مجله محاسبات نرم، جلد ۱۲، شماره ۲ (پاییز و زمستان ۱۴۰۲)، صفحه ۱۴۶ـ۱۶۴
https://doi.org/10.22052/scj.2023.248702.1118

مجله محاسبات نرم
SOFT COMPUTING JOURNAL
تارنمای مجله: scj.kashanu.ac.ir

دانشگاه کاشان
University of Kashan

بهینه‌سازی مسیریابی وسایل نقلیه مبتنی بر ترکیب الگوریتم‌های کلونی مورچه و ازدحام ذرات با تابع ابتکاری کسینوس زوایا✧

حمیدرضا شفیعی[۱]، کارشناسی ارشد، وحید رافع[۱]، دانشیار، مریم امیری[۱*]، استادیار
[۱] گروه کامپیوتر، دانشکده مهندسی فنی و مهندسی، دانشگاه اراک، اراک، ایران.

اطلاعات مقاله

تاریخچه مقاله:
دریافت ۲۶ آذر ماه ۱۴۰۱
پذیرش ۴ مرداد ماه ۱۴۰۲

کلمات کلیدی:
شبکه‌های حمل‌ونقل
برنامه‌ریزی مسیر پویا
کلونی مورچگان
بهینه‌سازی ازدحام ذرات
تابع ابتکاری کسینوس زوایا

چکیده

وضعیت شلوغی جاده‌ها عامل بسیار مهمی در ترافیک شهری است. برای حل این مشکل پژوهش‌های زیادی انجام شده است که عمده راهکارهای ارائه شده بر مبنای الگوریتم‌های فراابتکاری می‌باشند. در بسیاری از این الگوریتم‌ها در ابتدای کار مسیریابی به صورت تصادفی و در محدوده‌ای بزرگ انجام می‌شود که خود موجب افزایش زمان جستجو می‌گردد. همچنین در تحلیل‌های صورت گرفته بر مبنای این الگوریتم‌ها تنها فاصله فیزیکی بین خودروها در نظر گرفته شده است. با توجه به اینکه عوامل محیطی نظیر ترافیک، در مسیریابی بسیار موثر هستند، باید تاثیر این عوامل را در مسیریابی در نظر گرفت. بدین منظور برای حل مشکلات مطرح شده، در این مقاله یک روش برنامه‌ریزی مسیر پویا مبتنی بر ترکیب الگوریتم‌های کلونی مورچگان و ازدحام ذرات با اعمال تابع ابتکاری کسینوس زوایا پیشنهاد شده است. این روش عوامل مختلف وضعیت جاده‌ها از جمله طول جاده شهری و ترافیک ورودی و خروجی تقاطع‌ها را در مسیریابی وسایل نقلیه در نظر می‌گیرد و در مسیریابی برای نقاطی که هم‌راستای مسیر پیمایش به سمت مقصد نهایی باشد، شانس بیشتری قائل می‌شود. نتایج حاصل از اعمال مدل پیشنهادی بر روی داده‌های کتابخانه معتبر TSPLIB که مبتنی بر فاصله فیزیکی بین خودروها است، نشان می‌دهد که زمان جستجوی روش پیشنهادی نسبت به سایر روش‌ها (ده ارزیابی انجام شده) بطور متوسط ۴۰/۷۴ درصد کاهش داشته است. بیشترین و کمترین میزان کاهش به ترتیب ۹۸/۱ و ۶/۰۲ درصد می‌باشد. همچنین آزمایش برنامه‌ریزی مسیر پویا تحت ترافیک جاده‌ها بر روی برخی از تقاطع‌های شهر پکن نشان می‌دهد که روش پیشنهادی به صورت میانگین ۱/۵۷ درصد تراکم و ازدحام ایجاد می‌کند.

© ۱۴۰۲ نویسندگان. مقاله با دسترسی آزاد تحت مجوز CC-BY

۱. مقدمه

براساس مطالعات انجام شده در مرجع [۱]، در بسیاری از شهرها سیستم حمل و نقل تلفیقی از سیستم سنتی و مدرن است. سیستم سنتی شامل حمل و نقل ناوگانی (تاکسی و اتوبوس) و سیستم مدرن شامل حمل و نقل ریلی (مترو و قطار شهری) می‌باشد که این استخوان‌بندی اصلی حمل و نقل کشورها را تشکیل می‌دهد. مشکلات بسیاری در حمل و نقل کشورها وجود دارد و زیرساخت‌ها در بسیاری از نقاط جوابگوی حجم ترافیک نیست.

✧ نوع مقاله: پژوهشی

* نویسنده مسئول
پست(های) الکترونیک: hr.shafiei1994@gmail.com (شفیعی)
v-rafe@araku.ac.ir (رافع)، m-amiri@araku.ac.ir (امیری)

نحوه ارجاع به مقاله: حمیدرضا شفیعی، وحید رافع، مریم امیری، «بهینه‌سازی مسیریابی وسایل نقلیه مبتنی بر ترکیب الگوریتم‌های کلونی مورچه و ازدحام ذرات با تابع ابتکاری کسینوس زوایا»، مجله محاسبات نرم، جلد ۱۲، شماره ۲، صفحات ۱۴۶ـ۱۶۴، پاییز و زمستان ۱۴۰۲.

در حمل و نقل شهری انتخاب مسیر بهینه برای رانندگان با توجه به پویایی مسیر و ازدحام شدید بسیار دشوار است و این خود می‌تواند باعث ایجاد سوانح رانندگی، اتلاف وقت و تشدید ترافیک شود. بنابراین در مواجهه با چنین زیرساختی پیدا کردن مسیر بهینه با بالاترین دقت و کمترین زمان جستجو از میان مسیرهای موجود که اغلب شلوغ هستند، یک مساله چالش‌برانگیز است. با توجه به افزایش روزافزون خودروها و مشکلات زیرساخت معابر، پیدا کردن مسیرهای بهینه با الگوریتم‌های فرا-ابتکاری در مرجع [۲] یکی از رویکردهای موثر در کنترل ترافیک است. بیشتر تحقیق‌هایی که در این زمینه انجام شده تنها با در نظر گرفتن فاصله فیزیکی بین خودروها سعی در حل مساله داشته‌اند. از آنجایی که فاصله فیزیکی تنها فاکتور موثر در پیدا کردن مسیر بهینه نیست، پژوهش جاری سعی دارد در یک برنامه‌ریزی پویا جریانات ترافیکی را لحاظ کند و با ارائه مسیر بهینه باعث یکدست شدن ترافیک معابر شود. الگوریتم‌های فراابتکاری زیادی برای برنامه‌ریزی مسیر وجود دارد. بسیاری از این الگوریتم‌ها به دلیل پیچیدگی‌های مسائل مسیریابی و سنگین بودن تحلیل‌ها برای برنامه‌ریزی مسیر مناسب نیستند. اگرچه یکی از پرکاربردترین الگوریتم‌ها برای برنامه‌ریزی مسیر الگوریتم کلونی مورچگان ACO[1] است، این الگوریتم در همگرایی به نقطه بهینه سراسری ضعف دارد و محدوده تنظیم متغیرهای آن بسیار وسیع است. بدین منظور برای رفع این محدودیت‌ها، در مرجع [۳] ترکیب الگوریتم ACO با سایر الگوریتم‌ها پیشنهاد شده است. با توجه به اینکه فاصله فیزیکی تنها فاکتور موثر در برنامه‌ریزی نیست و عوامل محیطی نظیر ترافیک تاثیرگذار است، در ترافیک شهری پیچیده این‌گونه برنامه‌ریزی‌ها مناسب نیستند. در این مقاله، با توجه به مشکلاتی که در حوزه تنظیم متغیرها و همگرایی الگوریتم ACO وجود دارد، الگوریتم ازدحام ذرات (PSO)[2] که دارای بار محاسباتی پایین و تعداد کم متغیرها است [۴]، با این الگوریتم ترکیب شده است. اختلاف پایه‌ای الگوریتم بهینه‌سازی ازدحام ذرات در مقایسه با سایر الگوریتم‌های ابتکاری در میزان سادگی پیاده‌سازی است. در صورتی که تابع هزینه بطور مناسب انتخاب شود، الگوریتم به نقطه اکسترمم همگرا می‌شود [۵]. همچنین برای افزایش دقت و کاهش زمان جستجوی الگوریتم ترکیبی، تابع ابتکاری کسینوس زوایا به آن اضافه شده است. این تابع، زاویه بین گره‌های مبدا و مقصد را در نظر می‌گیرد و گره دارای زاویه‌ی کمتر نسبت به مقصد را در اولویت انتخاب قرار می‌دهد. در روش پیشنهادی برخلاف روش ارائه شده در مرجع [۶]، الگوریتم PSO در تمامی تکرارهای الگوریتم ACO استفاده می‌شود و بهینه‌های فردی و سراسری الگوریتم PSO در رابطه احتمال ACO به صورت مداوم تاثیر داده می‌شوند و در نهایت با اعمال تابع ابتکاری کسینوس زوایا دقت الگوریتم ACOPSO افزایش و زمان مسیریابی هم کاهش داده می‌شود. به صورت خلاصه می‌توان گفت ایده اصلی مقاله بررسی تاثیر ترافیک جاده ها در برنامه‌ریزی مسیر همچنین تاثیر الگوریتم PSO بر روی متغیرهای ACO، اعمال تابع ابتکاری کسینوس زوایا و یافتن ترکیب جدیدی برای بروزرسانی فرمون روی مسیرها است، که در نهایت موجب می‌شود زمان جستجو برای پیدا کردن بهینه‌ترین مسیر و همچنین نرخ انحراف در پیدا کردن مسیر بهینه کاهش پیدا کند که این خود موجب افزایش دقت نهایی روش اجرایی می‌شود. روش پیشنهاد شده در این مقاله ACOPSOCa نامگذاری می‌شود. نوآوری این مقاله در موارد زیر خلاصه می‌شود.

- کاهش نرخ ازدحام و ترافیک مسیرها: انجام برنامه‌ریزی مسیر پویا با در نظر گرفتن عوامل محیطی (نظیر ترافیک جاده‌ها) به منظور یکدست کردن ترافیک مسیرها و کاهش نرخ ازدحام مسیرهای مختلف به عنوان یک هدف در روش پیشنهادی لحاظ شده است که در واقعیت و به صورت عملی منجر به کاهش هزینه و مصرف انرژی می‌شود.
- جلوگیری از همگرایی زودرس الگوریتم ACO: در این مقاله، مشکل بزرگ همگرایی زودرس الگوریتم ACO، با حذف استقلال انتخاب گره‌ها، ترکیب این الگوریتم با PSO و در نظر گرفتن بهینه‌های فردی و سراسری حل می‌شود. این امر، سبب جستجو در سطح جهانی خواهد شد.

