سنجش از دور و GIS ایران

سنجش از دور و GIS ایران سال پنجم، شماره سوم، پاییز ۱۳۹۲
Iranian Remote Sensing & GIS Vol.5, No.3, Autumn 2013

۷۹-۹۸

ارائه روش ترکیبی بهینه‌سازی جمعیت مورچه‌ها و الگوریتم ژنتیک برای حل مسئله فروشنده دوره‌گرد در GIS

ندا کفاش چرندابی[۱]*، علی‌اصغر آل‌شیخ[۲]، گلبرگ کامروز خدایار[۳]

۱. دانشجوی دکتری GIS، دانشکده مهندسی نقشه‌برداری، دانشگاه صنعتی خواجه نصیرالدین طوسی
۲. دانشیار دانشکده مهندسی نقشه‌برداری، دانشگاه صنعتی خواجه نصیرالدین طوسی
۳. کارشناس ارشد GIS، دانشکده مهندسی نقشه‌برداری، دانشگاه صنعتی خواجه نصیرالدین طوسی

تاریخ دریافت مقاله: ۱۳۹۱/۷/۱۹ تاریخ پذیرش مقاله: ۱۳۹۲/۱۲/۱۰

چکیده

انسان از دیرباز برای حل مسائل پیچیده، از جهان زندهٔ پیرامونش الهام گرفته است. این امر آشکارا در توسعهٔ الگوریتم‌های مختلف تقریبی، از نظریهٔ تکاملی داروین تا الگوریتم‌های مختلف هوش جمعی، دیده می‌شود. مسئلهٔ فروشنده دوره‌گرد از مسائلی است که می‌توان آن را با الگوریتم هوش جمعی به چالش کشید. در پژوهش حاضر ابتدا با ارزیابی و تنظیم صحیح پارامترهای مؤثر در الگوریتم بهینه‌سازی کلونی مورچه‌ها، الگوریتم هوش جمعی بهبود می‌یابد. سپس روشی ترکیبی برای حل دودویی مسئله فروشندهٔ دوره‌گرد در مقیاس بزرگ و برمبنای الگوریتم بهبودیافته کلونی مورچه‌ها و عملگرهای الگوریتم ژنتیک ارائه می‌شود. به‌منظور بررسی کیفیت جواب‌های به‌دست‌آمده، نتایج روش پیشنهادی با نتایج دو الگوریتم بهینه‌سازی کلونی مورچه‌ها و الگوریتم ژنتیک در مسیریابی بین مراکز استان‌ها مقایسه می‌شود. این مقایسه بهبود در جواب‌ها، کاهش زمان اجرای الگوریتم، و کاهش حجم لازم برای ذخیره‌سازی جواب‌های به‌دست‌آمده در شرایط گوناگون را نشان می‌دهد. با توجه به پایداری و بهینگی نتایج حاصل از الگوریتم مورچهٔ بهبودیافته و اهمیت افزایش خدمات در سامانه اطلاعات مکانی، کاربرد الگوریتم پیشنهادی در صنعت توریسم مطرح می‌شود.

کلیدواژه‌ها: GIS، TSP، الگوریتم بهینه‌سازی کلونی مورچه‌ها، الگوریتم ژنتیک، بهینه‌سازی ترکیبی.

* نویسنده مکاتبه‌کننده: تهران، خیابان ولی‌عصر، تقاطع میرداماد، دانشگاه صنعتی خواجه‌نصیرالدین طوسی، دانشکده نقشه‌برداری. تلفن: ۸۸۷۸۶۲۱۲
Email: n_kaffash@Yahoo.com

۱- مقدمه

هدف بهینه‌سازی، یافتن بهترین راه‌حل از میان تمام گزینه‌های ممکن است. مسائل بهینه‌سازی ترکیبی، فضای جست‌وجوی گسسته‌ای از راه‌حل‌های ممکن را ایجاد می‌کنند و در بسیاری از موارد پیچیدگی‌های محاسباتی بالایی دارند و در کلاس NP-hard[1] طبقه‌بندی می‌شوند (Mohammadi et al., 2009). مسئله فروشنده دوره‌گرد[2] (TSP) نمونه‌ای رایج از مسائل NP-Complete و از مشهورترین مسائل بهینه‌سازی ترکیبی است (Gutin and Punnen, 2002). مسئله TSP را ویلیام همیلتون و توماس کرکمن در قرن ۱۸ میلادی مطرح کردند و سپس در دهه ۱۹۳۰ ریاضی‌دانانی همچون کارل منگر و هاسلر ویتنی به بررسی آن پرداختند (Paralos and Resende, 2002).

مسئله فروشنده دوره‌گرد، یافتن کم‌هزینه‌ترین مسیری است که از یک شهر شروع شود و از تمامی شهرها دقیقاً یک‌بار عبور کند و به شهر شروع بازگردد. به عبارت دیگر، یافتن کم‌وزن‌ترین دور همیلتونی در یک گراف کامل و محدود، مسئله فروشنده دوره‌گرد را شکل می‌دهد. این مسئله از مسائل مهم در نظریه پیچیدگی محاسباتی الگوریتم‌ها به‌شمار می‌آید، که از راه‌حل آن به‌عنوان معیاری برای ارزیابی بسیاری از روش‌های آماری و ابتکاری[3] و در حل مسائل بهینه‌سازی ترکیبی استفاده می‌شود (Gutin and Punnen, 2002). هدف تکنیک‌های ابتکاری در مواجهه با مسائل بهینه‌سازی، یافتن راه‌حل نزدیک به بهینه (تقریبی) در زمانی منطقی است. فناوری‌های فراابتکاری[4] نیز راهبردهای سطح بالایی هستند که در فضاهای جست‌وجو با استفاده از روش‌های گوناگون میان تنوع و افزایش فضای جست‌وجو تعادل ایجاد می‌کنند (Blum and Roli, 2003). نمونه‌ای از فناوری فراابتکاری، محاسبات تکاملی[5] (EC) است که الگوریتم‌هایی همچون ژنتیک[6] (GA) و بهینه‌سازی کلونی مورچه‌ها[7] (ACO) را دربرمی‌گیرد.

استراتژی بهینه‌سازی کلونی مورچه‌ها برمبنای غریزه این جانور در یافتن مسیر بهینه بنا شده است. الگوریتم‌های ژنتیک نیز از اصول انتخاب طبیعی داروین مانند وراثت و جهش، برای یافتن فرمول بهینه به‌منظور پیش‌بینی یا تطبیق الگو استفاده می‌کنند. هر دو الگوریتم ACO و GA مبتنی بر جمعیت‌اند و مهم‌ترین مزیت‌شان به روش‌های قطعی، سرعت حل مسئله به‌ویژه در مسائلی است که حجم داده‌هایش زیاد است (Misevicius et al., 2004).

در مقالهٔ حاضر دو الگوریتم ACO و GA معرفی می‌شود و ضمن حل TSP برای مراکز استان‌های کشور، مقایسهٔ این دو الگوریتم صورت می‌گیرد و پارامترهای مؤثر بر آنها تحلیل می‌شود. سپس برمبنای تحلیل صورت‌گرفته و با استفاده از برخی ویژگی‌های الگوریتم ژنتیک، روشی ترکیبی برای حل TSP در مسائلی با ابعاد بزرگ پیشنهاد می‌شود و اعتبار آن به‌کمک داده‌های واقعی سنجیده می‌شود.

۱-۱- اهمیت مسئله و پیشینه تحقیق

مسئله فروشندهٔ دوره‌گرد، مسئله‌ای شناخته‌شده در مسائل بهینه‌سازی ترکیبی است که مدل‌سازی آن ساده اما راه‌حلش دشوار است. ازآنجاکه این مسئله در حوزه‌های مختلف علوم و مهندسی کاربرد دارد، روش‌های متعددی برای حل آن پیشنهاد شده است.

به‌طور کلی برای حل TSP به دو روش ناآگاهانه[8] و آگاهانه[9] عمل می‌شود (Gutin and Punnen, 2002; Paralos, and Resende, 2002). در روش ناآگاهانه تمامی حالت‌های ممکن بررسی می‌شود، به همین

1. Non-deterministic Polynomial-time hard
2. Traveling Salesman Problem
3. Heuristics
4. Metaheuristics
5. Evolutionary Computation
6. Genetic Algorithm
7. Ant Colony Optimization
8. Uninformed
9. Informed

خاطر لذا در بسیاری از موارد به‌دلیل گستردگی فضای جست‌وجو و پیچیدگی بالای آن، کارایی چندانی ندارد. به‌منظور پرهیز از بررسی تمام حالت‌های ممکن، استفاده از روش‌های آگاهانه پیشنهاد می‌گردد که خود به دو دسته الگوریتم‌های قطعی و تقریبی تقسیم می‌شود. الگوریتم‌های قطعی با صرف زمان و هزینه زیاد، مقدار بهینهٔ قطعی را برآورد می‌کنند. به‌دلیل زمان‌بر و پرهزینه بودن این روش‌ها، الگوریتم‌های تقریبی برای دستیابی به مقادیر نزدیک به بهینه پیشنهاد شده‌اند و الگوریتم‌های ابتکاری و فراابتکاری نیز در این دسته از روش‌های حل TSP جای می‌گیرند. الگوریتم اجتماع مورچه‌ها (Dorigo et al., 2006; Dorigo and Gambardella, 1997a; 1997b; Socha and Dorigo, 2008; Hung et al., 2007) الگوریتم ژنتیک (Mohammadi et al., 2009; Chong, 2001; Hashemi et al. 2010)، شبکه‌های عصبی مصنوعی (Kos and Nagórny, 2009; Hosseinali et al, 2009) و الگوریتم بهینه‌سازی ذرات (Clerc, 2004; Elbeltagi et al., 2005) از الگوریتم‌های فراابتکاری هستند که برای حل TSP به‌کار رفته‌اند. شکل ۱، دسته‌بندی روش‌های مختلف حل TSP را نشان می‌دهد.

