View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by Simorgh Research Repository

کاوش و بهره‌گیری موثر براساس رهیافت ابرابتکاری برای حل مسئله فروشنده دوره‌گرد

میترا منتظری[۱*]؛ حسین نظام آبادی پور[۲]؛ عباس بحرالعلوم[۳]

چکیده

الگوریتم‌های ابتکاری یکی از نامزدهای اساسی برای حل مسائل NP به شمار می آیند. این الگوریتم ها با ایجاد مصالحه بین قابلیتهای کاوش و بهره‌گیری سعی در یافتن جواب بهینه در یک زمان قابل قبول دارند. بر این اساس ایده استفاده از روشهای ابتکاری که ترکیبی از یک الگوریتم ابتکاری فرامحلی به منظور کاوش فضای جستجو و یک الگوریتم ابتکاری محلی به منظور بهره-گیری از جواب‌های خوب شکل گرفته است. در این الگوریتم‌های ابتکاری ترکیبی الگوریتم جستجوی محلی عموماً وابسته به نوع مساله است و به عبارتی این جستجوگرها مساله گرا هستند. این مسئله می تواند قدرت بهره‌گیری یک الگوریتم ترکیبی را به شدت کاهش داده و احتمال از دست دادن جواب بهینه را افزایش دهد.

در این مقاله حل مساله فروشنده دوره‌گرد با یک الگوریتم پیشنهادی ابرابتکاری که الگوریتم جستجوی محلی را مستقل ازنوع مساله در نظر می‌گیرد پیشنهاد و پیاده سازی می شود. رهیافت ابرابتکاری از دو سطح اصلی تشکیل شده است. در سطح پایین تعدادی جستجوی محلی، که پیمایشگرهایی در همسایگی راه حل هستند، تشکیل شده است. در سطح بالا یک تابع انتخاب وجود دارد که جستجوی محلی مناسب را باتوجه به ناحیه جستجو جاری انتخاب می‌کند. این انتخاب بر اساس تاریخچه عملکرد هر جستجوی محلی صورت می‌گیرد. در روش پیشنهادی از ۶ جستجوهای محلی استفاده شده است. تابع انتخاب استفاده شده بر اساس یادگیری تقویتی است به طوری که جستجوی محلی که تاریخچه عملکرد بهتری داشته باشد شانس انتخاب بالاتری دارد. به منظور افزایش کارآیی الگوریتم یک جستجوی فرامحلی مانند الگوریتم ژنتیک استفاده شده است. آزمایش های انجام شده بر روی پایگاه داده‌های استاندارد برای مسئله فروشنده دوره‌گرد، برتری روش پیشنهادی بر روش‌های ترکیبی را تأیید می کند.

کلمات کلیدی

رهیافت ابرابتکاری، الگوریتم فراابتکاری، الگوریتم ممتیک، جستجوی محلی، مساله فروشنده دوره‌گرد.

Exploring and Exploiting Effectively Based Hyper-Heuristic Approach for Solving Travelling Salesman Problem

Mitra Montazeri; Hossein Nezamabadi-pour; Abbas Bahrololoum

ABSTRACT

Heuristic algorithms are one of the major volunteer for solving NP problems. These algorithms by trading off between exploration and exploitation attempt to find an optimum solution in a reasonable time. Therefore, heuristic studies which are combination of global heuristic algorithm for exploring solution space and local heuristic algorithm for exploiting solution space have been attended. In these combinational heuristic algorithms, local heuristic algorithm is problem oriented. This issue can decrease a capability of exploiting of combinational heuristic algorithms and cause to decrease a probable of finding an optimal solution.

In this paper, we propose a new optimization algorithm based on Hyper-Heuristic for solving TSP which uses local searches with domain-independent. A hyper-heuristic approach has two levels. In low level, it has some local searches which search neighborhood of solution and in high level it has choice function which

[1] دانشجوی کارشناسی ارشد هوش مصنوعی، گروه مهندسی کامپیوتر، دانشگاه شهید باهنر، کرمان، mmontazeri@eng.uk.ac.ir
[*] عضو انجمن پژوهشگران جوان، دانشگاه شهید باهنر کرمان.
[۲] دانشیار، گروه مهندسی برق، دانشگاه شهید باهنر، کرمان، nezam@mail.uk.ac.ir
[۳] مربی، گروه مهندسی کامپیوتر، دانشگاه شهید باهنر، کرمان، A.Bahrololoum@mail.uk.ac.ir

select a proper local search depended on characteristics of the region of the solution space that is currently under exploration and also the performance history of local searches. In the proposed method, we use ٦ local searches and our choice function based on reinforcement learning. In our choice function, the local search that has better performance history has high chance to be chosen. In aim of improving efficiency of our method we use a global search algorithm, Genetic Algorithm. Empirical results on standard databases of TSP confirm the efficiency of the proposed method in comparison with combinational heuristic algorithms.