[1] Ant colony optimization
[2] Particle Swarm Optimization

- کاهش زمان جستجو در پیدا کردن مسیرهای بهینه در جستجوهای جهانی با اعمال تابع ابتکاری: در ترکیب چند الگوریتم برای جلوگیری از همگرایی زودرس به صورت طبیعی ممکن است زمان جستجو برای پیدا کردن بهینه‌ترین مسیر افزایش یابد که در روش پیشنهادی با اعمال تابع ابتکاری کسینوس زوایا تاثیر این مساله کاهش داده شده است.
- پیدا کردن مسیرهای بهینه با زمان جستجوی پایین: مطابق با نتایج ارزیابی‌ها، روش پیشنهادی قادر است بهینه‌ترین مسیر را در مقایسه با رایج‌ترین و جدیدترین روش‌های ارائه شده، با زمان جستجوی کمتری یافت نماید.
- پیدا کردن مسیرهایی با نرخ انحراف پایین: مطابق با نتایج ارزیابی‌ها، روش پیشنهادی قادر است بهینه‌ترین مسیرها را با نرخ انحراف پایین (دقت بالا) بیابد که در تکرارهای مداوم با ثبات کامل مسیر بهینه را تکرار می‌کند.

در ادامه این مقاله، در بخش دوم، روش‌های کلاسیک در مسیریابی، الگوریتم های موثر در مسیریابی و همچنین مطالعات انجام شده در حوزه مسیریابی مرور خواهند شد. همچنین در انتهای این بخش، به بررسی الگوریتم‌های پرکاربرد کلونی مورچه و الگوریتم ازدحام ذرات خواهیم پرداخت. در بخش سوم روش پیشنهادی به صورت گام به گام توضیح داده می‌شود. در بخش چهارم به ارزیابی و حل مساله می‌پردازیم و نتایج به دست آمده را با نتایج حاصل از اعمال جدیدترین و پرکاربردترین الگوریتم‌های موجود مقایسه می‌کنیم. بخش پنجم به ارائه نتایج کلی مقاله، جمع‌بندی و ارائه پیشنهاداتی برای کارهای آتی می‌پردازد.

۲. مبانی نظری و مرور روش‌های پایه

در این بخش ابتدا به مرور کلی رویکردهای موثر در مسیریابی می‌پردازیم. در ادامه، در بخش ۱.۲ رایج‌ترین و جدیدترین روش‌های مسیریابی و در بخش ۲.۲ الگوریتم‌های پایه روش ACOPSOCa بررسی می‌شوند.

برای حل مسائل مسیریابی وسایل نقلیه در طول سالیان روش‌های متفاوتی ارائه شده که بسیاری از این روش‌ها امروزه منسوخ شده‌اند و کاربرد مناسبی در حل مسائل ندارند. مسائل بهینه‌سازی مطابق شکل (۱) به دو دسته مسائل گسسته و پیوسته تقسیم می‌شوند. مسائل گسسته در دو دسته دقیق و تقریبی تقسیم‌بندی می‌شوند. الگوریتم‌های دقیق توانایی یافتن جواب بهینه در مسائل بسیار کوچک را به صورت دقیق دارند در الگوریتم‌های تقریبی تضمینی برای پیدا کردن جواب دقیق وجود ندارد. در این الگوریتم‌ها در هر تکرار با توجه به جواب‌های تکرار قبلی، جواب‌های تکرارهای بعدی مشخص می‌شوند و در نهایت جوابی نزدیک به واقعیت به دست می‌آید. الگوریتم‌های تقریبی به دو دسته ابتکاری و فراابتکاری تقسیم می‌شوند. الگوریتم‌های ابتکاری در مسائلی که پیچیدگی زیادی ندارند، می‌توانند جواب بهینه را بیابند. الگوریتم‌های فراابتکاری برای حل مشکلات الگوریتم‌های ابتکاری الگوهایی ارائه کرده‌اند. یکی از ویژگی‌های مهم برای این گونه الگوریتم‌ها این است که برای یک محدوده وسیع که جواب تحلیلی ندارند، نزدیکترین جواب به حل مساله را خواهند یافت [۷]. در واقع این نوع الگوریتم‌ها راهکارهای برون‌رفت از نقاط بهینه محلی و قابلیت کاربرد در طیف گسترده‌ای از مسائل را دارند. از آنجایی که الگوریتم‌های فراابتکاری بر جستجوی عمیق در نقاط مختلف از فضای جواب تاکید دارند، جواب‌های با کیفیت‌تری نسبت به الگوریتم‌های ابتکاری تولید می‌کنند.

شکل (۱): طبقه‌بندی الگوریتم‌ها در حوزه مسیریابی

۱.۲. بررسی روش‌های پیشین

مساله مسیریابی در سال ۱۹۵۹ توسط دانتزیک و رامسر مطرح شد [۸]. این مقاله مبنای بسیاری از تحقیقات بعدی قرار گرفت

و پس از آن، مسیریابی وسایل نقلیه به کمک الگوریتم‌های فراابتکاری بررسی شدند. دوریگوو نسخه اولیه الگوریتم ACO را برای مساله فروشنده دوره‌گرد در سال ۱۹۹۱ منتشر کرد [۹]. از سال ۲۰۰۲ به بعد مساله مسیریابی وسایل نقلیه با ترکیب الگوریتم‌های فراابتکاری مورد بررسی قرار گرفت. در مرجع [۱۰]، از PSO برای بهینه‌سازی متغیرهای الگوریتم ACO استفاده شد. در مرجع [۱۱]، از ترکیب ACO و PSO در مساله برنامه‌ریزی مسیر استفاده شد. در مقاله [۱۲]، روش هیبرید را با ACOPSO بهبود بخشیدند. در مرجع [۱۳]، یک الگوریتم بهبودیافته ACOPSO برای حل مشکل فروشنده دوره‌گرد در مقیاس بزرگ پیشنهاد شده است. در مقاله [۱۴]، برای مساله زمانبندی مسیریابی استراتژی حریص تطبیقی، در مقاله [۱۵] استراتژی خوشه‌بندی و در مقاله [۱۶] استراتژی تولید سلسله مراتبی با الگوریتم PSO ارائه شده است. در مرجع [۱۷]، با استفاده از منطق فازی و در مرجع [۱۸] با استفاده از الگوریتم زنبور عسل با تابع Multi insert مساله مسیریابی حل شده است. در جدول (۱) خلاصه فعالیت‌های انجام شده در زمینه مسیریابی وسایل نقلیه شامل الگوریتم‌های حل کلاسیک و الگوریتم‌های فراابتکاری و ترکیبی اخیر لیست شده است. لازم به ذکر است با توجه به محدودیت فضا امکان بررسی جامع کارهای پیشین وجود ندارد علاقمندان می‌توانند با مراجعه به مرجع [۱۹] توضیحات کاملتری را دریافت کنند.

۲.۲. روش‌های پایه

با توجه به اینکه روش ارائه شده در این مقاله مبتنی بر الگوریتم‌های ACO و PSO می‌باشد، در ادامه به شرح آنها می‌پردازیم.

۱.۲.۲. الگوریتم ACO

در مرجع [۲۰]، فرآیند جستجوی مورچه دو مدل احتمال انتقال و مدل به‌روزرسانی فرومون دارد. در این الگوریتم، مورچه‌ها در شبکه $n \times m$ قرار می‌گیرند که n تعداد گره‌ها و m تعداد یال‌ها می‌باشد. مورچه‌ها یک راس را از میان جواب‌های کاندید برای حرکت روبه‌جلو انتخاب می‌کنند. در رابطه (۱) که به رابطه احتمال معروف است، داریم:

$$P_{ij}^{k} = \begin{cases} \dfrac{\mathcal{T}_{ij}^{\alpha}\,\eta_{ij}^{\beta}}{\sum_k \mathcal{T}_{ij}^{k}\,\eta_{ij}^{k}} & j \in J_K(i) \\ 0 & j \notin J_K(i) \end{cases} \qquad (۱)$$

که P_{ij}^{k} احتمال حرکت مورچه k از نقطه i به نقطه j، $J_k(i)$ شهرهایی که هنوز مورچه k انتخاب نکرده و $\mathcal{T}_{ij}$ شدت غلظت فرومون یال (i,j) را مشخص می‌کند. همچنین، متغیرهای α و β به ترتیب اهمیت نسبی شدت غلظت فرومون مسیر و اطلاعات هیوریستیکی را اندازه‌گیری می‌کنند. هرچه میزان غلظت فرومون مسیر بیشتر باشد، احتمال انتخاب آن بیشتر می‌شود و تصادفی بودن جستجو کم می‌شود. متغیر β هرچه مقدار بیشتری داشته باشد، عملکرد مورچه‌ها بر اساس اطلاعات هیوریستیکی بیشتر خواهد بود. متغیر η_{ij} که براساس رابطه (۲) محاسبه می‌شود، اطلاعات هیوریستیکی مساله است که می‌تواند هر نوع داده اضافی مانند ترافیک جاده‌ها باشد به طوری که اولویت شهرها را برای حرکت انتخاب می‌کنند. در این رابطه، d_{ij} فاصله بین شهرهای i و j است که η_{ij} رابطه معکوسی با d_{ij} (تابع هزینه) دارد.

$$\eta_{ij}(t) = \frac{1}{d_{ij}} \qquad (۲)$$

به‌روزرسانی فرومون بر اساس رابطه (۳) محاسبه می‌شود.

$$\mathcal{T}_{ij}(t+n) = (1-\rho) \times \mathcal{T}_{ij}(t) + \Delta\tau_{ij} \qquad (۳)$$

در این رابطه ρ نرخ تبخیر فرومون است که از تجمع فرومون جلوگیری می‌کند. این امر سبب می‌شود تصمیمات نامناسب قبلی در نظر گرفته نشود و در نتیجه از همگرایی سریع جلوگیری می‌کند. متغیر $\Delta\tau_{ij}$ مقدار فرومونی است که توسط مورچه در یال (i,j) به‌جا گذاشته می‌شود و طبق رابطه (۴) محاسبه می‌شود. در این رابطه m تعداد مورچه‌ها را نشان می‌دهد و $\Delta\tau_{ij}^{k}$ مقدار فرومونی است که قرار است توسط مورچه k در یال (i,j) ریخته شود و مقدار آن توسط رابطه (۵) محاسبه می‌شود که مقدار Q معمولا ۱ در نظر گرفته می‌شود و L_k طول دور کامل ساخته شده توسط مورچه k است که با مقدار مشخصی تقویت می‌شود.