مطالعات زیادی در زمینه الگوریتم‌های تقریبی برای حل مسئله فروشنده دوره‌گرد در مقیاس کوچک صورت گرفته است. منظور از مسائل TSP در مقیاس کوچک (ابعاد کوچک) مسائلی با فضای جست‌وجوی کوچک و محدود است. با افزایش فضای مطالعاتی (افزایش تعداد نودها و یال‌های واصل) این نوع مسائل به مسائل TSP در مقیاس بزرگ (ابعاد بزرگ) بدل می‌شوند. سوچا و دوریگو (Socha and Dorigo, 2008) با بهره‌گیری از ACO به حل مسائل بهینه‌سازی گسسته و پیوسته در ابعاد کوچک (به‌طور نمونه بهینه‌سازی توابع تست معروف در ریاضی) پرداختند. دوریگو و گامباردلا (Dorigo and Gambardella, 1997a; 1997b) برای ارائه الگوریتم بهینه‌سازی کلونی مورچه‌ها، این الگوریتم را با سایر الگوریتم‌های تقریبی در حل TSP مقایسه کردند. نتایج این تحقیقات بر بهینه‌بودن نتایج ACO نسبت به سایر روش‌های بهینه‌سازی ـ به‌ویژه در مسائل با ابعاد کوچکـ تأکید دارد. در مقایسهٔ این الگوریتم‌ها، پارامترهای مؤثر و ترکیب الگوریتم‌ها بررسی نشده‌اند.

شکل ۱. دسته‌بندی روش‌های مختلف حل مسئله TSP

چونگ (Chong, 2001) و هانگ (Hung et al., 2007) نیز به حل مسئله مسیریابی در ابعاد کوچک با الگوریتم ACO پرداختند. همگی این مطالعات حاکی از آن است که الگوریتم بهینه‌سازی کلونی مورچه‌ها به‌تنهایی در حل مسائلی با ابعاد کوچک در مقایسه با سایر الگوریتم‌های تقریبی بسیار موفق عمل می‌کند. الگوریتم ACO در مقایسه با سایر روش‌های بهینه‌سازی برای حل مسائلی با فضای جست‌وجوی محدود و کوچک کارایی بهتری دارد و در زمان کوتاه و با دقت مناسب بهینه‌سازی را انجام می‌دهد. اما در مورد مسائلی با ابعاد بزرگ‌تر ـ به‌عنوان مثال داده‌های واقعی مراکز استان‌ها و راه‌های کشور که در این تحقیق استفاده شده است ـ این روش به تنهایی کارآمد نیست و باید به کمک ایده‌ها و روش‌های جدید بهبود یابد (Grotschel and Holland, 1991; Ghassiri and Sarhadi, 2009; Hung et al., 2007; Chong, 2001).

بهبود الگوریتم ACO به‌کمک استفاده از قوانین به‌روزرسانی پویا (Hung et al., 2007) و ترکیب این الگوریتم با روش جست‌وجوی محلی (Ghassiri and Sarhadi, 2009) بخشی از تحقیقاتی بوده که برای بهبود الگوریتم ACO انجام شده است. به‌دلیل توانمندی ویژهٔ الگوریتم ژنتیک در حل مسائل بهینه‌سازی، برخی پژوهشگران به ترکیب این روش با الگوریتم ACO نیز توجه کرده‌اند؛ به‌عنوان نمونه گوانگدونگ و همکارانش (Guangdong et al., 2007) فرایندی ترکیبی از الگوریتم‌های ACO و GA را پیشنهاد کرده‌اند که در آن در ابتدا با الگوریتم GA جست‌وجوی اولیه‌ای در منطقه انجام گرفته و نتایج آن به‌عنوان جمعیت اولیه وارد الگوریتم ACO شده است. لی و همکارانش (Lee et al., 2008) در قالب فرایندی ترکیبی ابتدا جست‌وجو با الگوریتم ACO را انجام داده و سپس الگوریتم GA را اجرا کرده‌اند تا احتمال گرفتارشدن الگوریتم GA در بهینه‌های محلی کاهش یابد. در الگوریتم پیشنهادی نعمتی و همکارانش (Nemati et al., 2009) در هر تکرار هر دو الگوریتم ACO و GA اجرا می‌شوند. بهترین جواب از میان نتایج این دو روش انتخاب و الگوریتم وارد تکرار بعدی می‌شود. در تحقیق لی و همکارانش (Li et al., 2011) برای بهبود الگوریتم GA از روش ACO استفاده شده است، به این ترتیب که پارامترهای دو الگوریتم مقداردهی اولیه می‌شوند، سپس الگوریتم GA اجرا می‌شود و عملگرهای برازش، ادغام و جهش روی آن اعمال می‌گردد. آنگاه روی یال انتخاب‌شده میزان فرمون (Pheromone) به‌روزرسانی می‌شود تا احتمال گرفتارشدن الگوریتم GA در بهینه‌های محلی کمتر شود. الگوریتم پیشنهادی تحقیق لی (2011) در مقیاس کوچک و برای مسیریابی وسایل حمل‌ونقل اجرا شده است. در مطالعات انجام‌شده بیشتر بر اجرای هم‌زمان و موازی دو الگوریتم یا بر اجرای متوالی آنها برای کاهش فضای جست‌وجو تأکید شده، اما به ترکیب عملگرهای دو الگوریتم در قالب الگوریتمی واحد برای کاهش زمان اجرای الگوریتم و بهبود نتایج ـ به‌ویژه در تحلیل مسائل بزرگ‌مقیاس ـ پرداخته نشده است.

در پژوهش حاضر با تنظیم پارامترهای مؤثر در الگوریتم بهینه‌سازی کلونی مورچه‌ها و ترکیب آن با برخی ویژگی‌ها و عملگرهای الگوریتم ژنتیک، روشی ترکیبی و بهبودیافته برای حل مسئله TSP در ابعاد بزرگ برای داده‌های واقعی مراکز استان‌ها و راه‌های کشور ارائه شده است. در پایان نیز آنالیز حساسیت برای تحلیل پارامترهای مؤثر در الگوریتم پیشنهادی و مقایسهٔ آن با الگوریتم اصلی کلونی مورچه و ژنتیک انجام شده است.

۱-۲- الگوریتم ژنتیک در حل TSP

الگوریتم ژنتیک یکی از روش‌های جست‌وجوی تصادفی برگرفته از طبیعت است که امروزه در حل مسائل بهینه‌سازی و فرایندهای یادگیری کاربرد گسترده‌ای دارد (Sheta and Turbaieh, 2006). در طبیعت از ترکیب کروموزوم‌های مناسب، نسل‌های بهتری پدید می‌آیند؛ در این بین گاهی جهش‌هایی نیز در

کروموزوم‌ها رخ می‌دهند که ممکن است باعث بهتر شدن نسل بعدی شوند. الگوریتم ژنتیک با استفاده از این ایده اقدام به حل مسائل می‌کند. در ادامه، عملگرهای مختلف این الگوریتم بررسی می‌شود (Ravagnani et al., 2005):

- برازش[1]

میزان بهینگی هر کروموزوم با استفاده از عملگر برازش تعیین می‌گردد. در این مرحله با معرفی معیار، بهتربودن یا نبودن هر جواب ممکن از جمعیت اولیه نسبت به سایر جواب‌ها بررسی می‌شود، که به آن میزان بهینگی هر کروموزوم می‌گویند. عملگر برازش به هر کروموزوم مقداری را نسبت می‌دهد که همان احتمال ترکیب آن برای تولید نسل‌های آینده است. بدیهی است که کروموزوم‌های بهینه شانس بیشتری برای ترکیب با دیگر کروموزوم‌ها دارند. بنابراین احتمالی که به آنها نسبت داده می‌شود نیز بیشتر است.

- انتخاب[2]

پس از آنکه عملگر برازش روی جمعیت انجام پذیرفت، عملگر انتخاب شروع به کار می‌کند. وظیفهٔ این عملگر انتخاب کروموزوم‌هایی از میان جمعیت موجود برای ترکیب‌شدن است. کروموزوم‌های با مقادیر برازندگی بیشتر، شانس زیادتری برای انتخاب دارند. روش‌های گوناگونی برای انتخاب کروموزوم‌ها وجود دارد، که یکی از معمول‌ترین آنها روش رقابتی است. در روش رقابتی دو یا چند کروموزوم به‌طور تصادفی انتخاب می‌شوند و از میان‌شان کروموزومی که برازش آن بهتر از دیگران است، برگزیده می‌شود. این عمل به تعداد کروموزوم‌های جمعیت اولیه تکرار می‌شود. محصول نهایی عملگر انتخاب، جمعیت میانی است که از آن در مراحل بعدی استفاده خواهد شد.

- ادغام[3]

عملگر ادغام، کروموزوم‌های جمعیت میانی را با هم ترکیب می‌کند تا شاید کروموزوم فرزند حاصل از ترکیب آنها از کروموزوم والد بهتر باشد. ادغام کروموزوم‌ها روش‌های مختلفی دارد که از میان سه روش ادغام تک‌نقطه‌ای، ادغام دونقطه‌ای، و ادغام به‌وسیلهٔ ماسک، در این تحقیق از ادغام تک‌نقطه‌ای به‌دلیل سادگی آن استفاده شد.

- جهش[4]

وقتی عملگر جهش روی کروموزومی اعمال می‌شود باعث بروز جهش در آن کروموزوم می‌گردد. روش معمول برای جهش، تغییر تصادفی یک یا چند ژن از کروموزوم است.

برای حل TSP به‌کمک الگوریتم ژنتیک، ابتدا گراف نودهای مسیر برمبنای مختصات تقریبی تشکیل می‌شود، سپس ماتریس مجاورت[5] از نودهای مسیر شکل می‌گیرد و طول مسیر به‌عنوان ملاکی برای ارزیابی مسیر انتخابی به هریک از یال‌ها نسبت داده می‌شود (رابطه (۱)). در رابطه (۱)، n تعداد نودهای مسیر و l_{ij} طول مسیر میان نود i و j به کیلومتر است که در این تحقیق به‌عنوان تابع هزینه درنظر گرفته شده است.

