

حل مساله فروشنده دوره گرد با استفاده از الگوریتم ابتکاری

همایون موتمنی*

استادیار، دانشگاه آزاد اسلامی واحد ساری، گروه مهندسی کامپیوتر، ساری، ایران

رسید مقاله: ۱۰ بهمن ۱۳۹۲

پذیرش مقاله: ۱۸ خرداد ۱۳۹۳

چکیده

مساله فروشنده دوره گرد به عنوان یکی از مسایل پرکاربرد در علوم کامپیوتر و نیز در حوزه بهینه سازی ترکیبی شناخته شده و جزء مسایل ان پی کامل (NP-complete) می باشد. بنابراین نمی توان از الگوریتم های قطعی برای بهبود آن استفاده نمود. در این مقاله یک الگوریتم ابتکاری به نام الگوریتم جستجوی تصادفی تقلید نیروی گرانشی، برای حل مسئله فروشنده دوره گرد پیشنهاد شده است. این الگوریتم بر پایه مفاهیم جستجوی تصادفی، دو تا از چهار پارامتر اصلی سرعت و نیروی گرانشی در فیزیک استفاده می کند. الگوریتم پیشنهادی را GELSTSP نامیده و برای تصدیق کارایی الگوریتم پیشنهادی، آن را پیاده سازی نموده و با چندین نمونه از کتابخانه استاندارد TSPLIB مقایسه کرده ایم. نتایج حاصل از شبیه سازی الگوریتم پیشنهادی نشان دهنده کارایی مناسب این الگوریتم می باشد.

کلمات کلیدی: مساله فروشنده دوره گرد، سرعت، قانون نیوتن، نیروی گرانشی.

۱ مقدمه

مساله فروشنده دوره گرد (TSP) یکی از مسایل مهم در تئوری پیچیدگی محاسباتی الگوریتم ها و یکی از مسایل NP-Complete محسوب می شود [۱]. مساله فروشنده دوره گرد تعمیم یافته مساله مشهور سیکل همیلتونی است. فرض کنید یک گراف کامل داریم که هر یال یک هزینه و وزن صحیح نامنفی دارد. در مساله فروشنده دوره گرد، رأس های گراف معادل شهرها و یال های گراف معادل مسیرهای بین شهرها است. فروشنده باید با شروع از یک شهر به عنوان مبدأ، تمامی شهرهای دیگر را دقیقاً یک بار ملاقات کرده و به شهر مبدأ باز گردد، به طوری که هزینه کل تور حداقل باشد [۲-۵]. در واقع به دنبال یک دور همیلتونی بهینه هستیم این مساله را می توان با نوشتن همه دورهای همیلتونی ممکن با نقطه شروع و پایان از رأس و محاسبه کل مسافت پیموده شده برای هر دور حل کرد. اما این کار در عمل برای حتی تعداد کم شهرها بسیار زمان بر است به همین دلیل از روش های پویا برای

* عهده دار مکاتبات

پست الکترونیکی: motameni@iausari.ac.ir

حل این مساله استفاده می شود و با توجه به اینکه روش پویا برای حل این مساله برای تعداد شهرهای زیاد زمانبر می باشد، بنابراین باید از الگوریتم های ابتکاری برای حل این مساله استفاده شود. تاکنون روش های ابتکاری مختلفی برای حل مساله فروشنده دوره گرد بکار گرفته شده اند، که عبارت اند از الگوریتم ژنتیک، الگوریتم تپه نوردی، الگوریتم مورچگان و... در این مقاله می خواهیم مساله فروشنده دوره گرد متقارن را با استفاده از الگوریتم جستجوی گرانشی حل نماییم. ساختار مقاله به صورت زیر سازماندهی شده است: در بخش ۲ به بیان مسایل آن پی کامل پرداختیم. در بخش ۳ به شرح شرح مساله فروشنده دوره گرد پرداخته و در بخش ۴ به جزئیات بیشتر الگوریتم جستجوی تقلید نیروی گرانشی برای مساله فروشنده دوره گرد پرداخته و الگوریتم جدید را GELSTSP نامیدیم. در بخش ۵ مراحل الگوریتم پیشنهادی و در بخش ۶ نتایج شبیه سازی را نشان دادیم و با نتیجه گیری مقاله را به پایان رساندیم.

