اولین کنفرانس مهندسی فناوری اطلاعات مکانی
دانشگاه صنعتی خواجه نصیرالدین طوسی
دانشکده مهندسی نقشه‌برداری
۲۹ و ۳۰ دی ماه ۱۳۹۴

The 1st National Conference on
Geospatial Information Technology
K.N.Toosi University of Technology
Faculty of Geomatics Engineering
19 - 20 January 2016

کان‌یابی کیوسک‌های پلیس راهنمایی و رانندگی در شبکه راه‌های شهری با استفاده از بهینه‌سازی ذرات ازدحامی

سینا ابوالحسینی [1*]، رضا محمدی سلیمانی [1]

۱- گروه سیستم اطلاعات مکانی، دانشکده مهندسی نقشه‌برداری (ژئوماتیک)، دانشگاه صنعتی خواجه نصیرالدین طوسی

چکیده :

مسئله ترافیک در شهرهای بزرگ کشور به یکی از مهم‌ترین مسائل حوزه حمل‌ونقل شهری تبدیل شده است. زیرا بر بسیاری دیگر از حوزه‌های مهم تأثیر مستقیم و غیرمستقیم دارد. به طور مثال در حوزه محیط زیست و آلودگی آب و هوا، بهداشت جسمی افراد جامعه و همینطور بهداشت روانی آن‌ها تاثیرات ترافیک بر هیچ کسی پوشیده نیست. یکی از راه‌حل‌های موجود برای حل این مشکل، استفاده از پلیس‌های راهنمایی و رانندگی در محل وقوع ترافیک به‌منظور هدایت مؤثر خودروها است. هدف این مقاله مکان‌یابی بهینه کیوسک‌های پلیس راهنمایی و رانندگی در شبکه‌های حمل‌ونقل شهری با استفاده از الگوریتم بهینه‌سازی ذرات ازدحامی است. در این مسئله سعی شده است که امکان دسترسی پلیس‌ها به نقاط ترافیکی تسهیل شود تا بتوان با بهره‌گیری از آن‌ها مشکل ترافیک را بهبود داد. بدین منظور برای به کارگیری الگوریتم ذرات ازدحامی در این مسئله، تغییراتی در آن صورت گرفت تا بتوان در این مسئله از این الگوریتم استفاده کرد. در پایان مقاله نتایج حاصل، مورد بررسی قرارگرفته‌است.

واژه‌های کلیدی : مکان‌یابی، کیوسک پلیس، ترافیک، الگوریتم بهینه‌سازی ذرات ازدحامی.

نویسنده مکاتبه کننده : سینا ابوالحسینی
آدرس پستی : تهران، خیابان ولیعصر، تقاطع میرداماد، دانشکده مهندسی ژئوماتیک، دانشگاه خواجه نصیرالدین طوسی
تلفن :
آدرس پست الکترونیک : abolhoseini.sina@gmail.com

۱- مقدمه

امروزه با افزایش جمعیت و گسترش شهرها، ترافیک تبدیل به یکی از معضلات در کلان‌شهرها شده است. مسئله ترافیک یکی از مسائلی است که گریبان گیر بسیاری از شهرهای بزرگ دنیا شده است. ترافیک می‌تواند نقش بسیار مخربی را ازلحاظ اقتصادی، سلامتی و محیط زیستی در این کلان‌شهرها بازی کند.

یکی از کارهایی که می‌توان برای برخورد با این معضل انجام داد، استفاده از نیروهای پلیس راهنمایی و رانندگی است. یکی از شروطی که برای استفاده از این نیروها باید در نظر گرفت، استفاده بهینه از این افراد است. استفاده بهینه از نیروهای پلیس راهنمایی و رانندگی را می‌توان با استفاده بجا و به‌موقع از آن‌ها در نقاط ترافیکی لحاظ کرد. یکی از ساده‌ترین کارهایی که می‌توان انجام داد، ساخت کیوسک‌های پلیس به‌صورتی بهینه است، زیرا با این کار پلیس راهنمایی و رانندگی بیشتر در موقعیت‌های بهتری قرار می‌گیرند. از طرف دیگر ساخت این کیوسک‌ها هزینه‌بردار است و نیازمند اختصاص دادن فضای شهری است. پس نمی‌توان تعداد زیادی از آن‌ها در همه نقاط ساخت.