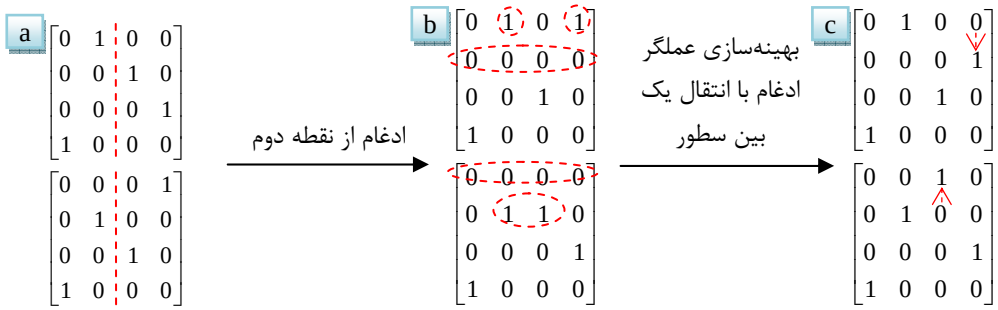
رابطه (۱)

$$L=\begin{bmatrix} \cdot & l_{12} & l_{13} & \dots & l_{1n} \\ l_{21} & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \vdots & \cdot & \cdot & \ddots & \cdot \\ l_{n1} & l_{n2} & l_{n3} & \dots & \cdot \end{bmatrix}$$

در ادامه، دو رشته از نودهای مسیر برمبنای بهینگی تابع هزینه به‌تصادف به‌وسیله عملگر انتخاب الگوریتم ژنتیک برگزیده می‌شود. سپس عملگر ادغام تک‌نقطه‌ای اجرا می‌شود، دو نقطه از دو رشته نودهای مسیر به‌تصادف انتخاب می‌گردد و ناحیه چپ یا راست نقطهٔ انتخابی در دو رشته جابه‌جا می‌شوند. به‌عنوان مثال، ادغام دو رشتهٔ ۱۲۳۴ و ۱۴۲۳ از نقطه دوم به‌صورت ۱۲۲۳ و ۱۴۳۴ درخواهد آمد.

رشتهٔ ۱۲۳۴ حرکت از نود یک به دو و از نود دو به نود سه و نهایتاً از آن به نود چهار را نشان می‌دهد.

1. Fitness
2. Selection
3. Crossover
4. Mutation
5. Adjacency Matrix



شکل ۲. روند اعمال عملگر ادغام در TSP

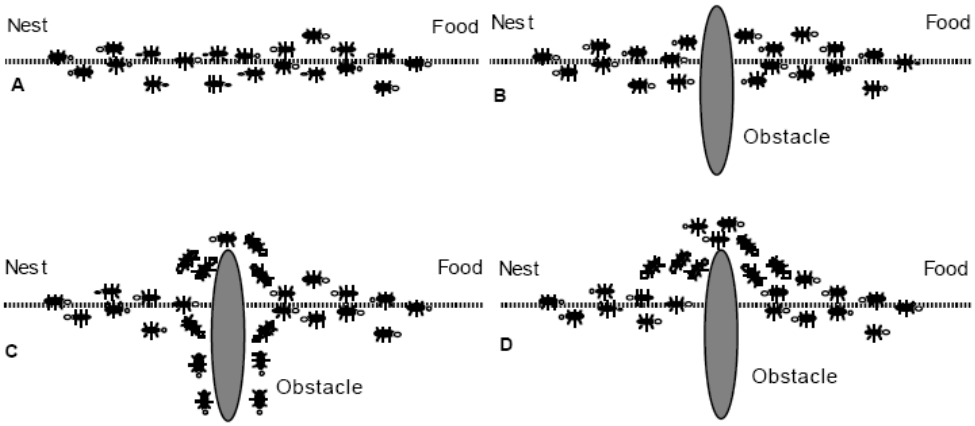
با اجرای عملگر سادهٔ ادغام تک‌نقطه‌ای امکان دارد که رشتهٔ اول فاقد نود ۴ و رشتهٔ دوم فاقد نود ۲ باشد و در مقابل به ترتیب نود ۲ و نود ۴ دوباره طی شود. برای رفع مشکلاتی از این دست و در مسائلی که نیاز به ایجاد یک دور مجاز با حضور تمامی نودهای انتخابی است، از روش ماتریس ادغام استفاده می‌شود (Khan Teymori, 2008). در این روش روی دو ماتریس اول (شکل ۲-a) که به ترتیب مبین دو رشته مسیر ۱۲۳۴ و ۱۴۲۳ است (مسیری متشکل از چهار نود) عملگر ادغام تک‌نقطه‌ای از نقطه دوم اعمال و ستون سوم و چهارم دو ماتریس جابه‌جا می‌شوند. سپس با توجه به اینکه برخی سطرها چندین یک دارند و برخی فاقد یک هستند (شکل ۲-b)، انتقال یک‌ها از سطرهایی با چندین یک به سطرهای فاقد یک به‌طور تصادفی صورت می‌گیرد (شکل ۲-c).

سپس عملگر جهش روی رشته مسیر انتخابی اعمال می‌گردد و بررسی می‌شود که آیا با صفرشدن نود یک یا یک‌شدن نود صفر در نودهای یک رشته (که به‌صورت تصادفی محل اعمال آن معین می‌گردد)، طول مسیر مربوط به آن رشته کوتاه می‌شود یا خیر. درصورت مثبت بودن پاسخ، نودها جابه‌جا می‌شوند و تابع هزینهٔ مربوط به آن بهینه می‌گردد. مراحل فوق تا رسیدن به حداکثر تکرار تعیین‌شده ادامه می‌یابد (Khan Teymori, 2008).

۱-۳-الگوریتم بهینه‌سازی کلونی مورچه‌ها در حل TSP

نخستین‌بار، دوریگو این خانواده از روش‌ها را به‌عنوان راه‌حلی چندعامله[1] برای حل مسائل بهینه‌سازی نظیر فروشنده دوره‌گرد مطرح کرد (Dorigo and Blum, 2005). در سال‌های اخیر کاربردهای فراوانی از به‌کارگیری این الگوریتم در مسائل بهینه‌سازی پیچیده نظیر انتخاب اشیا و تشخیص حالت چهره (Kanan et al., 2007)، مسیریابی وسایل نقلیه برای بهبود شبکه حمل‌ونقل (Yang et al., 2007)، و تکنیک‌های فراگیری ماشین (Mullen et al., 2009) ارائه گردیده است. از مهم‌ترین مزایای الگوریتم‌های هوش جمعی می‌توان به نبود کنترل متمرکز، تعاملات توزیع‌شدهٔ موجودات، سرعت انتقال و کارکرد موازی اشاره کرد (Beckers et al., 1992). نکتهٔ درخور توجه دیگر، توانایی سازگاری مورچه‌ها با تغییرات محیط پیرامون‌شان است؛ به‌عنوان مثال زمانی که مانعی کوتاه‌ترین مسیر را بسته باشد، مورچه‌ها کوتاه‌ترین مسیر جدید را پیدا می‌کنند (شکل ۳) (Dorigo and Gambardella, 1997).

1. Multi Agent



شکل ۳. نمایش مراحل یافتن کوتاه‌ترین مسیر به‌وسیله مورچه‌ها
منبع: Dorigo and Gambardella, 1997

شکل (A)۳ مورچه‌ها را در خط مستقیمی میان لانه و مکان غذا نشان می‌دهد. نخستین چیزی که در تشکیل و حفظ این خط به مورچه‌ها کمک می‌کند اثر فِرمون است. مورچه‌ها هنگام راه‌رفتن مقدار معینی فرمون ترشح می‌کنند و هر مورچه از روی غریزه و احتمال مسیری را که دارای فرمون بیشتری است، دنبال می‌کند. حال اگر مانعی در راه مسیر اولیه قرار گیرد (شکل (B)۳)، مورچه‌هایی که دقیقاً مقابل مانع هستند نمی‌توانند اثر فرمون را دنبال کنند، بنابراین نصف مورچه‌ها چرخش به راست و نصف دیگرشان چرخش به چپ را انتخاب می‌کنند (شکل (c)۳). مورچه‌هایی که به‌صورت اتفاقی راه کوتاه‌تر (حول مانع) را انتخاب می‌کنند در مقایسه با مورچه‌هایی که مسیر طولانی‌تر را برگزیده‌اند، فرمون قبلی را سریع‌تر تقویت می‌کنند. بدین ترتیب در ادامه مورچه‌های بیشتری مسیر کوتاه‌تر را ـ که مقدار فرمون بیشتری دارد ـ انتخاب می‌کنند (شکل (D)۳).

برای حل TSP، مورچه‌هایی مصنوعی طراحی می‌شوند که روی گراف TSP از یک شهر به شهر دیگر حرکت کنند. آنها برمبنای فرمولی احتمالی ـ که تابع فرمون تجمعی روی یال‌ها و مقدار اکتشافی ـ که خود تابعی از طول یال‌هاست ـ جابه‌جا می‌شوند (Dorigo et al., 2006). مورچه‌های مصنوعی به‌صورت احتمالی شهرهایی را که به وسیلهٔ یال‌هایی با فرمون بسیار زیاد به هم متصل و به یکدیگر نزدیک‌اند، انتخاب می‌کنند. ویژگی‌های کلی‌ای که برای مورچه‌های مصنوعی در نظر گرفته شده در جدول ۱ آمده است.