۲ مسایل NP-Complete

مجموعه "ان پی کامل" شامل چند هزار مساله مختلف با کاربردهای فراوان است که تاکنون برای آن ها راه حل سریع و قابل انجام در زمان معقول پیدا نشده است و به احتمال زیاد در آینده نیز یافت نخواهد شد. این که راه حل سریعی برای آن ها وجود ندارد هم اثبات شده است. البته ثابت شده است که اگر فقط برای یکی از این مساله ها راه حل سریعی پیدا شود این راه حل موجب حل سریع بقیه مساله ها خواهد شد. البته احتمال پیدا شدن چنین الگوریتمی ضعیف است. منظور از راه حل سریع آن است که زمان اجرا با اندازه ورودی مساله به صورت چندجمله ای رابطه داشته باشد. برخی از این مسایل عبارت اند از: مساله فروشنده دوره گرد، مساله مسیر همیلتونی، مساله کوله پشتی، مساله رنگ آمیزی گراف و غیره [۶-۸]. راه های معمول مقابله با چنین مسایلی عبارت اند از:

- طراحی الگوریتم هایی برای پیدا کردن جواب های دقیق که استفاده از این روش فقط برای مسایل با اندازه کوچک صورت می گیرد.
- استفاده از الگوریتم های مکاشفه ای یا ابتکاری که جواب هایی به دست می دهد که احتمالاً درست هستند.
- پیدا کردن زیرمساله هایی از مساله یعنی تقسیم مساله به مساله های کوچک تر تا بتوان الگوریتم های مکاشفه ای بهتر و دقیق تری ارائه کرد.

۳ شرح مساله فروشنده دوره گرد

مساله فروشنده دوره گرد به انگلیسی (Travelling salesman problem=TSP) مساله ای مشهور است که ابتدا در سده ۱۸ مسایل مربوط به آن توسط ویلیام همیلتون و توماس کرکمن مطرح شد و سپس در دهه ۱۹۳۰ شکل عمومی آن به وسیله ریاضیدانانی مثل کارل منگر از دانشگاه هاروارد و هاسلر ویتنی از دانشگاه پرینستون مورد مطالعه قرار گرفت.

شرح مساله بدین صورت است: تعدادی شهر وجود دارد به طوری که هزینه رفتن مستقیم از یک شهر به شهر دیگر را می‌دانیم. مطلوب است کم‌هزینه‌ترین مسیری که از یک شهر شروع شود و از تمامی شهرها دقیقاً یکبار عبور کند و به شهر شروع بازگردد. تعداد جواب‌های حل مساله، برابر است با $(n-1)!/2$ برای $n > 2$ که n تعداد شهرها می‌باشد. در واقع این عدد برابر است با تعداد دورهای همیلتونی در یک گراف کامل با n رأس. مساله فروشنده دوره گرد متقارن به صورت زیر به روش ریاضی فرموله می‌شود [۹]:

$$\begin{aligned} \min \quad & \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \\ \text{such that} \quad & \sum_{i=1}^n x_{ij}, \quad j = 1, \dots, n, \\ & \sum_{j=1}^n x_{ij} = 1, \quad i = 1, \dots, n, \\ & \sum_{i \in P} \sum_{j \in P, i \neq j} x_{ij} \leq |P| - 1, \quad \forall P \subset \{1, 2, \dots, n\}, 2 \leq |P| \leq n - 2 \\ & \text{(sub tour elimination constraint)} \\ & x_{ij} \in \{0, 1\}, \quad i = 1, \dots, n, \quad j = 1, \dots, n. \end{aligned}$$

