

یافتن کوتاهترین مسیر شبکه با استفاده از الگوریتم بهینه سازی فاخته در سیستم اطلاعات مکانی

عالیه سرگلزائی^۱، علیرضا وفائی نژاد^{۲*}

^۱ کارشناس ارشد سنجش از دور و سیستم‌های اطلاعات مکانی - دانشکده محیط زیست و انرژی - دانشگاه آزاد

اسلامی واحد علوم و تحقیقات

sargolzaee_a@yahoo.com

^۲ استادیار دانشکده مهندسی عمران، آب و محیط زیست - دانشگاه شهید بهشتی

a_vafaei@sbu.ac.ir

(تاریخ دریافت مرداد ۱۳۹۵، تاریخ تصویب اسفند ۱۳۹۵)

چکیده

امروزه مسیریابی در شبکه‌های شهری با افزایش حجم اتومبیل‌ها و نیز محدودیت‌های ترافیکی گوناگون امری ضروری محسوب می‌شود. روشی که در این پژوهش به منظور حل مسأله کوتاهترین مسیر برای نخستین بار پیشنهاد شده است، استفاده از الگوریتم بهینه‌سازی فاخته می‌باشد. علت انتخاب این الگوریتم، جدید بودن و نیز پاسخ مناسبی است که این الگوریتم برای حل مسأله مسیریابی، نسبت به سایر الگوریتم‌های فراابتکاری داده است. در این راستا و برای انجام مسیریابی، با ایجاد تغییراتی در نسخه دودویی الگوریتم فاخته، از الگوریتم یاد شده استفاده گردید. بدین منظور، برای ایجاد جمعیت اولیه از یک روش کنترل شده استفاده شد، که از ایجاد جمعیت‌های تصادفی که درصد بسیار ناچیزی از آنها ممکن است مسیری را تشکیل دهند، جلوگیری می‌کند. به این صورت که متغیرهای جمعیت که در واقع همان نقاط شبکه و موقعیت‌های فاخته‌ها هستند، به صورت تصادفی انتخاب نشده بلکه به صورت کنترل شده انتخاب می‌شوند و انتخاب نقاط بعدی از میان نقاطی است که با نقطه مورد نظر ارتباط دارند. در تمام مراحل اجرای الگوریتم، موقعیت فاخته‌ها به صورت اعداد باینری درمی‌آیند و اگر نقطه‌ای در مسیر حضور داشته باشد عدد یک و در غیر اینصورت عدد صفر می‌گیرد. در فاز مهاجرت فاخته‌ها نیز، از یک تابع سیگموئید استفاده شده است که موقعیت جدید فاخته‌ها را به بازه‌ی بین صفر و یک برده و در نتیجه موقعیت‌های جدید نیز تبدیل به اعداد صفر و یک می‌شوند. به منظور آزمون میزان کارایی الگوریتم پیشنهادی، از سه نوع شبکه فرضی، محلی و واقعی استفاده شده است. نتیجه اجرای این الگوریتم در دو نوع شبکه فرضی و محلی با تعداد نقاط ۲۰ و ۳۱، نتیجه‌ای مشابه یک الگوریتم قطعی بود، اما در شبکه‌ای که بخشی از یک شبکه شهری واقعی بود و از ۶۱۷ نود و ۹۹۵ یال تشکیل شده بود، با اختلاف اندکی بهتر از الگوریتم قطعی، توانست مسیر بهینه را نشان دهد. نتایج حاصله نشان داد که این الگوریتم توانایی انجام مسیریابی در شبکه را دارا می‌باشد و با ایجاد تغییراتی بر اساس ساختار شبکه، می‌تواند برای داده‌های بزرگ و شبکه‌های واقعی شهری مورد استفاده قرار گیرد.

واژگان کلیدی: یافتن کوتاهترین مسیر در شبکه، سیستم اطلاعات مکانی، الگوریتم بهینه‌سازی فاخته، کدگذاری دودویی، جمعیت کنترل شده

* نویسنده رابط

۱- مقدمه

توسعه‌های شهری و به دنبال آن افزایش مسیرها و ارتباطات، نقش جابه‌جایی را پررنگ‌تر می‌کند. آنچه در این میان، اهمیت بیشتری می‌یابد، مسیریابی و مدیریت و هدایت حرکت در این مسیرها است که از یک سو روان‌سازی و نظم را در پی داشته و از سوی دیگر در شرایط اضطرار، نیاز را مرتفع سازد.

مسئله کوتاهترین مسیر^۱ در یک شبکه از جمله اساسی‌ترین مسائل تحلیل شبکه است. در واقع برای انجام تحلیل‌های سطوح بالاتر، یافتن پاسخ این مسئله از ضروریات است. در این راستا، مدیریت حجم بالای داده‌های مورد استفاده در مسیریابی، استفاده از توانمندیهای سامانه‌های اطلاعات مکانی که امکان مدلسازی و تحلیل داده‌های مکانی را در اختیار قرار می‌دهد، اجتناب‌ناپذیر می‌سازد [۱]، [۲].

از طرف دیگر و به طور کلی، روش‌های حل مسائل با استفاده از الگوریتم‌های بهینه‌سازی، به دو دسته الگوریتم‌های دقیق و تخمینی تقسیم‌بندی می‌شوند. در الگوریتم‌های دقیق، دستیابی به جواب بهینه، نیازمند صرف زمان و در نتیجه، هزینه بیشتری است. اما در طرف مقابل، الگوریتم‌های تخمینی قرار دارند که در یک زمان اندک به جواب می‌رسند.