جدول ۱. مقایسهٔ خصوصیات مورچه‌های واقعی با مورچه‌های مصنوعی

رفتار مورچه‌های مصنوعی برگرفته از مورچه‌های واقعی	تفاوت رفتار مورچه‌های مصنوعی با مورچه‌های واقعی برای کارآمدشدن TSP
حق تقدم انتخاب مسیرهایی که سطح فرمون بالایی دارند.	تشخیص مسافت شهرها به‌وسیله مورچه‌های مصنوعی
نرخ بالاتر افزایش مقدار فرمون در مسیرهای کوتاه‌تر	برخورداری از نعمت حافظه برای مورچه‌های مصنوعی- به‌منظور ذخیرهٔ شهرهای بازدیدشده
نقش واسطهٔ ارتباطی ردپا، میان مورچه‌ها	

ابتدا m مورچهٔ مصنوعی در شهرهایی که به‌طور تصادفی انتخاب شده‌اند قرار داده می‌شوند، به‌گونه‌ای‌که هر مورچه روی یک رأس قرار گیرد. مورچه‌ها به‌صورت موازی با هم اقدام به پیمایش تور می‌کنند. زمانی که مورچهٔ kام در رأس iام قرار دارد، براساس رابطه (۲) عدد تصادفی q در فاصله [۱ و ۰] تولید می‌شود و با q_0، که عدد تصادفی دیگری در فاصله [۱ و ۰] است، مقایسه می‌شود (Dorigo and Blum, 2005).

رابطه (۲)

$$s=\begin{cases}\text{Max}\,(\tau_{ij})^{\alpha}\times(\eta_{ij})^{\beta} & q\le q_0\\ S & q>q_0\end{cases}$$

در رابطه فوق، S متغیری تصادفی است که از روی توزیع احتمال رابطه (۳) محاسبه می‌شود. این فرمول احتمال جابه‌جایی یک مورچه را از گره i به گره j محاسبه می‌کند تا مورچه برمبنای احتمالات به‌دست‌آمده جابه‌جا شود:

رابطه (۳)

$$p_{ij}^{k}(t)=\frac{\left[\tau_{ij}(t)\right]^{\alpha}\times\left[\eta_{ij}(t)\right]^{\beta}}{\sum_{l\in J_i^k}\left[\tau_{il}(t)\right]^{\alpha}\times\left[\eta_{il}(t)\right]^{\beta}}$$

τ_{ij} مقدار فرمون موجود روی مسیر ij، α پارامتری برای کنترل تأثیر τ_{ij}، $\eta_{ij}(t)$ میزان مطلوبیت گره j به‌عنوان گره بعدی و در حل TSP برابر $1/d_{ij}$، و β نیز پارامتری برای کنترل تأثیر $\eta_{ij}(t)$ است.

مورچه‌ها در هر مرحله زمانی، براساس قانون تغییر حالت به شهرهای جدید می‌روند و اثر فرمون روی یال‌های به‌کاررفته را تغییر می‌دهند، که به این کار بهنگام‌سازی محلی اثر فرمون گفته می‌شود (Dorigo and Stüzle, 2004). با به‌کارگیری فرمول بهنگام‌سازی محلی (رابطه (۴))، فرمون یال‌ها تغییر می‌کند که هدف از آن، اجتناب از انتخاب یک یال بسیار قوی به‌وسیله همه مورچه‌هاست.

رابطه (۴)

$$\tau_{ij}(t)\leftarrow(1-\rho).\tau_{ij}(t)+\rho.\tau_0$$

در رابطه فوق τ_0 مقدار اولیه مفروض برای فرمون و ρ میزان تبخیر فرمون است.

اثر فرمون به‌صورت محلی و کلی تغییر می‌کند. وقتی هر مورچه یک تور را طی می‌کند، مورچه‌ای که کوتاه‌ترین تور را ساخته است، فرمون یال‌های مربوط به تور خودش را با اضافه‌کردن مقداری فرمون (که با طول تور نسبت عکس دارد) تغییر می‌دهد. به این عمل بهنگام‌سازی کلی اثر فرمون گفته می‌شود (Dorigo and Stüzle, 2004)، که از رابطه (۵) محاسبه می‌شود.

رابطه (۵)

$$\tau_{ij}(t)\leftarrow(1-\rho)\cdot\tau_{ij}(t)+\rho\cdot\Delta\tau_{ij}(t)$$

در رابطه فوق، $\Delta\tau_{ij}$ مجموعه فرمون اضافه‌شده به‌وسیله مورچه بر لبهٔ i,j در این مرحله است و از رابطه (۶) محاسبه می‌گردد.

رابطه (۶)

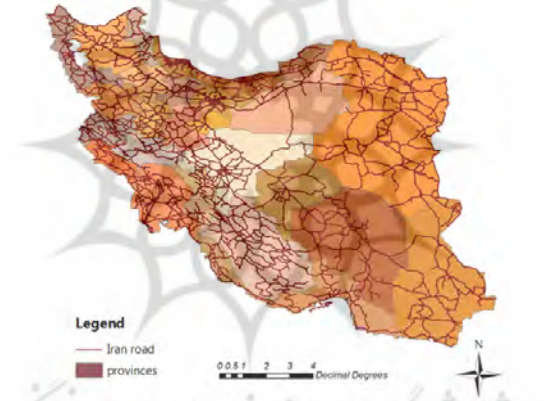
$$\Delta\tau_{ij}(t)=Q/L^{K}(t)\quad \text{if}\quad (i,j)\in T^{k}(t)\quad \text{else } 0$$

که در آن L^K مجموعهٔ مسافت‌ها برای همهٔ گره‌ها و لبه‌هایی است که مورچهٔ K ام در مسیر از آنها گذشته، Q مقداری ثابت، و $T^k(t)$ مجموعه لبه‌های طی‌شده به‌وسیله مورچه K ام است. این مراحل تا رسیدن به حد آستانه (حداکثر تکرار) ادامه می‌یابد.

۲- مواد و روش‌ها

شکل ۴ روند اجرایی این پژوهش را برای حل TSP در مقیاس بزرگ نشان می‌دهد.

شکل ۴. روند اجرای پژوهش



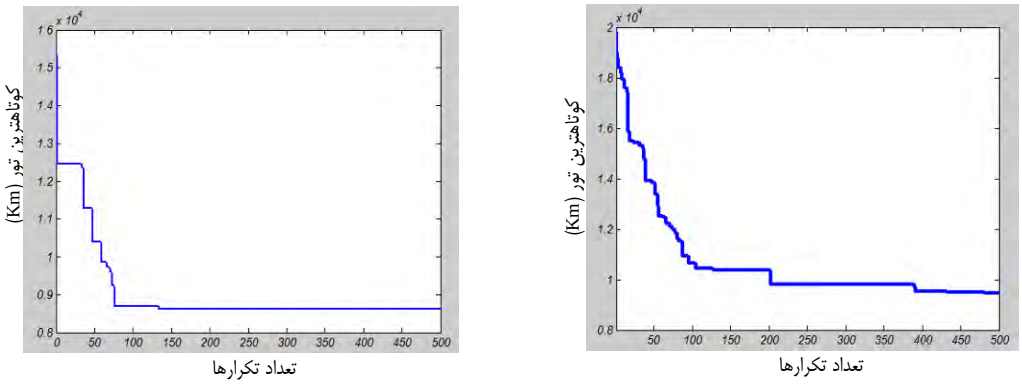
شکل ۵. نقشهٔ استان‌های کشور به‌همراه راه‌های ارتباطی

۲-۱- منطقهٔ مطالعه‌شده

در این تحقیق مراکز استان‌های کشور برای مطالعهٔ موردی انتخاب شدند. در فاز جمع‌آوری داده‌ها، لایه‌های مربوط به مراکز استان‌ها و راه‌های ارتباطی میان آنها گردآوری شدند (شکل ۵). مراکز استان‌ها به‌عنوان گره‌ها و راه‌های ارتباطی (برمبنای نقشهٔ راه‌های کشور) به‌عنوان یال‌های واصل TSP فرض شدند. سپس به کمک ACO و GA مسئله فروشنده دوره‌گرد همان‌طور که در ادامه شرح داده می‌شود، حل گردید.

۲-۲- اجرای الگوریتم ACO و GA برای حل TSP

پس از جمع‌آوری داده‌های مورد نیاز، مراکز استان‌های کشور به‌عنوان نودها و نقشهٔ راه‌های کشور مبنای یال‌های واصل میان نودها قرار گرفت. به‌منظور پیاده‌سازی الگوریتم‌های ACO و GA برای حل TSP میان مراکز استان‌های کشور مطابق آنچه در اینجا گفته شد، از کدنویسی در محیط MATLAB 7 استفاده گردید. اجرای دو الگوریتم مذکور به همگرایی به طولی معادل ۹۵۸۴ کیلومتر در ۵۰۰ تکرار به‌وسیله GA (شکل ۶-الف) و طولی معادل ۸۸۵۷ کیلومتر در ۵۰۰ تکرار به‌وسیله ACO (شکل ۶-ب) منجر گردید.



ب) نمودار نتایج الگوریتم بهینه‌سازی کلونی مورچه‌ها

الف) نمودار نتایج حاصل از الگوریتم ژنتیک

شکل ۶. نتایج حاصل از اجرای الگوریتم ACO و GA برای حل TSP

۲-۳- الگوریتم پیشنهادی

با توجه به اینکه الگوریتم کلونی مورچه‌ها برای حل مسائل در مقیاس بزرگ باید بهبود یابد، در ادامه با الهام از برخی ویژگی‌های الگوریتم ژنتیک، الگوریتمی برای TSP پیشنهاد می‌شود. الگوریتم پیشنهادی که در زبان برنامه‌نویسی مطلب کدنویسی شده است، تحت عنوان حل باینری TSP در مقیاس بزرگ بر مبنای الگوریتم مورچهٔ بهبودیافته معرفی می‌گردد.

در الگوریتم پیشنهادی، ابتدا ماتریس هزینهٔ شهرها (در این تحقیق فاصله به‌عنوان هزینه انتخاب شده است) تشکیل می‌شود. در ادامه حداقل و حداکثر فاصله تعیین می‌گردد و مطابق روابط ذیل ماتریس فواصل در بازهٔ صفر تا یک نرمال می‌گردد. با توجه به اینکه تابع هزینهٔ این تحقیق فاصله است و با افزایش آن بهینگی تابع کاهش می‌یابد، از رابطه (۷) برای نرمال‌سازی ماتریس استفاده شد. این کار با هدف بی‌واحدکردن مقادیر تابع هزینه و قراردادن آن در بازه صفر تا یک انجام گردید، تا به مقادیر زیاد فاصله که میزان بهینگی کمتری دارند، مقادیر کوچک‌تری اختصاص یابد.