سه روش کلی برای کد کردن راه حل‌های مساله TSP ارایه شده است که در الگوریتم‌های مختلفی قابل استفاده هستند. این راه حل‌ها عبارت‌اند از:

- نمایش جواب به صورت رشته گسسته جایگشتی که در الگوریتم‌های زیر قابل استفاده است: الگوریتم‌های ژنتیک، شبیه سازی تبرید، جستجوی ممنوعه، بهینه سازی کلونی مورچگان و سایر الگوریتم‌های بهینه سازی گسسته.
- نمایش جواب به صورت کلیدهای تصادفی که در الگوریتم‌های زیر قابل استفاده است: الگوریتم‌های ژنتیک، بهینه سازی ازدحام ذرات، الگوریتم رقابت استعماری، تکامل تفاضلی.
- نمایش جواب به شکل ماتریس‌های شبیه فرومون که توسط تمامی الگوریتم‌های اشاره شده در مورد (ب) قابل استفاده می‌باشد.

باقر زارعی و محمدرضا میبیدی [۱۰] یک روش ترکیبی برای حل مساله فروشنده دوره گرد با استفاده از اتوماتای یادگیر و الگوریتم ژنتیک ارایه داده‌اند که به طور همزمان برای جستجو در فضای حالت استفاده می‌شود. اتوماتای یادگیر و الگوریتم ژنتیک هر دو ابزاری جستجوی عمومی می‌باشند که برای حل بسیاری از مسایل NP-Complete از جمله افراز اشیا، افراز گراف، بهینه سازی صفحه کلید و پیدا کردن ساختار بهینه شبکه عصبی به کار برده شده است. در این مقاله یک الگوریتم ترکیبی برای حل مساله فروشنده دوره گرد پیشنهاد شده است. این الگوریتم از دو روش الگوریتم‌های ژنتیک و اتوماتای یادگیر به طور همزمان برای جستجو در فضای حالت استفاده می‌نماید. نشان داده شده است که با استفاده همزمان از اتوماتای یادگیر و الگوریتم ژنتیک در فرایند جستجو سرعت رسیدن به جواب افزایش چشمگیری پیدا می‌کند. یادگیری در اتوماتاهای یادگیر انتخاب یک

اقدام بهینه از میان یک مجموعه از اقدام‌های مجاز آتوماتا می‌باشد. این اقدام روی یک محیط تصادفی اعمال می‌شود و محیط به این اقدام اتوماتا بوسیله یک پاسخ تصادفی از مجموعه پاسخ‌های مجاز جواب می‌دهد. پاسخ محیط به صورت آماری به اقدام آتوماتا وابسته است. اصطلاح محیط شامل اجتماع تمام شرایط خارجی و تاثیرات آن‌ها روی عملکرد آتوماتا می‌باشد. شبه کد ترکیبی که برای حل استفاده شده به صورت زیر می‌باشد.

کیوان قیصری و حسن سرحدی در [۱۱] به یافتن کوتاه‌ترین تور همیلتونی با استفاده از ترکیب الگوریتم سیستم اجتماع مورچه‌ها و جستجوی محلی پرداخته‌اند. در این مقاله با ترکیب الگوریتم سیستم اجتماع مورچه‌ها و جستجوی محلی (ACS+2-OPT) رویکردی موفقیت آمیز در حل این مساله ارائه داده است. الگوریتم جستجوی محلی بکار رفته در این ترکیب الگوریتم 2-OPT است. در این الگوریتم ابتدا تور در محل دو یال شکسته می‌شود. لبه‌های این دو یال به صورت ضربدری به هم متصل می‌شوند. اگر این تغییر منجر به بهبود و کاهش در تابع هدف شود به تور اعمال می‌شود. در غیر این صورت این تغییر در نظر گرفته نمی‌شود و به دنبال شکستن دو یال دیگر از این تور می‌رویم. این کار آنقدر تکرار می‌شود تا با شکستن هیچ دو یالی دیگر نتوان تابع هدف را کاهش داد. در این حالت به کمینه محلی رسیده‌ایم.