KEYWORDS

Hyper-Heuristic Approach, Meta-Heuristic, Memetic Algorithm, Local Search, Travelling Salesman Problem.

۱. مقدمه

مساله فروشنده دوره گرد[۱]،TSP، یک مساله بهینه سازی ترکیبیاتی است. در این مسئله n تا شهر وجود دارد و هزینه رفتن مستقیم از هر شهر به شهر دیگر مشخص است، حال هدف بدست آوردن کم هزینه ترین مسیر است به طوری که سفر از یک شهر آغاز شود، از تمام شهرها دقیقاً یکبار گذشته و به شهر اول باز گردد.

این مساله اولین بار در سال ۱۹۳۰به عنوان یک مساله ریاضی مطرح شد، بعدها به عنوان یک مساله بهینه سازی مورد مطالعه قرار گرفت. در ابتدا راه حل های دقیقی مانند برنامه نویسی پویا[۲] و برنامه نویسی دقیق[۳] برای این مساله ارائه شدند که می توانستند جواب دقیق را برای این مساله پیدا کنند. اما در تئوری محاسبات چون این مسائل جزء مسائل سخت هستند، پیچیدگی محاسباتی آن نمایی است[۱]. بنابراین با افزایش تعداد شهرها زمان اجرای آن به طور نمایی افزایش می یابد به طوری که حل این مساله با تعداد شهر های زیاد غیر عملی است. این موضوع باعث شده است که تمایل محققان به حل این مسائل از طریق الگوریتم های ابتکاری جلب شود.

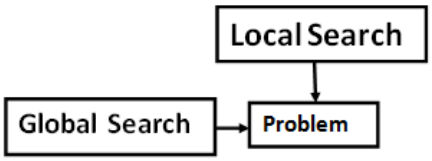
روش های ابتکاری روشهای هستند که جواب بهینه را در یک زمان معقول بدست می آورند، ولی این روش ها تضمینی برای بدست آوردن جواب بهینه ندارند. به طور دقیق تر می توان گفت که امکان گیر افتادن این الگوریتم ها در بهینه محلی وجود دارد. در نتیجه الگوریتم های متفاوتی با ایده های گوناگون سعی بر حداقل کردن این مشکل و در نتیجه یافتن بهینه عمومی که همان جواب مساله است، دارند. آنها با جستجو در فضای مساله(کاوش[۴]) و تمرکز بر روی راه حل های خوب (بهره گیری[۵])، سعی در پیدا کردن جواب بهینه مسئله دارند. این الگوریتم ها که الگوریتم های فرا ابتکاری[۶] نام دارند با استفاده از این روش توانسته اند احتمال گیر افتادن در بهینه محلی را تا حد قابل توجهی کاهش دهند[۲].

در مرجع [۳] از الگوریتم وراثتی برای یافتن مسیر بهینه در مساله فروشنده دوره گرد استفاده شده است. روش آنها از هم بری[۷] و جهش های[۸] خاصی مثل هم بری رتبه[۹]، هم بری جزئی[۱۰] و هم بری چرخه[۱۱] برای حل این مساله استفاده شده بود. روش آنها با استفاده از این عملگرها توانسته بود جوابهای مناسبی را در زمان معقول بدست آورد. در مرجع [۴] از مورچه های مصنوعی استفاده شده است تا با استفاده از اطلاعاتی که تحت عنوان فرمون برروی لبه های گراف مساله فروشنده دوره گرد قرار می گیرد، مولفه های سفر بهینه را به طور متوالی پیدا کنند. نتایج آنها در مقایسه با روش های دیگر رضایت بخش بوده است. در مرجع [۵] روشی ارائه شده است که بر مبنای الگوریتم سیستم ایمنی راه حلهای خوبی برای مساله فروشنده دوره گرد تولید کرده است و سرعت همگرایی آن در مقایسه با الگوریتم مورچه به طور قابل توجهی افزایش یافته است.