بدین منظور برای مسئله‌ای که پیش روی است، فرض‌هایی انجام‌شده است. یکی از این فرض‌ها، وقوع ترافیک تنها در تقاطع‌هاست. فرض دوم این است که حرکت نیروهای پلیس راهنمایی و رانندگی مستقل از وضعیت ترافیکی و جهت خیابان است. پلیس راهنمایی و رانندگی می‌تواند پیاده و یا سوار بر وسائل نقلیه دوچرخ، خود را به تقاطعی که ترافیک رخ داده‌است، برساند. سپس برای حل آن‌ها کافی ست که مدل‌سازی برای شبکه راه‌های شهری انجام شود تا بتوان با استفاده از الگوریتم‌های موجود کار بهینه‌سازی را انجام داد. بهینه‌سازی در اینجا به این معناست که از نیروهای پلیس موجود و در دسترس به‌گونه‌ای استفاده کنیم که این نیروها بیشترین نزدیکی را با تقاطع‌ها داشته باشند و همچنین در مدت‌زمان کوتاهی بتوانند خود را به تقاطع‌ها که محل وقوع گره‌های ترافیکی هستند، برسانند.

مدل‌سازی رایجی که برای شبکه راه‌ها انجام می‌شود، مدل کردن شبکه راه به‌وسیله گراف است. تقاطع‌ها با گره و خیابان‌ها با لینک مدل‌سازی می‌شوند. مسافت، زمان سفر و یا هزینه حرکت از یک گره به گره بعد نیز به‌عنوان وزن هر لینک معرفی می‌شود. پس‌ازاین می‌توان از الگوریتم‌های فرا ابتکاری برای بهینه‌سازی مکان کیوسک‌های پلیس استفاده کرد.

الگوریتم‌های انتخابی برای این مسئله الگوریتم ذرات ازدحامی[1] هستند. الگوریتم PSO به راحتی برای مسائل مکان‌یابی در فضاهای پیوسته استفاده می‌شود و نتایج بسیار خوبی را ارائه می‌دهد. اما برای استفاده از الگوریتم PSO در این مسئله باید با ابتکارهایی، الگوریتم را برای به کارگرفتن در فضای گراف آماده ساخت. همچنین خصوصیت مهمی که در الگوریتم PSO وجود دارد، حرکت تمامی ذرات به‌صورت هم‌زمان است. این خصوصیت سبب می‌شود که فضای جستجو با سرعت بیشتری مورد ارزیابی قرار بگیرد.

در ادامه روند مقاله، ابتدا نحوه مدل‌سازی مسئله در فضای گراف، به همراه تابع هدف شرح داده شده‌است. سپس الگوریتم PSO مورد بررسی قرار گرفته‌است. در آخر، نتایج الگوریتم PSO در این مسئله آورده شده‌است.

۲- مدل‌سازی مسئله

برای اینکه بتوانیم این مسئله را در فضای گراف مدل کنیم، نیاز است که از یکسری روش‌های ابتکاری استفاده بشود. در مقاله [1] برای حل مسئله مکان‌یابی از مدلی استفاده کرده است که مسئله را بسیار ساده‌تر کرده است. در این مقاله نیز از مدل ارائه‌شده در مقاله مذکور استفاده می‌کنیم که در ادامه مدل موردبحث ارائه می‌شود.

[1] Particle Swarm Optimization (PSO)

در این مدل، Y مجموعه کیوسک‌های پلیس راهنمایی و رانندگی است (معادله ۱). هر زیرمجموعه‌ای از Y (موقعیت کیوسک‌ها)، یک سه‌تایی است که به فرم معادله ۲ نمایش داده می‌شود. در این سه‌تایی، عدد اول و دوم نماینده لینکی هستند که کیوسک بر روی آن قرارگرفته است. عدد اول گره آغازی و عدد دوم گره پایانی لینک است. عدد سوم هم فاصله کیوسک از گره آغازی است.