رابطه (۷)

$$n_{ij} = \frac{r_j^{\otimes}}{r_{ij}} \qquad \leftrightarrow \qquad r_j^{\otimes} = \min_i r_{ij}$$

در رابطه (۷)، r_{ij} عناصر ماتریس فاصله و $r_j^{\otimes}$ مینیمم مقدار فاصله است.

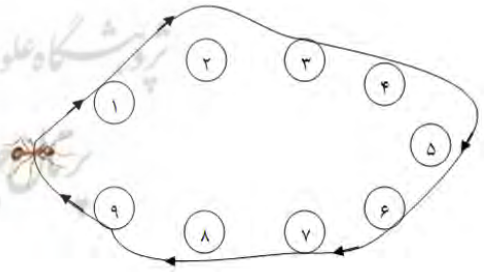
سپس پارامترهای مختلف الگوریتم مورچه به شرح جدول ۲ بهینه می‌گردد و مقادیر احتمال حرکت مورچه‌ها، فرمون برجای‌مانده، به روزرسانی محلی، مطابق مطالب بیان‌شده در قسمت ۱-۳ محاسبه می‌گردد. مقادیر پیشنهادی در جدول ۲، ویژهٔ مسئله تحقیق حاضر و برمبنای سعی و خطاست، که اعتبار آن و میزان حساسیت جواب حاصل از الگوریتم به آنها در بخش ۳-۱ ارزیابی خواهد شد.

در ادامه، پس از اجرای الگوریتم ACO، احتمال جابه‌جایی مورچه‌ها مطابق رابطه (۳) و به‌روزرسانی محلی مطابق رابطه (۴) انجام می‌شود و دنبالهٔ باینری شهرهایی که مورچه‌ها طی کرده‌اند، تشکیل می‌گردد. در الگوریتم پیشنهادی دنبالهٔ باینری، به‌جای رشتهٔ مسیر در هر تکرار و برای هر عضو جمعیت ذخیره می‌شود. در این ویژگی (که از الگوریتم ژنتیک الهام گرفته شده است) شهرهایی که مورچه‌ها طی کرده‌اند، مانند کروموزوم به‌شکل دنباله قرار می‌گیرند. در TSP رایج، دوره‌گرد بایستی تمام شهرها را طی کند و به شهر نخست بازگردد. در TSP این تحقیق، دوره‌گرد علاوه بر مورد فوق، حق انتخاب چند شهر را از میان تمامی شهرها و سپس طی‌کردن آنها را دارد که انتخاب‌شان برمبنای علایق و اولویت‌های کاربر صورت می‌گیرد.

جدول ۲. مقادیر پارامترهای الگوریتم مورچهٔ بهبودیافته

تعداد شهرها	این تحقیق برمبنای داده‌های واقعی مراکز استان‌های کشور و در چهار حالت مختلف (۳۰، ۲۵، ۲۰ و ۱۵ شهر) به بررسی بهبود الگوریتم و تأثیر المان‌های مؤثر می‌پردازد.
تعداد جمعیت مورچه‌های مصنوعی	در این تحقیق، تعداد جمعیت nبرابر تعداد شهرها فرض شد، که n عددی بین ۱ تا ۵ است. هرچه n بزرگ‌تر باشد، زمان دستیابی به جواب بهینه افزایش می‌یابد. برای دستیابی به جواب‌های بهتر، تعداد جمعیت سه‌برابر تعداد شهرها، و معادل ۹۰ نفر، فرض گردید.
ضریب تبخیر	در این تحقیق، مقدار ضریب تبخیر ثابت نیست و در هر تکرار افزایش می‌یابد. مقدار اولیهٔ ضریب تبخیر ۰/۱ فرض می‌شود، سپس در هر تکرار به میزان e^{-nt} افزایش می‌یابد، که t بیانگر تکرار tام و n ضریبی مثبت است. در این تحقیق n مساوی ۵ فرض شد (Dorigo and Blum, 2005; Dorigo and Gambardella, 1997a). قراردادن این رابطه و افزایش تبخیر در هر تکرار به دلیل خاصیت فرمون و تجمع فرمون در هر تکرار است.
تعداد تکرار	پس از تنظیم تعداد جمعیت و ضریب تبخیر مطابق مقادیر پیشنهادی، الگوریتم در تکرار کمتری به جواب بهینه دست می‌یابد (۲۰۰-۱۰۰ تکرار).
α , β , τ_0	این پارامترها مطابق مساوی $\alpha = 1, \beta = 2, \tau_0 = 1$ تنظیم شدند (Dorigo and Blum, 2005).

این مسئله می‌تواند کاربردهای فراوانی در صنعت گردشگری داشته باشد. برای ساختن دنبالهٔ باینری اولیه از رشته مسیر ساخته‌شده، در رشته‌ای که حاوی نود اول تا آخر است، به شهری که شخص آن را انتخاب کرده است و طی می‌کند، عدد یک و در غیر این صورت عدد صفر تعلق می‌گیرد، بدین ترتیب دنباله‌ای از صفر و یک‌ها ایجاد می‌شود (شکل ۷).



شکل ۷. نمایش باینری کد ۹ بیتی برای ۹ شهر
(۱۳۶۷۹ ← ۱۰۱۰۰۱۱۰۱)

سپس اولویت انتخاب نودها برای تسهیل در اعمال عملگر جهش به این دنبالهٔ باینری منتسب می‌گردد. به‌عنوان مثال برای کد باینری ساخته‌شده در شکل ۷ اولویت‌بندی اولیه برابر رشتهٔ ۱۰۲۰۰۳۴۰۵ است که معادل اولویت اول برای نود اول، اولویت دوم برای نود سوم، اولویت سوم برای نود ششم، اولویت چهارم برای نود هفتم و اولویت پنجم برای نود نهم است. مرحلهٔ ساختن دنبالهٔ باینری در الگوریتم پیشنهادی برای نمایش شهرهای انتخابی، اولویت طی‌کردن آنها و ترکیب با الگوریتم ژنتیک با هدف اعمال عملگرهای مختلف نظیر عملگر ادغام و جهش طراحی شده است. بدیهی است برای کدبرداری از نمایش باینری و دستیابی به رشته مسیر طی‌شده، عملیات فوق به‌صورت معکوس انجام شود.

با توجه به اینکه عملگر ادغام و جهش الگوریتم ژنتیک با جابه‌جایی تصادفی ترتیب نودها به بهبود سریع‌تر تابع هزینه می‌انجامد، در ادامه روی دنبالهٔ باینری ساخته‌شدهٔ مسیر و اولویت‌های منتسب به آن، ابتدا عملگر ادغام و سپس عملگر جهش مطابق مطالب قسمت ۱-۲ اعمال می‌گردد و اولویت نودها به‌طور تصادفی به‌منظور بهینه‌سازی تابع هزینه عوض می‌شود.

در ادامه، به‌روزرسانی کلی مطابق رابطه (۵) انجام می‌شود. در الگوریتم پیشنهادی، در آغاز الگوریتم معمول ACO با پارامترهای بهینه اجرا می‌شود و دنبالهٔ باینری شهرهای طی‌شده تشکیل می‌گردد، اما پیش از اعمال به‌روزرسانی کلی، دنبالهٔ طی‌شده به‌وسیله

مورچه‌ها از طریق عملگرهای ادغام و جهش الگوریتم ژنتیک بهبود می‌یابد، تا در هر تکرار به‌میزان بیشتری به بهینهٔ سراسری نزدیک‌تر شود و زمان اجرای الگوریتم کاهش یابد. با این‌کار، مشکل الگوریتم ACO برای تحلیل مسائل با ابعاد بزرگ تا حد زیادی حل می‌شود و زمان اجرای الگوریتم به‌شدت کاهش می‌یابد. سپس اجرای الگوریتم وارد تکرار بعدی می‌شود و پس از جابه‌جایی تمامی مورچه‌ها و رسیدن به ماکزیمم تکرار، اجرای الگوریتم به پایان می‌رسد. در نهایت تابع هزینهٔ بهینه و دنبالهٔ نودهای مسیر (با کدبرداری از رشته باینری تشکیل‌یافته) ارائه می‌گردد. نتایج حاصل از الگوریتم پیشنهادی نشان می‌دهد که زمان اجرای الگوریتم تا حد زیادی کاهش می‌یابد و جواب بهینه با درصد اطمینان بیشتری تعیین می‌شود. شبه‌کد الگوریتم پیشنهادی در شکل ۸ نشان داده شده است.

```
Procedure Ant colony system for TSP (one iteration)
Initialize pheromone trails
While termination condition does not meet do
Place each ant on initial node randomly
Normalize the cost function
Determine parameter values according to Table $
For i=1: n do (#nodes)
For k=1: m do (#ants)
Apply state transition rule (ACO)
End (ants)
Apply local update
Build binary suite for cities
End for (one iteration)
Genetic operator
Apply global update
End (while)
Decoding the variable values (cities sequences)
```

شکل ۸. شبه‌کد الگوریتم مورچهٔ بهبودیافته

۳- نتایج

نتایج حاصل از اجرای دو الگوریتم اصلی مورچه و ژنتیک روی یک کامپیوتر با سیستم عامل ویندوز ویستا، دو هسته و 2GB RAM، برای مراکز ۳۰ استان کشور، در حل TSP با تغییر پارامترها و در حالت‌های مختلف، به شرح زیر است (جدول‌های ۳ و ۴):

