استفاده از تبدیل فضا برای حل مسئله فروشنده دوره‌گرد با معیار فاصله جغرافیایی

رضا مرتضوی[۱]، استادیار

۱- دانشکده فنی و مهندسی - دانشگاه دامغان - دامغان - ایران – r_mortazavi@du.ac.ir

چکیده: یکی از مسائل بهینه سازی مهم در حوزه الگوریتم‌های ترکیباتی، مسئله فروشنده دوره‌گرد است. با توجه به کاربردهای فراوان، حل این مسئله مورد توجه پژوهشگران است و به‌طور خاص به‌عنوان یک مسئله مهم تحلیل شبکه در فناوری‌های اطلاعات مکانی کاربرد دارد. علی‌رغم صورت ساده، حل کلی این مسئله از درجه پیچیدگی NP- سخت است. به‌همین جهت روش‌های ابتکاری زیادی در کاربردهای عملی پیشنهاد شده است. در این مقاله از روش تصویر سازی نقشه برای تبدیل فضا از نسخه‌ای از مسئله فروشنده دوره‌گرد با معیار فاصله جغرافیایی به نسخه ساده‌تر با معیار فاصله اقلیدسی استفاده شده است. در ادامه مقاله یکی از روش‌های ابتکاری حل این مسئله بهبود داده شده است. نتایج تجربی بر روی مجموعه داده‌های واقعی در مقایسه با روش‌های مشابه، نشان‌دهنده برتری روش پیشنهادی به‌لحاظ کیفیت پاسخ و زمان دستیابی به آن است.

واژه‌های کلیدی: مسئله فروشنده دوره‌گرد، تبدیل فضا، تصویرسازی نقشه، فاصله غیر اقلیدسی، بهینه‌سازی.

Solving Geographic Travelling Salesman Problem Based on Space Transformation

R. Mortazavi, Assistant Professor[1]

1- School of Engineering, Damghan University, Damghan, Iran, r_mortazavi@du.ac.ir

Abstract: One of the most important optimization problem in combinatorial algorithms is Travelling Salesman Problem. This is due to its practical usages, especially in geospatial information technology. It is proved that the problem complexity is NP-Hard, but many heuristic methods are presented for practical applications. In this paper, the space transformation method is used to transform a geographic instance of travelling salesman problem into a simpler Euclidean one, such that traditional heuristics can be applied more successfully. Additionally, a heuristic technique to solve the transformed instance is introduced. The results confirm the superiority of the proposed method in terms of both the tour length and running time of the method in comparison with similar techniques.

Keywords: Travelling Salesman Problem, space transformation, map projection, non-Euclidean distance, optimization.

تاریخ ارسال مقاله: ۱۳۹۵/۰۵/۱۲
تاریخ اصلاح مقاله: ۱۳۹۵/۰۷/۲۲
تاریخ پذیرش مقاله: ۱۳۹۵/۰۹/۱۲
نام نویسنده مسئول: رضا مرتضوی
نشانی نویسنده مسئول: ایران – دامغان – میدان دانشگاه – دانشگاه دامغان - دانشکده فنی و مهندسی – گروه مهندسی کامپیوتر

۱- مقدمه

مسـئله فروشــنده دوره‌گرد (TSP) یکی از مســائل معروف الگوریتم‌های ترکیباتی[۱] است. هدف این مسئله در بیانی تمثیلی بدین صورت است که فروشنده‌ای باید تعداد مشخصی شهر را با شروع از یکی از آن‌ها ملاقات کند و به نقطه شـروع برگردد، به‌طوری‌که هر شـهر دقیقاً یک‌بار مشـاهده شود و طول مسیر طی‌شده کمینه باشد.

این مسئله اولین بار در ۱۹۳۰ به‌صورت ریاضی بیان شده است [۱] و در حوزه‌های مختلف مانند مســیریابی، شــبکه‌های کامپیوتری، طراحی مدارات الکترونیکی و مدل‌ســازی به‌کمک کامپیوتر [۴-۲] کاربرد دارد. مسـئله TSP دارای قدمت و کاربردهای زیادی اسـت. ولی در حالت کلی دارای راه‌حل دقیق و کارآمد در زمان چندجمله‌ای برحسب تعداد شهرها (n) نیسـت و در دسـته مسـائل NP-سـخت تقسـیم‌بندی می‌شـود [۵]. بنابراین با توجه به اهمیت حل آن در کاربردهای عملی، روش‌های ابتکاری مختلفی برای حل تقریبی آن پیشنهاد شده است که سعی می‌کنند در زمان قابل‌قبولی جواب نزدیک به بهینه را بیابند [۶].

با استفاده از روش‌های بهینه‌سازی مبتنی بر تقسیم و حل با شروع از جواب‌های نزدیک به بهینه می‌توان راحت‌تر به جواب‌های بهینه رســید [۱]. یکی از معروف‌ترین ابزارها در این بخش Concorde[۲] اســت که برای دســتیابی به حل بهینه از روش شـاخه و برش اسـتفاده می‌کند. این ابزار می‌تواند مسـائلی به بزرگی ۸۵۹۰۰ شـهر را در مدت ۱۳۶ پردازنده- سال حل کند [۷] و برای نســخه‌هایی با چند میلیون شــهر نیز جواب آن در نزدیکی جواب بهینه اسـت [۶]. دسـتیابی به جواب اولیه نزدیک به بهینه در زمان معقول، از اهمیت بالایی در حل این مسئله برخوردار است.

یکی از روش‌های موفق برای حل مسـئله فروشـنده دوره‌گرد با عنوان الگوریتم ابتکاری پس‌انداز شـناخته می‌شـود [۸]. اما نسـخه اولیه از این الگوریتم برای مجموعه داده‌های بزرگ قابل استفاده نیست و با محدودیت حافظه و زمان اجرا مواجه اسـت. بهبودهایی از این الگوریتم برای حل این مشکلات پیشنهاد شده است که مبتنی بر استفاده از ساختمان داده‌ای با نام kd-tree [۹] است [۱۰]. ولی این بهبودها برای نمونه‌هایی از مسئله با معیار فاصله اقلیدسی ارائه شده‌اند [۱۲-۱۰].

نسخه‌هایی از مسئله فروشنده دوره‌گرد در فضای غیر اقلیدسی تعریف می‌شــوند. به‌بیان‌دیگر با توجه به کاربرد، از توابعی خاص برای محاســبه فاصله بین گره‌ها استفاده می‌شود [۱۵-۱۳]. برای مثال، Faigl و همکاران برای محاسبه فاصله بین دو گره به‌دلیل وجود موانع میانی، طولی بیش از مســافت مســتقیم بین آن گره‌ها تعریف کرده‌اند و از نقشــه‌های خود سازمانده برای حل مسئله استفاده کرده‌اند [۱۳]. به‌عنوان مثالی دیگر، در مسئله فروشنده دوره‌گرد با معیار فاصله جغرافیایی (GEOTSP)، با فرض قرار گرفتن شـهرها بر روی کره زمین، فاصـله جغرافیایی آن‌ها بر اسـاس طول و عرض جغرافیایی آن‌ها محاسـبه و منظور می‌شـود. Curtin ضـمن اشــاره به کاربردهای گســترده GEOTSP در مســائل تحلیل شــبکه، به استفاده از روش‌های برنامه‌ریزی عدد صحیح برای حل نسخه‌های با اندازه متوسـط این مسـئله (برحسـب تعداد شـهرها) در سـیسـتم‌های اطلاعات جغرافیایی[۳] اشـاره کرده اسـت [۱۴]. علی‌رغم اینکه نسـخه‌های اقلیدسی TSP نیز برای حل مشکل هستند، اما راه‌حل‌های ابتکاری برای مواجهه با آن‌ها موفق‌تر هســتند [۱۶]. با توجه به ویژگی‌های نامتعارف در فضـاهای غیر اقلیدســی معمولاً نمی‌توان از روش‌های ابتکاری رایج در فضــای اقلیدســی مانند الگوریتم ابتکاری پس‌انداز برای تســریع یا بهبود حل اســتفاده کرد. بنابراین حل این نســخه‌ها به‌دلیل نیاز به اســتفاده از حل‌کننده‌های عمومی‌تر نیاز به صــرف منابع بیش‌تری دارد [۱۴]. برای مثال حل نسخه GEOTSP در سطح جهان با عنوان World TSP به‌روش تقریبی به ۲۵۶ روز زمان نیاز داشته است [۴].

