هایی از کروموزوم‌ها با یکدیگر تعویض می‌شوند. این موضوع باعث می‌شود که فرزندان ترکیبی از خصوصیات والدین خود را به همراه داشته باشند و دقیقاً مشابه یکی از والدین نباشند. هدف تولید فرزند جدید می‌باشد به این امید که خصوصیات خوب دو موجود در فرزندشان جمع شده و یک موجود بهتری را تولید کند [۲۲]. سه نوع عملیات تقاطع قابل انجام است:

- I تقاطع تک نقطه ای^۸
- II تقاطع دو نقطه ای^۹
- III تقاطع چند نقطه ای^{۱۰}

۳) عملگر جهش^{۱۱}: ویژگی تصادفی بودن و امکان فرار از نقاط بهینه محلی را فراهم می‌آورد [۲۳] (شکل ۱)

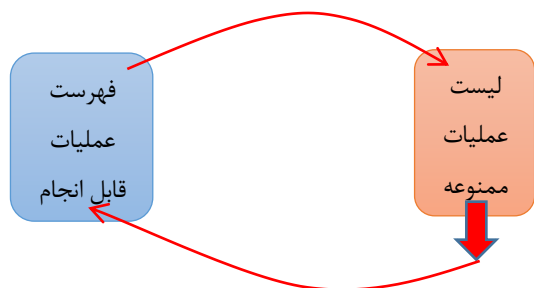
۲-۳- الگوریتم جستجوی ممنوع^{۱۲}

این الگوریتم مبتنی بر جمعیت نیست، بلکه مبتنی بر نقطه است، یعنی این که یک جواب را می‌گیرد و سعی می‌کند که همان جواب را با اعمال یک سری اپراتورها به سمت جواب بهینه مسئله هدایت کند و از این نظر شبیه الگوریتم SA^{۱۳} است [۲۴]. لغت Tabu در متون قدیمی بیشتر به معنی مقدس است ولی ممنوع هم ترجمه می‌شود [۲۵].

در الگوریتم TS به این شکل عمل می‌شود که الگوریتم از یک نقطه شروع می‌کند و این جواب را با یک سری از عملیات (اپراتورها) می‌توان به جواب بهینه تبدیل کرد [۲۶].

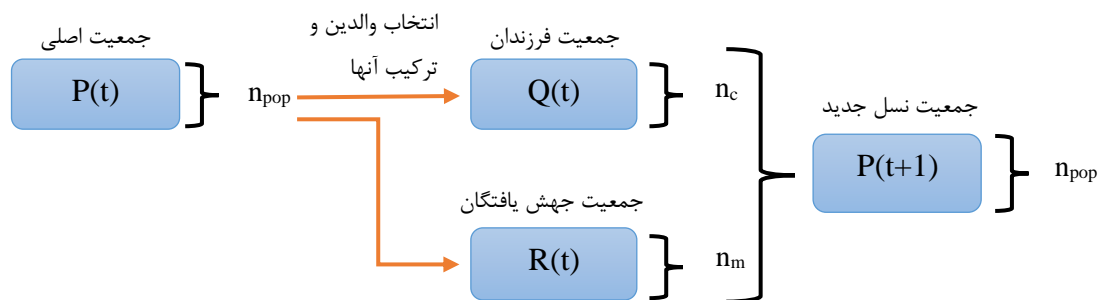
۵ Tournament Selection
 ۶ Recombination/Crossover
 ۷ partial solutions
 ۸ Single Point Crossover
 ۹ Two Point Crossover
 ۱۰ Uniform Crossover
 ۱۱ Mutation
 ۱۲ Tabu Search Algorithm
 ۱۳ Simulated Annealing

۱ Fitness Function
 ۲ Selection
 ۳ Random Selection
 ۴ Roulette Wheel Selection



شکل ۲- الگوریتم جستجوی ممنوع

لیست Tabu شامل یک سری عمل (حرکت) است که رفته رفته اضافه می‌شوند. جدیدترین عمل به بالای لیست می‌آید و یک عمل از پایین لیست حذف می‌شود [۲۷]. در الگوریتم جستجوی ممنوع همیشه بحث جهت حرکت نیست، بلکه شدت یک حرکت، نحوه‌ی انجام یک حرکت و ... می‌تواند به عنوان عمل‌های ما باشند. در بسیاری از موارد همین که گیرافتادن در مینیمم محلی اتفاق نیفتد، برای ما کافی است و نیازی به رسیدن به جواب نیست، یعنی ممکن است به جواب نرسید [۲۸] (شکل ۲).



شکل ۱- الگوریتم ژنتیک به صورت گرافیکی

۳-۲- پیاده‌سازی الگوریتم برای یافتن کوتاه-

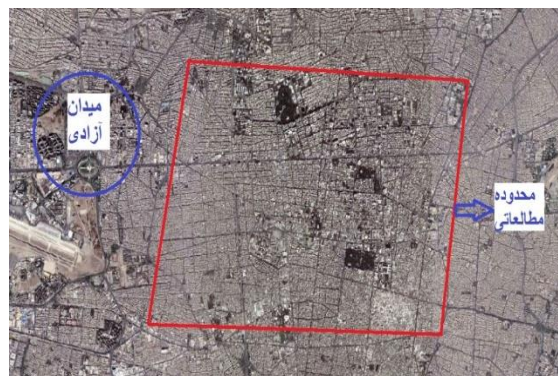
ترین مسیر

برای پیاده‌سازی ابتدا یک سری پیش پردازش هندسی شامل انتقال مبدا مسیر به مبدا مختصات و دوران شبکه حول مبدا بر روی شبکه انجام می‌شود. جمعیت اولیه در الگوریتم پیشنهادی به صورت تصادفی از بین نودهای شبکه انتخاب می‌شود. با توجه به تصادفی بودن جمعیت اولیه این امکان وجود دارد که نودهای تشکیل دهنده مسیرهای اولیه بسیار پراکنده واقع شوند. برای رفع این مشکل، یک محدوده‌ی جستجو حول نقطه‌ی مبدا و مقصد تعریف خواهد شد. در ادامه تابع هزینه برای این مسئله به صورت یک عدد مختلط تعریف می‌شود. در صورتی که معیار توقف تعریف شده برای مسئله برآورده شده باشد الگوریتم خاتمه می‌یابد، در غیر این صورت فرآیند حل مسئله وارد الگوریتم ژنتیک می‌شود و اپراتورهای تقاطع و جهش برای مسئله‌ی مسیریابی تعریف می‌شود. برای اعمال جهش در الگوریتم پیشنهادی از الگوریتم جستجوی ممنوع استفاده خواهد شد. فرآیند اعمال اپراتورهای تقاطع و جهش آنقدر ادامه پیدا می‌کند تا معیار توقف تعریف شده برآورده شود. شکل زیر گام‌های