مجید یوسفی و فرزاد دیده‌ور و فرهاد رحمتی [۱۲] از روش رقابت استعماری اصلاح شده برای حل این مساله ارائه کرده‌اند. این الگوریتم برای رسیدن به جواب از سیستم‌های تکاملی استفاده می‌کند و از نظر قابلیت تعمیم پذیری در مسایل موفق می‌باشد و در بسیاری از کاربردها ظاهر می‌شود. در این مسایل با توجه به اینکه الگوریتم اطلاعات اندکی مانند فضای جستجو و تعریف جواب شدنی برای مساله اصلی دارد می‌تواند با ایجاد جواب‌های تصادفی در مسیر یافتن جواب هاب بهتر حرکت کند و در فضای جستجو تا حد امکان به خوبی پیشروی نماید. محمد باقر دولتشاهی و حسین نظام آبادی پور و ماشالله ماشین چی [۱۳] این مساله را با استفاده از یک الگوریتم جستجوی گرانشی گسسته ترکیبی حل کرده‌اند.

۴ الگوریتم جستجوی تقلید نیروی گرانشی

در سال ۱۹۹۵ وادوریس و تسانگ [۱۴] برای اولین بار الگوریتم GLS برای جستجو در یک فضای جستجو و حل مسایل NP-complete پیشنهاد داد و در سال ۲۰۰۴ بری وبستر [۱۵] آن را به عنوان یک الگوریتم نیرومند ارائه داد و آن را GELS نام گذاری کرد. ایده این الگوریتم بر اساس اصل نیروی گرانشی است که در طبیعت سبب می‌شود اشیاء به سمت یکدیگر جذب شوند به طوری که شیئی سنگین تر دارای نیروی گرانش بیشتر است و آن را بر اشیاء دیگر اعمال کرده و اشیاء با وزن کمتر را به سوی خود جذب می‌کند. البته فاصله دو شیئی بر اندازه این نیرو بسیار موثر است به طوری که، دو جسم با وزن یکسان و با فاصله‌های مختلف نسبت به شیئی با وزن کمتر را در نظر بگیرید جسمی که دارای فاصله کمتری با شیئی کم وزن است نیروی جاذبه بیشتری بر آن اعمال می‌کند. در الگوریتم جستجوی تقلید نیروی گرانشی، فرمول قانون نیروی گرانشی نیوتن بین دوشی عبارت است از:

$$F = \frac{Gm_1m_2}{r^2} \quad (۱)$$

که در آن m_1 و m_2 به ترتیب جرمشی اول و دوم می باشد. G نیز برابر مقدار ثابت نیروی گرانشی $6/۶۷۲$ ، r نیز پارامتر شعاع و فاصله بین دوشی است.

الگوریتم جستجوی تقلید نیروی گرانشی نیز برای جستجو در میان یک فضای جستجو از این فرایند طبیعت تقلید می کند. بطوری که فضای جستجو دنیا و اشیای در این دنیا پاسخ های ممکن برای جستجو می باشند. هر یک از این اشیا دارای وزن می باشد، وزن هر شیئی عبارت است از کارایی و یا معیار جستجو که در آن بهترین پاسخ دارای بیشترین وزن است و هیچ یک از اشیا نیز نمی توانند دارای وزن صفر باشند [۱۴-۱۶]. در این روش، پاسخ های ممکن در فضای جستجو بر اساس معیاری که به نوع مساله بستگی دارد به دسته هایی تقسیم می شوند که هر یک از این دسته ها یک بعد از پاسخ مساله نامیده می شود و برای هر بعد از پاسخ مساله یک مقدار با نام سرعت اولیه در نظر گرفته می شود که در ادامه توضیح داده خواهد شد.