در این مقاله برای حل مشکل فاصله غیر اقلیدسی برای نمونه‌های با معیار فاصله جغرافیایی، از روش تصویرسازی نقشه[۴] استفاده شده است تا فضای مسئله را از جغرافیایی (روی کره) به اقلیدسی (در صفحه) تقریب بزند. این روش برای حل نمونه‌هایی در حد یک کشــور با حداکثر $n=۷۱۰۰۹$ شــهر به‌کار رفته اســت[۵]. در حد اطلاع نویســنده مطالعه این روش در قالب چارچوب ارائه‌شده در این مقاله و در گستره جغرافیایی کل زمین صورت نگرفته است. در ادامه بهبودی در الگوریتم ابتکاری پس‌انداز [۸] همراه با پیش‌پردازشی اضافی نسبت به [۱۷] ارائه شده است که به بهبود کیفیت جواب نهایی منجر می‌شــود. نهایتاً این روش بر روی نمونه‌هایی از مســئله GEOTSP با اندازه‌های متفاوت و نیز نمونه‌های اقلیدسی ارزیابی شده است و نتایج به‌لحاظ کیفیت جواب نهایی و زمان اجرا ارائه و مقایسه شده است.

ادامه این مقاله به‌صــورت زیر بخش‌بندی شــده اســت: در بخش ۲ مفاهیم پایه بیان می‌شوند. این مفاهیم شامل بیان رسمی GEOTSP است که در این مقاله بررسی شده است. همچنین روش‌های ابتکاری حل TSP که در بخش مقایسه به آن‌ها اشاره شده است، معرفی می‌شوند. در ادامه نسخه اولیه روش ابتکاری پس‌انداز تشریح می‌شود که برای حل TSP در فضای اقلیدسی در این مطالعه بهبود داده شده است. در بخش ۳ روش پیشنهادی ارائه شده است و در ادامه در بخش ۴ نتایج تجربی حاصل از ارزیابی روش ارائه می‌شــود. بخش ۵ نیز به نتیجه‌گیری و پیشــنهادهایی برای ادامه کار اختصاص دارد.

۲- مفاهیم پایه

در این بخش ابتدا مســئله فروشــنده دوره‌گرد با معیار فاصــله جغرافیایی به‌صورت رسمی بیان می‌شود. سپس روش‌های ابتکاری حل TSP به‌اجمال مرور می‌شود. در ادامه روش ابتکاری پس‌انداز توضیح داده می‌شود که در

این مقاله از نسخه بهبودیافته آن برای حل TSP در فضای اقلیدسی استفاده کرده‌ایم. همان‌طور که در بخش مقدمه بیان شد، در این مقاله برای اینکه بتوان در حل GEOTSP از ابتکارات مختص فضای اقلیدسی استفاده کرد، از تصویرسازی نقشه استفاده می‌شود. در پاسخ به این سوال که کدام روش‌های تصویرسازی نقشه برای حل GEOTSP مناسب هستند، در این بخش توضیح داده می‌شود.

۲-۱- معرفی مسئله فروشنده دوره‌گرد با معیار فاصله جغرافیایی

فرض کنید شبکه‌ای از مکان‌های جغرافیایی (شهرها) به‌صورت گراف کامل $G=(V,E)$ در اختیار است که $V=\{v_۱,v_۲,...,v_n\}$ نشان‌دهنده شهرها و $E=V\times V$ یال‌های بین آن‌هاست. وزن هر یال بین دو گره v_i و v_j به‌صورت $c(v_i,v_j)$ نشان داده می‌شود. هدف مسئله فروشنده دوره‌گرد کمینه کردن دوری روی این گراف است که از همه گره‌ها دقیقاً یک‌بار می‌گذرد (و به مکان شروع برمی‌گردد). به‌صورت رسمی‌تر، هدف یافتن ترتیبی از شهرهاست که تابع هدف رابطه (۱) کمینه شود.

$$min\sum_{i=۲}^{n} c(v_{(i-۱)},v_i)+c(v_n,v_۱) \qquad (۱)$$

نسخه‌ای از مسئله که در این مقاله به آن پرداخته شده است فواصل بین شهرها را به صورت متقارن تعریف می‌کند، یعنی $c(v_i,v_j)=c(v_j,v_i)$. برای نمونه‌هایی از TSP با معیار فاصله جغرافیایی و در گستره‌ای به وسعت کره زمین در مجموعه داده استاندارد TSPLib معیار فاصله جغرافیایی بین دو شهر v_i و v_j یعنی $c(v_i,v_j)$ برحسب متر بر روی کره‌ای به شعاع $R=۶۳۷۸۳۸۸$ متر به‌کمک تابع $GEOM(i,j)$ محاسبه می‌شود. فرض کنید طول و عرض جغرافیایی شهر v_i در ارائه‌های متناظر به ترتیب به‌صورت $long[i]$ و $lat[i]$ و برحسب رادیان ذخیره شده‌اند. محاسبه فاصله در شکل ۱ نشان داده شده است.

```
GEOM(i,j) {
    R = 6378388
    q1 = cos(lat[j])*sin(long[i] – long[j])
    q3 = sin((long[i] – long[j]) / 2.0)
    q4 = cos((long[i] – long[j]) / 2.0)
    q2 = sin(lat[i] + lat[j])*q3*q3 - sin(lat[i] – lat[j])*q4*q4
    q5 = cos(lat[i] – lat[j])*q4*q4 - cos(lat[i] + lat[j])*q3 * q3
    return (int)(R * atan(sqrt(q1 * q1 + q2 * q2), q5) + 1.0)
}
```

شکل ۱: محاسبه فاصله جغرافیایی بین دو شهر v_i و v_j برحسب متر

در شکل ۱، توابع atan، sqrt و int به‌ترتیب نشان‌دهنده معکوس تانژانت، جذر و جزء صحیح هستند.

۲-۲- روش‌های ابتکاری حل TSP

روش‌های ابتکاری برای حل TSP شامل روش‌های ساختنی و روش‌های بهبود تدریجی می‌شوند. در روش‌های ساختنی، بر اساس قاعده‌ای تعریف شده دور ساخته می شود و بعد از ساختن (بخشی از) آن، تلاشی برای بهبود آن انجام نمی‌شود و یال‌ها بدون تغییر می‌مانند. در روش‌های بهبود تدریجی، با شروع از یک جواب اولیه (مثلاً با استفاده از روش‌های ساختنی) به دفعات با تغییر یال‌ها سعی می‌شود دور نهایی بهبود یابد. به‌طورکلی روش‌های ساختنی خیلی سریع‌تر از روش‌های بهبود تدریجی هستند ولی معمولاً کیفیت جواب پایین‌تری نسبت به روش‌های بهبود تکراری دارند [۱۸].