۳- پیاده‌سازی

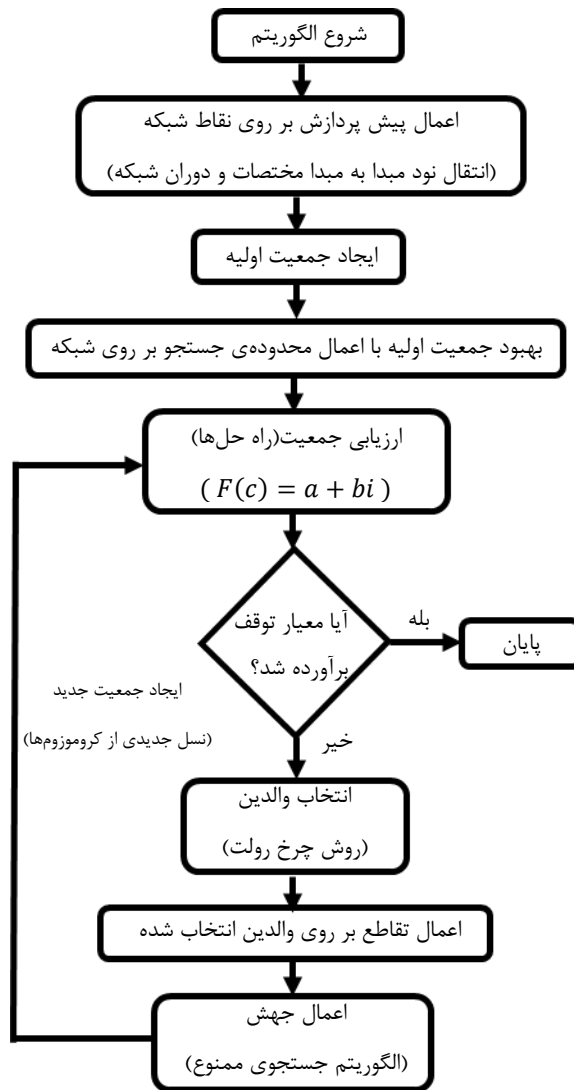
۱-۳- محدوده‌ی مطالعاتی

محدوده‌ی مورد مطالعه در این پژوهش، بخشی از شبکه-ی راه‌های اصلی شهر تهران به ابعاد تقریبی ۶*۷ کیلومتر می‌باشد که از نظر موقعیت جغرافیایی تقریباً در ۳۵ درجه و ۴۱ دقیقه و ۴۸ ثانیه عرض شمالی و ۵۱ درجه و ۲۳ دقیقه و ۴۷ ثانیه طول شرقی و در ارتفاعی معادل ۱۲۶۱ متر از سطح دریا واقع شده است (شکل ۳). شهر تهران به عنوان پرجمعیت‌ترین شهر ایران جمعیتی بالغ بر ۸ میلیون و ۷۳۷ هزار و ۵۱۰ نفر را در خود جای داده است.



شکل ۳- محدوده‌ی مطالعاتی

الگوریتم پیشنهادی را نشان می‌دهد که با جزئیات بیشتر در ادامه بحث خواهد شد (شکل ۴).



شکل ۴- گام‌های الگوریتم پیشنهادی

۳-۲-۱- اعمال پیش‌پردازش بر روی نقاط شبکه

قبل از شروع حل مسئله با الگوریتم ژنتیک، پردازش‌ها و تغییراتی بر روی شبکه انجام می‌شود. با اعمال این پردازش‌ها، اپراتورهای الگوریتم ژنتیک آسان‌تر برای مسئله تعریف می‌گردند و می‌توان برای الگوریتم یک محدوده جستجو تعیین کرد. این پردازش‌ها به صورت زیر بر روی شبکه اعمال می‌شوند:

در ابتدا نقطه شروع و پایان مسیر مشخص می‌گردد. با توجه به مختصات نقطه شروع، تمام نقاط موجود در شبکه در راستای محور x و y جابجا می‌شوند به طوری

که مختصات جدید نقطه شروع (0,0) گردد. این جابجایی به صورت رابطه (۲) تعریف می‌گردد:

$$\begin{aligned} & Start_Point(X_{Start}, Y_{Start}) \\ & End_Point(X_{End}, Y_{End}) \\ & Shift_{Point} \rightarrow \\ & \left\{ \begin{array}{l} \text{for All Points} \\ (x_i, y_i) \rightarrow (x_i, y_i) - (X_{Start}, Y_{Start}) \\ \text{end} \end{array} \right. \end{aligned} \quad (2)$$

مرحله دیگر پردازش، دوران شبکه می‌باشد. شبکه موجود حول نقطه شروع به مختصات (0,0) دوران داده می‌شود به طوری که نقطه انتهایی مسیر در جهت مثبت محور x قرار گیرد. در این صورت مختصات جدید نقطه دوم به صورت $(X_{End}^{new}, 0)$ می‌باشد که مقدار X_{End}^{new} یک مقدار مثبت می‌باشد [۲۹].

این دوران به صورت روابط (۳) و (۴) تعریف می‌گردد:

$$\begin{aligned} \theta &= Atan\left(\frac{Y_{End} - Y_{Start}}{X_{End} - X_{Start}}\right) \\ & \left\{ \begin{array}{l} \text{for All Points} \\ \alpha_i = Atan\left(\frac{Y_i - Y_{Start}}{X_i - X_{Start}}\right) \\ Rotated_Angle_{(i)} = \alpha_i - \theta \\ Distance_{(i)} = \sqrt{(X_i - X_{Start})^2 + (Y_i - Y_{Start})^2} \\ \text{end} \end{array} \right. \end{aligned} \quad (3)$$

$$\begin{aligned} & Transferred_{Point} \rightarrow \\ & \left\{ \begin{array}{l} X_{new(i)} = Distance_{(i)} * Cos(Rotated_Angle_{(i)}) \\ Y_{new(i)} = Distance_{(i)} * Sin(Rotated_Angle_{(i)}) \\ \text{end} \end{array} \right. \end{aligned} \quad (4)$$

۳-۲-۲- ایجاد جمعیت اولیه

برای حل یک مسئله بهینه‌سازی لازم است تا مقادیر متغیرهای مسئله به فرم یک آرایه شکل گیرند. در الگوریتم ژنتیک این آرایه با نام «کروموزوم» مشخص می‌شود. در یک مسئله بهینه‌سازی $Nvar$ -بعدی تعداد متغیرهای مسئله که باید بهینه شوند، یک کروموزوم یک آرایه $1 \times Nvar$ خواهد بود (رابطه ۵)، که در مسئله یافتن کوتاه‌ترین مسیر هر ژن یک کروموزوم یک نود(گره) از شبکه را نشان می‌دهد [۳۰]. به عنوان مثال:

$$pop(i). Position = [N_{start}, N_2, N_3, N_4, \dots, N_{nvar-1}, N_{end}]_{1 \times Nvar} \quad (5)$$

در این مقاله، با توجه به ابعاد شبکه که دارای ۷۳۹ نود و ۱۱۶۰ یال می‌باشد، تعداد کروموزوم‌ها برابر ۸۰ و

تعریف شود که این ارتباط در آن مشخص شود. در این مسئله تابع برازش به صورت یک عدد مختلط تعریف می-شود (رابطه ۷):

$$F(c) = a + bi \quad (7)$$

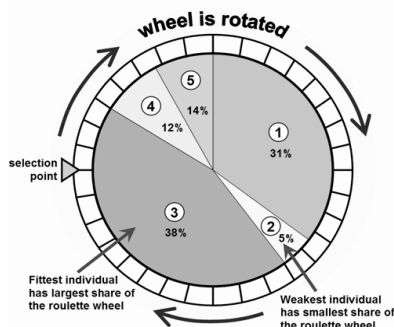
که در آن:

a : برابر با جمع وزن تمام یال‌های موجود در داخل مسیر، در صورتی که بین دو گره مجاور ارتباط واقعی وجود داشته باشد.