الگوریتم جستجوی تقلید نیروی گرانشی به دو صورت نیروی گرانشی میان پاسخ ها یا اشیا در فضای جستجو را محاسبه می کند. در روش اول یک پاسخ از فضای همسایگی محلی پاسخ جاری انتخاب شده و نیروی گرانشی بین این دو پاسخ محاسبه می شود. در روش دوم، نیروی گرانشی بین تمام پاسخ های همسایه در فضای همسایگی پاسخ جاری محاسبه می شود و به یک پاسخ همسایه محدود نمی شود. در حرکت درون فضای جستجو نیز الگوریتم جستجوی تقلید نیروی گرانشی به دو روش عمل می کند، روش اول حرکت از پاسخ جاری به پاسخ هایی را اجازه می دهد که در فضای محلی پاسخ جاری هستند و در روش دوم حرکت به پاسخ هایی در خارج از فضای همسایگی محلی پاسخ جاری را اجازه می دهد. هر کدام از این روش های انتقال می تواند با هر یک از روش های محاسبه نیروی گرانش بکار برده شوند که در نتیجه آن چهار مدل برای الگوریتم جستجوی تقلید نیروی گرانشی ایجاد می شود.

الگوریتم جستجوی تقلید نیروی گرانشی شامل یک بردار است که سائز آن مشخص کننده تعداد ابعاد پاسخ می باشد. مقادیر این بردار نشان دهنده سرعت نسبی در هر بعد است. الگوریتم با یک پاسخ اولیه، بردار سرعت اولیه و جهت حرکت شروع می شود. برای هر بعد در بردار سرعت، یک عدد تصادفی بین یک و ماکزیمم سرعت انتخاب می شود و این مقدار هر عنصر در هر بعد می باشد. پاسخ اولیه به عنوان پاسخ جاری توسط کاربر و یا تصادفی ایجاد می شود. برای هر بعد در بردار سرعت اولیه، با توجه به بردار سرعت اولیه بعدهای پاسخ، جهتی برای حرکت انتخاب می شود که این جهت برابر است با بعد پاسخی که دارای بیشترین سرعت اولیه در بردار سرعت اولیه است.

الگوریتم شامل یک شیئی اشاره گر است که می تواند در میان فضای جستجو حرکت کند و وزنی که برای شیئی اشاره گر در نظر گرفته می شود در تمام محاسبات ثابت است و این شیئی همیشه به پاسخی با بیشترین وزن اشاره می کند. الگوریتم با رخ دادن یکی از دو شرط فوق پایان می پذیرد: تمامی عناصر بردار سرعت اولیه برابر صفر شود و یا تعداد تکرار الگوریتم به ماکزیمم تعداد خود برسد.

در هر تکرار الگوریتم به روش اول یک پاسخ نامزد از فضای همسایگی محلی پاسخ جاری بر حسب جهت حرکت جاری انتخاب می‌شود و نیروی گرانش بین پاسخ جاری و پاسخ نامزد محاسبه شده و سپس بردار سرعت اولیه با توجه به این نیرو به روزرسانی می‌شود. برای تکرار بعدی نیز بردار سرعت اولیه چک شده و با توجه به آن جهت حرکتی برای ادامه جستجو انتخاب می‌شود. هر تکرار الگوریتم به روش دوم نیز کاملاً مشابه روش اول است با این تفاوت که به جای محاسبه نیروی گرانش و به روز رسانی بردار سرعت اولیه برای فقط یک پاسخ نامزد به دست آمده از جهت جاری، نیروی گرانش و به روزرسانی سرعت اولیه برای هر یک از پاسخ‌های نامزد محاسبه می‌شود.