جدول (۱): خلاصه پژوهش‌های انجام شده در حوزه مسیریابی

مرجع	سال	روش پیشنهادی	ابزار شبیه‌سازی-مجموعه داده	معیارهای کارایی (مزایا و معایب)
[۱۲]	۲۰۰۲	شبیه‌سازی تبرید	-	از الگوریتم شبیه‌سازی تبرید با انتخاب گروهی از جواب‌ها شروع می‌شود و در هر تکرار با جواب‌های همسایه مقایسه می‌شود. عیب روش همگرا شدن سریع و عدم توانایی در پیدا کردن بهینه‌ترین جواب است.
[۱۳]	۲۰۰۵	الگوریتم ژنتیک	-	از الگوریتم ژنتیک استاندارد استفاده شده که به شدت میل به همگرایی زودرس دارد. عیب این الگوریتم جواب‌های غیربهینه با نرخ انحراف بالا است.
[۲۰]	۲۰۰۹	روش ابتکاری دوفازی برای vrp	-	در این مقاله برای بهبود متغیرهای ACO از ژنتیک استفاده شده است. نتایج به دست آمده نشان می‌دهد موفقیت آنچنانی ندارد و نرخ انحراف جواب‌ها بالا است.
[۲۱]	۲۰۱۰	بهینه‌سازی متغیرهای ACO براساس PSO	MATLAB TSPLIB	با الگوریتم PSO از همگرایی زودرس جلوگیری می‌کند و جواب اولیه را در اختیار ACO قرار می‌دهد. مزیت این روش عدم همگرایی زودرس اولیه است و معایب آن تعداد تکرار بالا برای پیدا کردن مسیر بهینه و بالا بودن نرخ انحراف جواب‌ها است.
[۶]	۲۰۱۹	برنامه‌ریزی مسیر پویا ACOPSO	MATLAB TSPLIB	این الگوریتم از PSO برای جلوگیری از همگرایی زودرس ACO، جواب اولیه را پیدا کرده و در اختیار ACO قرار می‌دهد. مزیت این روش جستجوی جهانی به نسبت قوی و اختصاص حافظه مجزا به هر مورچه برای نگه داشتن جواب بهینه فردی و عیب آن استفاده از PSO فقط برای جواب اولیه است که با گذر زمان از جواب‌های بهینه فاصله می‌گیرد. همچنین پیچیدگی آن تا حدودی بالا است.
[۱۴]	۲۰۲۲	استراتژی حریص تطبیقی	MATLAB TSPLIB	این الگوریتم از استراتژی تطبیقی استفاده می‌کند. مزیت روش جستجوی در سطح جهانی است، ولی عیب روش استفاده از روش حریصانه در بروزرسانی مسیرها است که در نهایت در پیدا کردن مسیرهای بهینه تاثیر منفی دارد.
[۱۵]	۲۰۲۲	استراتژی خوشه‌بندی	MATLAB TSPLIB	این الگوریتم از سه مفهموم خوشه‌بندی، تبخیر فرومون تطبیقی و شرایط پایان جدید بر اساس تنوع راه‌حل در جمعیت برای بهبود عملکرد کلی و کاهش زمان اجرا استفاده می‌کند. مزیت روش تنوع جواب‌ها در سطح جهانی است. از معایب این روش می‌توان به تعداد تکرار بالا در پیدا کردن مسیر بهینه اشاره کرد.
[۱۶]	۲۰۲۲	استراتژی تولید سلسله مراتبی با الگوریتم PSO	MATLAB TSPLIB	این روش با استفاده از الگوریتم PSO از همگرایی زودرس جلوگیری می‌کند و برای بروزرسانی مسیر از یک استراتژی سلسله مراتبی دولایه مبتنی بر یادگیری حریصانه استفاده می‌کند. مزیت آن عدم همگرایی اولیه است و عیب آن هم نرخ انحراف بالای جواب‌ها به دلیل استفاده از روش حریصانه است.
[۱۷]	۲۰۲۲	استفاده از منطق فازی	TSPLIB-JAVA MATLAB	این روش با منطق فازی سعی در حل مشکل همگرا شدن زودرس ACO دارد. مزایای روش هم این است که تا حدودی توانسته با منطق فازی و استفاده از کنترل‌کننده FACS از همگرا شدن روش ACO جلوگیری کند. عیب این روش هم پیچیدگی بیش از حد آن است که در هر تکرار برای تابع ابتکاری باید ورودی و خروجی‌هایی تعیین شود که این کار خود موجب افزایش تعداد تکرار برای پیدا کردن مسیر بهینه می‌شود.
[۱۸]	۲۰۲۳	الگوریتم زنبور عسل	MATLAB TSPLIB	این روش از تابع ابتکاری MULTI-INSERT استفاده می‌کند و بهترین همسایه را در اولویت قرار می‌دهد. مزیت روش تعداد بسیار کم متغیرها می‌باشد و عیب آن تکیه بر جواب‌های بهینه ابتدایی و نادیده گرفتن جستجوی جهانی در هر تکرار است.

$$\Delta\tau_{ij} = \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \quad (۴)$$

$$\Delta\tau_{ij}^{k} = \begin{cases} \frac{Q}{L_K} & \text{اگر یال } (i,j) \text{ توسط مورچه k طی شود} \\ 0 & \text{در غیراینصورت} \end{cases} \quad (۵)$$

۲.۲.۲. بهینه‌سازی ازدحام ذرات

در الگوریتم ازدحام ذرات [۲۱]، هر ذره دو ویژگی سرعت V_t و موقعیت X_t دارد که بر اساس تجربیات شخصی و گروهی

تغییر می‌کنند. تغییر مکان ذره تحت تاثیر همسایگان آن ذره قرار می‌گیرد و ذرات به سمت نواحی موفق حرکت می‌کنند. علت آن این است که ذرات از همدیگر می‌آموزند و با دانش کسب شده به سمت بهترین همسایگان حرکت می‌کنند و در ادامه، در حین جستجو مقادیر مناسب بهینه فردی انتخاب می‌شود و ذره آن را به خاطر می‌سپارد. همین روال ادامه می‌یابد تا بهینه سراسری انتخاب شود. سرعت با توجه به رابطه (۶) و موقعیت با توجه به رابطه (۷) به‌روزرسانی می‌شود.

$$v_i(t+1) = wv_i(t) + c_1r_1\big(pbest_i - x_i(t)\big) + c_2r_2(gbest - x_i(t)) \quad (۶)$$

$$x_i(t+1) = x_i(t) + v_i(t) \quad (۷)$$

در رابطه (۶)، w وزنی مثبت برای تنظیم محدوده جستجو سراسری است. c_1 فاکتور یادگیری مدل ادراکی و نشان‌دهنده میزان یادگیری فردی در هنگام جستجو در فضا است و c_2 فاکتور یادگیری مدل جمعی و میزان یادگیری گروهی و به اشتراک‌گذاری اطلاعات اعضا در فضا را نشان می‌دهد. r_1 و r_2 دو متغیر با مقادیر تصادفی در بازه [۰،۱] هستند که از همگرایی متغیرها ممانعت می‌کنند. $pbest_i$ بهترین موقعیت ذره i و $gbest$ بهترین موقعیت سراسری را نمایش می‌دهد. در رابطه (۷)، موقعیت جدید ذرات را با توجه به موقعیت فعلی $x_i(t)$ و رابطه سرعت $v_i(t)$ به‌روزرسانی می‌کند.

۳. روش پیشنهادی

الگوریتم‌های فراابتکاری در حل مساله مسیریابی نقش بسزایی دارند. با توجه به خاصیت جستجوی هیوریستیک در این دسته از مسائل، الگوریتمی باید انتخاب شود که در شرایط پیچیده و غیرقطعی که نیازمند همکاری و هوشمندی اعضا است، بتواند پاسخگوی موضوع باشد. برای این منظور از ترکیب الگوریتم کلونی مورچگان و الگوریتم ازدحام ذرات استفاده می‌شود. در ادامه برای افزایش دقت الگوریتم‌های ذکر شده باید به دنبال تابع ابتکاری بود که تعداد تکرارها را برای پیدا کردن مسیر بهینه کاهش دهد. برای این هدف تابع ابتکاری کسینوس زوایا به الگوریتم ترکیبی ACOPSO اضافه شده است. محدوده تنظیم متغیر الگوریتم کلونی مورچگان شامل سه متغیر کلیدی α، β و ρ است. از آنجایی که دامنه این متغیرها گسترده است، تنظیم آنها چالش برانگیز است و انتخاب نامناسب متغیرها منجر به افزایش زمان جستجو و کاهش دقت در پیدا کردن مسیر بهینه می‌شود. در مرجع [۶]، از ترکیب ACO با PSO که دارای ویژگی‌های مطلوبی مانند متغیرهای کمی، جستجو مشارکتی و همگرایی سریع است بدین شکل استفاده شده است که PSO فقط جواب اولیه مناسب را پیدا می‌کند و در اختیار ACO قرار می‌دهد ولی در این مقاله PSO به عنوان بخشی ثابت در تمامی تکرارها استفاده می‌شود. مورچه‌های ACO مثل ذرات PSO از تجربیات همدیگر استفاده می‌کنند. استقلال انتخاب مقصد توسط مورچه‌ها وجود ندارد و مورچه‌ها ملزم به استفاده از تجربیات فردی همدیگر هستند. برای هر مورچه یک حافظه اختصاص داده می‌شود که بهترین تجربه شخصی خود را در آن ذخیره کند و حالا علاوه بر اثر فرومون مسیر، تجربیات بقیه مورچه‌ها (بهینه فردی و سراسری) در به‌روزرسانی فرومون تاثیر داده می‌شود. اعمال PSO برای بهینه‌سازی متغیرهای ACO می‌تواند تاثیر محدودیت‌های الگوریتم ACO را تا حد زیادی کاهش دهد و در نتایج بسیار موثر باشد [۲۲]. برای توضیح بیشتر روش پیشنهادی گرافی با n ایستگاه با مختصات مختلف در نظر می‌گیریم. مطابق شکل (۲) قبل از اقدام به مسیریابی متغیرهای ورودی مساله تعیین می‌شوند. ابتدا تعداد مورچه‌ها (خودروها) و تعداد تکرارهای الگوریتم تعیین و با توجه به توضیحات ذکر شده در بخش ۱.۲.۲، متغیرهای الگوریتم کلونی مورچگان شامل میزان بزرگنمایی فرومون (α)، میزان بزرگنمایی هیورستیک (β) و شدت تبخیر فرومون را ابتدا باید مقداردهی کرد. همچنین یک ماتریس فرومون تشکیل و در هر مرحله از تکرار، میزان فرومون هر مسیر تعیین می‌شود. با توجه به تاثیر دادن بهینه‌های فردی و بهینه سراسری الگوریتم PSO در الگوریتم ACO این مقادیر به عنوان دو متغیر اساسی در پیدا کردن مسیر بهینه باید دارای مقادیر اولیه مناسب باشند که با تاثیر دادن آنها و اثر ترافیک جاده‌ها به عنوان متغیرهای

موثر در رابطه (۱) می‌توان اقدام به تولید جواب‌های مناسب کرد.