اگرچه الگوریتم های فرا ابتکاری نسبتاً برای حل مسائل دنیای واقعی مثل TSP موثر واقع شده است؛ اماچند مشکل وجود دارد که باعث شده است این الگوریتم ها نتوانند جواب گوی خوبی برای حل این مسائل باشند. از جمله این مشکلات این است که روشهای فرا ابتکاری اغلب مساله گرا هستند. همان طوری که در [۲] بحث شده است، بکارگیری رویه های فرا ابتکاری نیازمند داشتن تخصص هم در زمینه مساله و هم در تعیین پارامترهای الگوریتم برای حل مساله دارند که این خود نیازمند داشتن مهارت و تجربه زیاد است. این الگوریتم ها نیازمند دانش و تجربه درباره حوزه و مشخصات مساله و تنظیم پیشرفته پارامترها هستند . از طرف دیگر، طبق تئوری No Free Lunch [۶]، هیچ الگوریتمی وجود نداردکه برای همه مسائل بهترین باشد و همه الگوریتم ها میانگین اجرایی یکسانی برروی همه مسائل دارند. این تئوری به این معناست که هر الگوریتم برای مجموعه ای از مسائل، کارایی خوب و برای مسائل دیگر کارایی خوبی ندارد . در نتیجه میانگین اجرای الگوریتم ها برای همه مسائل یکسان است که بیانگر این مطلب است که هر الگوریتم نقاط ضعف و قوت خاص خود را دارد. در نتیجه نیاز به الگوریتمی است که بتواند قدرت الگوریتم ها را ترکیب و ضعف آنها را کاهش دهد ودر نتیجه بتواند برای تعداد زیادی از مسائل مفید باشد.

از طرفی اگرچه الگوریتم فرا ابتکاری توانایی جستجوی فرامحلی[۱۲] و جستجو ناحیه های جدید فضای راه حل[۱۳] برای پیدا کردن راه حل های خوب را دارند، اما آنها پس از یافتن یک راه حل خوب دیگر تمرکز بیشتری بر این ناحیه انجام نمی دهند[۷]، که این خود امکان از دست دادن راه حل خوب است. چون هنگامی که راه حل خوبی پیدا می شود احتمال وجود جوابهای بهتر در اطراف این راه حل وجود دارد. در نتیجه الگوریتم های ممتیک بر روی کار آمدند[۸]، قالب کلی الگوریتمهای ممتیک در شکل (۱) نشان داده شده است. الگوریتم های ممتیک الگوریتم های فرا

ابتکاری را با یک جستجوی محلی در اطراف راه حل های خوب بهبود می دهند. تحقیقات نشان می دهد ترکیب جستجوی محلی با الگوریتم فرا ابتکاری باعث کارآمد شدن جستجو می شود[۹]



نمودار (۱) قالب کلی الگوریتم ممتیک

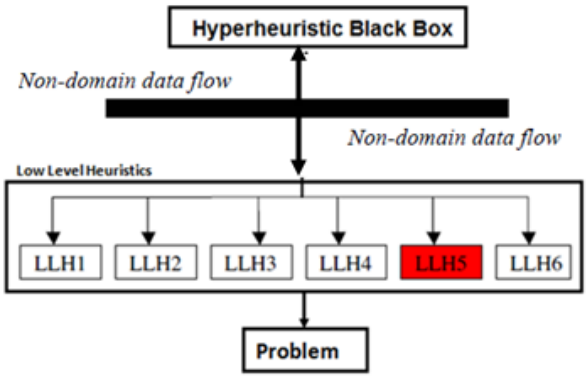
الگوریتم های ممتیک در تمام مراحل جستجو در فضای راه حل از یک جستجوی محلی استفاده می کنند که این خود انعطاف پذیری الگوریتم را به شدت کاهش می دهد. با توجه به اینکه، در هنگام جستجو در فضای راه حل، هر ناحیه از فضای راه حل ویژگی های خاص خود را داردوهر فضا با فضای دیگر متفاوت است، در نتیجه به الگوریتمی نیاز است که متشکل از چندین جستجوی محلی و یک تابع انتخاب باشد تا در هر مرحله از جستجو الگوربتم جستجوی محلی مناسب را انتخاب کند. از این رو، برای حل مشکلات بیان شده رهیافت جدیدی تحت عنوان ابرابتکار ایجاد شده است که می تواند نیازهای فوق را برطرف کند. تحقیقات نشان می‌دهد به‌کاربستن رهیافت ابرابتکاری باعث تولید جواب‌های بهتری می گردد[۲۳ ,۲۲]. در این مقاله مبتنی بر رهیافت ابر ابتکاری روشی برای حل مساله فروشنده دوره گرد ارائه شده است.