معادله ۱ $$Y = \{y_1, y_2, \dots, y_m\}$$

معادله ۲ $$y_j = \{a, b, c\}$$

مجموعه تقاطع‌ها نیز با X نمایش داده می‌شود (معادله ۳).

معادله ۳ $$X = \{x_i, x_2, \dots, x_n\}$$

برای محاسبه تابع بهینگی نیز به این صورت عمل شده است که ابتدا کوتاه‌ترین فاصله هر کیوسک تا تمامی تقاطع‌ها محاسبه می‌شود (معادله ۴). *Dpp(u,w)* در حقیقت کوتاه‌ترین فاصله نقطه u و w است.

معادله ۴ $$Dpp(x_i, y_j) = \min \{Dpp(a, x_i) + c, Dpp(b, x_i) + Dpp(a, b) - c\}$$

در معادله ۴ کوتاه‌ترین فاصله بین تقاطع x_i و y_j محاسبه‌شده است. لازم به ذکر است که در این معادله *Dpp(a,b)* همان فاصله بین دو تقاطع a و b که وزن لینک است خواهد بود. بعدازاینکه برای هر کیوسک کوتاه‌ترین فاصله آن تا هر تقاطع به دست آمد، کوچک‌ترین آن‌ها انتخاب می‌شود. (معادله ۵).

معادله ۵ $$Dps(x_i, Y) = \min Dpp(x_i, y_j)$$

در حقیقت در معادله ۵ نزدیک‌ترین تقاطع به هر کیوسک را پیدا می‌کنیم. سپس در مرحله بعد، بیشترین فاصله موجود در بین فواصل میان کیوسک‌ها و تقاطع‌ها را پیدا می‌کنیم (معادله ۶).

معادله ۶ $$D(X, Y) = \max Dps(x_i, Y)$$

حال تابع هدف را روی همین معادله ۶ تعریف می‌کنیم. همان‌طور که در معادله ۷ مشاهده می‌شود، هرچه مقدار D در یک ذره کمتر باشد، مقدار بهینگی آن بیشتر خواهد بود.

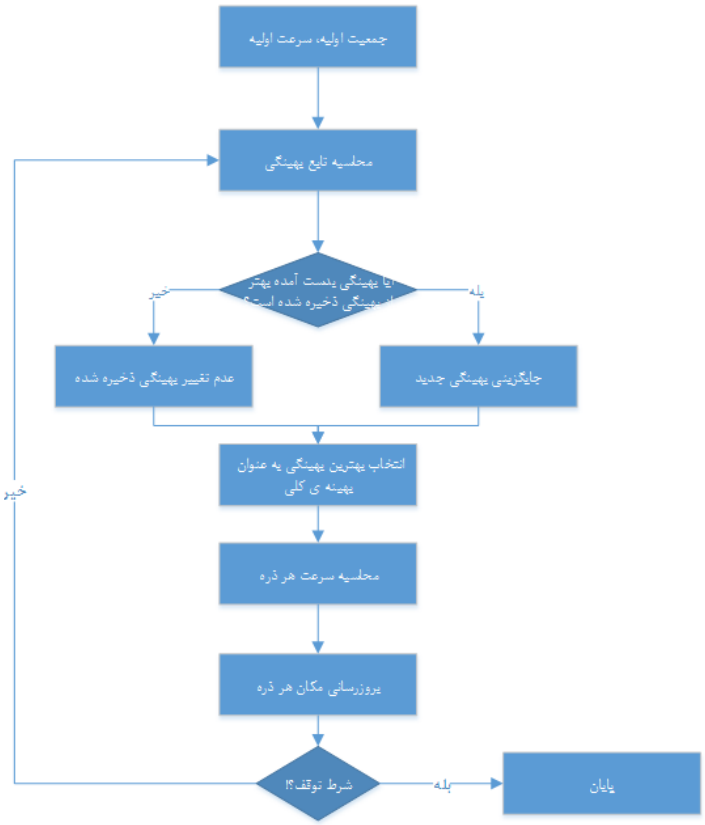
معادله ۷ $$f(Y) = \min D(X, Y)$$

این تابع بهینگی در الگوریتم PSO بهینه‌سازی می‌شود.