- در مقیاس بزرگ، برای مجموعه داده‌های گوناگون و در شرایط متفاوت، در اکثر موارد جواب الگوریتم مورچه از الگوریتم ژنتیک بهتر است.
- زمان اجرای الگوریتم مورچه در مقایسه با الگوریتم ژنتیک بیشتر است (در حدود چند ده ثانیه).
- پایداری جواب‌های حاصل از الگوریتم مورچه بیشتر است. براساس آزمایش‌های صورت‌گرفته، در تکرارهای مختلف الگوریتم ژنتیک، جواب‌های تولیدشده فاقد نزدیکی منطقی به یکدیگرند؛ درحالی‌که الگوریتم مورچه جواب‌های نزدیک‌تری تولید می‌کند. احتمال وجود جواب‌های بد در الگوریتم ژنتیک بسیار بالاتر از الگوریتم مورچه است که این امر احتمالاً به‌دلیل وجود فرمون روی مسیر و جلوگیری از انحراف سایر مورچه‌ها به جواب غلط است.
- بررسی دقیق المان‌های مؤثر در الگوریتم‌های مذکور نظیر تعداد شهرها، تعداد جمعیت، ضریب تبخیر و تعداد تکرار نشان می‌دهد که با تنظیم دقیق و صحیح این پارامترها، می‌توان در زمان کمتر به جواب‌های بهتری دست یافت. براساس یافته‌های این تحقیق نمی‌توان به‌طور قطع ادعا کرد که الگوریتم مورچه جواب بهتری در همه موارد تولید می‌کند. تنظیم پارامترهای دخیل در هر الگوریتم با توجه به شرایط مسئلهٔ مورد مطالعه برمبنای سعی و خطا یا نظر کارشناس ضروری است، تا بهترین مقادیر برای آنها تعیین گردد.

جدول ۳. نتایج الگوریتم مورچه با تغییر المان‌های مختلف

طول کوتاه‌ترین مسیر (کیلومتر)	زمان اجرای الگوریتم (ثانیه)	تعداد جمعیت	ضریب تبخیر	تعداد تکرار	تعداد شهرها
۸۸۵۷	۱۳۲	۶۰	۰.۰۱	۵۰۰	۳۰
۸۶۲۲	۱۸۰	۶۰	۰.۰۱	۸۰۰	۳۰
۸۶۲۲	۲۰۲	۹۰	۰.۰۱	۵۰۰	۳۰
۸۶۲۲	۸۰	۹۰	۰.۱	۲۰۰	۳۰
۱۰۳۰۰	۶۶	۶۰	۰.۰۱	۲۰۰	۲۵
۸۵۱۳	۱۹۱	۹۰	۰.۱	۱۵۰	۲۵
۷۳۵۸	۵۲	۹۰	۰.۱	۲۰۰	۱۵

جدول ۴. نتایج الگوریتم ژنتیک با تغییر المان‌های مختلف

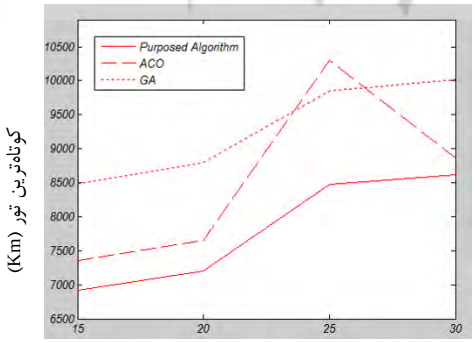
طول کوتاه‌ترین مسیر (کیلومتر)	زمان اجرای الگوریتم (ثانیه)	تعداد جمعیت	تعداد تکرار	تعداد شهرها
۱۰۰۱۹	۴۰	۶۰	۵۰۰	۳۰
۹۱۱۹	۶۵	۶۰	۸۰۰	۳۰
۹۵۸۴	۱۰۰	۶۰	۱۲۰۰	۳۰
۹۲۳۸	۱۶	۹۰	۲۰۰	۳۰
۹۸۵۳	۱۶	۹۰	۲۰۰	۲۵
۲۶۶۴	۱۷	۹۰	۲۰۰	۱۰
۹۴۶۰	۱۲۰	۶۰	۱۵۰۰	۳۰
۸۷۲۸	۱۳۴	۶۰	۲۰۰۰	۳۰
۸۶۲۸	۱۶۳	۹۰	۱۵۰۰	۳۰

در الگوریتم پیشنهادی این تحقیق، تنظیم پارامترهای الگوریتم مورچه مطابق موارد پیشنهادی در جدول ۲ انجام می‌گیرد، به بهبود اجرای الگوریتم مورچه می‌انجامد (نتایج بخش ۳-۱). نرمال‌کردن تابع هزینه و ذخیره‌سازی دنبالهٔ باینری و اولویت‌های شهرهای طی‌شده (به‌جای ذخیرهٔ خود شهرها به‌همراه تابع هزینهٔ اصلی) و اجرای عملگر جهش روی آنها سبب می‌شود که در هر تکرار الگوریتم مورچهٔ اصلی، بهینه‌سازی مضاعفی صورت بگیرد و منجر به دستیابی سریع‌تر به جواب بهینه، کاهش زمان لازم برای اجرای الگوریتم و حافظهٔ لازم برای ذخیرهٔ شهرها و تابع هزینهٔ مربوط به آن شود. این ادعا را می‌توان با استفاده از نتایج اجرای الگوریتم پیشنهادی و مقایسهٔ آن با نتایج الگوریتم اصلی اثبات کرد (مقایسهٔ نتایج جدول‌های ۳ و ۵).

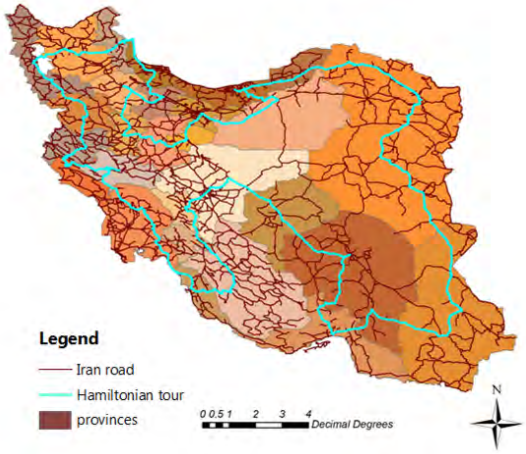
جواب الگوریتم مورچهٔ بهبودیافته در مدت زمان ۴۰ ثانیه، برای ۳۰ شهر ۸۶۲۲ کیلومتر است. این اعداد در مقایسه با نتایج حاصل از الگوریتم اولیه ۲۰۲، ۱۸۰، ۱۳۲ و ۸۰ ثانیه، برتری الگوریتم پیشنهادی را نشان می‌دهد. بهینگی و برتری الگوریتم پیشنهادی برای ۲۵ و ۱۵ شهر آشکارا بیان می‌کند که این برتری هم در زمان اجرای الگوریتم و هم در بهبود جواب بهینه است (شکل ۹). شایان ذکر است که زمان مربوط به اجرای الگوریتم‌ها نسبی است و با تعویض سیستم، تغییر خواهد کرد و تنها با اجرای هر سه الگوریتم روی یک سیستم، مقایسهٔ زمانی معنادار خواهد بود.

جدول ۵. نتایج اجرای الگوریتم مورچهٔ بهبودیافته برای داده‌های مختلف

دنبالهٔ باینری ۳۰ بیتی اولیهٔ طی‌شده به‌وسیله مورچه برای مراکز استان‌های کشور (مبین رشته مسیر طی‌شده)	طول کوتاه‌ترین مسیر به کیلومتر	زمان اجرای الگوریتم به ثانیه	تعداد جمعیت	ضریب تبخیر اولیه	تعداد تکرار	تعداد شهرها
۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱	۸۶۲۲	۴۰	۹۰	۰/۱	۱۰۰	۳۰
۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۱۰۰۰۰۰	۸۴۷۹	۶۵	۹۰	۰/۱	۱۵۰	۲۵
۱۱۱۱۱۱۰۰۰۰۰۰۰۰۱۱۱۱۱۱۱۱۱۱۱۱۱۰	۷۲۰۵	۳۵	۹۰	۰/۱	۱۰۰	۲۰
۰۱۱۰۱۰۱۰۱۱۰۰۰۰۱۰۰۱۱۰۰۱۰۰۱۱۰۱۱۱	۶۹۱۲	۳۱	۹۰	۰/۱	۱۰۰	۱۵



شکل ۹. مقایسهٔ کوتاه‌ترین تور محاسبه‌شده به‌وسیلهٔ الگوریتم پیشنهادی ACO و GA



شکل ۱۰. نمایش کوتاه‌ترین طول انتخابی به‌وسیله الگوریتم مورچهٔ بهبودیافته بین مراکز استان‌های کشور

شکل ۱۱. پنجرهٔ برنامهٔ راهنمایی گردشگر در بازدید از مراکز استان‌های کشور، به‌وسیله الگوریتم مورچهٔ بهبودیافته

در قسمت پایانی این تحقیق برای تعیین راه‌های ارتباطی سازندهٔ کوتاه‌ترین تور بین مراکز استان‌های کشور به‌وسیله الگوریتم مورچهٔ بهبودیافته، لایه‌های موردنظر وارد نرم‌افزار ArcMap می‌شود. نتایج انجام این آنالیز در شکل ۱۰ آمده است.

با توجه به تولید جواب‌های بهینه به‌وسیله الگوریتم پیشنهادی برای حل TSP و برای دستیابی به اهداف سامانهٔ اطلاعات مکانی در افزایش و بهبود خدمات به کاربران، در اینجا برنامه‌ای برای راهنمایی گردشگران طراحی و معرفی می‌شود (شکل ۱۱). ابتدا گردشگر تعداد شهرها و نام شهرهایی را که می‌خواهد از آنها بازدید کند، برمی‌گزیند و سپس کوتاه‌ترین طول سفر برای شهرهای انتخابی به‌وسیلهٔ الگوریتم پیشنهادی محاسبه و به کاربر اعلام می‌شود. پس از آن، شهری که از نظر تغییر در طول تور کمترین و از لحاظ جاذبه‌های توریستی بیشترین وزن را دارد (ولی کاربر آن را انتخاب نکرده است)، از طریق برنامه به وی پیشنهاد می‌شود. بدین ترتیب، شهری که مقدار مجموع مسافت طی‌شدهٔ گردشگر را چندان تغییر نمی‌دهد ولی از نظر جاذبه‌های گردشگری و علایق کاربر در اولویت بالاتری قرار دارد، تعیین و به کاربر پیشنهاد می‌شود. افزایش طول تور ناشی از اضافه‌شدن این شهر به شهرهای انتخابی به‌وسیله گردشگر، به‌کمک الگوریتم مورچهٔ بهبودیافته محاسبه و به کاربر اعلام می‌شود.