ادامه این مقاله این گونه سازمان دهی شده است که در بخش دوم الگوریتم ابرابتکاری به طور کامل توضیح داده می شود. در بخش ۳ مساله فروشنده دوره گرد شرح داده می شود. در بخش ۴ الگوریتم پیشنهادی ارائه می شود. در انتها، در بخش های ۵ و۶ به ترتیب پیاده سازی و جمع بندی بیان می شود.

۲– الگوریتم های ابرابتکاری

۲–۱– مفاهیم پایه

رهیافت ابرابتکار اولین بار در سال ۲۰۰۰در مرجع[۱۰] ارائه شده است. نمودار بلوکی این رهیافت در شکل (۲) نشان داده شده است. این الگوریتم از دو لایه تشکیل شده است : لایه اول شامل تعدادی جستجوی محلی است که به آن ابتکارات سطح پایین[۱۴]،LLH، گفته می شود. این جستجوگرهای محلی شامل قواعد یا راهبردهای متفاوتی برای حل مساله اند و وابسته به نوع مساله می باشند. اگرچه ابتکارات سطح پایین می توانند خود الگوریتم فرا ابتکاری باشند، ولی آنها اغلب ابتکارات ساده ای هستند که با توجه به مساله طراحی می شوند. در نتیجه وابسته به مساله (مساله گرا) بوده و از یک مساله تا مساله دیگر فرق می کنند.



نمودار (۲) قالب کلی رهیافت ابرابتکاری

دومین لایه ابرابتکار جعبه سیاه است که بدون هیچ مسئله خاصی طراحی می شود. این لایه اغلب به اطلاعاتی که به مساله وابسته نیستند (مثل اختلاف در تابع هدف ، تاریخچه اجرای هر ابتکار، حالات راه حل و....) دسترسی دارد. وظایف این لایه پذیرش یا عدم پذیرش راه حل جدید و انتخاب ابتکارات سطح پایین بعدی برای هدایت راه حل است. این لایه می تواند یک فرا ابتکاری یا تابع انتخاب[۱۵] باشد. این دو لایه با مرز دامنه جدا می شود.

در واقع جعبه سیاه به عنوان ناظری که انتخاب جستجوهای محلی را در هر زمان از مرحله جستجو را مدیریت می کند. این انتخاب با توجه به مشخصات ناحیه ای از فضای راه حل جاری در حال جستجو است و در هر مرحله با توجه به ناحیه جستجو و تاریخچه عملکرد ابتکارات سطح پایین، ابتکار سطح پایین مناسبی انتخاب و به راه حل جاری اعمال می کند.

در نتیجه رهیافت ابر ابتکاری با طراحی چنین سیستمی ، باعث بهبودی قابل توجهی در الگوریتم های گذشته شده است. این رهیافت با داشتن تعدادی جستجوی محلی و اعمال مناسب آنها در هر مرحله از جستجو، باعث بهبود در الگوریتم های ممتیک شده است. در این مقاله هدف این است که مکانیزم جستجوی محلی موجود در الگوریتم ممتیک با رهیافت ابرابتکار جایگزین شود.