و b : نشان‌دهنده‌ی تعداد یال‌های مجازی و در واقع تعداد عدم اتصال بین گره‌های مجاور داخل مسیر می‌باشد. با توجه به تابع برازش تعریف شده، در صورتی که قسمت موهومی عدد مختلط (b) صفر شود، بدین معناست که مسیر تولید شده یک مسیر واقعی بین نقاط مبدا و مقصد می‌باشد و در بین مسیرهایی که مقدار b آن‌ها صفر باشد، هدف یافتن مسیری است که پارامتر a آن (که نشان‌دهنده‌ی جمع وزن یال‌ها است) کمترین مقدار باشد و آن مسیر به عنوان کوتاه‌ترین مسیر معرفی خواهد شد.

۳-۲-۵- عملگر انتخاب در الگوریتم ژنتیک

همانطور که قبلاً گفته شد این عملگر نسل جدیدی از راه حل‌ها را با انتخاب والدینی که بالاترین $Fitness$ را دارند تولید می‌کند. روش‌های مختلفی برای انتخاب والدین وجود دارد که در بخش مبانی نظری به آن‌ها اشاره کردیم. در این مقاله از روش انتخاب چرخ رولت استفاده می‌گردد (شکل ۵).



شکل ۵- روش انتخاب چرخ رولت [۳۱]

به این منظور ابتدا مقدار تابع هزینه (برازش) برای تمام کروموزوم‌ها محاسبه می‌شود و بخش موهومی تابع هزینه به عنوان معیار برای تعیین کیفیت کروموزوم‌ها در نظر گرفته می‌شود (رابطه ۸):

تعداد ژن‌های هر کروموزوم (در واقع تعداد متغیرهای مسئله) برابر ۲۰۰ در نظر گرفته می‌شوند که این ژن‌ها در هنگام ایجاد جمعیت اولیه به صورت کاملاً تصادفی از بین نودهای شبکه انتخاب می‌شوند.

۳-۲-۳- اعمال محدوده‌ی جستجو بر روی شبکه

با توجه به تصادفی بودن انتخاب نودهای شبکه برای ایجاد جمعیت اولیه الگوریتم، این امکان وجود دارد که نودهای تشکیل دهنده‌ی یک کروموزوم بسیار پراکنده واقع شده و از تمامی فضای شبکه انتخاب شوند. در صورتی که به نودهایی که نسبت به نقطه‌ی مبدا و مقصد خیلی فاصله داشته باشند، با توجه به نوع مسئله که یافتن کوتاه‌ترین مسیر می‌باشد، نیازی نیست و بایستی حذف شوند. بنابراین فضای انتخاب نودها برای تشکیل جمعیت اولیه محدود می‌گردد. بدین منظور فاصله‌ی بین نقطه مبدا و مقصد که برابر مقدار X_{end}^{new} می‌باشد بر یک عدد صحیح تقسیم می‌شود (رابطه ۶).

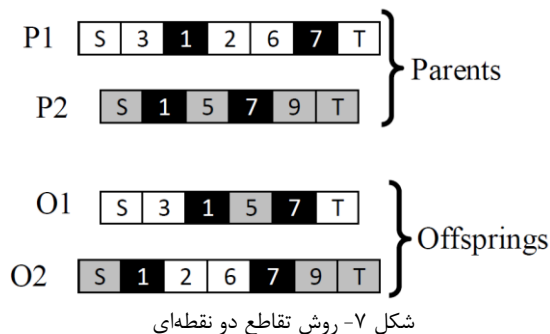
$$Limited_Distance = \frac{(X_{end}^{new} - X_{start}^{new})}{Div} \quad (6)$$

در این رابطه Div به صورت تجربی و وابسته به مسئله است که عددی صحیح بین ۱ تا ۵ در نظر گرفته می‌شود. فضای جستجو در راستای محور X ها به اندازه $Limited_Distance$ در سمت راست نقطه‌ی مقصد و به همین اندازه در سمت چپ نقطه‌ی مبدا محدود می‌شود. در راستای محور Y ها نیز، به اندازه $Limited_Distance$ در بالا و پایین محور X ها، یعنی نقاطی که قدرمطلق مختصه‌ی Y آن‌ها از مقدار $Limited_Distance$ کمتر باشد، داخل فضای جستجو قرار می‌گیرند و بقیه نودهایی که خارج از فضای جستجو در تشکیل کروموزوم‌های اولیه قرار دارند، فیلتر شده و حذف می‌شوند. این کار باعث سریع‌تر شدن فرایند جستجوی جواب‌ها و یافتن کوتاه‌ترین مسیر بین گره مبدا و مقصد خواهد شد [۵].

۳-۲-۴- تابع هزینه (برازش)

هر کروموزوم به عنوان یک مسیر بین نقطه‌ی مبدا و مقصد، یک مجموعه از نودها می‌باشد که به صورت تصادفی انتخاب شده‌اند. ارتباط بین این نودها می‌تواند واقعی و یا مجازی باشد. بنابراین تابع برازش باید به نحوی

۲) روش تقاطع دو نقطه‌ای: برای اعمال این عملگر نیز همانند روش تقاطع تک نقطه‌ای عمل می‌شود با این تفاوت که در این روش دو نود از بین نودهای مشترک به تصادف انتخاب می‌شوند. نقطه‌ای که مختصه X^{new} کمتری دارد به عنوان نقطه‌ی تقاطع اول و نقطه‌ای که X^{new} آن بیشتر است به عنوان نقطه‌ی تقاطع دوم انتخاب می‌شود و نودهای بین این دو نقطه‌ی مشترک در دو کروموزوم جابجا می‌شوند. (شکل ۷)



۳) روش تقاطع چند نقطه‌ای: اعمال این عملگر نیز مانند دو عملگر قبلی می‌باشد، با این تفاوت که در این عملگر تمام نقاط مشترک بین دو کروموزوم انتخاب می‌شوند و نودهای موجود بین هر دو نقطه‌ی مشترک به صورت یک در میان در دو کروموزوم جابجا می‌شوند. اعمال عملگرهای تقاطع (تک نقطه‌ای، دو نقطه‌ای و چند نقطه‌ای) بر روی کروموزوم‌های والد انتخاب شده با استفاده از روش انتخاب چرخ رولت صورت می‌گیرد و در تولید هر دو فرزند از دو والد انتخاب شده با توجه به احتمالی که برای هر کدام از این عملگرها در نظر گرفته شده است، یکی از این عملگرها برای انجام تقاطع انتخاب می‌شوند.