روش‌های ساختنی شامل ابتکارات مبتنی بر نزدیک‌ترین همسایه (NN)، ابتکارهای درج یال [۷]، ابتکارهای مبتنی بر درخت‌های پوشا، ابتکار حریصانه (GH) [۱۹]، MTS [۱۲] و ابتکار پس‌انداز می‌شود. در روش NN، با شروع از یک گره، نزدیک‌ترین همسایه بعدی از گره جاری که هنوز مشاهده نشده به انتهای دور کنونی اضافه می‌شود. در روش GH، همواره کوتاه‌ترین یال به دور ناکامل جاری افزوده می شوند، به شرط اینکه این افزودن باعث ایجاد زیردور نشود و درجه هر دو گره این کوتاه‌ترین یال در دور ناکامل کم‌تر از ۲ باشد [۱۹]. بهبودهایی از این روش با استفاده از صف اولویت پیشنهاد شده است [۲۰]. روش برووکای سریع[۶] (QB) توسعه‌ای از روش GH و مبتنی بر درخت پوشاست که به‌جای استفاده از صف‌های اولویت، از مرتب سازی داده‌ها استفاده می کند [۲۰]. روش MTS [۱۲] مبتنی بر تطابق عمل می‌کند. در این روش ابتدا دورهای کوچک‌تر ساخته می‌شوند و با استفاده از نسخه مناسب‌سازی‌شده از مسئله تطابق، دور نهایی به‌دست می‌آید.

```
function tour = savings(V) {
    hub ← index of the most centeral city in V
    V_h ← {1, …, n} − hub
    foreach i, j ∈ V_h, i < j do
        S[i, j] ← D(hub, i) + D(hub, j) − D(i, j)
    while |V_h| > 2 do
        add (v_i, v_j) with the best S[i, j] to partial tour provided that no
        subtour is created
        if degree of v_i is 2, remove i from V_h
        if degree of v_j is 2, remove j from V_h
        stitch remaining two cities in V_h through the hub node
        return tour
}
```

شکل ۲: روش ابتکاری پس‌انداز برای حل TSP– نسخه اولیه

روش‌های بهبود تدریجی شامل روش ۲-opt [۲۱] و روش‌های تکاملی مانند شبیه سازی ذوب فلزات [۲۲] و الگوریتم ژنتیک [۲۳] می شود. در روش ۲-opt به‌عنوان یک روش جستجوی محلی، جای گره‌های انتهایی دو یال از دور داده شده به صورت ضربدری تغییر می کند تا طول کلی دور کاهش یابد. توسعه هایی نیز از این روش برای در نظر گرفتن تعداد بیش‌تری یال ارائه شده است [۲۴]. در روش تکاملی ذوب فلزات تغییرات محدودی در دور داده شده پیشنهاد داده می شود. در صورتی که این تغییر منجر به بهبود شود، تغییر پذیرفته می شود و درصورتی که منجر به بهبود نشود، با احتمالی این تغییر پذیرفته می شود. این احتمال در طول اجرای الگوریتم به‌مرور کاهش می‌یابد. در روش تکاملی مبتنی بر الگوریتم ژنتیک

برای حل TSP چندین جواب کاندید در قالب یک نسل در نظر گرفته می شود و علاوه‌بر تغییرات مختصر بر روی اعضای یک نسل برای بهبود آن‌ها، از ترکیب جواب‌های خوب نیز برای تولید جواب جدید استفاده می شود. به‌طورکلی استفاده از روش‌های تکاملی بر روی نمونه‌های بزرگ TSP به‌علت سرعت پایین آن‌ها، به‌عنوان اولویت اول مطرح نمی‌شود.

۲-۳- روش ابتکاری پس‌انداز و بهبودهای آن

یکی از روش‌های ابتکاری موفق برای حل TSP ابتکار پس‌انداز است [۸]. این روش در اصل برای حل مسائلی در مسیریابی پیشنهاد شده و نسخه‌های متنوعی دارد. شکل ۲ نسخه اولیه این روش را نشان می‌دهد.

در شکل ۲، در خط ۲ مرکزی‌ترین شهر از مجموعه V با عنوان گره قطب انتخاب می شود و اندیس آن در متغیر ذخیره می شود. اندیس همه شهرها به‌جز *hub* در خط ۳ در V_h ذخیره می‌شود. در خطوط ۴ و ۵، برای همه جفت شهرها در V_h مقداری به‌عنوان پس‌انداز محاسبه و در متغیر متناظر یعنی $S[i,j]$ ذخیره می‌شود. در خط ۵، تابع $D(i,j)$ نشان‌دهنده فاصله بین v_i و v_j است. در حلقه اصلی این روش، یعنی خطوط ۶ تا ۹ تا زمانی که تعداد عناصر V_h بیش‌تر از دو است، یال متصل‌کننده شهرهای i -ام و j -ام یعنی (v_i,v_j) که مقدار پس‌انداز مربوط به آن از همه بیش‌تر است، درصورتی‌که به ایجاد دوری منجر نشود، به مجموعه یال‌های دور ناکامل اضافه می شود (خط ۷). در خط ۸ و ۹، اندیس مربوط به شهرهای v_i و v_j درصورتی‌که به‌طور کامل از دو طرف در tour متصل شده باشند، از V_h حذف می‌شود. در پایان و در خط ۱۰، گره قطب با اندیس *hub* به دو گره باقی‌مانده که اندیس‌هایشان در V_h قرار دارد، متصل شده و دور را تکمیل می‌کند. دور تکمیل‌شده در خط ۱۱ برگشت داده می‌شود.

پیاده سازی این الگوریتم اولیه آسان است. ولی مرتبه زمان اجرای آن $O(n^2 \log n)$ است و همچنین فضای موردنیاز برای ذخیره‌کردن مقادیر پس‌انداز از مرتبه $O(n^2)$ است. این بدان معنا ست که اجرای این الگوریتم برای مقادیر بزرگ n میسر نیست. بهبودهایی برای رفع این مشکلات پیشنهاد شده است. Golden و همکاران نشان داده‌اند که کافی است فقط برخی جفت گره‌های خاص را در ضمن ساخت دور در نظر بگیریم و بر این اساس روالی برای مرتب سازی پس‌اندازها را با استفاده از ساختمان داده کپه[۷] پیشنهاد کرده‌اند [۲۵].