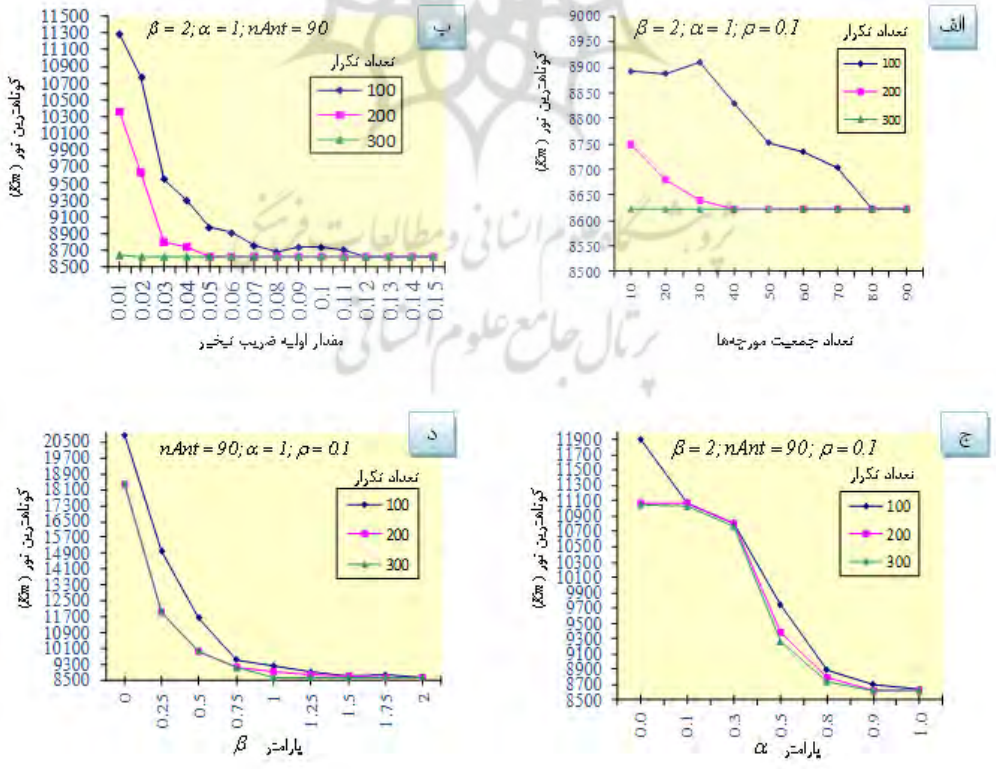
۳-۱- آنالیز حساسیت

با توجه به تأثیرگذاری پارامترهای مختلف در الگوریتم‌های فراابتکاری، آنالیز حساسیت از مهم‌ترین ابزارهای ارزیابی این الگوریتم‌ها به‌شمار می‌آید. در این بخش، تحلیل حساسیت طول کوتاه‌ترین مسیر به‌دست‌آمده از الگوریتم پیشنهادی نسبت به تغییر مقادیر مفروض در این تحقیق برای ضریب تبخیر، تعداد مورچه‌های مصنوعی و پارامتر α و β (برحسب رابطه (۳)) صورت می‌گیرد.

شکل ۱۲-الف نشان می‌دهد که با ثابت نگه‌داشتن ضریب تبخیر و پارامتر α و β، در صورتی که تعداد مورچه‌ها از ۱۰ تا ۹۰ نفر افزایش یابد، طول کوتاه‌ترین مسیر کاهش می‌یابد و بهینه می‌گردد. مشابه این حالت با ضریب تبخیر و ثابت نگه‌داشتن جمعیت مورچه‌ها و پارامتر α و β در شکل ۱۲-ب، تغییر پارامتر α و ثابت نگه‌داشتن جمعیت مورچه‌ها، ضریب تبخیر و β در شکل ۱۲-ج، و تغییر پارامتر β و ثابت نگه‌داشتن جمعیت مورچه‌ها، ضریب تبخیر و α در شکل ۱۲-د نیز رخ می‌دهد. بازهٔ تغییرات در زمان تغییر تعداد مورچه‌ها بسیار کمتر از سه حالت دیگر است، از این رو حساسیت بهینگی تابع هزینه به تغییرات پارامتر β، α و ضریب تبخیر بیشتر است و انتخاب نادقیق پارامتر β و α حتی در تکرارهای بالا نیز جبران‌ناپذیر است.

میزان تغییر طول کوتاه‌ترین تور در تعداد تکرارهای کم (۱۰۰ تکرار) در هر چهار حالت بیشتر است و با افزایش تعداد تکرارها میزان این تغییر کاهش می‌یابد. در تعداد تکرارهای زیاد (۳۰۰ تکرار و بیشتر) حساسیت جواب بهینهٔ حاصل از الگوریتم پیشنهادی به تغییر پارامترهای مذکور کاهش می‌یابد.

بنابراین می‌توان نتیجه گرفت که پارامترهای مفروض در این تحقیق $\alpha=۱;\ \beta=۲;\ nAnt=۹۰;\ \rho_{\circ}=۰/۱$ در ۱۰۰ تکرار بهینه هستند. همچنین می‌توان با افزایش تعداد تکرارها از حساسیت تغییر طول کوتاه‌ترین تور با تغییر پارامترهای مذکور کاست و به جواب بهینه با درصد اطمینان بیشتری دست یافت.



شکل۱۲. نتایج آنالیز حساسیت

۴- بحث و نتیجه‌گیری

سیستم‌های بسیاری در طبیعت وجود دارند که از تعدادی عامل غیرهوشمند یا باهوش اندک تشکیل شده‌اند. به‌رغم وجود نداشتن ساختار کنترل متمرکز، تعاملات ساده میان عوامل باعث به‌وجود آمدن نوعی رفتار جمعی هوشمندانه می‌شود. با الهام از این سیستم‌ها، روش‌هایی برای حل مسائل بهینه‌سازی ابداع و در بسیاری موارد با موفقیت به کار گرفته شده‌اند، که ازجمله مسائل بهینه‌سازی ترکیبی، می‌توان به TSP اشاره کرد. مزیت عمده این نوع از هوشمندی علاوه بر کاهش حجم محاسبات و تمامی مزایای مهندسی دیگر، قابلیت تعمق زیستی آنهاست، به این معنی که طبیعت آنها را طی میلیون‌ها سال به‌عنوان روش بهینه آزمون و انتخاب کرده است.

در این پژوهش از میان الگوریتم‌های تقریبی مختلف، دو الگوریتم ژنتیک و بهینه‌سازی کلونی مورچه‌ها برای حل مسئله فروشنده دوره‌گرد و به‌منظور مقایسه انتخاب گردید. نتایج اجرای این دو الگوریتم در حالت‌های مختلف نشان داد که در حالت کلی و شرایط مساوی، جواب الگوریتم بهینه‌سازی کلونی مورچه‌ها از الگوریتم ژنتیک بهتر و پایدارتر است، اما در مقیاس‌های بزرگ کارایی ندارد. بهبود زمان و جواب اجرای این الگوریتم در مقیاس بزرگ نیازمند نوآوری است، که با اضافه‌کردن ایده‌هایی معتبر و جدید حاصل می‌گردد. با استفاده از نقاط قوت الگوریتم ژنتیک ـ در قالب الگوریتمی ترکیبی ـ تا حدودی می‌توان مشکلات اجرای الگوریتم بهینه‌سازی کلونی مورچه‌ها در مقیاس بزرگ را برطرف کرد.

در این تحقیق، با تنظیم دقیق پارامترهای مؤثر در الگوریتم بهینه‌سازی کلونی مورچه‌ها نظیر تعداد جمعیت، تعداد تکرار، تعداد شهرها و ضریب تبخیر و با الهام از برخی ویژگی‌های الگوریتم ژنتیک (ترکیب عملگرهای الگوریتم ژنتیک)، الگوریتمی برای حل TSP در مقیاس بزرگ پیشنهاد گردید. الگوریتم پیشنهادی در زبان برنامه‌نویسی مطلب کدنویسی شد و تحت عنوان حل باینری TSP در مقیاس بزرگ برمبنای الگوریتم مورچهٔ بهبودیافته معرفی گردید.

در الگوریتم پیشنهادی، تنظیم پارامترهای الگوریتم ACO مطابق جدول ۲، نرمال‌کردن تابع هزینه، اختصاص کد باینری به دنبالهٔ شهرهای طی‌شده به‌همراه اولویت آنها و اعمال عملگر ادغام و جهش الگوریتم ژنتیک در رشته مسیر انتخابی هر مورچه در هر تکرار، به بهبود نتایج، کاهش زمان لازم برای اجرای الگوریتم و کاهش حافظهٔ لازم برای ذخیرهٔ شهرها و بهینگی تابع هزینهٔ مربوط به آن در مقایسه با الگوریتم اصلی ACO انجامید.

نتایج آنالیز حساسیت، کاهش بیش از ۵۰ درصد در زمان اجرا و بهبود ۵ الی ۲۵ درصدی در نتایج حاصل از الگوریتم پیشنهادی در مسیریابی استان‌های کشور (برمبنای شکل ۹ و نتایج قسمت ۳) دلیل بر این ادعاست. نتایج آنالیز حساسیت حاکی از آن است که حساسیت الگوریتم بهینه‌سازی کلونی مورچه‌ها به پارامتر β، α و ضریب تبخیر بیشتر و به تغییر تعداد جمعیت مورچه‌ها ناچیز است، اما این حساسیت با افزایش تعداد تکرارها تا حد زیادی کاهش می‌یابد و درصد اطمینان به جواب حاصل از الگوریتم بالا می‌رود.