۳-۲-۷- اعمال جهش بر روی کروموزوم‌ها با استفاده از الگوریتم جستجوی ممنوع

در این پژوهش، هر کدام از جهش‌هایی که قرار است در الگوریتم ژنتیک مورد استفاده قرار گیرند، به عنوان یک عملیات (حرکت) در الگوریتم جستجوی ممنوع تعریف می‌شوند و در واقع اعمال جهش بر روی کروموزوم‌ها به کمک الگوریتم جستجوی ممنوع انجام می‌شود و این موضوع نوآوری مهم این پژوهش می‌باشد. پس از اعمال تقاطع بر روی یک سری از کروموزوم‌ها نوبت به اعمال جهش بر روی یک سری دیگر از کروموزوم‌ها می‌رسد.

$$cost(i) = imag(F(c_i)) \quad (8)$$

و این مقدار بر اندازه کروموزوم نظیر آن تقسیم می‌گردد تا شایستگی کروموزوم را بهتر نشان دهد (رابطه ۹) (چون بعد از اعمال محدوده‌ی جستجو بر روی کروموزوم‌ها، طول کروموزوم‌ها با یکدیگر متفاوت خواهد بود).

$$p(i) = \frac{cost(i)}{Length(Chromosome(i))} \quad (9)$$

با توجه به اینکه مقدار p کوچکتر نشان‌دهنده‌ی جواب بهتر می‌باشد، برای اینکه در چرخ رولت احتمال انتخاب کروموزوم‌های بهتر (با مقدار p کوچکتر) افزایش یابد، تابع احتمال انتخاب به صورت رابطه (۱۰) تعریف می‌شود:

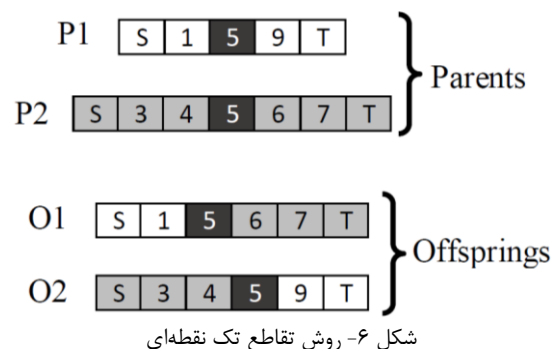
$$PS = \frac{1-p}{\sum(1-p)} \quad (10)$$

با این تعریف مقادیر احتمال انتخاب، نرمالیزه می‌شوند ($0 \leq PS \leq 1$) و جمع مقادیر آن‌ها برابر با یک خواهد شد.

۳-۲-۶- اعمال تقاطع بر روی کروموزوم‌های انتخاب شده

در اعمال عملگر تقاطع بر روی کروموزوم‌ها، از هر دو کروموزوم والد دو فرزند ایجاد می‌شود به این امید که از لحاظ مقدار تابع هزینه نسبت به والدین خود بهتر باشند. برای تولید هر دو فرزند ابتدا دو کروموزوم والد با استفاده از روش انتخاب چرخ رولت انتخاب می‌شوند و عملیات تقاطع بر روی این دو والد انجام می‌شود و در نتیجه‌ی آن دو فرزند ایجاد می‌شود. اعمال تقاطع بر روی کروموزوم‌ها به سه روش انجام می‌شود:

۱) روش تقاطع تک نقطه‌ای: برای اعمال این عملگر بر روی کروموزوم‌های والد، تمام نودهای مشترک بین دو کروموزوم (مسیر) مشخص می‌گردد و یکی از این نودها به تصادف انتخاب می‌شود. در این نود مشترک انتخاب شده، ژن‌های قبل و بعد در دو کروموزوم جابجا می‌شوند. (شکل ۶)



برای اعمال الگوریتم جستجوی ممنوع یک فهرست از عملیات قابل انجام که شامل جهش‌های ذکر شده می‌باشد، تشکیل می‌شود و هر کدام از جهش‌ها بعد از اعمال بر روی کروموزوم انتخاب شده به لیست عملیات ممنوعه انتقال داده می‌شوند و با توجه به طول لیست ممنوعه تا یک تعداد تکراری استفاده از آن جهش بر روی کروموزوم انتخاب شده ممنوع می‌شود و بعد از اضافه شدن جهش -های دیگر به لیست ممنوعه و پر شدن لیست، یک عملیات (جهش) از انتهای لیست آزاد می‌شود و دوباره می‌توان از آن جهش استفاده کرد. با این کار از احتمال اعمال شدن یک جهش تکراری به صورت متوالی بر روی کروموزوم‌ها شده جلوگیری می‌شود.

بعد از اعمال اپراتورهای تقاطع و جهش بر روی یک سری از کروموزوم‌ها و تولید فرزندان جدید، نیاز است تا کروموزوم‌هایی که از لحاظ مقدار تابع هزینه از بقیه بهتر هستند، باقی بمانند و بقیه کروموزوم‌ها حذف شوند. بنابراین به اندازه همان جمعیت اولیه (۸۰ کروموزوم) نگه داشته می‌شوند و بقیه کروموزوم‌ها حذف می‌شوند. این نسل جدید از کروموزوم‌ها به عنوان جمعیت اولیه برای تکرار بعدی الگوریتم انتخاب می‌شوند. این فرآیند تا زمانی ادامه می‌یابد که معیار توقف الگوریتم برآورده شود.

۴- ارزیابی نتایج

در این مرحله کارایی الگوریتم پیشنهادی بر روی شبکه‌ی موجود در محدوده‌ی مطالعاتی بررسی می‌شود. شبکه مذکور یک شبکه‌ی واقعی جهت‌دار با ۷۳۹ نود و ۱۱۶۰ یال می‌باشد. نقاط شروع و پایان مسیر به ترتیب نقاط ۱۰۰ و ۵۰۰ می‌باشد. هدف از مسیریابی در این شبکه، یافتن مسیری است که از لحاظ معیار فاصله کمترین وزن را داشته باشد. الگوریتم پیشنهادی بر روی شبکه مورد نظر چندین بار تکرار شد و نتایج حاصل از این تکرارها به همراه مقادیری که برای پارامترهای الگوریتم تنظیم شد در جدول (۱) ارائه شده است.

بهترین جواب به دست آمده از الگوریتم پیشنهادی با تعداد جمعیت ۸۰ کروموزوم و تعداد ۲۰۰ ژن برای هر کروموزوم با درصد ترکیب (تقاطع) ۸۰ درصد، درصد جهش ۳۰ درصد، نرخ جهش ۵ درصد در الگوریتم ژنتیک و تعداد عمل (حرکت) برابر با ۶ و طول لیست ممنوعه

برای اعمال جهش ابتدا یک کروموزوم به صورت تصادفی انتخاب می‌شود و با توجه به نرخ جهش تعیین شده (برای بعضی از انواع جهش)، تعدادی از ژن‌های کروموزوم برای اعمال جهش انتخاب می‌شوند و عملگر جهش بر روی آن ژن‌ها اعمال می‌شود. در این مقاله، جهش‌های زیر برای اعمال به کروموزوم تعریف می‌شوند:

(۱) حذف^۱: در این نوع جهش، نرخ جهش در طول کروموزوم ضرب می‌شود. عددی که به دست می‌آید، تعداد ژن‌هایی است که باید از کروموزوم حذف شوند. در واقع تعداد نودهایی است که بایستی از مسیر به دست آمده حذف شوند.