در این میان، الگوریتم‌های فراابتکاری^۲، از جمله الگوریتم‌های تخمینی هستند که توانایی یافتن جواب‌های بهینه و یا نزدیک به بهینه را در زمان معقول دارا می‌باشند [۳]. در تحقیق حاضر، حل مسئله مسیریابی، با استفاده از الگوریتم فراابتکاری بهینه‌سازی فاصله انجام می‌شود، که سرعت و دقت دستیابی به جواب بهینه توسط آن، در حل بعضی مسائل، نسبت به سایر الگوریتم‌های فراابتکاری بالاتر است.

تنوع مسائل مربوط به مسیریابی که از نوع مسائل بهینه‌سازی تام می‌باشند، آن‌قدر زیاد است که دسته‌بندی آن‌ها و بیان حالت‌های مختلفی که در آن رخ می‌دهد، مشکل و زمان‌گیر است [۳]. برای حل این نوع مسائل، که زمان حل آن‌ها با بزرگ‌تر شدن ابعاد مسئله، به طور نمایی رشد می‌کند، استفاده از الگوریتم‌های فراابتکاری مناسب

می‌باشند. این الگوریتم‌ها برای حل مسائل با ابعاد بزرگ، از کارایی لازم برخوردار هستند [۴]. در این راستا و تاکنون، روش‌های بهینه‌سازی و فراابتکاری برای حل مسائل مسیریابی، زیاد مورد استفاده قرار گرفته‌اند که در ادامه به برخی از آنها اشاره شده است.

میتسو و همکاران، از الگوریتم ژنتیک برای حل مسائل کوتاهترین مسیر استفاده کردند. که طی آن، کار اصلی توسعه الگوریتم ژنتیک برای مسیریابی، چگونگی کدگذاری یک مسیر در یک گراف به صورت کروموزوم بود. رویکرد پیشنهادی ایشان، روی سه مسئله با اندازه‌های متفاوت مورد آزمایش قرار گرفت و نتایج آزمون، یافتن مسیر بهینه با احتمال مناسب و سرعت بالا بود [۵].

در تحقیق دیگر، سلطانی و همکاران، به ارزیابی عملکرد سه الگوریتم دایجسترا، A^* و ژنتیک، به منظور برنامه‌ریزی مسیر در ایجاد سایت، پرداختند. دقت انتخاب مسیر و پیچیدگی‌های زمانی این الگوریتم‌ها، در این تحقیق، مورد ارزیابی و مقایسه قرار گرفت. آنچه که دو الگوریتم دایجسترا و A^* را دچار مشکل می‌کرد، اثر ابعاد مسئله بود، که عملکرد آن‌ها را محدود به مسائل کوچک و یا متوسط می‌نمود. که در این میان، الگوریتم ژنتیک توانست راه‌حل‌های بهینه و نزدیک بهینه را در زمان کمتری پیدا کند [۶].

در مقاله‌ای دیگر، آتیراتانا و همکاران به تحلیل زمان تداوم الگوریتم بهینه‌سازی کلونی مورچگان، برای یافتن کوتاه‌ترین مسیر پرداخته‌اند و نتایج و دستاوردهای خود را در این زمینه ارائه نموده‌اند [۷]. قصیری و همکاران نیز، از الگوریتم بهینه‌سازی کلونی مورچگان برای حل مسئله کوتاه‌ترین مسیر دو هدفه استفاده نمودند. این الگوریتم در مقایسه با الگوریتم تصحیح برچسب، توانست جواب‌هایی با کیفیت بالاتر و در زمان سریع‌تری را ارائه دهد [۸].

همچنین ژین لو و همکاران، در مقاله‌ای برای یافتن کوتاه‌ترین مسیر، با ترکیب مکرر نزدیک‌ترین گره به گره مبدأ، شیوه‌ای جدید (ترکیب گره به جای عملگرهای برچسب‌گذاری)، برای اجرای الگوریتم دایجسترا پیشنهاد می‌دهند [۹]. لینزانگ لیو و همکاران نیز، از یک درخت پوششی جهت‌دار (OST) مبتنی بر الگوریتم ژنتیک، برای حل مسائل یافتن کوتاه‌ترین مسیر چندمعیاری (MCSPP)، استفاده کردند [۱۰]. در پژوهشی دیگر، کیوری و همکاران، برای مسیریابی از الگوریتم فراابتکاری

^۱ Shortest path problem

^۲ Metaheuristic

۲- مبانی نظری

۲-۱- مسئله کوتاهترین مسیر

مسئله کوتاهترین مسیر، 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 منبع و مقصد با یکسری خطوط داده شده می‌باشد. تعیین کوتاهترین مسیر، می‌تواند به صورت زیر، به عنوان یک مسئله برنامه‌ریزی خطی مشخص شود.

S به عنوان نقطه منبع، t به عنوان نقطه هدف، و $c_{ij} > 0$ هزینه یا فاصله ارتباط کمان یا یال (i,j) می‌باشد. هدف جستجوی حداقل مقدار z است [۱۴].

$$\begin{aligned} Z = z &= \sum \sum c_{ij} x_{ij} \\ \text{Subject to} \\ \sum x_{ji} - \sum x_{ik} &= m, \text{ where} \\ m &= 0 \text{ for } i \neq s, \\ m &= 1 \text{ for } i = t \\ m &= -1 \text{ for } i = s \\ x_{ij} &\in \{0,1\} \end{aligned} \quad (1)$$

۲-۲- الگوریتم بهینه‌سازی فاخته

مانند سایر الگوریتم‌های تکاملی، این الگوریتم نیز با جمعیتی از فاخته‌ها کار خود را شروع می‌کند. فاخته‌ها تعدادی تخم در لانه بعضی پرندگان می‌گذارند. بعضی از این تخم‌ها که شباهت بیشتری به تخم‌های پرند می‌زیان دارند، برای رشد و پرند بالغ شدن، فرصت بیشتری دارند. سایر تخم‌ها توسط پرندگان می‌زیان شناسایی و از بین می‌روند. تخم‌های رشد یافته در یک منطقه، شایستگی لانه‌ها را در آنجا نشان می‌دهد. تخم‌های بیشتری که در یک ناحیه زنده مانده‌اند، سود بیشتری است که در آن ناحیه بدست آمده است. بنابراین، موقعیتی که در آن تخم‌های بیشتری زنده بمانند، وضعیتی است که COA در حال بهینه شدن خواهد بود [۱۳].