شکل (۲): مراحل الگوریتم ACOPSOCA

فرض کنید تعداد m مورچه روی مسیر هستند و الگوریتم‌ها K مرتبه تکرار می‌شوند. برای تولید مسیرهای تصادفی اولیه توسط مورچه‌ها به ازای فرومون و ترافیک اولیه ثابت ماتریسی به ابعاد $n \times m$ ایجاد می‌شود. هر سطر نماینده یک مسیر مورچه و هر ستون نماینده یک ایستگاه است. مورچه‌ها اولین ایستگاه (مبدا) را انتخاب می‌کنند. سپس به ازای هر تکرار به تعداد ایستگاه‌ها تکرار فرعی دارند. در این تکرارها مورچه‌ها قدم به قدم به سمت مقصد، ایستگاه‌ها را انتخاب می‌کنند. در تکرارهای فرعی مورچه اول تا مورچه m-اُم یک ایستگاه را بین گره‌های $\{0,1,\dots,m\}$ انتخاب می‌کنند. همچنین به ازای هر مورچه i یک متغیر بهینه فردی $(pbest_i)$ مطابق با رابطه (۸) که شامل کمینه امتیاز $(score)$ مورچه‌ها در هر تکرار (k) است، تعریف می‌شود.

$$pbest_i = \min_{k\in\{1,\dots,K\}} score(i,k) \qquad (۸)$$

به ازای کل مورچه‌ها یک متغیر بهینه سراسری $(gbest)$ مطابق با رابطه (۹) که بهترین بهینه فردی را در خود نگه می‌دارد، نیز تعریف می‌شود.

$$gbest = \min_{i\in\{1,\dots,m\}} pbest(i) \qquad (۹)$$

برای به‌روزرسانی ترافیک، فرض کنید وضعیت ترافیک مطابق شکل (۳) است. که از نقطه مبدا p_0 بین مسیرهایی موجود $\{p_1,p_2,p_3\}$، نقطه p_2 که دارای ترافیک (C_{02}) ۸ است، به عنوان مقصد بعدی انتخاب شود. حال در نقطه p_2 از بین مسیرهای $\{p_3,p_4,p_5\}$، فرض کنید مسیر p_5 برای ادامه مسیر انتخاب شود. بدین ترتیب مقدار ترافیک مسیر p_0 به p_2 یک واحد کم $(C_{02}=7)$ و مقدار ترافیک مسیر p_2 به p_5 یک واحد افزایش $(C_{25}=7)$ پیدا می‌کند. بعد از به‌روزرسانی ماتریس ترافیک، مقادیر آن را در ماتریس اصلی فرومون تاثیر داده می‌شود.

شکل (۳): وضعیت ترافیکی فرضی یک منطقه‌ی شهری

در فرآیند مسیریابی که انجام می‌دهیم، برای به‌روزرسانی ماتریس فرومون $T[i,j]$ مطابق با رابطه (۱۰) علاوه بر فرومون روی مسیر، بهینه‌های فردی و سراسری مورچه‌ها نیز تاثیر داده می‌شوند.

$$T[i,j] = T[i,j] + c_1 \times \sum_{i=1}^{m} pbest_i + c_2 \times (gbest) \qquad (۱۰)$$

در ادامه برای افزایش دقت و کاهش زمان مسیریابی، تابع ابتکاری کسینوس زوایا اعمال می‌شود. در مسیریابی که از یک

مبدا به یک مقصد مشخص صورت می‌گیرد، همیشه مقصد نهایی معلوم است و جهت بین مبدا و مقصد به عنوان یک بردار راهنما در نظر گرفته می‌شود. اگر مقصد نهایی مشخص نباشد، محاسبه کسینوس زوایا امتیازی محسوب نمی‌شود. الگوریتم ACO مراحل اولیه را کاملا تصادفی و بدون نظم خاصی جستجو می‌کند و هرچه جلوتر می‌رود و تعداد تکرارهای بیشتری انجام می‌شود، انتخاب مسیرها از حالت تصادفی خارج شده و هدفمند می‌شود. چون مراحل اولیه کاملا تصادفی است، می‌توان از خاصیت کسینوس زوایا که فقط روی نقاطی تاکید می‌کند که هم راستای مسیر پیمایش به سمت مقصد نهایی است، استفاده کرد تا دقت الگوریتم ACOPSO برای انتخاب مسیرها در ابتدای مسیریابی بالا برود و همچنین تعداد تکرارها برای پیدا کردن مسیر بهینه کاهش یابد. هرچه زاویه بین شهر مورد نظر و بردار راهنما بزرگ‌تر باشد، مقدار کسینوس به سمت صفر میل می‌کند و احتمال انتخاب آن کمتر می‌شود. اگر زاویه کمتر باشد، مقدار کسینوس به سمت یک میل می‌کند و احتمال انتخاب آن بالا می‌رود. در مسیریابی بر اساس الگوریتم ACO، هنگام حرکت از مبدا به مقصد، با افزایش طول مسیر و بعد از گذشت تکرارهای طولانی تفاوت بین مسیر کوتاه و مسیر بهینه کاهش می‌یابد. ولی این تفاوت ممکن است بعد از گذشت زمان زیادی به صفر برسد. بدین منظور برای افزایش دقت محاسبات در مراحل ابتدایی و همچنین برای کاهش زمان جستجو در مسیریابی، تابع ابتکاری کسینوس زوایا را با الگوریتم تلفیقی ACOPSO ترکیب می‌کنیم. در تابع ابتکاری ACOPSOCa، مطابق شکل (۴)، خط‌چین نارنجی به عنوان بردار راهنما در نظر گرفته می‌شود و زاویه بین هر شهر و بردار راهنما مطابق رابطه $cos(\theta) = \frac{u.v}{|u||v|}$ محاسبه می‌شود که u و v به ترتیب گره مبدا و گره مقصد هستند. به این ترتیب در مرحله اول مسیریابی زاویه بین بردار راهنما و مسیرهای p_0p_1 (که با θ_{01} نشان داده می‌شود)، p_0p_2 (θ_{02}) و p_0p_3 (θ_{03}) محاسبه می‌شود و مسیر دارای زاویه کمتر شانس بیشتری برای انتخاب می‌گیرد. این رابطه با توجه به ویژگی‌های جغرافیایی مساله عمل می‌کند و می‌تواند تصمیم‌گیری‌هایی متناسب با شرایط را انجام دهد. در واقع در هر مرحله کسینوس زاویه بین گره مقصد و همه گره‌ها محاسبه و در ماتریس احتمال الگوریتم کلونی مورچگان تاثیر داده می‌شود. با این کار مسیرها و گره‌های نامربوط بیشتری از دایره محاسبات خارج می‌شوند و احتمال انتخاب مسیر دارای کمترین زاویه با مقصد، افزایش می‌یابد. در ادامه نحوه عملکرد تابع ابتکاری کسینوس زوایا را در قالب یک مثال بررسی خواهیم کرد.

شکل (۴): محاسبه کسینوس زوایا بین نقاط مبدا و مقصد

مطابق شکل (۵)، ده گره فرضی با مختصات متفاوت را از یک منطقه شهری در نظر می‌گیریم. بین تمامی گره‌ها مسیرهایی با طول متفاوت وجود دارد. در ابتدا باید مقدار فرومون روی مسیر مشخص شود. برای این منظور، مقدار همه فرومون‌های روی مسیر را برابر عدد یک قرار می‌دهیم. مسیریابی مورد نظر از یک مبدا به یک مقصد مشخص صورت می‌گیرد. مطابق شکل (۵) می‌توانیم از این مورد استفاده کنیم و یک بردار راهنما داشته باشیم و در مسیریابی که در هر مرحله انجام می‌دهیم بردار راهنما را مدنظر قرار دهیم. در ادامه فرض می‌کنیم p_0 شهر مبدا است و قرار است به شهر p_6 به عنوان شهر مقصد برویم. کاندیدای مقصد بعدی p_0 شهرهای $\{p_1, p_2, p_3\}$ هستند که برای انتخاب شهر بعدی کسینوس زاویه بین شهرهای مقصد و شهرهای کاندید محاسبه می‌شود. فرض کنید کسینوس زاویه بین بردار راهنما و مسیر p_0p_1، ۶۰ درجه و $\cos(60) = 0.5$، کسینوس زاویه بین بردار راهنما و مسیر p_0p_2 برابر ۱۵درجه و $\cos(15) = 0.96$ و کسینوس زاویه بین بردار راهنما و مسیر p_0p_3 برابر با ۳۵ درجه است و $\cos(35) = 0.81$ می‌باشد.

بعد از مشخص شدن مسیرهای ارتباطی بین شهر p_0 و سایر

شهرها، مطابق شکل (۶)، مقدار کسینوس به دست آمده از هر مسیر با فرومون روی مسیر جمع می‌شود و در ماتریس تاثیر داده می شود. حال مورچه (خودرو روی مسیر) بر اساس ماتریس محاسبه شده که شامل تاثیر فرومون مسیر، حجم ترافیک مسیر و مقدار کسینوس زوایا است، عمل می‌کند و شهری را به عنوان شهر بعدی انتخاب می‌کند که دارای مقدار نهایی بزرگ‌تری است. بنابراین، مسیر بین شهر p_0 به شهر p_2 با مقدار کسینوس ۱/۹۶، احتمال بیشتری برای انتخاب به عنوان مقصد بعدی دارد ($p_2 = 1.96 > p_3 = 1.81 > p_1 = 1.5$). نکته حائز اهمیت این است که لزوما مقدار کسینوس بزرگ‌تر به معنای انتخاب همان مسیر نیست بلکه مقدار کسینوس بزرگ‌تر یعنی شانس بیشتری برای انتخاب مسیر متناظر وجود دارد. حال فرض می‌کنیم مقدار کسینوس زاویه مسیر شهر p_2 در ماتریس احتمالات تاثیر داده شده و به عنوان شهر بعدی انتخاب شده است. بعد از رسیدن به شهر p_2 کاندیدای انتخاب مقصد بعدی شهرهای $\{p_1, p_3, p_4, p_5\}$ هستند. با توجه به اینکه زاویه بین شهر p_2 و p_5 و بردار راهنما نسبت به سایر مسیرها کمتر است و کسینوس زاویه آن دارای مقدار بزرگ‌تری است، احتمال انتخاب شهر p_5 به عنوان مقصد بعدی نسبت به سایر شهرها بیشتر است. فرض می‌کنیم شهر p_5 به عنوان مقصد بعدی انتخاب شود. در ادامه کاندیدای انتخاب مقصد بعدی $\{p_4, p_7, p_8, p_9\}$ هستند که با توجه به محاسبات کسینوس زاویه، احتمال انتخاب شهر p_7 نسبت به سایر شهرها بیشتر است. فرض می‌کنیم، شهر p_7 به عنوان مقصد بعدی انتخاب شود. در نهایت در شهر p_7 کاندیدای مقصد بعدی شهرهای $\{p_8, p_6\}$ هستند که با توجه به محاسبات کسینوس زاویه، احتمال انتخاب شهر p_6 به عنوان مقصد نهایی بیشتر از سایر شهرها است.