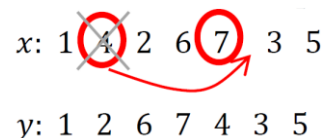
(۲) جابجایی^۲: در این جهش، یک ژن به صورت تصادفی بین نقطه‌ی شروع و پایان انتخاب می‌شود. در صورتی که ارتباط این ژن با ژن قبلی یا بعدی اش واقعی نباشد، این ژن با یک ژن دیگر در داخل محدوده‌ی جستجوی بین ژن قبل و بعد از ژن انتخاب شده که ارتباط آن با هر دو ژن قبلی و بعدی یک ارتباط واقعی باشد، جابجا می‌شود.

(۳) اضافه کردن^۳: در این جهش، در داخل کروموزوم انتخاب شده برای اعمال جهش، ارتباط بین هر نود با نود بعدی اش بررسی می‌شود. در صورتی که این ارتباط واقعی نباشد، یک نود به صورت تصادفی از داخل محدوده‌ی جستجوی بین دو نود انتخاب می‌شود و به کروموزوم انتخاب شده اضافه می‌گردد.

(۴) Swap: این اپراتور دو ژن از کروموزوم را به تصادف انتخاب کرده و جای آن‌ها را با هم عوض می‌کند.

(۵) معکوس کردن^۴: این اپراتور نیز دو ژن از کروموزوم را به تصادف انتخاب کرده و علاوه بر خود دو ژن، مسیری که بین آن‌ها قرار دارد نیز معکوس می‌شود.

(۶) حذف و انتقال^۵: در این اپراتور دو ژن به تصادف انتخاب می‌شوند و ژن اول حذف می‌شود و به جایگاه بعد از ژن دوم انتقال پیدا می‌کند. به عنوان مثال:



^۱ Remove Mutation

^۲ Shift Mutation

^۳ Insert Mutation

^۴ Reversion

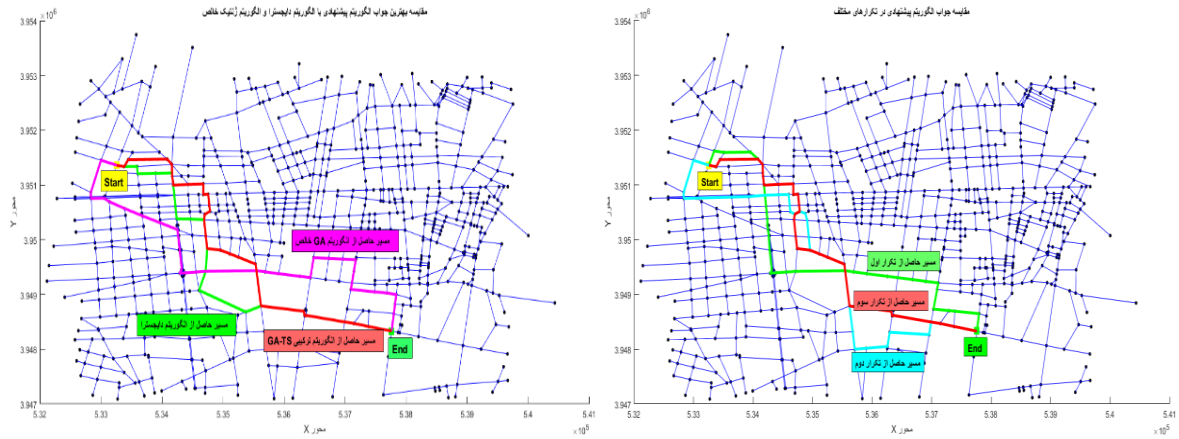
^۵ Insertion

برابر با ۳ برای الگوریتم جستجوی ممنوع و با تعداد ۱۰۰ تکرار برای حلقه‌ی اصلی الگوریتم پیشنهادی حاصل شد. وزن مسیر نهایی حاصل از بهترین جواب الگوریتم پیشنهادی برابر با ۷۱۱۳ متر به دست آمد که بسیار نزدیک به جواب حاصل از الگوریتم قطعی دایجسترا برابر ۷۰۷۳ متر می باشد. از نظر زمان اجرا، الگوریتم پیشنهادی بسیار سریع تر از الگوریتم دایجسترا مسئله‌ی مسیریابی را در این شبکه حل می‌کند و نسبت زمان اجرای الگوریتم پیشنهادی نسبت به الگوریتم دایجسترا به طور میانگین برابر با ۰/۱۹۵۲۴۸ می‌باشد. همچنین نتایج حاصل از الگوریتم پیشنهادی با نتایج حاصل از یک الگوریتم ژنتیک خالص (غیر ترکیبی) مقایسه شد. از لحاظ زمان اجرا پاسخ دو الگوریتم نزدیک به یکدیگر بوده اما از نظر طول مسیر (وزن نهایی مسیر) پاسخ الگوریتم

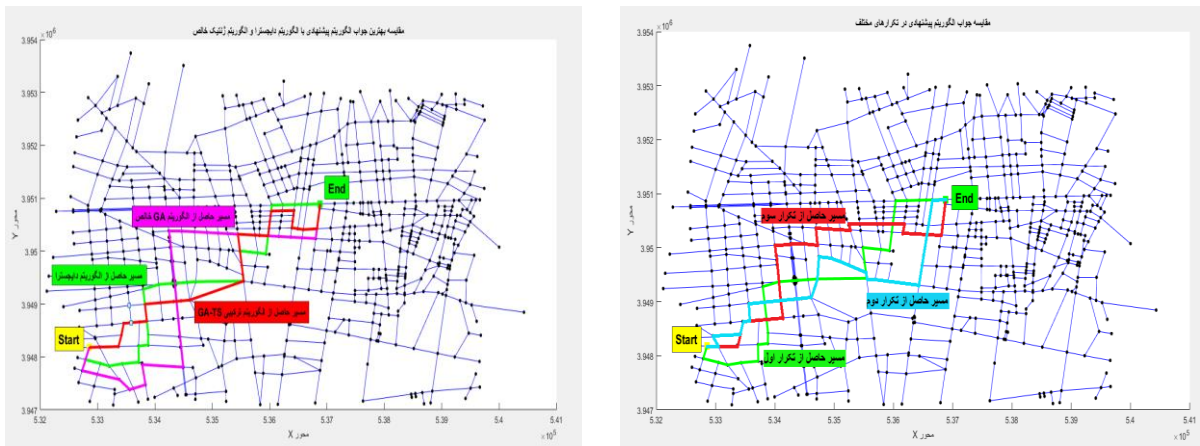
پیشنهادی با اختلاف قابل توجهی از الگوریتم ژنتیک خالص بهتر عمل کرده است. به لحاظ قابلیت تکرار پذیری الگوریتم نیز الگوریتم پیشنهادی در طی ۳ تکرار بین نودهای ۱۰۰ و ۵۰۰ از شبکه به جواب هایی رسید که تقریباً نزدیک بهم می‌باشند. در نهایت نیز بهترین جواب حاصل از الگوریتم پیشنهادی با جواب حاصل از الگوریتم دایجسترا و الگوریتم ژنتیک خالص مقایسه شد (شکل ۸). به منظور بررسی بیشتر کارایی الگوریتم پیشنهادی، این الگوریتم بین دو نود ۲۰۰ و ۴۳۵ از شبکه نیز اجرا شد و نتایج حاصل از آن کارایی الگوریتم را برای این ۲ گره نیز نشان می‌دهد (شکل ۹). همچنین کارایی الگوریتم بر روی یک شبکه بزرگ تر به ابعاد ۲۸۵۳ گره و ۴۴۸۲ یال نیز بررسی شد که نتایج حاصل از آن در شکل ۱۰ قابل مشاهده می باشد.