۳- الگوریتم بهینه‌سازی ذرات ازدحامی PSO

الگوریتم PSO ابتدا در مقاله [2] ارائه شد که از حرکت پرندگان الهام گرفته است. در این روش بهینه‌سازی، هر ذره به‌عنوان یک پرنده در نظر گرفته می‌شود، و این پرندگان در فضای جواب پرواز می‌کنند. با انتقال اطلاعات و داده‌ای که بین آن‌ها صورت می‌گیرد، کل ذرات به سمت بهترین جواب حرکت می‌کند. حرکتی که ذرات انجام می‌دهند متأثر از چند عامل است و همین باعث تغییر اندازه سرعت و جهت آن برای هر ذره می‌شود. هر ذره در مسیری که طی کرده است، از موقعیتی عبور کرده است که بهینگی آن تا آن لحظه از سایر مکان‌ها بهتر بوده است. این مکان ذخیره می‌شود. همچنین در میان ذرات، ذره‌ای وجود دارد که بهترین جواب را در میان تمام ذرات تاکنون پیداکرده است. سرعت اولیه هر ذره نیز در معادله سرعت تأثیر دارد. حتی در برخی مقالات، یک همسایگی برای هر ذره تعریف می‌شود و هر ذره علاوه بر این متغیرها، قابلیت حرکت به سمت بهترین جواب محلی خود را نیز دارد.

در این مسئله، هر ذره متشکل از مجموعه کیوسک‌ها (Y) است. روند کلی الگوریتم PSO در شکل ۱ ذکرشده است.



شکل ۱: فلوچارت الگوریتم PSO

به دلیل اینکه ذات الگوریتم PSO برای حل مسائل پیوسته است، و مسئله‌ای که پیش رو است، به‌نوعی ترکیبی از فضای گسسته و پیوسته است، باید از روشی برای بهبود این الگوریتم برای رسیدن به جواب استفاده شود. مسئله‌ای که در اینجا با آن برخورد پیش می‌آید، نحوه حرکت ذرات است. سرعت در الگوریتم PSO به فرم معادله ۸ تعریف می‌شود [3].

$$v(t+1) = v(t) \times w + \varphi_1(pbest(t) - x(t)) + \varphi_2(gbest(t) - x(t)) \quad \text{معادله ۸}$$

قسمت اول معادله، ذره را به حرکت در جهت قبلی ترغیب می‌کند. قسمت دوم حرکت را به سمت بهترین موقعیت ذخیره‌شده خود ذره (pbest[۲]) سوق می‌دهد و آخرین قسمت، حرکت به سمت بهترین موقعیت میان تمام ذرات (gbest[۳]) است. این سه حرکت فرصت جستجو در فضا و اکتشاف را به هر ذره می‌دهد. لازم به ذکر است که ذرات از طریق همین پارامتر سرعت و جابجایی با یکدیگر ارتباط پیدا می‌کنند. φ_1 و φ_2 نیز اعدادی تصادفی هستند که میزان حرکت به سمت pbest و gbest را تعیین می‌کنند.

در این مسئله به دلیل گسستگی که در فضای مسئله وجود داشت، این معادله سرعت به‌راحتی قابل پیاده‌سازی نبود و باید با روشی ابتکاری این کار انجام می‌شد. بدین منظور هرکدام از سه بخش ذکر شده در معادله ۸ به نحوی ابتکاری پیاده سازی شده‌اند. بخش اول بدون تغییر پیاده‌سازی شده‌است، بدین صورت که سرعت قبلی ذره، با سرعت ناشی از حرکت به سمت pbest و gbest جمع می‌شود.