در ادامه، مراحل الگوریتم ACOPSOCa مطابق شکل (۲) طی می‌شود. سرانجام، بعد از اتمام تعداد تکرارهای تعیین شده الگوریتم، بر اساس ماتریس تابع هدف (هزینه) که در هر گام بروزرسانی شده، بهترین مسیر نهایی انتخاب شده و فرآیند مسیریابی به پایان می‌رسد.

شکل (۵): کاندیدای مقصد بعدی هر مورچه به کمک کسینوس زوایا

شکل (۶): اضافه کردن مقادیر کسینوس زاویه به فرومون روی مسیر

۱.۳. پیچیدگی محاسباتی روش پیشنهادی

در این مقاله مقرون به صرفه بودن الگوریتم ACO با تعداد مورچه‌ها، تقاطع‌ها و زمان‌های تکرار، رابطه خطی دارد. وقتی زمان تکرار به مقدار معین از پیش تعیین شده برسد و پیشرفتی در جواب‌های بهینه حاصل نشود، الگوریتم از اجرا صرف نظر می‌کند. بنابراین پیچیدگی محاسبات را می‌توان به صورت $O(n \times (n-1)^2 \times m \times \frac{T}{2})$ بیان کرد که n تعداد تقاطع‌ها و m تعداد مورچه‌ها را نشان می‌دهد و T نشان‌دهنده زمان تکرار است. در آزمایش صورت گرفته زمان اجرا به زمان مصرف شده از نقطه شروع تا نقطه پایان یا زمان مورد نیاز برای ایجاد بهترین مسیر در تعداد تکرارها اشاره دارد. از بیان پیچیدگی محاسبات به راحتی می‌توان مشاهده کرد که وقتی تعداد مورچه‌ها و تعداد تقاطع‌ها ثابت است، پیچیدگی محاسبه و زمان اجرا با زمان تکرار رابطه مثبت دارد. به عبارت دیگر، هرچه تعداد تکرارها بیشتر باشد، پیچیدگی محاسباتی بیشتر و به زمان بیشتری نیاز است.

۴. ارزیابی

برای ارزیابی روش پیشنهادی، نتایج حاصل از اجرای الگوریتم ACOPSOCa با جدیدترین و رایج‌ترین روش‌ها بر روی داده‌های کتابخانه معروف TSPLIB [۲۳] مقایسه می‌شود. برای این منظور دوازده ارزیابی مبتنی بر فاصله فیزیکی طراحی شده که در هر ارزیابی روش پیشنهادی با نتایج روش‌های ACOPSO [۱]، ICMPACO[1] [۲۴]، ACSGA[2] [۲۵]، DSOS[3] [۲۶]، DPC-ACO-KOPT[4] [۲۷]، EHO[5] [۲۸]، SOS-SA[6] [۲۹]، DA[7] [۳۰]، GSACO[8] [۱۴]، BA[9] [۱۸] مقایسه شده و نتایج حاصل با توجه به داده‌های موجود در دو بخش اصلی ارزیابی می‌شوند.

در بخش اول، نرخ انحراف (دقت الگوریتم) و در بخش دوم میانگین تعداد تکرار (زمان محاسبات)، برای رسیدن به بهترین جواب بررسی می‌گردد. در ارزیابی یازدهم، بر روی نقاط شلوغ و دارای ترافیک سنگین شهر پکن [۱] ارزیابی دیگری انجام خواهیم داد. در ارزیابی دوازدهم، برای بررسی تاثیر تابع کسینوس زوایا بر روی الگوریتم ACOPSO، بر اساس مرجع [۶] مسیرهایی از گراف شبکه‌ای شهر پکن انتخاب می‌شود و هزینه زمان سفر بر روی هر مسیر محاسبه می‌شود. در این آزمایش‌ها چهار معیار مهم برای ارزیابی وجود دارد که Mean-iter میانگین تعداد تکرار برای به دست آوردن بهترین نتیجه نهایی، Mean-best میانگین بهترین نتیجه راه‌حل، best بهترین نتیجه از میان نتایج Mean-best و dev_{rate} نرخ انحراف برای مقایسه دقت الگوریتم‌ها است که مطابق رابطه (۱۱) محاسبه می‌شود.

$$dev_{rate} = \frac{|result_{avr} - result_{best}|}{result_{best}} \times 100 \qquad (۱۱)$$

در رابطه (۱۱)، $result_{avr}$ میانگین حاصل از بهترین نتایج به دست آمده و $result_{best}$ بهترین نتیجه از میان نتایج به دست آمده است. در این رابطه هرچه مقدار نرخ انحراف پایین‌تر باشد، نشان‌دهنده عملکرد صحیح و هدفمند الگوریتم است و هرچه مقدار آن بالاتر باشد، عملکرد نامنظم الگوریتم را نشان می‌دهد.

توجه داشته باشید که در ارزیابی‌های صورت گرفته در هر نمودار، به ازای هر مجموعه داده روش پیشنهادی مقاله با روش‌های سایر مقالات در دو بخش نرخ انحراف Accuracy و تعداد تکرارها Mean-iter مقایسه شده است. در توضیحات هر نمودار، میانگین تمام حالت‌ها به صورت درصد بیان شده است. علاقمندان می‌توانند به مجموعه داده‌های استفاده شده و کدهای شبیه‌سازی مقاله در مرجع [۱۹] دسترسی داشته باشند.

۱.۴. ارزیابی اول

در مرجع [۴]، آزمایشی برای بهینه‌سازی متغیرهای ACO بر اساس PSO طراحی شده است. خود این ارزیابی شامل سه نوع آزمایش متفاوت است که دو آزمایش برای انتخاب تصادفی متغیرهای ACO و یک آزمایش برای بهینه‌سازی متغیرها ACO در نظر گرفته شده است. در آزمایش انتخاب تصادفی متغیرهای ACO، مقادیر α، β و ρ به طور تصادفی در محدوده‌ای تنظیم می‌شوند و مقادیر سه متغیر با هر تکرار تغییر می‌کنند. در آزمایش بهینه‌سازی متغیرهای ACO، مقادیر سه متغیر α، β و ρ قبل از تکرار تعیین می‌شود و در طول فرآیند تکرار تغییر نمی‌کنند. در این آزمایش‌ها، برای انتخاب متغیرها طبق نتیجه‌گیری به دست آمده در مرجع [۳۱]، محدوده متغیرها α، β و ρ طبق جدول (۲) تنظیم می‌شود.

جدول (۲): محدوده متغیرهای ارزیابی اول

آزمایش	تعداد دفعات اجرا	حداکثر تعداد تکرارها	عدم تغییر راه‌حل مطلوب	α	β	ρ
آزمایش اول	100	1000	200	0.1-0.3	3-6	0.7-0.9
آزمایش دوم	100	1000	200	0.5-1.5	1-5	0.5-1
آزمایش سوم	100	1000	200	0.5-1.5	1-5	0.5-1

[1] Multi-Population Strategy
[2] Ant Colony System & Genetic Algorithm
[3] Discrete Symbiotic Organisms Search
[4] Density Peaks Clustering & Ant Colony Optimization
[5] Elephant Herding Optimization
[6] Symbiotic Organisms Search & Simulated Annealing
[7] Dragonfly Algorithm
[8] Ant Colony Optimization Algorithm based on Adaptive Greedy Strategy
[9] Best Algorithm

نرخ انحراف در مجموعه داده‌های مختلف به طور متوسط در آزمایش اول مطابق شکل (۷)، ۸۵/۲۸٪، در آزمایش دوم مطابق شکل (۹)، ۸۱/۵۴٪ و در آزمایش سوم مطابق شکل (۱۱)، ۸۹/۸۹٪ است که دقت روش ACOPSOCa به نسبت روش ACO و ACOPSO بیشتر بوده است. روش ACOPSOCa مطابق شکل (۸) در کاهش تعداد تکرارها در آزمایش اول نسبت به روش ACO متوسط ۱۱/۵۸٪، در آزمایش دوم مطابق شکل (۱۰)، ۱۰/۸۹٪ و در آزمایش سوم مطابق شکل (۱۲)، ۸/۲۱٪ بهبودی داشته است.

شکل (۷): مقایسه نرخ انحراف (دقت) الگوریتم‌ها در آزمایش اول

شکل (۸): مقایسه میانگین تعداد تکرار در آزمایش اول

شکل (۹): مقایسه نرخ انحراف (دقت) الگوریتم‌ها در آزمایش دوم

شکل (۱۰): مقایسه میانگین تعداد تکرار در آزمایش دوم

شکل (۱۱): مقایسه نرخ انحراف (دقت) الگوریتم‌ها در آزمایش سوم

شکل (۱۲): مقایسه میانگین تعداد تکرار در آزمایش سوم

۲.۴. ارزیابی دوم

در مرجع [۲۴]، یک الگوریتم بهبودیافته کلونی مورچگان بر اساس رویکرد چند جمعیتی برای بهبود عملکرد مسیریابی پیشنهاد شده است. مقایسه نرخ انحراف با توجه به شکل (۱۳) نشان می‌دهد در مجموعه داده‌های مختلف به جز ch130 به طور متوسط ۶۲/۱۶٪، دقت روش ACOPSOCa به نسبت روش ICMPACO بیشتر بوده است.

۳.۴. ارزیابی سوم

در مرجع [۲۵]، ترکیب الگوریتم کلونی مورچگان و الگوریتم

ژنتیک برای بهبود عملکرد مسیریابی پیشنهاد شده است. مقایسه نرخ انحراف با توجه به شکل (۱۴) نشان می‌دهد در مجموعه داده‌های مختلف دقت روش ACOPSOCa به نسبت روش ACSGA به طور متوسط ۲۷/۶۶٪ بیشتر بوده است.

شکل (۱۳): مقایسه نرخ انحراف الگوریتم‌ها در ارزیابی دوم

شکل (۱۴): مقایسه نرخ انحراف الگوریتم‌ها در ارزیابی سوم

۴.۴. ارزیابی چهارم

در مرجع [۲۶]، یک الگوریتم جستجوی موجودات همزیست گسسته DSOS برای یافتن راه‌حل مناسب برای مسائل مسیریابی پیشنهاد شده است. الگوریتم DSOS یک الگوریتم فراابتکاری طبق اکوسیستم بقا و تکثیر است. مقایسه نرخ انحراف با توجه به شکل (۱۵) نشان می‌دهد در مجموعه داده‌های مختلف به طور متوسط دقت روش ACOPSOCa به میزان ۴۶/۶٪ به نسبت روش DSOS بیشتر بوده است. در شکل (۱۶) مقایسه زمان محاسبات برای به دست آوردن مسیر بهینه نشان داده شده است. بر اساس نتایج حاصل شده، روش ACOPSOCa در کاهش زمان محاسبات نسبت به روش DSOS به طور متوسط ۴/۱۶٪ بهبودی داشته است.