جدول ۱- نتایج حاصل از اجرای الگوریتم ترکیبی GA-TS و الگوریتم دایجسترا و الگوریتم ژنتیک خالص بر روی یک شبکه‌ی واقعی

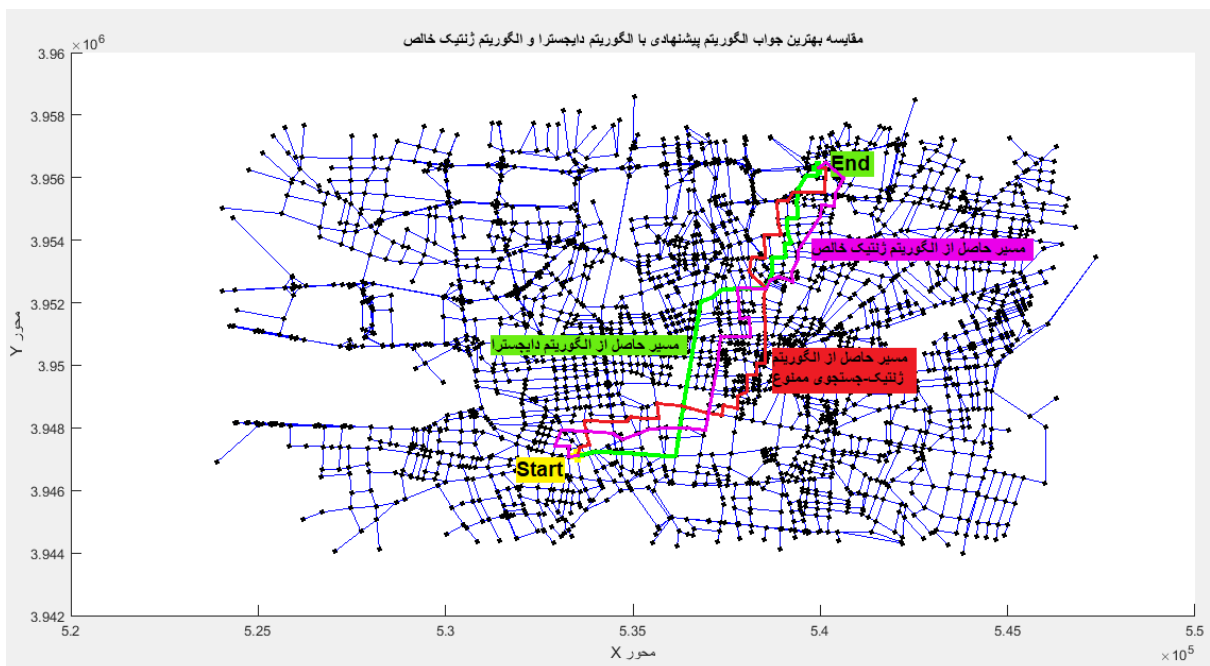
نوع الگوریتم	الگوریتم GA-TS (تکرار ۱)		الگوریتم GA-TS (تکرار ۲)		الگوریتم GA-TS (تکرار ۳)		الگوریتم ژنتیک خالص (غیر ترکیبی)		الگوریتم قطعی (دایجسترا)	
	۲/۴۴۹	۲/۹۰۳	۳/۵۸۸	۲/۹۰۷	۱۵/۲۶۵					
مدت زمان اجرای الگوریتم (ثانیه)	۱۰۰	۴۷	۱۰۰	۴۳	۳۷	۹۶	۴۳	۳	۱۰۰	۹۶
مسیر نهایی	۹۹	۸۱	۲۳۰	۲۳۰	۹۴	۲۳۰	۳۹۶	۲۲۶	۹۶	۹۷
	۹۵	۸۶	۱۶۶	۱۰۲	۹۲	۱۶۶	۵۵۰	۱۷۴	۲۲۶	۲۲۶
	۹۳	۱۴۰	۱۶۷	۲۳۷	۱۲۷	۱۶۷	۶۸۴	۱۷۴	۱۷۴	۱۷۴
	۹۲	۳	۲۳	۹	۲۳۷	۲۳۷	۴۸۲	۱۷۳	۱۷۳	۱۷۳
	۱۲۷	۳۹۶	۳	۳	۹	۲۳	۵۴۸	۲۲۷	۲۲۷	۱۵
	۱۷۳	۵۵۰	۲۳۲	۲۸۹	۳	۲۳۲	۳۶۸	۱۸۹	۲۲۷	۲۸۹
	۲۲۷	۶۸۴	۱۹۳	۱۶	۱۸۹	۱۳۷	۲۸۶	۴۶	۱۸۹	۲۹۱
	۱۸۹	۴۸۲	۱۳۵	۱۱	۴۶	۱۳۸	۲۸۶	۳۶۷	۱۳۷	۳۶۷
	۴۶	۲۸۵	۱۳۱	۱۱	۲۴۱	۱۹۶	۳۱۳	۱۳۱	۱۳۱	۲۸۸
	۱۳۱	۶۸۵	۱۳۰	۳۱۷	۱۵۴	۱۹۶	۳۱۲	۴۵	۱۳۱	۲۸۸
	۴۵	۳۱۱	۱۳۲	۳۱۵	۱۵۴	۹۰	۳۱۱	۴۵	۱۹۷	۵۰۶
	۱۹۷	۲۹۷	۲۷۱	۶۸۳	۱۵۵	۸۹	۲۹۷	۵۵	۱۹۷	۵۹۴
	۵۵	۴۹۸	۲۱۵	۲۸۷	۱۵۶	۸۰	۴۸۸	۵۵	۱۳۶	۵۰۵
	۵۴	۵۹۵	۲۱۹	۵۰۶	۲۱۱	۷۴	۴۸۵	۳۴	۱۳۶	۵۰۰
	۸۹	۵۰۱	۲۱۶	۵۹۴	۲۱۶	۷۶	۴۸۵	۲۲۰	۳۴	۳۴
	۸۰	۴۹۹	۲۱۷	۵۰۵	۲۱۹	۴۷	۴۸۶	۲۲۰	۲۳۹	۲۳۹
	۷۸	۷۶	۲۶۸	۵۰۰	۳۴	۸۱	۴۹۹	۱۴۰	۱۴۰	۱۴۰
		۲۶۷		۲۲۰	۱۴۰	۵۰۰	۱۱۳	۱۱۳	۱۱۳	
وزن مسیر (متر)	۷۵۳۸	۷۷۵۲	۷۱۱۳	۸۲۵۱	۷۰۷۳					
تعداد جمعیت در الگوریتم GA	۵۰	۷۰	۸۰	۸۰						
درصد ترکیب	۸۵	۸۰	۸۰	۸۰						
درصد جهش	۳۵	۲۵	۳۰	۳۰						
نرخ جهش	۰/۰۶	۰/۰۴	۰/۰۵	۰/۰۵						