هر ذره شامل مجموعه‌ای از لینک‌ها است که کیوسک‌ها بر روی آن‌ها قرار گرفته‌اند. برای هر ذره بهترین موقعیت آن در لیستی جداگانه ذخیره شده‌است. در حرکت به سمت pbest، حرکت با اعمال تغییرات بر روی c (معادله ۲) صورت

[2] Particle Best position
[3] Global Best position

می‌گیرد. برای حرکت کیوسک به سمت بهترین موقعیت موجود، در صورت مشترک بودن لینکی که کیوسک بر روی آن قرار گرفته‌است، سرعت به‌صورت معادله ۹ تعریف می‌شود.

$$v(t+1) = \left(c_{pbest(t)} - c_{y_j}\right)$$ معادله ۹

در صورتی که لینک مشترک بین موقعیت کنونی ذره و بهترین موقعیت آن وجود نداشته باشد، یکی از لینک‌های موقعیت کنونیِ ذره با بهترین موقعیت ذره جایگزین می‌شود. همچنین برای فرار از بهینه محلی، جهشی نیز در این قسمت انجام می‌پذیرد. به این صورت که لینکی به‌صورت تصادفی جایگزین یکی از لینک‌های موقعیت کنونی ذره می‌شود.

در حرکت به سمت gbest، اعمال تغییرات بر روی خود لینک‌ها صورت می‌گیرد، به این صورت که بطور تصادفی، یکی از لینک‌های موقعیت کنونی ذره، با یک لینک از gbest جایگزین می‌شود. موقعیت کیوسک در لینک تعویض شده به‌صورت تصادفی تعریف می‌شود تا با حرکت ذره در تکرار بعدی به سمت بهترین ذره، فضا نیز جستجو شود. در ابتدای تکرارها نخبه‌گرای در حرکت به سمت gbest کم انتخاب شده‌است تا فضا به خوبی جستجو شود. با افزایش تکرارها، نخبه‌گرایی بیشتر می‌شود تا الگوریتم بهتر همگرا شود.

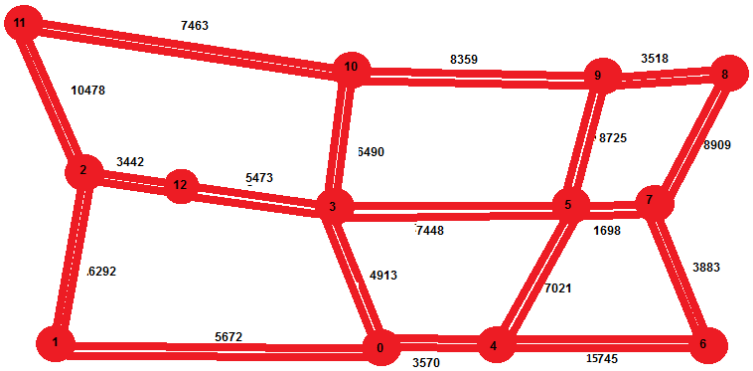
همچنین برای محاسبه کوتاهترین فاصله میان کیوسک‌ها و تقاطع‌ها (معادله ۴)، به منظور افزایش سرعت اجرا الگوریتم PSO، از الگوریتم فلوید-وارشال برای محاسبه کوتاهترین فاصله استفاده شد. الگوریتم فلوید-وارشال، الگوریتمی است که کوتاهترین فاصله را بین تمامی زوج گره‌های گراف محاسبه می‌کند [4].

۴- نتایج عددی الگوریتم

برای آزمون الگوریتم از میان خیابان‌های شهر تهران منطقه مشخص‌شده در شکل ۲ انتخاب شد. گراف خیابان‌های اصلی این منطقه تشکیل شد و الگوریتم بر روی آن اجرا گردید. به دلیل اینکه حرکت پلیس راهنمایی و رانندگی وابسته به جهت خیابان و یا ترافیک نیست، تمامی لینک‌ها دوطرفه در نظر گرفته‌شده‌اند.