شکل (۱۵): مقایسه نرخ انحراف الگوریتم‌ها در ارزیابی چهارم

شکل (۱۶): مقایسه زمان محاسبات الگوریتم‌ها در ارزیابی چهارم

۵.۴. ارزیابی پنجم

در مرجع [۲۷]، یک الگوریتم ترکیبی با توجه به ایده تقسیم و غلبه برای مسیریابی پیشنهاد شده است. مقایسه نرخ انحراف نشان می‌دهد در مجموعه داده‌های مختلف در آزمایش اول با توجه به شکل (۱۷) به طور متوسط ۹۸/۰۱٪ و در آزمایش دوم مطابق با شکل (۱۸)، با درصد بسیار بالا دقت روش ACOPSOCa به نسبت روش DPC-ACO-KOPT در انواع مقیاس متغیرها بیشتر بوده است.

شکل (۱۷): مقایسه نرخ انحراف الگوریتم‌ها در ارزیابی پنجم

شکل (۱۸): مقایسه نرخ انحراف الگوریتم‌ها در ارزیابی پنجم

۶.۴. ارزیابی ششم

در مرجع [۲۸]، برای بهینه‌سازی مسیر از الگوریتم گله فیل که از رفتار طبیعی فیل‌ها الهام گرفته شده، استفاده می‌شود. مقایسه نرخ انحراف با توجه به شکل (۱۹) نشان می‌دهد روش پیشنهادی ACOPSOCa عملکرد بهتری داشته است و در مجموع به طور متوسط ۱۱/۶۳٪ دقت روش ACOPSOCa به نسبت روش Improved Discrete EHO بیشتر بوده است.

شکل (۱۹): مقایسه نرخ انحراف الگوریتم‌ها در ارزیابی ششم

۷.۴. ارزیابی هفتم

در مرجع [۲۹]، الگوریتم جست‌وجوی موجودات همزیست با شبیه‌سازی تبرید ترکیب شده و روش جدیدی با عنوان SOS-SA معرفی شده است. مقایسه نرخ انحراف با توجه به شکل (۲۰) نشان می‌دهد در مجموعه داده‌های berline52، kroc100، Pr299 و Pr124 روش SOS-SA نسبت به روش ACOPSOCa عملکرد بهتری داشته است ولی در مجموع به طور متوسط ۷/۵۹٪ دقت روش ACOPSOCa نسبت به روش SOS-SA بیشتر بوده است.

شکل (۲۰): مقایسه نرخ انحراف الگوریتم‌ها در ارزیابی هفتم

۸.۴. ارزیابی هشتم

در مرجع [۳۰]، برای بهینه‌سازی مسیر از الگوریتم فرااکتشافی سنجاقک استفاده شده است. مقایسه نرخ انحراف با توجه به نتایج شکل (۲۱) نشان می‌دهد در مجموعه داده‌های مختلف به طور متوسط ۶۳/۳۴٪ دقت روش ACOPSOCa به نسبت روش DA بیشتر بوده است. شکل (۲۲) زمان محاسبات برای به دست آوردن مسیر بهینه را نشان می‌دهد. روش پیشنهادی ACOPSOCa در کاهش زمان محاسبات نسبت به روش DA به طور متوسط ۱۵/۹۵٪ بهبودی داشته است.

شکل (۲۱): مقایسه نرخ انحراف در ارزیابی هشتم

شکل (۲۲): مقایسه زمان محاسبات در ارزیابی هشتم

۹.۴. ارزیابی نهم

در مرجع [۱۴]، به منظور بهینه‌سازی مسیر از الگوریتم ترکیبی کلونی مورچه بر اساس رویکرد حریص تطبیقی (GSACO) استفاده شده است. مقایسه نرخ انحراف (دقت) با توجه به نتایج شکل (۲۳) نشان می‌دهد در مجموعه داده‌های مختلف به طور متوسط ۶۲/۱۴٪ دقت روش پیشنهادی ACOPSOCa به نسبت روش GSACO بیشتر بوده است.

شکل (۲۳): مقایسه نرخ انحراف در ارزیابی نهم

۱۰.۴. ارزیابی دهم

در مرجع [۱۸]، برای بهینه‌سازی مسیر نسخه جدیدی از الگوریتم بهینه‌سازی زنبور عسل با جایگزینی تابع Multi-Insert بجای Swap ارائه شده است. مقایسه نرخ انحراف (دقت) در شکل (۲۴) نشان می‌دهد در مجموعه داده KroA100 عملکرد الگوریتم BA بهتر بوده و در سایر مجموعه داده‌ها عملکرد روش ACOPSOCa برتری دارد. در مجموع به طور متوسط ۶/۰۲٪ دقت روش ACOPSOCa به نسبت روش BA بیشتر بوده است.

شکل (۲۴): مقایسه نرخ انحراف در ارزیابی هفتم

۱۱.۴. ارزیابی یازدهم

به منظور بررسی اثربخشی روش برنامه‌ریزی مسیر پویا بر اساس الگوریتم ACOPSOCa از شبکه‌های ترافیکی انتخاب شده از یک منطقه شهری برای شبیه‌سازی استفاده می‌شود. در بین شهرهای بزرگ چین، شبکه جاده‌ای پکن بالاترین درصد ازدحام را دارد. بنابراین مطابق شکل (۲۵) برخی از تقاطع‌های شهر پکن به عنوان ناحیه‌ای برای آزمایش برنامه‌ریزی مسیر پویا بر اساس الگوریتم بهبود یافته کلونی مورچگان انتخاب می‌شود [۳۲].

شکل (۲۵): شبکه شهری پکن

در این آزمایش، ابتدا سه متغیر الگوریتم ACO، به صورت $\alpha = 1$، $\beta = 2.5$ و $\rho = 0.8$ تنظیم می‌شوند. همچنین نقاط شروع و پایان به طور تصادفی انتخاب می‌شوند. در آزمایش برنامه‌ریزی مسیر پویا بر اساس الگوریتم ACOPSOCa در شرایط ترافیک جاده‌ای ابتدا مسیر اولیه $Tour_{int}$، مسیر ایستا $Tour_{static}$، مسیر پویای $Tour_{Aco}$ و مسیر پویا $Tour_{opt}$ را تعریف می‌کنیم. مسیر اولیه $Tour_{int}$ نشان‌دهنده شرایط جاده بدون ترافیک است که توسط الگوریتم ACOPSO [۶] محاسبه می‌شود. مسیر ایستا $Tour_{static}$ نشان‌دهنده مجموع عوامل شرایط جاده که توسط الگوریتم ACO به دست می‌آید. مسیر پویا $Tour_{Aco}$ نشان‌دهنده مجموع عوامل شرایط جاده است که توسط الگوریتم ACOPSO ذکر شده در مقاله به دست می‌آید. مسیر پویا $Tour_{opt}$ نشان‌دهنده مجموع عوامل شرایط جاده است که با روش برنامه‌ریزی مسیر پویا بر اساس الگوریتم ACOPSOCa به دست می‌آید. مجموع عوامل وضعیت جاده

در مسیر اولیه $Tour_{int}$ ثابت و در مسیر $Tour_{static}$، $Tour_{Aco}$ و $Tour_{opt}$ با توجه به روابط (۱۲) تا (۱۵) به دست می‌آید.

$$Tour_{int} = \frac{\sum_{i=0}^{n} Tour_{int}(i)}{n} \quad (۱۲)$$

$$Tour_{static} = \frac{\sum_{i=0}^{n} Tour_{static}(i)}{n} \quad (۱۳)$$

$$Tour_{Aco} = \frac{\sum_{i=0}^{n} Tour_{Aco}(i)}{n} \quad (۱۴)$$

$$Tour_{opt} = \frac{\sum_{i=0}^{n} Tour_{opt}(i)}{n} \quad (۱۵)$$

شاخص‌های ارزیابی آزمایش‌های برنامه‌ریزی مسیر پویا در شرایط ترافیک جاده‌ها CR_{STATIC}، CR_{ACO} و CR_{OPT} است. CR_{STATIC} میزان تراکم ایجاد شده توسط الگوریتم ACO، CR_{ACO} میزان تراکم ایجاد شده توسط الگوریتم ACOPSO و CR_{OPT} میزان ازدحام ایجاد شده توسط الگوریتم پیشنهادی ACOPSOCa است. نحوه محاسبات شاخص‌ها بر اساس روابط (۱۶) تا (۱۸) نشان داده شده است.

$$CR_{STATIC} = \frac{|Tour_{static} - Tour_{int}|}{Tour_{int}} \quad (۱۶)$$

$$CR_{ACO} = \frac{|Tour_{Aco} - Tour_{int}|}{Tour_{int}} \quad (۱۷)$$

$$CR_{OPT} = \frac{|Tour_{opt} - Tour_{int}|}{Tour_{int}} \quad (۱۸)$$

جدول (۳) نتایج مسیر پویای شهر پکن با روش ACOPSO و جدول (۴) نتایج مسیر پویای این شهر با روش ACOPSOCa را نشان می دهد. از نتایج تجربی جدول (۴) می‌توان به راحتی دریافت که وقتی ضریب تراکم ترافیک از وضعیت ترافیکی شکل (۲۵) تبعیت می‌کند، شاخص ارزیابی CR_{STATIC}، به طور متوسط منجر به نرخ ازدحام ۱۵/۲۴، شاخص ارزیابی CR_{ACO}، منجر به نرخ ازدحام ۱۷/۴۲ و شاخص ارزیابی CR_{OPT} منجر به نرخ ازدحام ۱/۵۷ می‌شود. حال آنکه با توجه به جدول (۳) روش برنامه‌ریزی مسیر پویا بر اساس روش ACOPSO [۱]، به طور متوسط منجر به نرخ ازدحام ۲/۰۸ می‌شود. مقایسه نرخ ازدحام به دست آمده با الگوریتم پیشنهادی ACOPSOCa و روش ACOPSO را در شکل (۲۶) می‌توان مشاهده کرد.