فاخته‌ها در جستجوی یافتن مناسب‌ترین منطقه برای تخم‌گذاری، به منظور افزایش نرخ بقای تخم‌هایشان هستند. بعد از این‌که، تخم‌های باقی‌مانده رشد کردند و تبدیل به یک پرند بالغ شدند، اجتماعاتی را تشکیل می‌دهند. هر

جستجوی ممنوعه^۱ استفاده نمودند [۱۱]. محمد و همکاران نیز، در مقاله خود به حل مسئله یافتن کوتاهترین مسیر در شبکه با استفاده از الگوریتم بهینه‌سازی جمعی ذرات پرداخته‌اند [۱۲].

اما رجبیون، در مقاله خود به معرفی الگوریتم بهینه سازی فاخته^۲ و مقایسه آن با الگوریتم های ژنتیک و بهینه‌سازی جمعی ذرات در چند مسئله (غیر از مسیریابی) پرداخته است. نتایج تحقیق او نشان داد که الگوریتم فاخته در زمان کمتر و دقت بالاتر و نیز با متغیرهای بیشتر به جواب بهینه دست یافته است [۱۳].

با توجه به مقایسه انجام شده و مزیت‌های الگوریتم بهینه‌سازی فاخته، از جمله همگرایی سریع‌تر، سرعت بیشتر، دقت بالاتر، توانایی جستجوی محلی در کنار جستجوی کلی، احتمال کمتر گیرافتادن در نقاط جمعیت محلی، جستجوی با جمعیت متغیر (بدلیل نابودی جمعیت در مناطق نامناسب)، حرکت کلی جمعیت به سمت نقاط بهتر با نابود شدن جواب‌های نامناسب‌تر و توانایی حل سریع مسائل بهینه‌سازی با ابعاد بالا [۱۳]، به نظر می‌رسد که این الگوریتم در حل مسائل مربوط به مسیریابی نیز از توانمندی لازم برخوردار باشد.

در این راستا، هدف از انجام این تحقیق، قبل از هر چیزی، پاسخ به این سؤال است که آیا این الگوریتم، می‌تواند برای حل مسائل مرتبط با مسیریابی در شبکه مورد استفاده قرار گیرد (زیرا الگوریتم‌های فراابتکاری، در بعضی مسائل ممکن است قابل استفاده نباشند و در بعضی مسائل هم به نتیجه مطلوب دست پیدا نکنند). سپس پاسخ به این مهم که، آیا این الگوریتم برای حل مسئله مسیریابی، نسبت به سایر الگوریتم‌های فراابتکاری رایج نتیجه مناسبی می‌دهد یا خیر.

بدین منظور و در ادامه پس از تشریح الگوریتم بهینه‌سازی فاخته، ابتدا به تنظیم پارامترهای الگوریتم، برای حل مسئله کوتاهترین مسیر پرداخته می‌شود. سپس الگوریتم نهایی بر روی سه نوع شبکه شماتیک، محلی و واقعی پیاده‌سازی می‌شود. در انتها نیز، نتایج اجرای الگوریتم روی شبکه‌ها مورد بحث و بررسی قرار گرفته و نتیجه‌گیری تحقیق ارائه خواهد شد.

^۱ Tabu Search

^۲ Cuckoo Optimization Algorithm

$$\text{Profit} = f_p(\text{habitat}) = f_p(x_1, x_2, \dots, x_{Nvar}) \quad (3)$$

همان طور که مشاهده می شود، COA الگوریتمی است که تابع سود را ماکزیمم می کند. برای استفاده از COA برای حل مسائل کمینه سازی، کافی است یک علامت منفی در تابع هزینه ضرب شود [۱۳]:

$$\text{Profit} = -\text{Cost}(\text{habitat}) = -f_c(x_1, x_2, \dots, x_{Nvar}) \quad (4)$$

برای شروع الگوریتم بهینه سازی، یک ماتریس Habitat به سائز $N_{pop} \times N_{var}$ تولید می شود. سپس برای هر کدام از این Habitat ها تعدادی تصادفی تخم، تخصیص می یابد. در طبیعت، هر فاخته بین ۵ تا ۲۰ تخم می گذارد. این اعداد به عنوان حد بالا و پایین تخصیص تخم به هر فاخته در تکرارهای مختلف استفاده می شود.

در مسأله مسیریابی تعداد این متغیرها (N_{var}) به اندازه تعداد نقاط موجود در شبکه می باشد. بنابراین قبل از هر کاری باید نقاط شبکه مورد نظر کدگذاری شوند. به هر نقطه داخل شبکه عددی از یک تا n (تعداد نقاط داخل شبکه) اختصاص داده می شود.

برای ایجاد جمعیت اولیه، برای هر فاخته باید اعدادی از یک تا n به صورت تصادفی اختصاص داده شود. این اعداد به صورت رشته ای می باشد که یک مسیر را نشان می دهد. اما در انتخاب این اعداد دو شرط زیر باید رعایت شود:

۱- متغیر اول و آخر به صورت ثابت و همان نقاط ابتدا و انتها می باشد.

۲- انتخاب هر نقطه باید به نحوی باشد که به نقطه قبلی خود متصل باشد.