شکل ۲: منطقه انتخاب‌شده جهت پیاده‌سازی الگوریتم‌ها

شبکه راه‌های نقشه بالا را با استفاده از نظریه گراف‌ها مدل‌سازی می‌کنیم (شکل ۳).



شکل ۳: گراف شبکه راه‌های منطقه موردنظر

نتایج به‌دست‌آمده از ده اجرای الگوریتم در شبکه بالا به شرح جدول زیر است (جدول ۱). پارامترهایی که در این جدول موردبررسی قرارگرفته است، مقدار تابع هدف نهایی به‌دست‌آمده از الگوریتم، و همچنین واریانس‌جواب‌های به‌دست آمده است. منظور از این عدد بیشترین فاصله موجود بین کیوسک‌ها و نزدیک‌ترین تقاطع است (معادله ۶). لازم به ذکر است که در این آزمون تعداد ذرات الگوریتم PSO ثابت در نظر گرفته شده‌اند.

جدول ۱: میانگین تابع هدف در الگوریتم PSO با ۱۰ آزمایش با ۵۰۰۰۰ تکرار و ۳۵ ذره

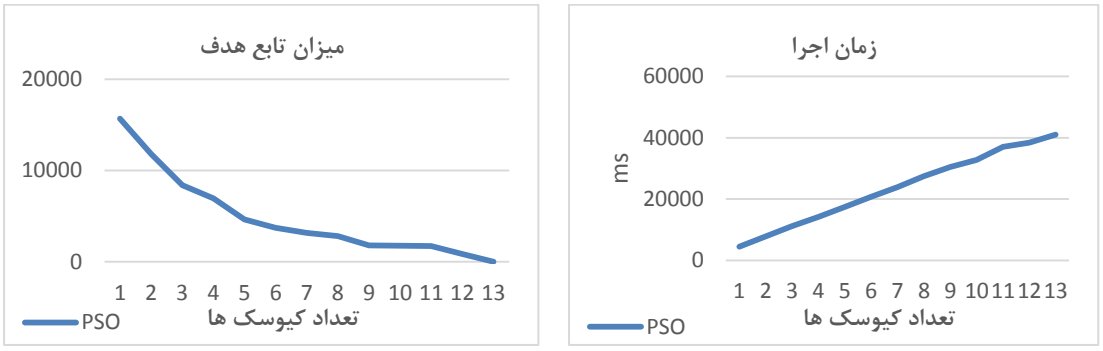
تعداد کیوسک	PSO	Variance	تعداد کیوسک	PSO	Variance
۱	۱۵۶۹۸.۰۰۱۳۷۱۸۶۷۸۹۳	2.47E-06	۸	۲۷۹۰.۵۰۱۷۸۴۲۲۷۵۶	1.82762E-06
۲	۱۱۸۰۷.۰۰۰۴۶۴۵۱۲۸۰۷	3.53668E-07	۹	۱۷۸۵.۳۰۵۷۸۹۶۴۴۶۲۵۷	0.038038624
۳	۸۳۸۵.۰۴۱۱۴۵۷۵۱۴۶۶	0.001994366	۱۰	۱۷۵۹.۰۹۴۵۳۵۹۳۸۸۴۰۶	0.005401959
۴	۶۹۶۰.۰۰۲۵۷۹۴۹۷۹۳	5.92E-06	۱۱	۱۷۲۱.۰۰۶۴۲۲۲۷۲۹۶۲	0.000261889
۵	۴۶۲۱.۰۱۷۶۸۶۴۱۷۹۸۱	0.000248	۱۲	۸۴۹.۰۰۴۶۷۹۰۷۶۷۵۵۹	1.29177E-05
۶	۳۷۳۱.۵۲۰۶۰۶۰۴۸۰۵۸	0.001059305	۱۳	۰.۰	0
۷	۳۱۴۶.۰۱۸۴۷۱۴۷۴۱۵۸	0.000275803			

همچنین زمان اجرای الگوریتم با شرایطی که گفته شد، نیز ثبت شد (جدول ۲).