در فرمول نیوتن استفاده شده، با تعویض کردن دو جرم در صورت کسر معادله و جایگزین نمودن با مقدار تفاوت بین هزینه پاسخ نامزد و پاسخ جاری، نیروی گرانش بین دو شیئی با استفاده از رابطه زیر محاسبه می‌شود:

$$f = \frac{G(CU - CA)}{R^2} \quad (2)$$

که در آن CU و CA به ترتیب هزینه پاسخ جاری و پاسخ نامزد می‌باشد. این فرمول مقدار مثبت دارد اگر هزینه پاسخ جاری بزرگ‌تر از هزینه پاسخ نامزد باشد و مقدار منفی، اگر هزینه پاسخ نامزد بزرگ‌تر باشد. سپس مقدار این نیرو، منفی یا مثبت، به بردار سرعت در وضعیت مسیر حرکت جاری اضافه می‌شود. اگر این عمل موجب شود که مقدار پارامتر سرعت از ماکزیمم تنظیمات تجاوز کند، آن مقدار ماکزیمم را می‌گیرد. اگر بروزرسانی موجب منفی شدن مقدار شود، آن مقدار صفر را می‌گیرد.

پارامترهایی که در GELSTSP قابل دسترس باشند عبارت‌اند از:

Maximum Velocity: ماکزیمم مقداری که می‌توان به هریک از عناصر بردار سرعت اولیه تخصیص داد و این پارامتر از بزرگ شدن بیش از حد این عنصر جلوگیری می‌کند.

Radius: شعاعی است که در فرمول محاسبه نیروی گرانشی استفاده می‌شود.

Iteration: ماکزیمم تعداد تکرار الگوریتم را مشخص می‌کند که تضمین می‌کند الگوریتم پایان می‌پذیرد.

تنظیمات این پارامترها را می‌توان با انجام آزمایش‌ها و آزمون و خطا به دست آورد.

با این توضیحات شبه کد الگوریتم جستجوی GELSTSP در شکل ۱ نشان داده شده است.

procedure GELSTSP is

--Number of solution dimensions is a predefined number "n"

integer dim; --Dimension iteration counter

integer cnt; --Loop iteration counter

begin

for dim = 1 **to** n **do**

--Assign a predefined starting solution component as the current solution component for

-- dimension "dim" and as the best solution component seen thus far for that dimension

--Randomly assign an initial velocity in the dimension "dim" within the bounds of 1 to n,
number of city

end;

cnt = 0;

--Calculate an initial vector velocity sum, based on the random initial velocity components

-- assigned in the previous step

while (the velocity sum <> 0) **and** (cnt < ITER) **do**

--Reset the velocity sum to 0

for dim = 1 **to** n **do**

--Calculate the solutions adjacent to the current solution and their respective RF values

--If any of these is better than the best solution seen thus far, then make that solution

-- the new best solution

--Calculate the net difference in gravitational "force" between the adjacent solutions and

-- the current solution for the current dimension "dim", using the Newtonian equation for

-- gravitational attraction

--Calculate change in acceleration for the current dimension "dim"

--Calculate change in velocity for the current dimension "dim"

--Calculate new current solution component for the dimension "dim", which will be the next

-- adjacent node in the dimension "dim" in the current direction of movement as indicated

-- by the velocity component for the dimension "dim"

end;

--Calculate the new velocity sum

cnt = cnt + 1;

end;

return best solution found for TSP is the sequence of visit city, its RF value, and the iteration
count (cnt)

end

شکل ۱. شبه کد الگوریتم GELSTSP

۵ مراحل الگوریتم پیشنهادی

به منظور بررسی کارایی الگوریتم پیشنهادی GELSTSP آن را با نمونه مسایل TSPLIB [۱۷] مقایسه می کنیم.

جدول ۱ نتایج را برای هر یک از مسایل حل شده نشان داده شده است.

روش بیان شده را توسط نرم افزار C#.net و اجرای آن بر روی کامپیوتری با پردازنده پنتیوم ۴ انجام شده است.

که این سیستم شامل بخش ها و امکانات زیر است:

در سیستم شبیه سازی شده در ابتدا تعداد مراحل، تعداد شهرها و مسافت بین شهرها مشخص می شود.

با توجه به تعداد شهرها، شعاع مشخص می شود.

مقادیر بردار سرعت را به صورت تصادفی از بازه مقادیر ۱ تا تعداد کل شهرها انتخاب می کنیم.