پس از بررسی و تنظیم دقیق پارامترهای مؤثر، نرمال‌سازی تابع هزینه برای بی‌مقیاس‌کردن آن و اجرای سریع الگوریتم، آشکار شد که ذخیرهٔ دنبالهٔ باینری به‌جای رشته مسیر انتخابی مورچه، اجرای عملگر ادغام و جهش با جابه‌جایی تصادفی اولویت‌ها به امید بهینه‌سازی هرچه بیشتر تابع هزینه در هر تکرار و کاهش زمان اجرای الگوریتم برای TSP به‌ویژه در مقیاس بزرگ از مهم‌ترین دلایل برتری الگوریتم پیشنهادی بر الگوریتم اصلی مورچه و ژنتیک در مسیریابی میان مراکز استان‌های کشور در تعداد شهرهای مختلف است (شکل ۸).

با توجه به بهینگی حاصل در مسیریابی استان‌های کشور به‌وسیله الگوریتم پیشنهادی، بخش پایانی این تحقیق سامانه‌ای را برای راهنمایی گردشگران پیشنهاد

می‌کند که هدفش افزایش خدمت در سامانهٔ اطلاعات مکانی است. این کار با هدف ایجاد انگیزه در علاقه‌مندان به‌منظور طراحی سامانهٔ حامی تصمیم‌گیری مکانی برای گردشگران به‌کمک ترکیب روش‌های مختلف تصمیم‌گیری چندمعیاره و الگوریتم‌های فراابتکاری انجام گردید. در این تحقیق پیشنهاد می‌شود که در دو مرحله، ابتدا وزن‌دهی به مناطق برمبنای علایق کاربر با روش‌های مناسب تصمیم‌گیری انجام شود، سپس مسیریابی و هدایت کاربران به‌کمک الگوریتم‌های فراابتکاری صورت گیرد. بدین ترتیب انتخاب شهرها به‌طور مستقیم به‌وسیله کاربر صورت نمی‌گیرد، بلکه برمبنای علایق وی که از طریق واسط کاربر دریافت می‌شود و به‌کمک اولویت‌بخشی با روش‌های تصمیم‌گیری، انتخاب انجام می‌شود.

موضوعاتی از این دست می‌تواند زمینهٔ پژوهش‌های آتی را شکل دهد: افزودن سایر متغیرها نظیر جاذبه‌های توریستی به تابع هزینه، انجام عملگر ادغام و جهش با سایر روش‌های موجود و مقایسهٔ نتایج، استفاده از ایده‌ها و روش‌های دیگر برای تنظیم پارامترهای الگوریتم بهینه‌سازی کلونی مورچه‌ها، مقایسهٔ نتایج الگوریتم‌ها به‌کمک سایر شاخص‌ها نظیر NFE[1]، و ترکیب این الگوریتم با سایر الگوریتم‌های تقریبی به‌منظور بهره‌گیری از مزایای آنها و بهبود این الگوریتم برای حل مسائلی با مقیاس بزرگ‌تر.

۵- منابع

Beckers, R., Deneubourg, J.L. and Goss, S., 1992, **Trails and U-turns in the Selection of the Shortest Path by the Ant Lasius Niger**, Journal of Theoretical Biology, Vol. 159, PP. 397–415.

Blum C., Roli, A., 2003, **Metaheuristics in Combinatorial Optimization: Overview and Conceptual Comparison**, ACM Computing Surveys, Vol. 35, NO. 3, PP. 268-308.

Chong, Y.N., 2001, **Heuristic Algorithms for Routing Problems**, Ph.D. Thesis, School of Mathematics and Statistics, Curtin University of Technology, Australia.

Clerc, M., 2004, **Discrete Particle Swarm Optimization, Illustrated by the Traveling Salesman Problem**, in New Optimization Techniques in Engineering, Springer, Heidelberg, PP. 219-239.

Dorigo, M., Birattari, M., StÜzle, T., 2006, **Ant Colony Optimization**, IEEE Computational Intelligence Magazine 1 (4), PP. 28–39.

Dorigo, M., Blum, C., 2005, **Ant Colony Optimization Theory: A Survey**, Theoretical Computer Science, 344, PP. 243–278.

Dorigo, M., Gambardella, L.M., 1997a, **Ant Colony System: A Cooperative Learning Approach to the Traveling Salesman Problem**, IEEE Trans. on Evolutionary Computation, Vol. 1, PP. 53-66.

Dorigo, M., Gambardella, L.M., 1997b, **Ant Colonies for the Traveling Salesman Problem**, BioSystems 43, PP. 73-81.

Dorigo, M., Stűzle, T., 2004, **Ant Colony Optimization**, MIT Press, USA, July.

Elbeltagi, E., Hegazy, T., Grierson, D., 2005, **Comparison Among Five Evolutionary Based Optimization Algorithms**, Adv. Eng. Inform. Vol. 19, PP. 43–53.

Ghoseiri, K., and Sarhadi, H., 2009, **Determination of the Shortest Hamiltonian Tour for Iranian Network Using Ant Colony System and Local Search**, Transportation Journal, Vol 6, pp. 149-161. (In Persian).

Grotschel, M., Holland, O., 1991, **Solution of Large-scale Symmetric Traveling Salesman Problems**, Mathematical Programming, 51, PP. 141–202.

Guangdong, H., Ping, L., Qun, W., 2007, **A Hybrid Metaheuristic ACO-GA With an Application in Sports Competition Scheduling**, Eighth ACIS International Conference on Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, IEEE, Computer Society, DOI 10.1109/SNPD. 2007.402, PP. 611-616.

Gutin, G., Punnen, A. P., 2002, **The Traveling Salesman Problem and Its Variations**, Combinatorial Optimization, Vol. 12, Kluwer Academic.

Hashemi, M., Alesheikh, A. and Behzadi, S., 2009, **Path Finding in Roads Network with Constrained Genetic Algorithm**, Journal of GeoInformatics, Vol. 1, PP. 25-35.

Hosseinali, F., Alesheikh, A. and Rajabi, M.A., 2009, **Evaluation of Various Methods for Weighting Spatial Information in GIS (Case Study: Mineral Potential Mapping)**, Journal of RS/GIS, Vol. 1, PP. 75-90.

Hung, K.S., Su, S.F., Lee, Z.J., 2007, **Improving Ant Colony Optimization Algorithms for Solving Traveling Salesman Problems**, Journal of Advanced Computational Intelligence and Intelligent Informatics 11 (4) PP.433–434.

Kanan, H. R., Faez, K., Taheri, S. M., 2007, **Feature Selection Using Ant Colony Optimization (ACO): A New Method and Comparative Study in Application of Face Recognition System**, Lecture Notes in Computer Science, 4597, PP. 63–76.

Khan teymori, P., 2008, **Genetic Algorithm and Solving of the Traveling Salesman Problem**, ICT conference of computer department, Iran University of Science and Technology, Tehran, Iran.

Kos, A., Nagórny, Z., 2009, **Application of a Modified Hopfield Network to the Traveling Salesman Problem**, Proceedings of the 1st Conference Tools of Information Technology, Rzeszów, Poland, Vol 6(4), PP. 58 – 66.

Lee, Z.J., Su, Sh.F., Chuang, Ch.Ch., Liu, K.H., 2008, **Genetic Algorithm With Ant Colony Optimization (GA-ACO) for Multiple Sequence Alignment**, Applied Soft Computing, 8, PP. 55–78.

Li, N., Wang, Sh., Li, Y., 2011, **A Hybrid Approach of GA and ACO for VRP**, Journal of Computational Information Systems, 7(13), PP. 4939-4946.

Misevicius A., Blazauskas T., Blonskis J., 2004, **An overview of Some Heuristic Algorithms for Combinatorial Optimization Problems**, In: ISSN 1392-124X Information technologijos ir valdymas, Nr. 1(30).

Mohammadi, N., Malek, M.R. and Alesheikh, A., 2009, **Evaluation of Location/Allocation Methods in a Continuous Space Using Genetic Algorithm**, International Review on

Computers and Software. Vol. 4, No. 6, PP. 743-748.

Mullen, R.J., Monekosso, D., Barman, S., Remagnino, P., 2009, **A Review of Ant Algorithms**, Expert Systems, 36, PP. 9608–9617.

Nemati, Sh., Basiri, M.E., Ghasem-Aghaee, N., Hosseinzadeh Aghdam, M., 2009, **A novel ACO–GA Hybrid Algorithm for Feature Selection in Protein Function Prediction**, Expert Systems with Applications, 36, PP. 12086–12094.

Paralos P.M., Resende, M.G.C., 2002, **Handbook of Applied Optimization**, Oxford University Press, PP. 616-624.

Ravagnani, M.A.S.S., Silva, A.P., Arroyo, P.A. Constantino, A.A., 2005, **Heat Exchanger Network Synthesis and Optimisation Using Genetic Algorithm**, Applied Thermal Engineering, 25 (7), PP. 1003–1017.

Sheta, A., Turabieh, H., 2006, **A Comparison Between Genetic Algorithms and Sequential Quadratic Programming in Solving Constrained Optimization Problems**, ICGST International Journal on Artificial Intelligence and Machine Learning (AIML), 6(1), PP. 67–74.

Socha, K., Dorigo, M., 2008, **Ant Colony Optimization for Continuous Domains**, European Journal Research, 185, PP. 1155–1173.

Yang, Z., Yu, B., Cheng, C., 2007, **A Parallel Ant Colony Algorithm for Bus Network Optimization**, Computer-Aided Civil and Infrastructure Engineering, 22, PP. 44–55.

پژوهشگاه علوم انسانی و مطالعات فرهنگی
پرتال جامع علوم انسانی