۲-۲- مرور بر کارهای مرتبط

در مرجع [۱۱] الگوریتم ابرابتکار بر مبنای جستجوی ممنوع ارائه شده است. در الگوریتم آنها، ابتدا به هر ابتکار یک وزن داده می شود و سپس برای ارتقاء راه حل جاری، ابتکار سطح پایینی که بیش ترین وزن را داراست، انتخاب می شود. در صورتی که این ابتکار سطح پایین باعث بهبودی در راه حل شود وزن آن افزایش ، در غیر این صورت وزن آن کاهش می یابد و وارد لیست ممنوعه می شود. در هر دو صورت جواب جاری جایگزین جواب قبلی می شود. در مرجع [۱۲] الگوریتم ابرابتکاری بر مبنای SA ارائه شده است که بهبود یافته نسخه الگوریتم [۱۳] است. در الگوریتم آنها ابتکارات سطح پایین به صورت احتمالی انتخاب و به کار بسته می شود. در این الگوریتم ابتدا به هر ابتکار سطح پایین یک وزن مناسب داده شده، سپس بر مبنای احتمال، یک ابتکار انتخاب می شود وبه راه حل جاری اعمال می شود. این ابتکار متناسب با اینکه که باعث ارتقاء یا کاهش شایستگی راه حل می شود تنبه شده یا پاداش می گیرد. در سال ۲۰۰۶ یک ابرابتکار استدلال مبتنی بر حالت[۱۶] برای مساله زمان بندی درسی ارائه شد[۱۴] که می توانست بیش از اینکه مستقیماً برای یافتن جواب استفاده گردد، برای تولید راه حل هایی با کیفیت خوب به کار رود. نتایج حاصل از آزمایش ها نشان می دهد که این سیستم می تواند به طور کارآمد و هوشمندانه در تولید جداول زمان بندی به کار رود.

در مرجع [۲] یک ابرابتکار بر مبنای الگوریتم مورچه، برای حل مساله زمان بندی ارائه پروژه[۱۷] ومساله مسابقه سیار[۱۸] ارائه شده است. در [۱۵] یک ابرابتکار برمبنای الگوریتم وراثتی ارائه شده است، که می تواند به راحتی برای مسائل مختلف مورد استفاده قرار گیرد. هر فرد در جمعیت رشته ای از ابتکارات سطح پایین است که نشان می دهد که کدام ابتکار وبه چه ترتیبی به کاربسته گردد. این روش برای حل مساله زمان بندی کامیون به کار رفته است که منجر به تولید جوابهای موثر شده است. رویه آنها در [۱۶] ارتقاء یافت به طوری که به جای داشتن کروموزوم ها با طول ثابت از کروموزوم ها با طول متغیر استفاده می شد. یک ابرابتکار جدید بر مبنای رتبه بندی ایستا ابتکارات سطح پایین در [۱۷] ارائه شده است. که در آنها از تابع انتخابی استفاده شده است که اطلاعات مربوط به عملکرد اخیر را جمع آوری می کند. تابع انتخاب شامل سه مجموعه اطلاعات است. اطلاعات اول مربوط به عملکرد هر ابتکار سطح پایین است ، اطلاعات دوم مربوط به عملکرد هر جفت ابتکار سطح پایین است و اطلاعات سوم مربوط به زمان اجرای هر ابتکار سطح پایین است. در [۱۸] از ابرابتکار مبتنی بر الگوریتم همسایگی متغیر[۱۹] استفاده کرده است ، که باعث تولید جواب های خوب برای مساله مورد تست شده است.

۳- مساله فروشنده دوره گرد

مساله فروشنده دوره گرد یک مساله بهینه سازی ترکیبیاتی است. در این مسئله، فروشنده ای می خواهد با شروع از یک شهر و عبور از n شهر، اجناس خود را بفروشد به طوری که از هر شهر یک بار و فقط یک بار گذشته و به شهر اول باز گردد. در این مساله هدف پیدا کردن کم هزینه ترین مسیر است.

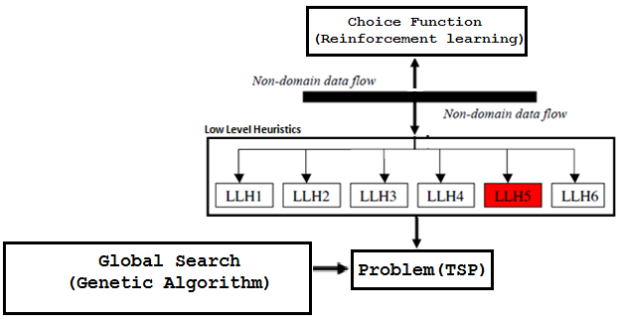
به بیان ریاضی، TSP به وسیله یک گراف وزن دار کامل به صورت G(V,E,d) قابل بیان است که در آن V شامل مجموعه ای ازگره ها (شهرها)، E مجموعه ای از یال ها و $d: E \to R$ یک تابع وزن دار است که عدد مثبت $d(c_i,c_j)$ را به یال متصل کننده گره i ام و گره j ام نسبت می دهند. در این مساله، هدف از بهینه سازی، یافتن کوتافتن کوتاهترین مسیر بسته ای است که از تمام شهرها و از هریک تنها یک بار گذشته باشد. در TSP قصد داریم کوتاه ترین سفر را که یک جایگشت[۲۰] π از شهرهای $\{c_1,c_2,...c_n\}$ است را بگونه ای بیابیم که $f(\pi)$ آن کمینه است. $f(\pi)$ با رابطه (۱) محاسبه می شود.