جدول (۳): نتایج مسیر پویای شهر پکن با روش ACOPSO

n	$Tour_{int}$	$Tour_{static}$	$Tour_{Aco}$	$Tour_{opt}$	CR_{STATIC}	CR_{ACO}	CR_{OPT}
100	1699	1950	2024	1727	14.8	19.12	1.64
200	1742	1998	2070	1800	14.7	18.8	3.32
500	1753	2037	2039	1791	16.2	16.31	2.16
1000	1758	2042	2080	1784	16.2	18.3	1.47
1406	1723	2003	2041	1754	16.3	18.5	1.79
Avg	1735	2006	2050	1771	15.61	18.2	2.08

جدول (۴): نتایج مسیر پویای شهر پکن با روش ACOPSOCA

n	$Tour_{int}$	$Tour_{static}$	$Tour_{Aco}$	$Tour_{opt}$	CR_{STATIC}	CR_{ACO}	CR_{OPT}
100	1704	1921	2001	1720	12.7	17.4	0.94
200	1730	1973	2043	1780	14.0	18.1	2.89
500	1743	2022	2026	1773	16.0	16.2	1.72
1000	1749	2040	2071	1769	16.6	18.4	1.14
1406	1753	2046	2050	1773	16.7	16.9	1.14
Avg	1735.	2000.	2038.	1763	15.24	17.42	1.57

شکل (۲۶): مقایسه نرخ ازدحام ACOPSO و ACOPSOCA

۱۲.۴. ارزیابی دوازدهم

در این قسمت به منظور بررسی تاثیر اعمال کسینوس زوایا در الگوریتم ACOPSO، بر روی شبکه شهری شکل (۲۷) آزمایشی به صورت مجزا بر روی یال‌های مشخص با متغیرهای $\alpha = 1$، $\beta = 3$، $\rho = 0.1$ و تعداد تکرار ۵۰ به ازای الگوریتم ACO، ACOPSO و ACOPSOCa انجام داده‌ایم. نتایج حاصل از این آزمایش در جدول (۵) نمایش داده شده است. با بهره‌گیری از مقایسه هزینه‌ای (هزینه زمان) با توجه به شکل (۲۷) می‌توان مشاهده کرد که الگوریتم ترکیبی ACOPSOCa به طور میانگین بر روی مسیرهای مختلف ۲/۶۲ ثانیه زمان مصرف می‌کند در

حالی که الگوریتم ACOPSO، ۳/۱۸ و الگوریتم ACO، ۵/۹۹ ثانیه زمان می‌برد تا مسیرهای مختلف را طی کند. از بررسی نتایج زمان طی شده بر روی مسیرهای مختلف می‌توان به این نتیجه رسید که با وجود ترافیک روی مسیرها، دقت الگوریتم ACOPSOCa نسبت به سایر روش‌ها بهتر است.

جدول (۵): هزینه‌های سفر بر روی مسیرهای متفاوت شهر پکن

ACOPSOCa	ACOPSO	ACO	Dest	Start
2.21	3.01	6.11	28	7
2.4	3.23	6.81	37	15
3.21	4.13	5.21	34	3
3.15	3.86	8.02	0	36
2.89	3.26	7.26	6	30
2.46	2.85	5.46	14	24
2.15	2.36	4.32	6	16
2.51	2.76	4.7	32	12
2.62	3.18	5.99	AVE	

شکل (۲۷): مقایسه زمان محاسبات مسیرهای متفاوت شهر پکن

۱۳.۴. آزمون‌های آماری

در آزمون‌های آماری هدف بررسی یک حدس یا فرضیه بر اساس داده‌های نمونه است. برای مقایسه نرخ انحراف و میانگین تعداد تکرارها با توجه به جواب‌های حاصله در بخش‌های قبلی آزمون آماری مطابق رابطه (۱۹) انجام می‌شود [۳۳]. که در این رابطه منظور از dev_{base} نرخ انحراف الگوریتم‌های پایه است که در بخش‌های قبلی با روش اجرایی مقاله بررسی شده است. همچنین منظور از $dev_{ACOPSOCA}$ میزان نرخ انحراف روش اجرایی مقاله است.

$$\begin{cases} h_0: dev_{base} = dev_{ACOPSOCA} \\ h_1: dev_{base} \neq dev_{ACOPSOCA} \end{cases} \quad (۱۹)$$

به طور معادل می‌توان مطابق رابطه (۲۰) عمل کرد:

$$\begin{cases} h_0: dev_{base} - dev_{ACOPSOCA} = 0 \\ h_1: dev_{base} - dev_{ACOPSOCA} \neq 0 \end{cases} \quad (۲۰)$$

نتایج جدول (۶) از آزمون T زوجی حاصل شده است. در این جدول بر روی تمامی ارزیابی‌های صورت گرفته در بخش‌های قبلی آزمون‌های آماری انجام گرفته است. ستون‌های LOWER و UPPER را می‌توان نوعی برآورد فاصله‌ای در نظر گرفت که آن را فاصله اطمینان گویند. ستون t آماره آزمون و df درجه آزادی (Degree of Freedom) است که بیانگر تعداد مقادیری است که در یک محاسبه مرتبط با شاخص می‌توانند آزادانه تغییر کنند. ستون آخر نیز، مقدار Sig یا مقدار احتمال (p-Value) برای آزمون است. در صورتی که مقدار Sig کمتر از احتمال خطا (در سطح اطمینان ۹۵٪ مقدار $\alpha = 0.05$ و در سطح اطمینان ۹۰٪ مقدار $\alpha = 0.1$ است) باشد، فرض صفر را رد خواهیم کرد و میتوان گفت روش پیشنهادی اختلاف معناداری ایجاد کرده است. به عنوان نمونه در سطر اول جدول (۶) که بر روی آزمایش اول از ارزیابی اول مقایسه انجام شده است، فرض صفر رد شده و روش پیشنهادی توانسته است اختلاف معناداری را ایجاد نماید.

مطابق با نتایج جدول (۶)، روش ACOPSOCa از نظر آماری تفاوت معناداری در مقایسه با سایر روش‌ها ایجاده کرده و به طور متوسط نرخ انحراف و میانگین تعداد تکرارها را کاهش داده است. بنابراین، ترکیب الگوریتم‌های یادگیری ماشین [۳۴]-[۳۷] می‌توانند به خوبی برای برنامه‌ریزی مسیر استفاده گردند.

۵. نتایج

رشد بی‌رویه تولید خودرو از یک سو و عدم گسترش صحیح زیرساخت‌های جاده‌ای از سوی دیگر باعث ایجاد تراکم شدید، تاخیرهای طولانی و سوانح زیاد رانندگی به خصوص در شهرهای شلوغ شده است. برای حل مشکلات فوق پژوهش‌های زیادی انجام شده که عمده این روش‌ها بر رویکردهای

فراابتکاری استوار هستند. در بسیاری از این الگوریتم‌ها در ابتدای کار مسیریابی به صورت تصادفی و در محدوده‌ای بزرگ انجام می‌شود. این موضوع سبب افزایش زمان جستجو می‌گردد. همچنین در تحلیل‌های صورت گرفته بر مبنای این الگوریتم‌ها تنها فاصله فیزیکی بین خودروها در نظر گرفته شده است. با توجه به اینکه عوامل محیطی نظیر ترافیک، در مسیریابی بسیار موثر هستند، باید تاثیر این عوامل را در مسیریابی در نظر گرفت. بدین منظور برای حل مشکلات مطرح شده یک روش برنامه‌ریزی مسیر پویا مبتنی بر ترکیب الگوریتم‌های کلونی مورچگان و ازدحام ذرات با اعمال تابع ابتکاری کسینوس زوایا در این مقاله پیشنهاد شد.

جدول (۶): هزینه‌های سفر بر روی مسیرهای متفاوت شهر پکن

ارزیابی	روش	Lower	Upper	t	df	Sig. (2-tailed)
اول	آزمایش اول نرخ انحراف	8.93	17.48	8.57	4	0.001
اول	آزمایش اول تعداد تکرارها	11.37	31.43	5.92	4	0.004
اول	آزمایش دوم نرخ انحراف	5.10	11.85	6.97	4	0.002
اول	آزمایش دوم تعداد تکرارها	9.73	19.87	8.10	4	0.001
اول	آزمایش سوم نرخ انحراف	0.05	2.00	2.90	4	0.044
اول	آزمایش سوم تعداد تکرارها	4.74	16.06	5.10	4	0.007
دوم	مقایسه نرخ انحراف	0.41	6.90	2.89	5	0.034
سوم	مقایسه نرخ انحراف	-0.14	1.03	2.13	4	0.100
چهارم	مقایسه نرخ انحراف	0.10	0.47	3.39	10	0.007
چهارم	مقایسه تعداد تکرارها	1.94	10.09	3.29	10	0.008
پنجم	مقایسه نرخ انحراف	1.85	5.62	6.30	3	0.008
پنجم	مقایسه نرخ انحراف	-1.13	11.42	1.89	8	0.095
ششم	مقایسه نرخ انحراف	0.01	0.23	2.45	10	0.034
هفتم	مقایسه نرخ انحراف	0.00	0.18	2.24	10	0.049
هشتم	مقایسه نرخ انحراف	-0.99	8.08	2.17	4	0.096
هشتم	مقایسه تعداد تکرارها	2.00	6.98	5.00	4	0.007
نهم	مقایسه نرخ انحراف	0.06	1.69	2.99	4	0.040
دهم	مقایسه نرخ انحراف	0.04	0.66	2.66	7	0.032
یازدهم	مقایسه نرخ انحراف	0.31	0.71	7.23	4	0.002
دوازدهم	مقایسه نرخ انحراف	0.32	0.80	5.57	7	0.001

روش پیشنهادی عوامل مختلف وضعیت جاده‌ها از جمله طول جاده شهری و ترافیک ورودی و خروجی تقاطع‌ها را در مسیریابی وسایل نقلیه در نظر می‌گیرد و در مسیریابی برای نقاطی که هم راستای مسیر پیمایش به سمت مقصد نهایی باشد شانس بیشتری قائل می‌شود. برای انجام بهینه‌سازی مسیریابی از الگوریتم کلونی مورچگان استفاده شد. با توجه به ضعف این الگوریتم در همگرایی و جستجوی تصادفی در مراحل اولیه و همچنین با توجه به زمان‌بر بودن تنظیم محدوده متغیرهای این الگوریتم، الگوریتم کلونی مورچگان با الگوریتم ازدحام ذرات ترکیب و برای بهبود دقت و کاهش زمان مسیریابی، تابع ابتکاری کسینوس زوایا به آن افزوده شده است. نتایج آزمایش‌های صورت گرفته بر اساس کتابخانه TSPLIB و در آزمایش برنامه‌ریزی مسیر پویا بر اساس داده‌های ترافیکی واقعی شهر پکن نشان می‌دهد روش پیشنهادی مقاله دقت محاسباتی و همچنین زمان اجرا محاسبات را نسبت به روش‌های پیشین بهبود بخشیده است.