در هر مرحله اندیس بزرگترین مقدار بردار سرعت انتخاب می شود و سپس شهر تخصیص داده شده به آن بعد

را تغییر می دهیم. درواقع شی جدید بوجود می آید، سپس هزینه پاسخ جاری محاسبه می شود. اگر هزینه پاسخ

جاری از بهترین هزینه پاسخ به دست آمده تاکنون بیشتر بود آن را به عنوان بهترین شیء تا این مرحله انتخاب می‌کنیم.

نیروی گرانش بر اساس هزینه پاسخشی جاری و نامزد محاسبه می‌شود.

سپس نیروی گرانشی به دست آمده در مرحله ۵ را به مقدار بردار سرعت در هر اندیس اضافه می‌کنیم.

مراحل ۱ تا ۶ را تا زمانی که یکی از دو شرط زیر اتفاق نیافتد ادامه می‌یابد: تمامی عناصر بردار سرعت اولیه برابر صفر شود و یا تعداد تکرار الگوریتم به ماکزیمم تعداد خود برسد.

در جدول ۱ الگوریتم GELSTSP را اجرا کرده، سپس با الگوریتم ژنتیک، تپه نوردی، تبرید فلزات شبیه‌سازی شده مقایسه نمودیم.

جدول ۱. نمونه‌های استفاده شده برای ارزیابی الگوریتم پیشنهادی با ۱۱ نمونه TSPLIB

نام نمونه	مقدار هدف بهینه	تبرید فلزات شبیه‌سازی شده	تپه نوردی	الگوریتم ژنتیک	الگوریتم پیشنهادی GELSTSP
5sample1	۲۱	۲۱	۲۱	۲۱	۲۱
6sample2	۵۶	۵۶	۵۶	۵۶	۵۴
10sample3	۳۲۳	۳۲۴	۳۲۳	۳۲۴	۳۲۳
10sample4	۷۲۵	۷۴۲	۷۲۵	۷۶۱	۷۲۰
11eil51	۱۷۴	۱۷۷	۱۹۵	۲۲۷	۱۷۰
16eil76	۲۰۹	۲۵۵	۲۴۷	۲۵۵	۲۳۰
20kora100	۹۷۱۱	۱۱۷۹۲	۱۱۷۲۳	۱۱۷۹۲	۱۱۸۲۰
Gr21	۲۷۰۷	۳۳۳۳	۳۳۰۳	۳۳۳۳	۲۷۷۰۰
Gr24	۱۲۷۲	۱۵۵۳	۱۴۲۳	۱۵۲۸	۱۲۵۱
Eil51	۴۲۶	۴۹۷	۴۹۵	۴۹۷	۴۲۳
Eil76	۵۳۸	۶۶۱	۶۵۱	۶۶۱	۵۳۳

همچنین در جدول ۲ الگوریتم GELSTSP را با الگوریتم ژنتیک مقایسه نمودیم.

جدول ۲. نمونه‌های استفاده شده برای ارزیابی الگوریتم پیشنهادی با ۴ نمونه TSPLIB

نام نمونه	مقدار هدف بهینه	الگوریتم ژنتیک	الگوریتم پیشنهادی GELSTSP
kroA100	۲۱۲۸۲	۲۱۸۰۲	۲۱۲۸۰
kroB100	۲۲۱۴۱	۲۲۶۴۱	۲۲۱۳۴
kroB200	۲۹۴۳۷	۳۰۲۷۷	۲۹۵۴۵
Pr152	۷۳۶۸۲	۷۴۹۷۶	۷۳۵۶۷

۶ نتیجه گیری

مساله فروشنده دوره گرد جزء مسایل ان پی کامل (NP-complete) محسوب می شود. بنابراین نمی توان از الگوریتم های قطعی برای بهبود آن استفاده نمود. در این مقاله روش ابتکاری جدیدی به نام الگوریتم جستجوی تقلید نیروی گرانشی را برای حل مساله فروشنده دوره گرد نشان داده ایم. نتایج پیاده سازی الگوریتم پیشنهادی و مقایسه آن با سایر الگوریتم ها، نشان دهنده کارایی مناسب الگوریتم پیشنهادی می باشد. در کارهای آینده می توان این الگوریتم را بر روی مساله فروشنده دوره گرد نامتقارن و یا مسایل ان پی کامل اجرا و اعمال کرد.