نکته اساسی در ایجاد جمعیت اولیه برای حل مسأله کوتاهترین مسیر، طول رشته ای است که در الگوریتم فاخته با عنوان موقعیت فاخته ها و در الگوریتم ژنتیک با عنوان طول کروموزوم مطرح می شود. در مسأله کوتاهترین مسیر، به دلیل اینکه حرکت از مبدأ به مقصد از تعداد نقاط یکسانی انجام نمی شود، لذا چالش اصلی در این مسأله تعیین طول موقعیت فاخته ها می باشد.

برای ایجاد جمعیت اولیه ابتدا ماتریسی از اتصالات شبکه ایجاد می شود، یعنی هر جا که نقطه ای به نقطه دیگر متصل می باشد عدد یک و در غیر اینصورت عدد صفر منظور می شود. سپس رشته ای به طول متغیرهای مسأله که همان تعداد نقاط شبکه می باشد، ایجاد می شود. متغیرهای ابتدا و انتهای این رشته از همان ابتدا مشخص می شود.

گروه محدوده سکونت مختص خودش را برای زندگی دارد و بهترین منطقه سکونت تمام گروه ها مقصد بعدی فاخته ها در سایر گروه ها خواهد بود. بنابراین آن ها به سمت بهترین زیستگاه، مهاجرت خواهند کرد. آن ها در جایی نزدیک بهترین زیستگاه، ساکن خواهند شد [۱۳]. با توجه به تعداد تخم هایی که هر فاخته دارد و نیز فاصله فاخته تا نقطه هدف (بهترین زیستگاه)، تعدادی شعاع تخم گذاری نسبت به آن مشخص می شود. سپس پرند شروع به تخم گذاری تصادفی در لانه هایی درون شعاع تخم گذاری خود می کند. این فرآیند تا زمانی که بهترین موقعیت با ارزش سود حداکثر کسب شود ادامه می یابد و بیشتر جمعیت فاخته ها اطراف موقعیت یکسان جمع می شوند [۱۳].

۳- یافتن کوتاهترین مسیر با استفاده از الگوریتم بهینه سازی فاخته

۳-۱- طراحی مسأله

مسأله مسیریابی در شبکه در واقع یافتن کوتاهترین مسیر بین دو نقطه می باشد. یک مسیر از نقطه ابتدا شروع می شود و با عبور از تعدادی نقطه، به نقطه پایان می رسد. اما شرطی که باید رعایت شود این است که هیچ نقطه تکراری در این مسیر نباشد.

۳-۲- ایجاد جمعیت اولیه

برای حل یک مسأله بهینه سازی لازم است تا مقادیر متغیرهای مسأله به فرم یک آرایه شکل گیرند. در GA^1 و PSO^2 این آرایه ها با نام های «کروموزوم» و «موقعیت ذرات» مشخص می شوند. ولی در الگوریتم بهینه سازی فاخته (COA)، به این آرایه ها «Habitat» یا «محل سکونت» گفته می شود. در یک مسأله بهینه سازی N_{var} -بعدی، یک Habitat، یک آرایه $1 \times N_{var}$ خواهد بود که موقعیت فعلی زندگی فاخته ها را نشان می دهد [۱۳]. این آرایه به شکل زیر تعریف می شود:

$$\text{Habitat} = [x_1, x_2, \dots, x_{Nvar}] \quad (2)$$

هر یک از ارزش های متغیر ($x_1, x_2, \dots, x_{Nvar}$)، شماره نقطه شناور است. مقدار سود در یک Habitat، با ارزیابی تابع سود (f_p) در Habitat به دست می آید [۱۳]. بنابراین:

^۱ Genetic Algorithm

^۲ Particle Swarm Optimization

می کنند. البته این تخمگذاری به صورتی انجام می شود که هر فاخته در شعاع تخمگذاری خود محدوده وسیع تری را جستجو کند. در پایان مرحله تخمگذاری، موقعیت های جدید تخمها به صورت صفر و یکی درمی آیند.

تخم هایی که در موقعیت های تکراری قرار دارند به دلیل اینکه در هر موقعیت تنها یک تخم می تواند حضور داشته باشد، از بین می روند. همچنین اگر تعداد تخمهای ایجاد شده از حد مجاز بیشتر باشد، تخمهایی که مقدار مناسبی ندارند از بین می روند. در این میان فاخته ای که بهترین موقعیت را داشته است برای ایجاد نسل بعد انتخاب می شود.

۴-۳- مهاجرت فاخته ها

تخم های باقیمانده شروع به رشد پیدا می کنند و تبدیل به فاخته بالغ می شوند. سپس خوشه بندی فاخته ها انجام می شود. در هر خوشه، میانگین سود براساس تابع هدف تعیین می شود. خوشه ای که میانگین بیشتری داشته است به عنوان نقطه هدف در نظر گرفته می شود. بعد از اینکه نقطه هدف تعیین شد مهاجرت فاخته های بالغ به سمت نقطه هدف برای تخمگذاری جدید آغاز می شود.

با توجه به اینکه مسائل مربوط به مسیریابی از نوع مسائل گراف مبنا هستند لذا باید در فضای گسسته به حل آنها پرداخت. در حل مسائل به روش دودویی، هر راه حل از فضای جستجو به وسیله ی رشته های صفر و یکی کد می شوند. نمایش دودویی در بهینه سازی گسسته خیلی رایج است، به این علت که بسیاری از مسائل، طبیعت فرموله شدن با متغیرهایی دودویی را دارند [۱۵].