در کارهای آتی می‌توان برای اجرای واقعی روش پیشنهادی مقاله در محیط حقیقی که نیازمند سخت‌افزار و نرم‌افزار لازم است، اقدام کرد و می‌توان علاوه بر تاثیر وضعیت جاده‌ها، تاثیر وضعیت آب و هوایی را به عنوان متغیرهای تاثیرگذار در الگوریتم پیشنهادی مقاله تاثیر داد. همچنین با توجه به عملکرد به نسبت خوب الگوریتم‌های SOS-SA و Discrete-EHO می‌توان از این دو در ترکیب با الگوریتم کلونی مورچگان یا سایر الگوریتم‌ها برای افزایش دقت و کاهش زمان مسیریابی در برنامه‌ریزی مسیر پویا استفاده کرد.

تعارض منافع: نویسندگان اعلام می‌کنند که هیچ تعارض منافعی ندارند.

مراجع

[1] J. Yang, A.-O. Purevjav, and S. Li, "The marginal cost of traffic congestion and road pricing: evidence from a natural experiment in Beijing," American Econ. J. Econ. Policy, vol. 12, no. 1, pp. 418-453, 2020, doi: 10.1257/pol.20170195.

[2] M. Dorigo, M. Birattari, and T. Stutzle, "Ant colony optimization," IEEE Comput. Intel. Mag., vol. 1, no. 4, pp. 28-39, 2006, doi: 10.1109/ci-m.2006.248054.

[3] I. Karaoglan, F. Altiparmak, I. Kara, and B. Dengiz, "The location-routing problem with simultaneous pickup and delivery: Formulations and a heuristic approach," Omega, vol. 40, no. 4, pp. 465-477, 2012, doi: 10.1016/j.omega.2011.09.002.

[4] S. Doostali and M. Khalily-Dermany, "A multi-hop PSO based localization algorithm for wireless sensor networks," Soft Comput. J., vol. 8, no. 1, pp. 58-69, 2019, doi: 10.22052/8.1.58 [In Persian].

[5] R. Ghasemi, H.R. Mohammadi, and S.A. Taher, "Frequency Control of an Islanded Microgrid based on Intelligent Control of Demand Response using Fuzzy Logic and Particle Swarm Optimization (PSO) Algorithm," Soft Comput. J., vol. 6, no. 2, pp. 18-31, 2018, dor: 20.1001.1.23223707.1396.6.2.2.6 [In Persian].

[6] C. Wu, S. Zhou, and L. Xiao, "Dynamic path planning based on improved ant colony algorithm in traffic congestion," IEEE Access, vol. 8, pp. 180773-180783, 2020, doi: 10.1109/access.2020.3028467.

[7] H.R. Mohammadi and A. Akhavan, "Performance Comparison of HSA, ICA and PSO Algorithms for Selective Harmonic Elimination in Cascaded Multilevel Inverter with Variable DC Sources," Soft Comput. J., vol. 3, no. 2, pp. 20-30, 2015, dor: 20.1001.1.23223707.1393.3.2.55.8 [In Persian].

[8] V. Chvatal, W. Cook, G.B. Dantzig, D.R. Fulkerson, and S.M. Johnson, "Solution of a Large-Scale Traveling-Salesman Problem," in M. Junger, et al. 50 Years of Integer Programming 1958-2008, Springer, Berlin, Heidelberg, 2010, doi: 10.1007/978-3-540-68279-0_1.

[9] A. Colorni, M. Dorigo, and V. Maniezzo, "Distributed optimization by ant colonies," in Proc. European Conf. Artif. Life (ECAL91), Paris, France, 1991, pp. 134-142.

[10] G. Shang, J. Xin-zi, T. Kezong, and Y. Jingyu, "Hybrid Algorithm Combining Ant Colony Optimization Algorithm with Particle Swarm Optimization," in Chinese Control Conf., Harbin, China, 2006, pp. 1428-1432, doi: 10.1109/CHICC.2006.280708.

[11] C.-x. Shi, B. Ying-yong, L. Zi-guang, and T. Jun, "Solving path planning problem by an aco-pso hybrid algorithm," in Int. Conf. Intell. Syst. Knowl. Eng., Atlantis Press, 2007, pp. 1-4, doi: 10.2991/iske.2007.91.

[12] D. Pal, P. Verma, D. Gautam, and P. Indait, "Improved optimization technique using hybrid ACO-PSO," in 2nd Int. Conf. Next Gener. Comput. Technol. (NGCT), Dehradun, India, 2016, pp. 277-282, doi: 10.1109/NGCT.2016.7877428.

[13] A.J. Ouyang and Y.Q. Zhou, "An improved PSO-ACO algorithm for solving large-scale TSP," Adv. Mater. Res., vol. 143-144, pp. 1154-1158, 2010, doi: 10.4028/www.scientific.net/amr.143-144.1154.

[14] W. Li, L. Xia, Y. Huang, and S. Mahmoodi, "An ant colony optimization algorithm with adaptive greedy strategy to optimize path problems," J. Ambient Intell. Humaniz. Comput., vol. 13, no. 3, pp. 1557-1571, 2022, doi: 10.1007/s12652-021-03120-0.

[15] P. Stodola, P. Otrisal, and K. Hasilova, "Adaptive ant colony optimization with node clustering applied to the travelling salesman problem," Swarm Evol. Comput., vol. 70, p. 101056, 2022, doi: 10.1016/j.swevo.2022.101056.

[16] R. Zheng, Y. Zhang, and K. Yang, "A transfer learning-based particle swarm optimization algorithm for travelling salesman problem," J. Comput. Des. Eng., vol. 9, no. 3, pp. 933-948, 2022, doi: 10.1093/jcde/qwac039.

[17] S.A. Al-Agamy, F.M. Ba-Alwi, and A.M. Mohsen, "A Fuzzy Logic for Parameter Adaptation in Ant Colony Optimization Approach," Int. J. Innov. Sci. Res. Technol. vol. 7, no. 5, pp. 1549-1560, 2022, doi: 10.5281/zenodo.6774879.

[18] M. Sahin, "Solving TSP by using combinatorial Bees algorithm with nearest neighbor method," Neural Comput. Appl., vol. 35, no. 2, pp. 1863-1879, 2023, doi: 10.1007/s00521-022-07816-y.

[19] Dataset Description (2022, Dec. 02), [Online]. Available: https://github.com/article2023-uni/code-dataset-Description.

[20] W. Zhangqi, Z. Xiaoguang, and H. Qingyao, "Mobile robot path planning based on parameter optimization ant colony algorithm," Procedia Eng., vol. 15, pp. 2738-2741, 2011, doi: 10.1016/j.proeng.2011.08.515.

[21] A.K.M.F. Ahmed and J.U. Sun, "An improved particle swarm optimization algorithm for the travelling salesman problem," Adv. Sci. Lett., vol. 22, no. 11, pp.3318-3322, 2016, doi: 10.1166/asl.2016.7864.

[22] X. Xie and P. Wu, "Research on the optimal combination of ACO parameters based on PSO," in Int. Conf. Netw. Digit. Soc., Wenzhou, China, 2010, pp. 94-97, doi: 10.1109/ICNDS.2010.5479311.

[23] TSPLIB (2022, Dec. 02), [Online]. Available: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/.

[24] W. Deng, J. Xu, and H. Zhao, "An improved ant colony

optimization algorithm based on hybrid strategies for scheduling problem," IEEE Access, vol. 7, pp. 20281-20292, 2019, doi: 10.1109/access.2019.2897580.

[25] E. Elsayed, A.H. Omar, and K. Elsayed, "Smart solution for STSP semantic traveling salesman problem via hybrid ant colony system with genetic algorithm," Int. J. Intell. Eng. Syst., vol. 13, no. 5, pp. 476-489, 2020, doi: 10.22266/ijies2020.1031.42.

[26] A.E. Ezugwu and A.O. Adewumi, "Discrete symbiotic organisms search algorithm for travelling salesman problem," Expert Syst. Appl., vol. 87, pp. 70-78, 2017, doi: 10.1016/j.eswa.2017.06.007.

[27] E. Liao and C. Liu, "A hierarchical algorithm based on density peaks clustering and ant colony optimization for traveling salesman problem," IEEE Access, vol. 6, pp. 38921-38933, 2018, doi: 10.1109/access.2018.2853129.

[28] A. Hossam, A. Bouzidi, and M.E. Riffi, "Elephants Herding Optimization for Solving the Travelling Salesman Problem," in M. Ezziyyani, (eds) Advanced Intelligent Systems for Sustainable Development (AI2SD), Advances in Intelligent Systems and Computing, vol. 912. Springer, Cham, 2019, doi: 10.1007/978-3-030-12065-8_12.

[29] A.E. Ezugwu, A.O. Adewumi, and M.E. Frincu, "Simulated annealing based symbiotic organisms search optimization algorithm for traveling salesman problem," Expert Syst. Appl., vol. 77, pp. 189-210, 2017, doi: 10.1016/j.eswa.2017.01.053.

[30] A.I. Hammouri, E.T.A. Samra, M.A. Al-Betar, R.M. Khalil, Z. Alasmer, and M. Kanan, "A Dragonfly Algorithm for Solving Traveling Salesman Problem," in 8th IEEE Int. Conf. Control Syst. Comput. Eng. (ICCSCE), Penang, Malaysia, 2018, pp. 136-141, doi: 10.1109/ICCSCE.2018.8684963.

[31] Z. Zhang, K. Zou, and J. Zhang, "Parameter Analysis for a Novel Ant Colony Optimization Algorithm," Eng. Technol. Res., 2016, doi: 10.12783/dtetr/icca2016/6052.

[32] Congest (2022, Dec. 02), [Online]. Available: https://report.amap.com/congest.do.

[33] R.S. Cochran, Book Reviews: J. Neter, W. Wasserman, and G.A. Whitmore, Applied Statistics, Boston: Allyn and Bacon, Inc., 1978. pp. xix + 743. Educ. Psychol. Meas., vol. 40, no. 1, pp. 267-270, 1980, doi: 10.1177/001316448004000148.

[34] M. Amiri, L. Mohammad-Khanli, and R. Mirandola, "A new efficient approach for extracting the closed episodes for workload prediction in cloud," Computing, vol. 102, no. 1, pp. 141-200, 2020, doi: 10.1007/s00607-019-00734-3.

[35] M. Amiri and H. Askari, "Illegal Miner Detection based on Pattern Mining: A Practical Approach," J. Comput. Secur., vol. 9, no. 2, pp. 1-10, 2022, doi: 10.22108/jcs.2022.133306.1096.

[36] F. Farnaghi-Zadeh, M. Rahmani, and M. Amiri, "Feature Selection Using Neighborhood based Entropy," J. Univers. Comput. Sci., vol. 28, no. 11, pp. 1169-1192, 2022, doi: 10.3897/jucs.79905.

[37] F. Jaryani, and M. Amiri, "A Pre-Trained Ensemble Model for Breast Cancer Grade Detection Based on Small Datasets," Iranian J. Health Sci., vol. 11, no. 1, pp. 47-58, 2023, doi: 10.32598/ijhs.11.1.883.1.