جدول ۲: بررسی میانگین زمان محاسبات الگوریتم PSO با شرایط بالا

تعداد کیوسک	PSO	تعداد کیوسک	PSO
۱	۴۴۱۹.۱	۸	۲۷۴۳۶.۲
۲	۷۸۲۶.۰	۹	۳۰۴۲۷.۶
۳	۱۱۱۹۹.۲	۱۰	۳۲۷۷۵.۰
۴	۱۴۱۳۳.۳	۱۱	۳۷۰۷۶.۲
۵	۱۷۳۹۵.۹	۱۲	۳۸۴۳۲.۳
۶	۲۰۷۳۹.۶	۱۳	۴۰۹۶۰.۵
۷	۲۳۸۷۳.۶		

در شکل ۴ می‌توان نمودار زمان اجرا و میزان تابع هدف نهایی الگوریتم را مشاهده کرد.



شکل ۴: نمودار بررسی الگوریتم PSO در زمان اجرا (سمت راست) و میزان تابع هدف نهایی (سمت چپ)

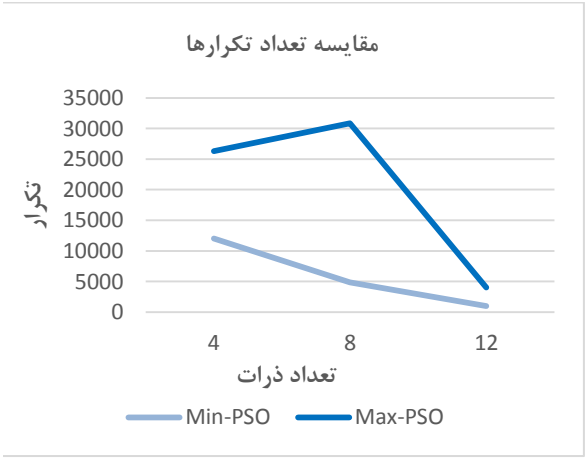
در جدول ۳ برای تعداد خاصی کیوسک، تعداد جمعیت اولیه و تکرار مورد نیاز برای الگوریتم مورد بررسی قرار گرفت. در الگوریتم‌ها معیار توقف تکرارها به‌صورت معادله ۰۸ در نظر گرفته شد. برای پر کردن جدول ۳، به ازای تعداد کیوسک مشخص شده، ۵ بار آزمایش انجام شد، اگر در این ۵ آزمایش یک‌بار مسئله به بهینه کلی نرسد، تعداد ذرات به اندازه ۵ واحد افزایش می‌یابد. شکل ۵ نمودار این تغییرات را نشان می‌دهد.

معادله ۰۸ $$Gfitness_j - Gfitness_{j-1} < 10e-2$$

جدول ۳: مقایسه تعداد تکرار و جمعیت اولیه در الگوریتم PSO

(اعداد درون پرانتز بیشترین مقدار تکرار و اعداد خارج از آن کمترین مقدار تکرار را نشان می‌دهند)

تعداد کیوسک	تعداد کیوسک = ۴	تعداد کیوسک = ۸	تعداد کیوسک = ۱۲
الگوریتم	PSO	PSO	PSO
تعداد تکرار	۱۲۰۳۳(۲۶۲۷۷)	۴۸۷۳(۳۰۸۵۵)	۹۵۷ (۴۰۳۲)
جمعیت اولیه	۲۰	۲۰	۲۰



شکل ۵: بررسی تعداد تکرارها با شرایط توقف معادله ۸

همان‌طور که از نتایج عددی آزمایش الگوریتم توسعه داده‌شده برمی‌آید، میزان پایداری[4] الگوریتم بالاست. بیشترین انحراف معیاری که از آزمون‌ها به دست آمده است، مقدار $\sigma = 0.1950349302048225$ است که در مقایسه با توابع هدف، مقدار بسیار مناسبی است [5]. افزایش زمان محاسبات در جدول ۲ امری اجتناب ناپذیر است، زیرا با افزایش تعداد کیوسک‌ها، اندازه ذرات افزایش می‌یابد و نیاز به محاسبات بیشتری است. مشاهده می‌شود که افزایش زمان محاسبات نیز، با توجه به نمودار شکل ۴، تقریباً خطی است.