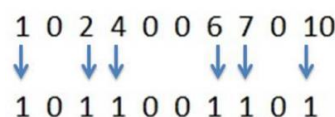
در گسسته سازی دودویی الگوریتم فاخته، جهت تبدیل COA پیوسته به فضای دودویی، عملگر مهاجرت COA به صورت زیر بازتعریف می شود. $X_{CurrentPosition}$ و X_{Goal} به ترتیب نقطه هدف جاری و موقعیت جاری یک فاخته در جمعیت می باشد. موقعیت بعدی فاخته ($X_{NextHabitat}$) به شکل زیر محاسبه می شود:

$$X_{NextHabitat} = X_{CurrentPosition} + rand \times (X_{Goal} - X_{CurrentPosition}) \quad (۶)$$

برای اینکه موقعیت جدید برای فضای دودویی مناسب باشد از تابع سیگموید رابطه (۷) برای نگاشت $X_{NextHabitat}$ به محدوده ۰ و ۱ به صورت زیر استفاده می شود. سپس براساس رابطه (۷) مقدار موقعیت به مقدار دودویی ۰ و ۱ تغییر می یابد [۱۵].

برای انتخاب نقاط بعدی، ابتدا مشخص می شود که هر نقطه با چه نقاطی ارتباط دارد، سپس به صورت رندم یکی از نقاط متصل به نقطه موردنظر انتخاب می شود.

این مراحل تا جایی ادامه پیدا می کند تا به نقطه انتهایی برسد. اما در صورتی که در طول مسیر به نقطه ای برسد که با نقاط دیگر ارتباطی نداشته باشد، به صورت خودکار نقطه انتها ثبت می شود و رشته ای به اندازه طول متغیرها و با دو نقطه ابتدا و انتها ایجاد می شود. اما این رشته به عنوان رشته غیرقابل استفاده از مجموعه جمعیت حذف می شود. بعد از اینکه رشته های موقعیت فاخته ها ایجاد شد، باید این رشته ها تبدیل به رشته های باینری شوند. برای این منظور در هر مسیر ایجاد شده، هر جا که شماره نقطه ای وجود دارد یک و در غیر اینصورت عدد صفر قرار می دهد. مثلاً برای رشته ای از موقعیت فاخته ها به طول ۱۰، رشته باینری آن به صورت شکل (۱) می شود.



شکل ۱- تبدیل موقعیت فاخته ها به رشته های باینری

۳-۳- تخم گذاری فاخته ها

در یک مسأله بهینه سازی با حد بالای متغیرهای var_{hi} و حد پایین var_{low} ، هر فاخته دارای، حداکثر دامنه تخم گذاری یا ELR^1 خواهد بود که متناسب است با تعداد کل تخمها، تعداد تخم های فعلی فاخته و همچنین حد بالا و پایین متغیرهای مسأله [۱۳].

بنابراین ELR به صورت زیر تعریف می شود:

$$ELR = \alpha \times (\text{Number of cuckoo's eggs} / \text{Total number of eggs}) \times (var_{low} - var_{hi}) \quad (۵)$$

α متغیری است که حداکثر مقدار ELR را با آن تنظیم می شود [۱۳].

بعد از ایجاد جمعیت اولیه، به هر کدام از فاخته های ایجاد شده در بازه بین حداقل و حداکثر تخمها، تعدادی تخم به صورت تصادفی اختصاص داده می شود. سپس مجموع تخمهای فاخته ها محاسبه می شود. در این مرحله، باید شعاع تخمگذاری فاخته ها براساس فرمول شعاع تخمگذاری (ELR)، تعیین شود. فاخته ها در محدوده شعاع تخمگذاری خود به صورت تصادفی شروع به تخمگذاری

^۱ Egg Laying Radius

$$S = 1 / (1 + e^{-X_{\text{NextHabitat}}})$$

$$\text{If } S \leq \text{rand Then } X_{\text{NextHabitat}} = 0 \quad (Y)$$

$$\text{If } S > \text{rand Then } X_{\text{NextHabitat}} = 1$$

مهاجرت فاخته‌ها براساس تابع سیگموئید انجام می‌شود به این صورت که ابتدا موقعیت فعلی فاخته‌ها با استفاده از این تابع تبدیل به اعداد بین ۰ و ۱ می‌شود. سپس با اعمال ضرب حرکتی و اختلاف فاصله نقطه هدف و موقعیت فعلی، نتایج حاصله به صورت اعداد ۰ یا ۱ نمایش داده می‌شود. استفاده از ضرب حرکتی باعث می‌شود که فاخته‌ها مستقیماً به سمت نقطه هدف مهاجرت نکنند و مقداری انحراف برای جستجوی بیشتر محدوده داشته باشند.

بهترین فاخته‌ای که در این مرحله به دست آمد که در واقع همان بهترین مسیر است که برای ایجاد نسل بعد مورد استفاده قرار می‌گیرد. بعد از مهاجرت فاخته‌ها مجدداً تخمگذاری و مراحل ذکر شده قبلی انجام می‌شود. این مراحل تا جایی ادامه پیدا می‌کند که نقطه هدف برابر یا نزدیک مقدار بهینه باشد. سپس شرط توقف اجرا می‌شود. و الگوریتم به انتهای خود می‌رسد. با توجه به اینکه جوابی که در پایان به دست آمده به صورت باینری می‌باشد باید مجدد تبدیل به شماره واقعی نقاط شود تا مسیر به دست آمده در شبکه مشخص شود.

در پایان الگوریتم مجدداً بررسی می‌شود که آیا جواب به دست آمده یک مسیر واقعی است و مسیری بین دو نقطه موردنظر وجود دارد یا خیر.