$$f(\pi)=\sum_{i=1}^{n} d(c_\pi(i),c_\pi(i+1)) + d(c_\pi(n),c_\pi(1)) \qquad (۱)$$

کنفرانس داده کاوی ایران

۴- الگوریتم پیشنهادی

بلوک دیاگرام رهیافت پیشنهادی در شکل (۳) ارائه شده است. همانطور که در این شکل نشان داده شده است، الگوریتم پیشنهادی از ۳ جزء اصلی تشکیل شده است. الف)ابتکارات سطح پایین که مجموعه ای از ابتکارات است که منجر به تغییرات کوچک در مسیر جاری برای حل مساله TSP می گردد. ب)ابتکار سطح بالا (جعبه سیاه) که انتخاب ابتکارات سطح پایین را بر مبنای یادگیری تقویتی انجام می دهد. ج)جستجوی فرامحلی که از الگوریتم وراثتی برای کاوش بهتر فضای جستجو استفاده می کند. در ادامه هر یک از این اجزا به طور کامل توضیح داده می شوند.



نمودار(۳) الگوریتم پیشنهادی

۴–۱– ابتکارات سطح پایین

رهیافت ابرابتکاری شامل مجموعه ای از ابتکارات سطح پایین است که توسط مهارت انسانی بر اساس مساله ایجاد می شوند. در این الگوریتم از ۶ ابتکارات سطح پایین استفاده شده است که راه حل جاری را می گیرند و به منظور ارتقاء راه حل ، آن را با روشی ساده تغییر می دهند. در ابتکار سطح پایین اول[۱۹] به طور تصادفی دو لبه از تور داده شده حذف می کنیم و دو مسیر به دست آمده را به هم وصل می کنیم که تنها به یک طریق می توان این کار را انجام داد. در ابتکار سطح پایین دوم [۱۹]، تغییر کوچکی در تور جاری انجام می دهد به طوری که دو لبه (a,b) و (c,d)که a#d و b#c، به طور تصادفی انتخاب می کند و به ترتیب با(b,d) و (a,c) جابه جا می کند. در ابتکار سطح پایین چهارم[۱۹] ابتکار دوم در صورتی اجرا می کند که تور کوتاهتری را تولید کند. در ابتکار سطح پایین پنجم[۲۰]، ابتکار سطح پایین اول را در صورتی اجرا می کند که تور کوتاهتری را تولید کند. . در ابتکار سطح پایین ششم [۲۰]سه لبه جدا از تور داده شده را به طور تصادفی حذف می کند سپس این سه مسیر به دست آمده دوباره به هم وصل می کند در نتیجه تور جدیدی حاصل می شود. در ابتکار سطح پایین سوم[۲۰]، ابتکار سطح پایین ششم را در صورتی اجرا می کند که تور کوتاهتری را تولید کند.

۴–۲– ابتکار سطح بالا (جعبه سیاه)

در این ابتکار از یک تابع انتخاب بر مبنای یادگیری تقویتی برای انتخاب و کنترل مناسب ابتکارات سطح پایین استفاده شده است به طوری که در ابتدای کار همه ابتکارات سطح پایین دارای وزن(پاداش) مساوی برای داشتن تعداد کروموزوم های جمعیت دارند. این وزن در طی نسلها تغییر می کند وبر اساس عملکرد هر یک از ابتکارات سطح پایین، افزایش یا کاهش می یابد. می توان مطالب فوق را به صورت روابط (۲) ، (۳) و (۴) نشان داد:

$$w_i(0) = \frac{1}{NL} \qquad i = 1,2,...,NL; \tag{۲}$$

$$w_i(t+1) = \frac{w_i(t) + R_i}{\sum_{i=1}^{NL} w_i(t+1)} \tag{۳}$$

$$R_i = \frac{RN_i}{TN_i} \tag{۴}$$

که در روابط فوق NL تعداد کل ابتکارات سطح پایین، $w_i(t+1)$ وزن مربوط به ابتکار سطح پایین i ام در زمان $t+1$ ، RN_i و TN_i به ترتیب تعداد پاداشهای کسب شده و کل راه حل های مربوط به ابتکار سطح پایین i ام است.