به‌علاوه، Paessens روشی با عنوان «یافتن تدریجی بیش‌ترین مقدار پس‌انداز» ارائه کرده است [۱۰]. در این روش ابتدا همه رکوردها بر اساس فاصله تا گره قطب مرتب می‌شوند و تنها آن دسته از جفت گره‌ها در محاسبه پس‌انداز وارد می‌شوند که فاصله هر دو گره آن‌ها تا گره قطب حداقل به‌اندازه نصف بیش‌ترین پس‌انداز موجود باشد، زیرا در فضای اقلیدسی حداکثر مقدار پس‌انداز برای یک جفت گره همواره مساوی یا کم‌تر از ۲ برابر حداقل فاصله‌های آن گره‌ها تا گره قطب است. به‌بیان‌دیگر برای دو جفت گره با اندیس i و j همواره نامساوی $S[i,j] \le min\{D(i,hub), D(j,hub)\}$ برقرار است [۱۰]. بنابراین نیازی نیست که همه جفت گره‌ها را در ضمن ساخت دور در نظر گرفت. به‌علاوه مرتضوی و جلیلی روشی برای بهبود این ابتکار ارائه کرده‌اند که از ساختمان داده kd-tree به همراه مفهومی بنام گره فعال استفاده می کند [۱۷]. در این روش برای محاسبه پس‌انداز به‌صورت تدریجی تنها نزدیک‌ترین همسایه فعال یک گره (یعنی همسایه‌ای با درجه کم‌تر از ۲ در دور ناکامل)، برای جفت شدن مورد بررسی قرار می گیرد، زیرا اصولاً احتمال جفت شدن دو گره نزدیک به هم در دور با طول کمینه بیش‌تر است. تمامی این بهبودها با فرض این قابل‌استفاده هستند که فاصله بین *hub* گره‌ها در فضای اقلیدسی محاسبه می‌شود.

۲-۴- روش‌های تبدیل فضا در مختصات جغرافیایی

به روش‌های تبدیل فضا در مختصات جغرافیایی اصطلاحاً تصویر سازی نقشه می‌گویند. به بیان دقیق‌تر به‌روش قاعده‌مندی که برای تبدیل نقاطی بر روی سطح کره یا بیضی برحسب طول و عرض جغرافیایی به نقاطی بر روی کاغذ استفاده می شود، تصویر سازی نقشه می‌گویند [۲۶]. همواره چنین تبدیلی همراه با نوعی اعوجاج در مساحت، شکل، جهت، فاصله یا مقیاس خواهد بود. همچنین نحوه مدل کردن زمین (به‌صورت کره، بیضی یا ...) نیز در ویژگی‌های نقشه نهایی تأثیر خواهد داشت.

به‌صورت طبیعی شباهت فواصل بین شهرها در تصویر ساخته‌شده در نقشه با اطلاعات اصلی قبل از تصویر سازی منجر به نتایج دقیق‌تری در حل GEOTSP می‌شود. به روش‌های تصویرسازی نقشه که (تا حد امکان) فواصل بین گره‌ها را پس از تصویرسازی حفظ می‌کنند، به اصطلاح هم‌فاصله[۸] می‌گویند. در ادامه این مقاله و با توجه به نحوه محاسبه فاصله در مجموعه داده های موردنظر در مسئله TSP (بخش ۲-۱)، به‌دنبال روش‌های تصویرسازی نقشه هم‌فاصله و با فرض کروی‌بودن زمین هستیم.

در این مقاله از روش‌های تصویر سازی نقشه هم‌فاصله Azimuthal equidistant، Equidistant Conic و Cassini استفاده شده است. در روش Azimuthal equidistant که توسط ابوریحان بیرونی در سال ۱۰۰۰ میلادی ارائه شده است، از نقطه‌ای مرکزی استفاده می شود که فواصل تا این نقطه حفظ می‌شوند. روش Equidistant Conic در سال ۱۰۰ میلادی پیشنهاد شده است و فواصل را بر روی نصف‌النهارها (همان فاصله مدارهای استاندارد عرض‌های جغرافیایی)، حفظ می‌کند. روش Cassini که در سال ۱۷۴۵ توسط César-François Cassini de Thury ارائه شده است از نوع تصویر سازی نقشه استوانه‌ای است و مقیاس‌ها را در طول نصف‌النهار و خطوط موازی با آن حفظ می‌کند.

۳- روش پیشنهادی

ایده اصلی این مقاله برای حل TSP با معیار فاصله جغرافیایی بر پایه تبدیل فضا هست. بدین منظور از روش‌های ترسیم نقشه که به‌نوعی فاصله‌ها را حفظ می‌کنند، برای تبدیل مختصات جغرافیایی شهرها به مختصات کارتزین در دو بعد استفاده می شود. برای حل نسخه دوبعدی، نسخه بهبودیافته‌ای از روش ابتکاری پس‌انداز استفاده می‌شود که می‌تواند با سرعت و کیفیت بالا نسخه‌هایی تا چند میلیون شهر از مسئله TSP را حل کند. در پایان می‌توان جواب نمونه‌های دوبعدی را به نسخه اولیه مسئله در فضای اصلی اعمال کرد تا طول مسیر جواب پیشنهادی به دست آید.

همان‌طور که در بخش ۲-۳ بیان شد، نسخه اولیه روش ابتکاری پس‌انداز برای حل TSP دارای مشکلاتی هست. به همین دلیل در این مقاله برای بهبود زمان و حافظه مصرفی این الگوریتم، بهبودهایی در آن داده شده است. فرض کنید $knn(i,k) \in \{1,\dots,n\}$ نشان‌دهنده اندیس k-امین نزدیک‌ترین همسایه قابل اتصال گره v_i در V (البته به‌جز خود v_i) باشد. شکل ۳ در قالب تابع savings2 روش پیشنهادی را با عنوان savings2 ارائه می‌دهد.

```
1.   function tour=savings2(V) {
2.       hub ← index of the most central city in V
3.       V ← sort(V, hub)
4.       KD ← kd-tree on V
5.       H ← empty Max Heap
6.       V_h ← {2, …, n}
7.       if knn(i,1) = j  ∀i ∈ V_h        // decreasing loop on i
8.           if knn(j,1) = i or (i < j and degree of v_j is 0 and
             knn(j,2) = i), add (i,j) to partial tour
9.       idx ← n
10.      while |V_h| > 2 do
11.          while H is empty or D(idx,hub) ≥ H.top/2 and idx > 1 do
12.              NN ← knn(idx,1)
13.              S ← D(hub,idx) + D(hub,NN) − D(idx,NN)
14.              H.push(S)
15.              idx ← idx − 1
16.          (i,j) ← H.pop
17.          if i ∉ V_h continue
18.          if j ∈ V_h and adding(i,j) to partial tour does not create a
             subtour
19.              add(i,j) to partial tour
20.              if degree of v_j is 2, remove j from V_h
21.              if degree of v_i is 2, remove i from V_h, and continue
22.          NN ← knn(i,1)
23.          S ← D(hub,i) + D(hub,NN) − D(i,NN)
24.          H.push(S)
25.      stitch remaining two cities in V_h through the hub node
26.      return tour
27.  }
```

شکل ۳: نسخه بهبودیافته و کامل ابتکار پس‌انداز برای حل TSP

در خط ۱ مختصات گره ها به تابع تحویل می‌شود. در خط ۲ مرکزی‌ترین گره بنام قطب انتخاب می‌شود. در خط ۳، همه گره‌ها در مجموعه V برحسب فاصله از گره قطب به‌صورت صعودی مرتب می‌شوند به‌نحوی‌که گره قطب با اندیس ۱ در V شناخته می‌شود. در خطوط ۴ و ۵ به ترتیب ساختمان داده‌های kd-tree و کپه ساخته می شوند. در خط ۶ اندیس همه گره‌ها به‌جز گره قطب (با اندیس ۱) در V_h ذخیره می‌شود.