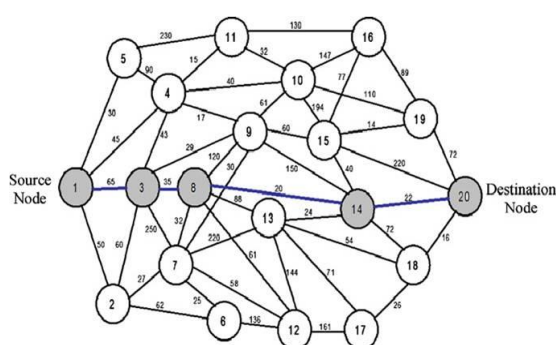
۴- پیاده سازی الگوریتم در شبکه های مختلف

در این مرحله کارایی الگوریتم روی شبکه‌های مختلفی امتحان می‌شود. ابتدا از یک شبکه فرضی با تعداد نقاط محدود استفاده می‌شود. سپس روی یک شبکه محلی و در نهایت برای آزمون اینکه آیا این الگوریتم روی شبکه‌های واقعی هم به نتیجه می‌رسد یا خیر، از یک شبکه شهری واقعی استفاده می‌شود.

۴-۱- شبکه فرضی با تعداد نقاط محدود

این شبکه از ۲۰ نقطه تشکیل شده است [۱۲]. نقاط شروع و پایان مسیر به ترتیب نقاط ۱ و ۲۰ می‌باشد. شبکه مفروض یک شبکه غیرجهت دار می‌باشد، یعنی اندازه یالها هم از مسیر رفت و هم برگشت به یک اندازه می‌باشد (شکل ۲).

هدف از مسیریابی در این شبکه یافتن مسیری است که طول آن حداقل باشد. بعد از پیاده‌سازی شبکه در نرم افزار متلب (MATLAB)، و اجرای این الگوریتم بر روی شبکه مفروض، نتایجی که از آن بدست آمد با نتایج حاصل از الگوریتم دایجسترا مقایسه شد. مقایسه نتایج دو الگوریتم نشان داد که هر دو از مسیرهای یکسانی برای رسیدن به نقطه مقصد استفاده کرده‌اند. با توجه به اینکه الگوریتم دایجسترا از نوع الگوریتمهای پیمایش گراف و یک الگوریتم قطعی می‌باشد به این معنی که بعد از اجرای آن به نتیجه دقیق دست پیدا می‌کند. لذا برای این شبکه این نتیجه حاصل شد که توانسته است با این تعداد نقاط به نتیجه قطعی دست پیدا کند.

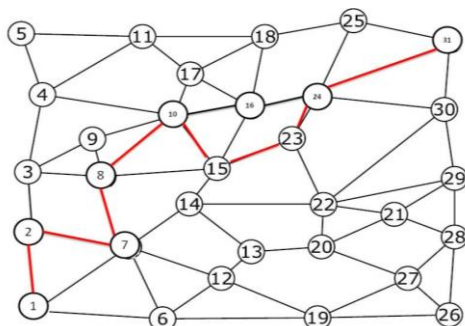


شکل ۲- نتیجه اجرای الگوریتم COA روی شبکه فرضی

۴-۲- مسیریابی در شبکه‌های محلی

شبکه مورد استفاده در این بخش [۱۶]، که با عنوان شبکه محلی معرفی شده است شبکه‌ای متشکل از ۳۱ نقطه و ۶۱ یال می‌باشد. یالها در این شبکه نیز به صورت غیرجهت دار فرض شده‌اند.

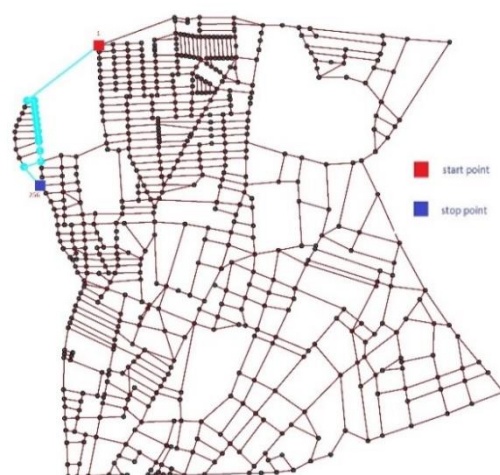
هدف از مسیریابی در این شبکه نیز یافتن مسیری است که مجموع یالهای آن مینیمم باشد. بعد از پیاده سازی شبکه در نرم افزار متلب نتایج (شکل ۳) حاصل شد. نتایجی که از این الگوریتم حاصل شد با نتایج بدست آمده از الگوریتم دایجسترا مشابه بودند. در نتیجه در این شبکه نیز الگوریتم توانست به نتایج دقیقی دست پیدا کند.



شکل ۳- نتایج اجرای الگوریتم COA روی شبکه محلی

۳-۴- مسیریابی در شبکه‌های واقعی

شبکه‌ای که در این بخش مورد استفاده قرار گرفته است، بخشی از یک شبکه حمل‌ونقل شهری تهران می‌باشد. تعداد نقاط این شبکه که حدود ۶۱۷ نقطه و ۹۹۵ یال می‌باشد در مقایسه با شبکه محلی بسیار بزرگتر می‌باشد. این شبکه نیز مانند شبکه‌های قبلی غیرجهت‌دار است و وزن یالها از هر دو طرف یکسان می‌باشد (شکل ۴). نقاط ۱ و ۲۵۶ به عنوان نقاط شروع و پایان انتخاب شدند.



شکل ۴- نتایج حاصل از اجرای الگوریتم COA روی بخشی از یک شبکه واقعی



شکل ۵- نتایج حاصل از اجرای الگوریتم دایجسترا روی بخشی از یک شبکه واقعی

۴-۴- نتایج پیاده‌سازی الگوریتم

با بررسی نتایج جدول (۱) معلوم می‌شود که الگوریتم بهینه‌سازی فاخته با موفقیت توانست مسیر بهینه را در شبکه‌های کوچک پیدا نماید. این امر نشان می‌دهد که تغییراتی که در ایجاد جمعیت اولیه و فاز مهاجرت این الگوریتم صورت گرفته است، به درستی پیش‌بینی شده بود.