۴–۳– جستجوی فرامحلی

جستجوی فرامحلی به کاربرده شده در این تحقیق الگوریتم وراثتی است. الگوریتم وراثتی می تواند فضای جستجو را به طور بسیار مناسب کاوش کند. اما این الگوریتم توانایی کافی برای بهره گیری ندارد. برای افزایش کارایی الگوریتم پیشنهادی، تعداد راه حل هایی که تحت تأثیر ابتکارات سطح پایین بهبود می یابد را به صورت تابعی از زمان در نظر می گیریم. تعداد این راه حل ها در ابتدا کم در نظر گرفته شده و با گذر زمان افزایش می یابد. برای این منظور از یک تابع خطی استفاده می شود.

۵– پیاده سازی و نتایج

این الگوریتم بر روی پایگاه داده TSPLIB [۲۱] و [۲۴] اجرا شده است که جواب‌های بدست آمده از اجرای این الگوریتم در جدول (۱) درج شده است. این الگوریتم طی ۲۰ اجرای با تعداد مراحل ۱۰۰۰۰و اندازه جمعیت ۱۴۸ اجرا شده است. به دلیل اینکه الگوریتم پیشنهادی یک الگوریتم تصادفی است و در هر مرحله ممکن است جواب متفاوتی بدهد، میانگین جواب‌ها، در این جدول درج شده است. به منظور نشان دادن برتری این الگوریتم با روش‌های مختلف [۲۴, ۲۵, ۲۶] مقایسه شده است. که در هر سطر بهترین جواب تیره شده است. روش پیشنهادی با

الگوریتم ممتیک [۲۴] مقایسه شده است. در این مقاله از دو نوع الگوریتم ممتیک استفاده شده است (TGMA-HC و SSMA-HC). همچنین روش پیشنهادی با الگوریتم‌های فراابتکاری مانند الگوریتم وراثتی و الگوریتم پرندگان،PSO مقایسه شده است. در آخر در مرجع [۲۶]، روش پیشنهادی با الگوریتم ترکیبی مقایسه شده است(IDGA). در این مقاله نشان داده شده است که روش ارائه شده بهتر از ترکیبی از SA (الگوریتم جستجوی محلی) و GA (الگوریتم جستجوی فرامحلی) (GASA) و تک تک روش‌ها (SA و GA) است. جوابهای بدست آمده حاکی از کارآمد بودن این روش است. در شکل (۴) عملکرد روش پیشنهادی با الگوریتم ژنتیک و الگوریتم ممتیک نشان داده شده است. این نمودارها نشان دهنده عملکرد بسیار مناسب الگوریتم پیشنهادی نسبت به الگوریتم های گذشته است.

لازم به تذکر است در جدول (۲) مقایسه روش پیشنهادی با IDGA منظور از علامت؟ نامشخص بودن پارامتر مدنظر است. به عنوانی دیگر در این مرجع تعدادی از پایگاه داده به منظور مقایسه متناسب با مراجع دیگر ۱۰۰ بار اجرا و میانگین گیری شده و تعدادی دیگر که مقایسه با الگوریتم-های پیاده سازی شده در این مقاله است صرفا بهترین جواب بدون ذکر کردن تعداد اجرا لحاظ شده است.

جدول(۱) میانگین جواب بدست آمده از اجرای روش پیشنهای

پایگاه داده	تعداد شهر	جواب بهینه	میانگین جواب‌ها
C۲۰	۲۰	۶۲۵۷۵	۶۲۵۷۵
C۳۰	۳۰	۶۲۷۱۶	۶۲۷۱۶
C۴۰	۴۰	۶۲۷۶۸	۶۲۷۶۸
F۳۲	۳۲	۸۴۱۸۰	۸۴۴۳۲
F۴۱	۴۱	۶۸۱۶۸	۶۸۱۶۸
S۲۱	۲۱	۶۰۰۰۰	۶۰۰۰۰
Eil۵۱	۵۱	۴۲۶	۴۳۳.۴۵
Att ۴۸	۴۸	۱۰۶۲۸	۱۰۷۶۰
KroA۱۰۰	۱۰۰	۲۱۲۸۲	۲۲۲۳۲