شکل ۸- الگوریتم GA-TS در تکرارهای مختلف و مقایسه بهترین جواب آن با الگوریتم دایجسترا و الگوریتم ژنتیک خالص بین ۲ گره ۱۰۰ و ۵۰۰



شکل ۹- الگوریتم GA-TS در تکرارهای مختلف و مقایسه بهترین جواب آن با الگوریتم دایجسترا و الگوریتم ژنتیک خالص بین ۲ گره ۲۰۰ و ۴۳۵



شکل ۱۰- بررسی کارایی الگوریتم GA-TS بر روی یک شبکه بزرگ با ۲۸۵۳ گره و ۴۴۸۲ یال

۵- بحث و نتیجه‌گیری

با توجه به گسترش شهرها و پیچیده‌تر شدن شبکه‌ی راه‌های درون شهری به ویژه در شهرهای بزرگ، مسئله‌ی مسیریابی و در واقع یافتن کوتاه‌ترین مسیر تبدیل به یکی از دغدغه‌های افراد در هنگام تصمیم‌گیری برای انتخاب مسیر در جابجایی از مبدا حرکت به یک مقصد مشخص شده است. در این پژوهش به بررسی ادغام الگوریتم‌های فراابتکاری ژنتیک و جستجوی ممنوع برای حل مسئله‌ی مسیریابی در یک شبکه‌ی حمل و نقل درون شهری واقعی و جهت‌دار پرداخته شد.

پس از بررسی مسئله‌ی کوتاه‌ترین مسیر به عنوان یک مسئله‌ی برنامه‌ریزی خطی، ساختار الگوریتم‌های ژنتیک و جستجوی ممنوع برای حل مسئله‌ی مسیریابی مورد مطالعه قرار گرفت. قبل از شروع حل مسئله با الگوریتم پیشنهادی یک سری پیش پردازش هندسی شامل انتقال مبدا مسیر به مبدا مختصات و دوران شبکه حول مبدا، بطوریکه نقطه انتهایی مسیر در جهت مثبت محور x ها قرار گیرد، بر روی شبکه‌ی مورد نظر انجام شد. جمعیت اولیه در الگوریتم پیشنهادی به صورت تصادفی از بین نودهای شبکه انتخاب شد. با توجه به تصادفی بودن جمعیت اولیه این امکان وجود دارد که نودهای تشکیل دهنده‌ی مسیرهای اولیه بسیار پراکنده واقع شوند. برای رفع این مشکل، یک محدوده‌ی جستجو حول نقطه‌ی مبدا و مقصد تعریف شد. این کار باعث سریع‌تر شدن فرآیند جستجوی جواب‌ها نیز می‌شود. در ادامه تابع هزینه برای این مسئله به صورت یک عدد مختلط تعریف شد که قسمت حقیقی آن بیان‌کننده‌ی مجموع وزن یال‌های واقعی و قسمت موهومی آن نشان‌دهنده‌ی تعداد یال‌های مجازی (تعداد عدم اتصالات بین نودهای تشکیل دهنده‌ی مسیر) می‌باشد. صفر شدن قسمت موهومی تابع هزینه، نشان‌دهنده‌ی یک مسیر واقعی بین نود مبدا و مقصد می‌باشد. برای انتخاب کروموزوم‌های والد از روش انتخاب چرخ رولت استفاده شد تا کروموزوم‌هایی که شایستگی بیشتری دارند، برای اعمال تقاطع و جهش انتخاب شوند. برای اعمال تقاطع از ۳ روش تقاطع یک نقطه‌ای، دو نقطه‌ای و چند نقطه‌ای استفاده شد که استفاده از هر کدام از آن‌ها با استفاده از روش انتخاب چرخ رولت صورت گرفت.

نوآوری اصلی در این پژوهش، ادغام دو الگوریتم فراابتکاری ژنتیک و جستجوی ممنوع بوده است. این ادغام

در بحث جهش الگوریتم ژنتیک صورت گرفته است و در واقع برای اعمال جهش بر روی کروموزوم‌ها از الگوریتم جستجوی ممنوع استفاده شده است، به این صورت که ۶ نوع جهش شامل حذف، جابجایی، اضافه کردن، Swap، معکوس کردن و جهش حذف و انتقال با استفاده از الگوریتم جستجوی ممنوع و با تعریف یک لیست ممنوعه برای اعمال انواع جهش‌ها بر روی کروموزوم‌های انتخاب شده اعمال شده است.

پس از تعریف روش پیاده‌سازی، الگوریتم پیشنهادی بر روی یک شبکه‌ی واقعی جهت‌دار (بخشی از شبکه راه‌های شهر تهران) پیاده‌سازی و اعمال شد. نتایج حاصل از الگوریتم ترکیبی GA-TS با نتایج حاصل از یک الگوریتم قطعی (دایجسترا) مقایسه شد. ضعف الگوریتم پیشنهادی در پارامتر طول مسیر (وزن مسیر) می‌باشد که با اختلاف اندکی از الگوریتم دایجسترا بیشتر است و نقطه‌ی قوت روش پیشنهادی پارامتر زمان اجرای الگوریتم می‌باشد که در زمان بسیار کوتاه‌تری نسبت به الگوریتم دایجسترا به جواب می‌رسد و برای مسائلی که نیاز به تصمیم‌گیری آنی دارند (مثل مسیریابی تحت وب و ...) این الگوریتم می‌تواند جایگزین مناسبی برای الگوریتم دایجسترا باشد. همچنین نتایج حاصل از الگوریتم پیشنهادی با نتایج حاصل از یک الگوریتم ژنتیک غیرترکیبی (GA خالص) مقایسه شد و نتایج بهتر بودن الگوریتم پیشنهادی را نسبت به الگوریتم ژنتیک غیرترکیبی و همچنین دایجسترا نشان می‌دهد.

به عنوان پیشنهاد آتی برای مسائل حوضه مسیریابی حمل و نقل، می‌توان برای بررسی بیشتر کارایی الگوریتم ترکیبی GA-TS، شبکه‌ی ورودی حمل و نقل را توسعه داد و علاوه بر شبکه‌ی خطوط راه‌های اصلی، ساخت‌های دیگر حمل و نقل مانند خطوط مترو، BRT و همچنین خطوط پیاده‌روی را وارد شبکه‌ی حمل و نقل کرد و همینطور علاوه بر معیار فاصله‌ی هندسی، معیارهای دیگری مثل زمان، هزینه و تعداد تغییر ساخت حمل و نقل را به عنوان معیارهای دیگر بهینه‌سازی در نظر گرفت. مورد دیگری که می‌تواند به عنوان پیشنهاد آتی مد نظر قرار بگیرد، استفاده از روش جستجوی همسایگی متغیر (VNS) برای ایجاد جمعیت اولیه‌ی الگوریتم ژنتیک می‌باشد. همچنین می‌توان جواب حاصل از الگوریتم ترکیبی GA-TS را با جواب سایر الگوریتم‌های تکاملی که مناسب حل مسائل گسسته بهینه‌سازی مثل مسئله یافتن کوتاه‌ترین مسیر هستند، مانند الگوریتم کلونی مورچگان (ACO) مقایسه کرد.