پس از انجام این مقدمات، فاز پیش‌پردازش شروع می‌شود که تفاوت اصلی روش پیشنهادی با روش ارائه شده در FDM [۱۷] محسوب می شود. این فاز پیش‌پردازش بر روی گره‌ها به ترتیب عکس فاصله آن‌ها تا گره قطب و در خطوط ۷ و ۸ انجام می‌شود. در این فاز، سعی می‌شود حتی‌الامکان بهترین انتخاب برای گره‌ها به صورت یک انتخاب محلی صورت بگیرد: اگر $knn(i,1) = j$ و $knn(j,1) = i$، در این صورت یال (v_i, v_j) قبل از شروع حلقه اصلی به یکدیگر متصل می شوند. همچنین در مورد گره‌های v_i و v_j که $j > i$، چنانچه $knn(i,1) = j$ و $knn(j,2) = i$ باشد و در مرحله قبلی گره j به گره دیگری متصل نشده باشد، در این صورت یال (v_i, v_j) به‌عنوان جزئی از دور نهایی افزوده می‌شود. درواقع این مراحل به‌نوعی تضمین می‌کنند که بهترین انتخاب برای گره v_i انجام شده است، زیرا به‌طور طبیعی اتصال به نزدیک‌ترین همسایه v_i به‌عنوان تصمیمی محلی بهترین انتخاب حریصانه برای v_i محسوب می‌شود. استفاده از این پیش‌پردازش باعث کاهش بار محاسباتی در مراحل بعدی الگوریتم می‌شود، زیرا زودتر همسایه‌های نهایی گره v_i در دور نهایی تعیین می شوند. در خط ۹ اشاره‌گر گره جاری برای محاسبه پس‌انداز به آخرین گره در با اندیس n تنظیم می‌شود. حلقه اصلی برنامه در خطوط ۱۰ تا ۲۴ به اضافه‌کردن یال‌ها تا زمانی که اندازه V_h بیش‌تر از ۲ هست، می‌پردازد. ابتدا در حلقه‌ای داخلی تا زمانی که امکان بهبود بهترین پس‌انداز باشد (خط ۱۱)، اطلاعات پس‌اندازها در کپه به‌روز می‌شود. دیس نزدیک‌ترین همسایه قابل اتصال به گره جاری با اندیس idx در متغیر NN ذخیره می‌شود (خط ۱۲) و مقدار پس‌انداز مربوط به یال (v_{idx}, v_{NN}) در S ذخیره می‌شود (خط ۱۳). این اطلاعات در کپه ذخیره می‌شوند (خط ۱۴) و مقدار idx برای اشاره به گره بعدی به‌روز می‌شود (خط ۱۵). در خط ۱۶ بهترین یال کاندید برای اضافه شدن به دور ناکامل در (i, j) ذخیره می‌شود و اطلاعات آن از بالای کپه حذف می‌شود. اگر v_i کاملاً متصل شده باشد و اندیس آن در V_h نباشد، ادامه حلقه نادیده گرفته می شود و مورد بعدی بررسی می شود (خط ۱۷). درصورتی‌که v_j به‌طور کامل متصل نشده باشد و افزودن یال (i, j) باعث ایجاد دور نشود، این یال به دور ناکامل اضافه می‌شود (خط ۱۹). در خط ۲۰ و ۲۱ اطلاعات V_h در صورت لزوم به‌روز می‌شود. در خطوط ۲۲ تا ۲۴ نیز اطلاعات مربوط به همسایگی گره v_i محاسبه و کپه به‌روز می شود. پس از پایان حلقه اصلی، خط ۲۵ وظیفه کامل کردن دور ناکامل را با افزودن یال‌هایی بین گره قطب و گره‌های با اندیس ذکرشده در V_h به عهده دارد. نهایتاً خط ۲۶ دور کامل شده را برمی‌گرداند.

۴- نتایج تجربی

در این بخش نتایج اجرای نسخه بهبود یافته ابتکار پس‌انداز بر روی نمونه‌های TSP با معیار فاصله جغرافیایی ارائه می‌شود. این ۱۰ نمونه از

۱۰۰۰۰ تا ۱۰۰۰۰۰ موقعیت جغرافیایی شهرهای کره زمین را دربر دارند و از ابتدای نمونه TSP جغرافیایی با عنوان World TSP با ۱۹۰۴۷۱۱ شـــهر انتخاب شـــده‌اند[۹]. اطلاعات این شـــهرها در قالب طول و عرض جغرافیایی ارائه شده است. کلیه محاسبات بر روی کامپیوتری شخصی با پردازنده Core i7 - 1.60 GHz، ســیســتم‌عامل ویندوز ۱۰ و ۸ گیگابایت حافظه اصلی انجام شده است. برای اعمال تصویرسازی نقشه نرم‌افزار ArcMap 10.3.1 و جعبه‌ابزار تصــویرســازی نقشـــه نرم‌افزار MATLAB اســتفاده شـــده اســت. در این مورد از مقادیر پیش‌فرض پارامترهای این نرم‌افزارها برای تصویرسازی نقشه استفاده شده است.

نتایج ارائه‌شـده در این بخش مربوط به سـه نوع تصـویرسـازی نقشـه معرفی‌شـــده در بخش ۲-۴ یعنی Azimuthal، Cassini و Conic اســت. نتایج این روش‌ها به‌ترتیب با AZ، CA و CO در جدول ۱ ارائه شده است. برای مشاهده اثر پیش‌پردازش پیشنهادی و مقایسه دقیق‌تر با روش ارائه شـــده در مرجع [۱۷]، نتایج طول دور بدون در نظر گرفتن پیش‌پردازش در الگوریتم پیشنهادی (خطوط ۷ و ۸ از شکل ۳) با عنوان نسخه ساده و با در نظر گرفتن پیش‌پردازش با عنوان نسخه کامل نشان داده شده است. این طول دورها با نتایج طول دور مربوط به روش‌های GH، NN و QB بر روی نمونه اصلی (بدون تصویرسازی نقشه)، مقایسه شده‌اند. پیاده‌سازی این روش‌ها در برنامه‌ای با عنوان linkern[۱۰] ارائه شـده اسـت. زمان اجرای روش‌های فوق (بدون در نظر گرفتن زمان کار با دیسـک) نیز در جدول ۲ ارائه شـــده اســـت. زمان اجرای savings2 به‌ازای هر نمونه با اندازه ثابت، برای نسخه‌های ساده و کامل بسیار کم و تقریباً یکسان است. بنابراین تنها یک زمان اعمال تصویرسازی نقشه ($t_{projection}$) همراه با زمان اجرای ابتکار پس‌انداز بهبودیافته ($t_{savings2}$) در جدول ۲ ارائه شده است.

جدول ۱: طول دور روش savings2 مبتنی بر تصویرسازی نقشه (نسخه ساده بدون پیش‌پردازش و نسخه کامل با پیش‌پردازش) در مقایسه با سایر روش‌های ابتکاری بدون استفاده از تصویرسازی نقشه (همه فاصله‌ها برحسب متر هستند.)