جدول ۱- نتایج حاصل از اجرای الگوریتم‌های COA و دایجسترا روی شبکه‌های مختلف

نوع شبکه	شبکه دارای ۲۰ نود		شبکه دارای ۳۱ نود		شبکه دارای ۶۱۷ نود	
	الگوریتم COA	الگوریتم دایجسترا	الگوریتم COA	الگوریتم دایجسترا	الگوریتم COA	الگوریتم دایجسترا
مسیر نهایی	۱	۱	۱	۱	۱	۱
	۳	۳	۲	۲	۲۲۵	۲
	۸	۸	۷	۷	۲۲۴	۳
	۱۴	۱۴	۸	۸	۲۲۶	۴
	۲۰	۲۰	۱۰	۱۰	۲۲۷	۵
			۱۵	۱۵	۲۲۸	۶
			۲۳	۲۳	۲۲۹	۷
			۲۴	۲۴	۲۳۰	۸
			۳۱	۳۱	۲۳۱	۹
					۲۳۲	۱۰
					۲۳۳	۱۱
					۲۳۴	۱۲
					۲۳۵	۱۳
					۲۵۶	۱۴
وزن مسیر نهایی	۱۴۲	۱۴۲	۲۸	۲۸	۱۸۹۲	۱۹۵۶
						۲۴۹
						۲۵۰
						۲۵۱
						۲۵۲
						۲۴۷

۵- بحث و نتیجه‌گیری

در پژوهش حاضر به بررسی توانمندی الگوریتم بهینه‌سازی فاخته در مسیریابی پرداخته شد. پس از بررسی مسأله یافتن کوتاهترین مسیر در شبکه به عنوان مسأله هدف، ساختار الگوریتم بهینه‌سازی فاخته و نسخه دودویی آن برای حل مسائل گسسته، مورد مطالعه قرار گرفت. در

بعد از اجرای الگوریتم‌های بهینه‌سازی فاخته و دایجسترا روی یک نمونه شبکه واقعی (شکل ۴ و ۵)، نتایجی که حاصل شد نمایانگر این است که الگوریتم بهینه‌سازی فاخته در یک شبکه واقعی هم توانست به نتیجه‌ای نزدیک و حتی با اختلافی اندک نسبت به الگوریتم دایجسترا، دست یابد.

نهایت با توجه به روشهای مختلف حل مسأله کوتاهترین مسیر و نیز ساختار الگوریتم بهینه سازی فاخته، روشی برای حل این مسأله پیشنهاد شد.

در این روش ابتدا در مرحله ایجاد جمعیت اولیه، تغییراتی به وجود آمد به این صورت که متغیرهای جمعیت که در واقع همان نقاط شبکه و موقعیت های هر فاخته می باشند، به صورت تصادفی انتخاب نمی شوند. این متغیرها به صورت کنترل شده انتخاب می شوند به این صورت که انتخاب نقاط بعد از هر نقطه به صورت تصادفی از میان نقاطی است که با نقطه مورد نظر ارتباط دارند. در تمام مراحل اجرای الگوریتم موقعیت فاخته ها به صورت اعداد باینری درمی آیند. به این صورت که اگر نقطه ای در مسیر حضور داشته باشد عدد ۱ و در غیر اینصورت عدد صفر می گیرد.

تغییر دیگری که در الگوریتم ایجاد شد تغییر در فاز مهاجرت بود که البته این تغییر در تحقیقات قبلی برای گسسته سازی دودویی این الگوریتم پیشنهاد شده است. با استفاده از این تغییر در فاز مهاجرت برای تعریف موقعیت های جدید فاخته ها، از یک تابع سیگموید استفاده می شود که موقعیت جدید را تبدیل به صفر و یک می کند. نتایجی که از اجرای الگوریتم پیشنهادی بر روی چند شبکه، بدست آمد و مقایسه آن با یک الگوریتم قطعی نشان داد که این الگوریتم توانایی دستیابی به مسیر بهینه در شبکه های با اندازه کوچک را دارد. در شبکه هایی که کمی بزرگتر هستند و در واقع ویژگی شبکه های واقعی را دارا می باشند، این الگوریتم توانست با اختلاف اندکی بهتر از الگوریتم قطعی، مسیر بهینه را پیدا نماید.

دستیابی به مسیر بهینه، حتی در یک شبکه فرضی و کوچک، مهمترین هدف استفاده از الگوریتم ابتکاری می باشد. بعد از دستیابی به این هدف است که تلاش برای پیاده سازی این الگوریتم روی شبکه های واقعی و بزرگتر آغاز می شود. با توجه به اینکه ساختار شبکه های واقعی اختلاف زیادی با شبکه های فرضی و یا شبکه های با اندازه های کوچکتر دارد، لذا ایجاد تغییراتی روی الگوریتم برای کار با داده های بزرگتر اجتناب ناپذیر است.

در اجرای این الگوریتم از عامل زمان اجرای الگوریتم صرف نظر شده است. این موضوع به دو دلیل مهم می باشد: اول اینکه مقایسه زمان اجرا بهتر است که با یک الگوریتم مشابه صورت گیرد. چرا که این الگوریتم ها، بدون هیچ

دانش اولیه ای به جستجو در محیط می پردازند و با آزمون و خطا درصدد یافتن جواب اولیه مناسب هستند تا بتوانند با بهبود بخشیدن جوابهای مناسب به جواب بهینه دست یابند. این الگوریتم ها به الگوریتم های غیرقطعی مشهور هستند چرا که سعی در یافتن جواب دقیق یا نزدیک به آن را دارند. این موضوع در مسائلی با اندازه های بزرگ بسیار مشهود است. در مقابل الگوریتم های قطعی در اندازه های کوچک با سرعت بالا به جواب دقیق دست پیدا می کنند. اما نقطه ضعف آنها در مسائل با اندازه ی بزرگ، نمود پیدا می کند. دلیل دوم اینکه در چنین اندازه های کوچک شبکه و با توجه به سرعت بالای پردازنده های رایانه ای، سرعت رسیدن به جواب معیار مناسبی برای سنجش کارایی یک الگوریتم نمی باشد.