- [1] Moradi Seloushi, B., & Vafaeinejad, A. R. (2017). Using Geospatial Information System (GIS) to Determine the Capacity of Iran Railway Networks. *Journal of Geomatics Science and Technology*, 6(3), 15-22.
- [2] Zafari, A., Tashakori, S. M., & Yousefi Khoshbakht, M. (2010). A Hybrid Effective Genetic Algorithm for Solving the Vehicle Routing Problem. *International Journal of Industrial Engineering & Production Management*, 21(2), 63-76.
- [3] Golnarkar, A., Alesheikh, A. A., & Malek, M. R. (2010). Solving best path problem on multimodal transportation networks with fuzzy costs. *Iranian Journal of Fuzzy Systems*, 7(3), 1-13.
- [4] Ambrosino, D., & Sciomachen, A. (2014). An algorithmic framework for computing shortest routes in urban multimodal networks with different criteria. *Procedia-Social and Behavioral Sciences*, 108, 139-152.
- [5] Behzadi, S., Alesheikh, A. A., & Poorazizi, E. (2008). Developing a genetic algorithm to solve shortest path problem on a raster data model. *Journal of Applied Sciences*, 8(18), 3289-3293.
- [6] Zakzouk, A. A., Zaher, H. M., & El-Deen, R. A. Z. (2010, March). An ant colony optimization approach for solving shortest path problem with fuzzy constraints. In *2010 The 7th International Conference on Informatics and Systems (INFOS)* (pp. 1-8). IEEE.
- [7] Abbaspour, R. A., & Samadzadegan, F. (2010). An evolutionary solution for multimodal shortest path problem in metropolises. *Comput. Sci. Inf. Syst.*, 7(4), 789-811.
- [8] Dib, O., Moalic, L., Manier, M. A., & Caminada, A. (2017). An advanced GA-VNS combination for multicriteria route planning in public transit networks. *Expert Systems with Applications*, 72, 67-82.
- [9] Attiratanasunthron, N., & Fakcharoenphol, J. (2008). A running time analysis of an ant colony optimization algorithm for shortest paths in directed acyclic graphs. *Information Processing Letters*, 105(3), 88-92.
- [10] Ghoseiri, K., & Nadjari, B. (2010). An ant colony optimization algorithm for the bi-objective shortest path problem. *Applied soft computing*, 10(4), 1237-1246.
- [11] Liu, L., Mu, H., Yang, X., He, R., & Li, Y. (2012). An oriented spanning tree based genetic algorithm for multi-criteria shortest path problems. *Applied soft computing*, 12(1), 506-515.
- [12] Kuri, J., Puech, N., Gagnaire, M., & Dotaro, E. (2002, November). Routing foreseeable lightpath demands using a tabu search meta-heuristic. In *Global Telecommunications Conference, 2002. GLOBECOM'02. IEEE (Vol. 3, pp. 2803-2807)*. IEEE.
- [13] Mohemmed, A. W., Sahoo, N. C., & Geok, T. K. (2008). Solving shortest path problem using particle swarm optimization. *Applied Soft Computing*, 8(4), 1643-1653.
- [14] De Smith, M. J., Goodchild, M. F., & Longley, P. (2007). *Geospatial analysis: a comprehensive guide to principles, techniques and software tools*. Troubador Publishing Ltd.
- [15] Lu, T., & Zhu, J. (2013). A genetic algorithm for finding a path subject to two constraints. *Applied Soft Computing*, 13(2), 891-898.
- [16] Poczeta, K., Kubus, L., & Yastrebov, A. (2019). Analysis of an evolutionary algorithm for complex fuzzy cognitive map learning based on graph theory metrics and output concepts. *Biosystems*.
- [17] Owais, M., & Osman, M. K. (2018). Complete hierarchical multi-objective genetic algorithm for transit network design problem. *Expert Systems with Applications*, 114, 143-154.
- [18] Ahmadi, E., Süer, G. A., & Al-Ogaili, F. (2018). Solving Stochastic Shortest Distance Path Problem by Using Genetic Algorithms. *Procedia Computer Science*, 140, 79-86.
- [19] Ruiz, E., Soto-Mendoza, V., Barbosa, A. E. R., & Reyes, R. (2019). Solving the Open Vehicle Routing Problem with Capacity and Distance Constraints with a Biased Random Key Genetic Algorithm. *Computers & Industrial Engineering*.
- [20] Shojaedini, E., Majd, M., & Safabakhsh, R. (2019). Novel adaptive genetic algorithm sample consensus. *Applied Soft Computing*, 77, 635-642.
- [21] Cox, E. (2005). *Fuzzy modeling and genetic algorithms for data mining and exploration*. Elsevier.

- [22] Podvalny, S. L., Chizhov, M. I., Gusev, P. Y., & Gusev, K. Y. (2019). The Crossover Operator of a Genetic Algorithm as Applied to the Task of a Production Planning. *Procedia Computer Science*, 150, 603-608.
- [23] Wang, J., Ersoy, O. K., He, M., & Wang, F. (2016). Multi-offspring genetic algorithm and its application to the traveling salesman problem. *Applied Soft Computing*, 43, 415-423.
- [24] Hu, Q., Luan, X., Corman, F., & Lodewijks, G. (2016). A Tabu Search Algorithm for Inter-terminal Container Transport. *IFAC-PapersOnLine*, 49(3), 413-418.
- [25] Qiu, M., Fu, Z., Eglese, R., & Tang, Q. (2018). A Tabu Search algorithm for the vehicle routing problem with discrete split deliveries and pickups. *Computers & Operations Research*, 100, 102-116.
- [26] Silvestrin, P. V., & Ritt, M. (2017). An iterated tabu search for the multi-compartment vehicle routing problem. *Computers & Operations Research*, 81, 192-202.
- [27] Lai, D. S., Demirag, O. C., & Leung, J. M. (2016). A tabu search heuristic for the heterogeneous vehicle routing problem on a multigraph. *Transportation Research Part E: Logistics and Transportation Review*, 86, 32-52.
- [28] Schermer, D., Moeini, M., & Wendt, O. (2019). A Hybrid VNS/Tabu Search Algorithm for Solving the Vehicle Routing Problem with Drones and En Route Operations. *Computers & Operations Research*.
- [29] Behzadi, S., & Alesheikh, A. A. (2008). A Pseudo Genetic Algorithm for solving best path problem. *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 3.
- [30] Rajabioun, R. (2011). Cuckoo optimization algorithm. *Applied soft computing*, 11(8), 5508-5518.
- [31] Lipowski, A., & Lipowska, D. (2012). Roulette-wheel selection via stochastic acceptance. *Physica A: Statistical Mechanics and its Applications*, 391(6), 2193-2196.