اندازه	نسخه	مبتنی بر تصویرسازی نقشه			بدون تصویرسازی نقشه		
(۱۰۰۰)	savings2	AZ	CA	CO	GH	NN	QB
۱۰	ساده	۱۵۸۴۰۱۵۲۴	۱۵۴۶۰۳۶۳۵	۱۵۳۰۵۳۵۳۲	۱۶۷۷۹۸۹۳۹	۱۶۰۶۰۲۰۵۰	۱۷۰۸۰۷۷۱۷
	کامل	۱۵۸۲۸۱۴۹۱	۱۵۴۲۱۵۵۴۰	۱۵۴۳۲۲۵۴۹			
۲۰	ساده	۲۴۷۵۹۴۶۲۱	۲۵۱۵۶۴۴۴۵	۲۴۷۵۱۱۴۶۶	۲۵۸۰۸۸۵۹۲	۲۴۹۷۷۹۸۸۵	۲۷۵۲۸۶۲۳۸
	کامل	۲۴۳۴۲۴۹۸۹	۲۴۸۳۳۲۶۲۱	۲۵۱۸۰۷۷۶۲			
۳۰	ساده	۳۴۹۱۷۱۴۲۸	۳۵۲۴۳۶۲۰۳	۳۴۶۴۹۲۰۲۳	۳۶۶۵۱۵۹۵۲	۳۷۰۲۲۸۴۱۶	۳۷۲۱۹۵۷۷۲
	کامل	۳۴۵۹۰۶۸۳۶	۳۴۶۶۹۱۳۷۹	۳۴۳۴۰۸۱۵۳			
۴۰	ساده	۴۱۶۸۱۵۵۰۴	۴۱۹۳۵۴۰۸۹	۴۱۴۸۷۶۷۷۲	۴۶۰۳۶۶۲۹۳	۴۴۷۷۶۴۵۶۳	۴۵۶۴۹۶۹۲۱
	کامل	۴۱۲۱۹۲۲۲۹	۴۱۵۸۸۶۱۸۷	۴۱۴۸۷۳۲۳۹			
۵۰	ساده	۴۸۰۹۲۷۴۰۷	۴۸۱۵۰۵۶۲۱	۴۷۶۳۲۹۵۹۶	۵۴۷۰۰۹۸۰۳	۵۱۷۱۶۴۹۷۵	۵۱۲۹۶۱۹۴۱
	کامل	۴۷۵۷۶۱۲۵۰	۴۷۶۳۳۹۸۱۲	۴۷۹۰۵۱۷۴۳			
۶۰	ساده	۵۴۳۸۱۵۹۹۷	۵۴۵۴۳۹۸۰۸	۵۴۰۹۰۴۶۵۲	۶۱۶۴۲۰۸۲۰	۵۹۳۷۴۵۹۰۵	۶۰۸۵۴۹۶۳۲
	کامل	۵۳۸۴۸۲۳۰۴	۵۴۰۱۷۱۳۷۰	۵۳۷۸۰۴۲۸۸			
۷۰	ساده	۶۰۵۰۴۹۰۸۴	۶۰۶۲۹۷۵۶۳	۶۰۶۳۶۷۹۳۵	۶۷۵۰۲۱۸۸۰	۶۷۵۲۹۵۷۲۴	۶۵۸۹۴۱۸۴۳
	کامل	۵۹۹۲۶۰۹۶۴	۶۰۰۶۳۷۵۷۹	۵۹۷۶۴۷۸۸۰			
۸۰	ساده	۶۷۳۷۷۶۷۸۸	۶۷۴۲۷۹۹۳۸	۶۷۴۹۰۱۱۰۸	۷۳۰۴۰۶۸۸۷	۷۴۵۳۱۵۱۹۰	۷۳۹۵۶۸۹۷۰
	کامل	۶۶۵۴۴۵۶۵۸	۶۶۷۹۸۷۱۱۶	۶۶۸۶۱۳۳۲۸			
۹۰	ساده	۷۳۴۷۶۰۳۵۳	۷۳۴۷۲۳۸۳۳	۷۳۷۶۴۹۳۴۳	۸۱۴۴۱۵۳۲۱	۸۰۶۹۰۱۲۱۷	۸۰۷۲۵۴۶۰۴
	کامل	۷۳۰۱۰۶۶۴۰	۷۲۸۳۶۷۲۲۵	۷۳۱۴۵۴۹۳۶			
۱۰۰	ساده	۷۹۳۳۲۲۸۲۰	۷۸۹۳۴۷۲۹۸	۷۹۱۷۸۳۹۳۷	۸۸۶۳۷۲۴۱۸	۸۷۲۹۹۳۱۴۰	۸۸۵۴۲۶۵۴۰
	کامل	۷۸۶۲۲۵۴۷۹	۷۸۹۷۷۲۲۵۲	۷۸۷۲۸۱۳۵۲			

جدول ۲: زمان اجرای روش مبتنی بر تصویرسازی نقشه در مقایسه با سایر روش‌های ابتکاری بدون استفاده از تصویرسازی نقشه (همه زمان‌ها برحسب ثانیه هستند.)

اندازه	مبتنی بر سیستم تصویر		بدون سیستم تصویر		
(۱۰۰۰)	$t_{projection}$	$t_{savings2}$	t_{GH}	t_{NN}	t_{QB}
۱۰	۰/۰۱	۰/۱۴	۰/۲۷	۳۳/۸۹	۰/۷۳
۲۰	۰/۰۱	۰/۱۸	۰/۸۷	۱۲۸/۷۶	۲/۹۳
۳۰	۰/۰۱	۰/۲۳	۱/۸۵	۱۸۴/۲۴	۶/۲۱
۴۰	۰/۰۲	۰/۲۵	۳/۲۱	۳۹۷/۷۵	۱۰/۶۸
۵۰	۰/۰۲	۰/۳۳	۴/۹۱	۵۹۲/۲۵	۱۶/۳۹
۶۰	۰/۰۲	۰/۳۷	۶/۷۶	۹۲۸/۷۹	۲۳/۴۳
۷۰	۰/۰۲	۰/۴۱	۹/۰۴	۱۲۵۷/۴۵	۳۰/۹۴
۸۰	۰/۰۲	۰/۴۳	۱۱/۶۱	۱۶۴۷/۵۶	۴۰/۹۳
۹۰	۰/۰۳	۰/۵۲	۱۴/۹۵	۱۶۴۹/۰۶	۴۹/۵۲
۱۰۰	۰/۰۳	۰/۵۸	۱۸/۲۷	۲۳۶۱/۲۱	۶۰/۶۷

نتایج جدول ۱ نشـــان می‌دهد که همواره بهترین جواب روش‌های مبتنی بر تصویرسازی نقشه از بهترین جواب سایر روش‌ها بهتر بوده است. این بهبود به‌طور میانگین بیش از ۹٪ بوده اســـت. برای مثال طول دور نهایی روش مبتنی بر تصویر سازی نقشه Azimuthal به‌ازای $n=۱۰۰۰۰$ نسبت به ابتکار QB بدون تصویرسازی نقشه بهبود ۱۳٪ را نشان می‌دهد. در بین ابتکارهای QB، NN و GH اکثراً کوتاه‌ترین دورها توســـط روش NN به دســـت آمده‌اند. البته با توجه به زمان اجرای ثبت‌شـــده برای این روش در جدول ۲ در مقایســـه با زمان روش‌های QB و GH، این روش ابتکاری کندترین روش اســـت. در هر صـــورت برتری مطلق به‌لحاظ زمان اجرا نیز با روش‌های مبتنی بر تصویر سازی نقشه است. درواقع این زمان اجرا (شامل تصویرسازی نقشه و اعمال روش ابتکاری پس‌انداز بهبودیافته) همواره کم‌تر از ۱ ثانیه بوده که برتری مطلق روش پیشـــنهادی را نشـــان می‌دهد.