حل مسأله کوتاهترین مسیر با استفاده از این الگوریتم گام مهمی در حل مسائل بزرگتر و پیچیده تر می باشد، چرا که مسأله کوتاهترین مسیر از جمله مسائل کلیدی آنالیز مسیریابی می باشد و بسیاری از مسائل دیگر که در زمره مسائل بهینه سازی تام و سخت قرار می گیرند زیرمجموعه این مسأله قرار می گیرند.

با توجه به اینکه در این پژوهش، برای اولین بار به حل مسأله یافتن کوتاهترین مسیر با استفاده از الگوریتم بهینه سازی فاخته پرداخته شده است، لذا می توان با ایجاد تغییرات دیگری در این الگوریتم به نتایج بهتری دست یافت. در این تحقیق از روش گسسته سازی باینری استفاده شده است، اما می توان از سایر روشهای گسسته سازی نیز برای حل این مسأله کمک گرفت. همچنین به نظر میرسد استفاده از روش اولویت مبنای، برای ایجاد جمعیت اولیه نیز نتایج مطلوبی داشته باشد. مورد دیگری که می تواند به عنوان پیشنهاد آتی مدنظر قرار گیرد، کاهش تعداد موقعیت فاخته ها یا همان متغیرهای مسأله به نصف این تعداد می باشد، با توجه به اینکه رسیدن به مسیر بهینه لزوماً عبور از تمام نقاط شبکه نمی باشد. همچنین کاربرد این الگوریتم در شبکه های بزرگتر و پیچیده تر و نیز مقایسه آن با سایر الگوریتم های فراابتکاری از جمله PSO می تواند میزان کارایی الگوریتم را نشان دهد.

- [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*, Vol. 6, No. 3, pp. 15-22.
- [2] Payam Rad, D., Vafaeinejad, A.R., (2015). "Using a GIS Based Decision Support System to Aid Earthquake Crisis Management with Site Selection of Temporary Housing Case Study: District 8 of Isfahan Municipality". *Journal of Geomatics Science and Technology*, Vol. 5, No. 2, pp. 231-246.
- [3] Zafari, A. Tashakori Hashemi, S., and Yousefi Khoshbakht, M. (2010). "A Hybrid Effective Genetic Algorithm for Solving the Vehicle Routing Problem". *International Journal of Industrial Engineering and Production Management*. Vol. 21, No.2, pp. 63-76
- [4] Ghoseiria, K., Ghanad poor, S.F., (2008). "Locomotive Routing Problem Using a Hybrid Genetic Algorithm". *Journal of Transportation Research*, Vol. 5, No. 3, p. 259
- [5] Mitsuo, G., Runwei, C., and Dingwei, W. (1997). "Genetic Algorithms for Solving Shortest Path Problems". *Evolutionary Computation*. April 1997. IEEE International Conference on. 401-406
- [6] Soltani, A.R., Tawfik, H., Goulernas, J.Y., and Fernando, T. (2002). "Path planning in construction sites: performance evaluation of the Dijkstra, A*, and GA search algorithms". *Advanced Engineering Informatics*. Vol. 16, No. 4, pp. 291-303.
- [7] Attiratanasunthron, N., Fakcharoenphol, J. (2007). "A running time analysis of an Ant Colony Optimization algorithm for shortest paths in directed acyclic graphs". *Information Processing Letters*. 105, pp. 88-92.
- [8] Ghoseiri, K., Nadjari, B. (2009). "An ant colony optimization algorithm for the bi-objective shortest path problem". *Applied Soft Computing*. Vol.10, No.4, 1237-1246.
- [9] Lu, X., Camitz, M. (2011). "Finding the shortest paths by node combination". *Applied Mathematics and Computation*. Vol.217, No.13, pp. 6401-6408.
- [10] Liu, L., Mu, H., Yang, X., and Li, Y. (2012). "An oriented spanning tree based genetic algorithm for multi-criteria shortest path problems". *Applied Soft Computing*, Vol.12, No.1, pp.506-515.
- [11] Kuri, J., Puech, N., Gagnaire, M., and Dotaro, E., "Routing foreseeable light path demands using a tabu search meta-heuristic", in: *Proceedings of the IEEE Global Telecommunication Conference*, 2003, pp. 2803-2807.
- [12] Mohemmed, A., Chandra Sahoo, N., and Kim Geok, T. (2008). "Solving shortest path problem using particle swarm optimization", *Applied Soft Computing*, Vol.8, No.4, pp. 1643-1653.
- [13] Rajabioun, R. (2011). "Cuckoo Optimization Algorithm". *Applied Soft Computing*, Vol.11, No.8, pp. 5508-5518.
- [14] Smith, M., Goodchild, M. and Longley, P. (2009). "Geospatial Analysis". Matador.
- [15] Mahmoudi, Sh., (2012). "Discrete Manufacturing Cuckoo Search algorithm, Case Study: Graph Coloring". A thesis submitted for the degree M.S. in Computer Engineering – Artificial intelligence, Faculty of engineering, University of Nabi Akram.
- [16] Behzadi, S., (2008). "Extending Route Finding Methods by Using Genetic Algorithm". A thesis submitted for the degree M.S. in GIS, Faculty of Surveying, K.N.Toosi University of Technology.