منابع

- [۱۰] زارعی، ب.، میدی، م.ر.، (۱۳۸۶). یک روش ترکیبی برای حل مساله فروشنده دوره گرد. سومین کنفرانس فناوری اطلاعات و دانش، ۶ تا ۸ آذر.
- [۱۱] قیصری، ک.، سرحدی، ح.، (۱۳۸۸). یافتن کوتاه ترین تور همیلتونی ایران با استفاده از ترکیب الگوریتم سیستم اجتماع مورچه ها و جستجوی محلی. پژوهشنامه حمل و نقل، ۶(۲).
- [۱۲] یوسفی خوشبخت، م.، دیده ور، ف.، رحمتی، ف.، (۱۳۹۰). کاربرد یک الگوریتم رقابت استعماری برای حل مساله فروشنده دوره گرد. مجله مدلسازی پیشرفته ریاضی، ۱(۲).
- [۱۳] دولت شاهی، م. ب.، نظام آبادی پور، ح.، ماشین چی، م.، (۱۳۹۰). حل نمونه های بزرگ مساله فروشنده دوره گرد متقارن با استفاده از یک الگوریتم جستجوی گرانشی پیوسته. نشریه علمی و پژوهشی انجمن کامپیوتر ایران، ۹(۱ و ۳) الف، ۱۱-۱.

- [1] Garey, M. R., Johnson, D. S., (1979). Computers and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman press.
- [2] Cirasella, J., Johnson, D. S., McGeoch, L. A., Zhang, W., (2001). The Asymmetric Traveling Salesman Problem: Algorithms, Instance Generators, and Tests, in Algorithm Engineering and Experimentation, Third International Workshop, ALENEX 2001, Lecture Notes in Computer Science, Vol. 2153, Springer, 32-59.
- [3] Grötschel, M., Holland, O. (1991). Solution of Large-Scale Symmetric Traveling Salesman Problems. Mathematical Programming, 51, 1991, 141-202.
- [4] Moscato, P., Norman, M. G., (1994). An Analysis of the Performance of Traveling Salesman Heuristics on Infinite- Size Fractal Instances in the Euclidean Plane, Oct.
- [5] Gutin, G., Punnen, A. P., (2002). Traveling Salesman Problem and Its Variations. Kluwer Academic Publishers, Dordrecht.
- [6] Lawler, E., Lenstra, J., Rinnooy Kan, A., Shmoys, D., (1985). The traveling salesman problem: a guided tour of combinatorial optimization. Wiley, New York.
- [7] Lin, S., Kernighan, B., (1973). An effective heuristic algorithm for the traveling salesman problem. Operations Research, 21, 498-516.
- [8] Litke, J., (1984). An improved solution to the traveling salesman problem with thousands of nodes. Communications of the ACM, 27, 1227-1236.
- [9] Raja Balachandar, S., Kannan, K., (2007). Randomized gravitational emulation search algorithm or symmetric traveling salesman problem. Applied mathematics and computation, 413-421.
- [14] Chris, V., Tsang, E., (1995). Guided Local Search. Technical Report CSM-247, Department of Computer Science, University of Essex, UK.
- [15] Webster, B. L., (2004). Solving Combinatorial Optimization Problems Using a New Algorithm Based on Gravitational Attraction, Ph.D., thesis, Melbourne, Florida Institute of Technology.
- [16] Webster, B., Bernhard, P. J., (2003). A Local Search Optimization Algorithm Based on Natural Principles of Gravitation. Proceedings of the International Conference on Information and Knowledge Engineering, 1, 255-261.

- [17] Reinelt, G. , (1991) . TSPLIB - A Traveling Salesman Problem Library. ORSA Journal on Computing, 376 – 384.