برای انجام مقایســـه‌ای عادلانه و کامل، نتایج روش ابتکاری ســـاختنی QB[11] و MTS [۱۲] بر روی نســـخه تصـــویرســـازی‌شـــده از مجموعه داده پیشـــنهادی اعمال شـــده اســـت. برای این منظور مانند آزمایش قبلی ابتدا انواع تصویرسازی نقشه انجام شده و روش ابتکاری بر روی آن اعمال شده است. سپس نتایج مربوط به دور نهایی بر روی مجموعه داده اصلی (قبل از تبدیل فضـا) اعمال شـده اسـت. بهترین نتایج مربوط به انواع روش‌های تصویر سازی نقشه در این آزمایش در جدول ۳ آمده است. در این آزمون از تنظیمات پیش‌فرض MTS که مبتنی بر ساخت درخت پوشای کمینه[12] هست، استفاده شده است. ازآنجاکه روش MTS برای نمونه‌های با تعداد شهر بزرگ‌تر از ۶۰۰۰۰ بسیار کند (بیش از ۱۰ دقیقه) عمل می‌کند، در جدول ۲ فقط بهترین نتایج طول دور برای نمونه‌های کوچک برای MTS گزارش شده است[13]. نتایج جدول ۳ مؤید برتری مطلق روش پیشنهادی به‌لحاظ طول دور نهایی است.

جدول ۳: بهترین نتایج طول دور مربوط به QB، MTS و savings2 در مورد مجموعه داده‌های جغرافیایی مبتنی بر تصویرسازی نقشه

اندازه (۱۰۰۰)	QB	MTS	savings2
۱۰	۲۰۰۵۹۷۳۲۲	۲۹۹۴۶۳۳۹۲	۱۵۳۰۵۳۵۳۲
۲۰	۳۳۴۸۳۳۸۶۵	۵۲۹۵۲۷۲۰۸	۲۴۳۴۲۴۹۸۹
۳۰	۵۱۵۱۹۸۹۸۶	۷۸۵۰۹۴۹۷۷	۳۴۳۴۰۸۱۵۳
۴۰	۵۸۳۲۲۳۷۳۰	Error	۴۱۲۱۹۲۲۲۹
۵۰	۶۹۵۵۶۸۱۰۴	۱۲۳۱۹۹۹۹۹۹	۴۷۵۷۶۱۲۵۰
۶۰	۸۴۱۴۴۷۸۶۸	۱۴۶۵۲۹۵۴۳۱	۵۳۷۸۰۴۲۸۸
۷۰	۹۳۴۷۶۱۴۷۸	-	۵۹۷۶۴۷۸۸۰
۸۰	۹۷۸۵۱۸۲۲۲	-	۶۶۵۴۴۵۶۵۸
۹۰	۱۰۷۰۸۳۸۳۳۰	-	۷۲۸۳۶۷۲۲۵
۱۰۰	۱۱۸۸۸۳۹۹۲۲	-	۷۸۶۲۲۵۴۷۹

برای مثال در آزمایش‌های انجام‌شـــده در این جدول، نتایج savings2 همواره کم‌تر از نصـف طول دور به‌دســـت آمده از MTS اســـت. همچنین طول دور روش QB همواره حـداقـل ۳۰٪ بیش‌تر از طول دور روش savings2 است. نکته قابل توجه دیگر روند تقریبی بهبود نسبی طول دور روش savings2 به طول دور روش QB است که به‌صورت صعودی می‌باشد و روش پیشـنهادی را به‌عنوان گزینه مناسـبی برای نمونه‌های بزرگ TSP معرفی می‌کند.

به‌منظور نشـــان دادن توانایی روش پیشـــنهادی در مقایســـه با ســـایر ابتکارهای ســـاختنی حل TSP، الگوریتم پیشـنهادی روی ۵ نمونه از TSP مربوط به نقاط جغرافیایی کشورها در فضای اقلیدسی دوبعدی اعمال شد. این نمونه‌ها از مجموعه داده TSPLIB[6] به‌عنوان مجموعه استاندارد گرفته شده است و در مراجع جدید مانند [۱۲] به آن‌ها اشاره شده است. نتایج طول دور savings2 در مقایسه با QB در جدول ۴ نشان داده شده است. زمان اجرا برای savings2 همواره کم‌تر از QB بود. چون این زمان ها

همواره حدود ۱ ثانیه بوده است، مقدار آن‌ها ذکر نشده است. مجدداً مشاهده می‌شود روش پیشنهادی نسبت به یکی از بهترین روش‌های ابتکار ساختنی موجود یعنی QB که در ابزار linkern استفاده شده است، به‌لحاظ طول دور نهایی برتری مطلق دارد.

جدول ۴: مقایسه روش‌های savings2 و QB برای نمونه‌هایی از کشورها

مجموعه داده	نام کشور	اندازه	savings2	QB
usa13509	آمریکا	۱۳۵۰۹	۲۱۶۵۴۹۰۰	۲۳۴۱۸۴۷۵
brd14051	آلمان	۱۴۰۵۱	۵۰۲۹۶۳	۵۴۵۳۸۰
d15112	آلمان	۱۵۱۱۲	۱۶۸۲۳۳۰	۱۷۹۴۱۸۸
d18512	آلمان	۱۸۵۱۲	۶۹۰۲۷۳	۷۳۵۲۶۵
ch71009	چین	۷۱۰۰۹	۴۹۱۴۵۴۰	۵۲۹۵۶۱۸

۵- نتایج و پیشنهادها

در این مقاله روشی برای حل مسئله فروشنده دوره‌گرد با معیار فاصله جغرافیایی بر روی کره زمین ارائه شد. این روش در ابتدا داده‌های مکانی را با استفاده از تصویرسازی نقشه از فضای جغرافیایی به فضای اقلیدسی منتقل می‌کند و سپس نسخه بهبودیافته‌ای از روش ابتکاری پس‌انداز را بر روی این نمونه جدید اعمال می‌کند. نتایج تجربی نشان‌دهنده برتری نسبی ۱۰٪ مربوط به‌روش پیشنهادی به‌لحاظ کیفیت جواب نهایی برحسب طول دور نسبت به‌روش QB است. همچنین روش پیشنهادی در مقایسه با سایر روش‌های ابتکاری ساختنی بسیار سریع‌تر هست و برای کاربردهای نزدیک به بلادرنگ و پویا [۲۷، ۲۸] مناسب به‌نظر می‌رسد. نتیجه اینکه این مطالعه نشان داد استفاده از روش تبدیل فضا برای حل GEOTSP مبتنی بر تصویر سازی نقشه نسبت به تلاش برای حل آن در فضای اولیه در صورت انتخاب مناسب روش تصویر سازی مورد استفاده موفقیت قابل‌توجهی به‌لحاظ کیفیت و زمان دستیابی به پاسخ خواهد داشت. استفاده از مقیاس‌گذاری چندبعدی (MDS) برای تبدیل فضای مسئله فروشنده دوره‌گرد با معیار فاصله جغرافیایی یکی از روش‌های توسعه ایده این مقاله است که در کارهای آتی به آن توجه خواهد شد. ایده دیگر استفاده از روش پیشنهادی برای کاربردهای دنیای واقعی است. نمونه‌ای از این کاربردها در [۲۹] در مورد برنامه‌ریزی مسیر برای هواپیمای بدون سرنشین ارائه شده است. ترکیب روش پیشنهادی با یکی از الگوریتم‌های تکاملی جدید مانند الگوریتم بهینه‌سازی فاخته [۳۰] نیز ازجمله ایده‌های ادامه کار جاری محسوب می‌شود.