[4] Robustness

در آزمون آخر که بر روی تعداد تکرارهای مورد نیاز برای رسیدن به جواب با شرط معادله ۱۰ انجام گرفته‌است، مشاهده می‌شود که پراکندگی تعداد این تکرارها زیاد است. این امر ناشی از افتادن در دام بهینه‌های محلی در برخی تکرارهاست. راه حلی که برای رها شدن از بهینه‌های محلی اتخاذ شده‌است، جهش در برخی ذرات است که در بخش ۳ توضیح داده‌شد.

۵- نتایج و پیشنهادات

در این مقاله برای مقابله با پدیده ترافیک، استفاده از پلیس‌های راهنمایی و رانندگی به منظور تسهیل آمدوشد وسائل نقلیه مورد توجه قرار گرفت. می‌توان با توزیع مکانی مناسب پلیس‌های راهنمایی و رانندگی در قسمت‌های مختلف شبکه راه، زمان رسیدن آن‌ها به مناطق ترافیکی را کاهش داد و بدین ترتیب وضعیت ترافیکی را بهبود داد. بر این اساس مکان‌یابی تعدادی کیوسک راهنمایی و رانندگی در سطح یک شبکه راه شهری با استفاده از الگوریتم PSO انجام شد. بررسی‌هایی که تاکنون انجام‌شده است، نشان می‌دهند که الگوریتم PSO بااینکه ذاتاً با این مسئله تفاوت دارد، اما توانسته است به خوبی جوابگوی این مسئله باشد.

پیشنهاد می‌شود که برای مسائل مکان‌یابی از تابع بهینگی بهتری برای این کار استفاده شود. تابع فوق به دلیل استفاده از کمینه کردن بیشینه فاصله نزدیک‌ترین فاصله بین تقاطع‌ها و کیوسک‌ها، نمی‌تواند به‌صورت کامل مکان تمامی کیوسک‌ها را بهینه کند. پیشنهاد می‌شود که در مطالعات مشابه از معادله‌ی ۱۱ استفاده شود. همچنین توسعه الگوریتمی بر مبنای الگوریتم کلونی زنبورعسل و مقایسه آن با الگوریتم PSO می‌تواند نتایج بهتر الگوریتم کلونی زنبورعسل را در پی داشته باشد. زیرا این الگوریتم به‌صورت ذاتی با این مسئله قرابت دارد.

معادله ۱۱ $$\sum_i Dps(x_i, Y)$$

مراجع

[1] N. Zhao and Z. Li, "Optimize Traffic Police Arrangement in Easy Congested Area Based on Improved Particle Swarm Optimization," in *Procedia - Social and Behavioral Sciences*, 2014.

[2] R. Eberhart and J. Kennedy, "A new optimizer using particle swarm theory," in *Proceedings of the sixth international symposium on micro machine and human science*, 1995.

[3] A. Ghaderi,. M. S. Jabalameli,. F. Barzinpour and. R. Rahmaniani, "An Efficient Hybrid Particle Swarm Optimization Algorithm for Solving the Uncapacitated Continuous Location-Allocation Problem," *Networks and Spatial Economics*, vol. 12, no. 3, pp. 421-439, 2012.

[4] T. H. Cormen, Introduction to algorithms, MIT press, 2009.

[5] A. P. Engelbrecht, Fundamentals of computational swarm intelligence, John Wiley & Sons, 2006.

[6] D. Karaboga and B. Basturk, "An idea based on honey bee swarm for numerical optimization," Technical report-tr06, Erciyes university, engineering faculty, computer engineering department, 2005.