مراجع

[1] E. L. Lawler, *The Travelling Salesman Problem: A Guided Tour of Combinatorial Optimization*: John Wiley & Sons, 1985.

[2] R. Dewil, P. Vansteenwegen, and D. Cattrysse, "A review of cutting path algorithms for laser cutters," *The International Journal of Advanced Manufacturing Technology*, vol. 87, no. 5, pp. 1865-1884, 2016.

[3] Z. Lin *et al.*, "Tool path generation for multi-axis freeform surface finishing with the LKH TSP solver," *Computer-Aided Design,* vol. 69, pp. 51-61, 2015.

[4] G. Laporte, "A concise guide to the traveling salesman problem," *Journal of the Operational Research Society,* vol. 61, no. 1, pp. 35-40, 2010.

[5] R. M. Karp, "Reducibility among combinatorial problems," *Complexity of Computer Computations: Proceedings of a symposium on the Complexity of Computer Computations*, pp. 85-103: Springer, 1972.

[6] C. Rego *et al.*,"Traveling salesman problem heuristics: leading methods, implementations and latest advances," *European Journal of Operational Research,* vol. 211, no. 3, pp. 427-441, 2011.

[7] D. L. Applegate *et al.*, *The Traveling Salesman Problem: A Computational Study: A Computational Study*: Princeton university press, 2011.

[8] G. Clarke, and J. W. Wright, "Scheduling of vehicles from a central depot to a number of delivery points," *Operations research,* vol. 12, no. 4, pp. 568-581, 1964.

[9] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Communications of the ACM,* vol. 18, no. 9, pp. 509-517, 1975.

[10] H. Paessens, "The savings algorithm for the vehicle routing problem," *European Journal of Operational Research,* vol. 34, no. 3, pp. 336-344, 1988.

[11] A. Czumaj, "Euclidean Traveling Salesman Problem," *Encyclopedia of Algorithms*, M.-Y. Kao, ed., pp. 1-6, Berlin, Heidelberg: Springer Berlin Heidelberg, 2014.

[12] A. B. Kahng, and S. Reda, "Match twice and stitch: a new TSP tour construction heuristic," *Operations Research Letters,* vol. 32, no. 6, pp. 499-509, 2004.

[13] J. Faigl *et al.*, "An application of the self-organizing map in the non-Euclidean Traveling Salesman Problem," *Neurocomputing,* vol. 74, no. 5, pp. 671-679, 2011.

[14] K. M. Curtin, "Network analysis in geographic information science: Review, assessment, and projections," *Cartography and Geographic Information Science,* vol. 34, no. 2, pp. 103-111, 2007.

[15] A. L. Walwyn, and D. J. Navarro, "Minimal paths in the city block: Human performance on Euclidean and non-Euclidean traveling salesperson problems," *The Journal of Problem Solving,* vol. 3, no. 1, pp. 5, 2010.

[16] T. Moore, *Implementing the Held-Karp Lower Bound Algorithm in Python*, 2015.

[17] R. Mortazavi, and S. Jalili, "Fast data-oriented microaggregation algorithm for large numerical datasets," *Knowledge-Based Systems,* vol. 67, pp. 195-205, 2014.

[18] H. Okano, S. Misono, and K. Iwano, "New TSP construction heuristics and their relationships to the 2-Opt," *Journal of Heuristics,* vol. 5, no. 1, pp. 71-88, 1999.

[19] J. L. Bentley, "Experiments on traveling salesman heuristics," *Proceedings of the first annual ACM-SIAM symposium on Discrete algorithms*, pp. 91-99, San Francisco, California, USA: Society for Industrial and Applied Mathematics, 1990.

[20] D. S. Johnson, and L. A. McGeoch, "Experimental analysis of heuristics for the STSP," *The traveling salesman problem and its variations*, pp. 369-443: Springer, 2007.

[21] G. A. Croes, "A method for solving traveling-salesman problems," *Operations research,* vol. 6, no. 6, pp. 791-812, 1958.

[22] S. Rees, and R. C. Ball, "Criteria for an optimum simulated annealing schedule for problems of the travelling salesman type," *Journal of Physics A: Mathematical and General,* vol. 20, no. 5, pp. 1239-1239, 1987.

[23] J. Knox, and F. Glover, *Comparative Testing of Traveling Salesman Heuristics Derived from Tabu Search, Genetic Algorithms, and Simulated Annealing*, 1989.

[24] K. Helsgaun, "General k-opt submoves for the Lin--Kernighan TSP heuristic," *Mathematical Programming Computation,* vol. 1, no. 2-3, pp. 119-163, 2009.

[25] B. L. Golden, T. L. Magnanti, and H. Q. Nguyen, "Implementing vehicle routing algorithms," *Networks,* vol. 7, no. 2, pp. 113-148, 1977.

[26] J. P. Snyder, "Album of Map Projections, United States Geological Survey Professional Paper," *United States Government Printing Office,* vol. 1453, 1989.

[27] H. Zhang *et al.*, "The Steiner traveling salesman problem with online advanced edge blockages," *Computers & Operations Research,* vol. 70, pp. 26-38, 2016.

[28] T. Cheong, and C. C. White, "Dynamic traveling salesman problem: Value of real-time traffic information," *IEEE Transactions on Intelligent Transportation Systems,* vol. 2, no. 13, pp. 619-630, 2012.

[۲۹] هوشنگ جعفری و حامد خراطی، «طراحی و توسعه سیستم برنامه‌ریزی مسیر سه‌بعدی برای هواپیمای بدون سرنشین،» مجله مهندسی برق دانشگاه تبریز، جلد ۴۶، شماره ۳، صفحه ۹۴-۸۳، ۱۳۹۵.

[۳۰] زینب صادقی چوینلی و سیدمحمدحسین معطر، «زمان‌بندی سیستم‌های تولید کارگاهی انعطاف‌پذیر با استفاده از الگوریتم جستجوی فاخته بهبودیافته با خوشه‌بندی مارکوف و پرواز لوی،» مجله مهندسی برق دانشگاه تبریز، جلد ۴۶، شماره ۳، صفحه ۱۹۴-۱۸۵، ۱۳۹۵.

[1] Combinatorial algorithms

[2] http://www.math.uwaterloo.ca/tsp/concorde/downloads / codes/cygwin/linkern.exe.gz

[3] Geographic Information System (GIS)

[4] Map projection

[5] http://www.math.uwaterloo.ca/tsp/world/countries.html

[6] Quick Boruvka (QB)

[7] Heap

[8] Equidistant

[۹] به این مجموعه داده در TSPLIB به‌عنوان معیار ارزیابی در مسابقات اشاره شده است.

[10] http://www.math.uwaterloo.ca/tsp/concorde/downloads/ codes/cygwin/linkern.exe.gz

[۱۱] در بین روش‌های ابتکاری روش QB جدیدتر است و به مصالحه بهتری بین زمان اجرا و کیفیت جواب نهایی برحسب طول دور دست می‌یابد. بنابراین در ادامه این بخش این روش برای سایر مقایسه‌ها انتخاب شد.

[12] Minimum Spanning Tree (MST)

[۱۳] برای اندازه ۴۰۰۰۰ برنامه MTS به‌دلیل نامعلومی با خطا مواجه شد.