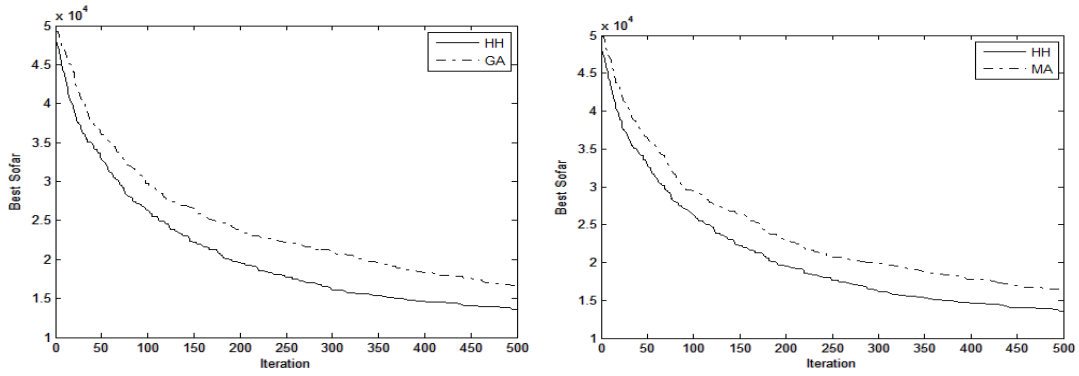
جدول (۲) مقایسه الگوریتم پیشنهادی با روش‌های موجود

روش‌های استفاده شده در مراجع		روش پیشنهادی		جواب بهینه	پایگاه داده
PSO [۲۵] میانگین ۱۰۰ اجرای	۶۳۲۷۶	میانگین ۱۰۰ اجرای	**۶۲۵۷۵**	۶۲۵۷۵	C۲۰
GA [۲۵] میانگین ۱۰۰ اجرای	۶۳۱۸۸	میانگین ۱۰۰ اجرای	**۶۲۵۷۵**		
TGMA-HC [۲۴] میانگین ۱۰۰ اجرای	۱۳۴۴۹۷	میانگین ۱۰۰ اجرای	**۶۲۵۷۵**		
SSMA-HC[۲۴] میانگین ۱۰۰ اجرای	۱۰۷۴۹۳	میانگین ۱۰۰ اجرای	**۶۲۵۷۵**		
IDGA[۲۶] میانگین ۱۰۰ اجرای	۱۱۶۳۶۷	میانگین ۱۰۰ اجرای	**۶۲۵۷۵**		
PSO [۲۵] میانگین ۱۰۰ اجرای	۶۰۷۸۶	میانگین ۱۰۰ اجرای	**۶۰۰۰۰**	۶۰۰۰۰	S۲۱
GA [۲۵] میانگین ۱۰۰ اجرای	۶۰۶۴۸	میانگین ۱۰۰ اجرای	**۶۰۰۰۰**		
TGMA-HC [۲۴] میانگین ۱۰۰ اجرای	۱۲۰۲۷۶	میانگین ۱۰۰ اجرای	**۶۰۰۰۰**		
SSMA-HC[۲۴] میانگین ۱۰۰ اجرای	۹۳۶۲۶	میانگین ۱۰۰ اجرای	**۶۰۰۰۰**		
IDGA[۲۶] میانگین ۱۰۰ اجرای	۱۲۱۴۳۹	میانگین ۱۰۰ اجرای	**۶۰۰۰۰**		
PSO [۲۵] میانگین ۱۰۰ اجرای	۶۳۶۲۵	میانگین ۱۰۰ اجرای	**۶۲۷۱۶**	۶۲۷۱۶	C۳۰
GA [۲۵] میانگین ۱۰۰ اجرای	۶۳۳۵۶	میانگین ۱۰۰ اجرای	**۶۲۷۱۶**		

کنفرانس داده کاوی ایران

		۶۲۷۱۶	میانگین ۱۰۰ اجرای	۱۶۵۷۹۵	TGMA-HC [۲۴] میانگین ۱۰۰ اجرای
		۶۲۷۱۶	میانگین ۱۰۰ اجرای	۱۱۶۶۳۳	SSMA-HC[۲۴] میانگین ۱۰۰ اجرای
		۶۲۷۱۶	میانگین ۱۰۰ اجرای	۱۲۰۰۲۵	IDGA[۲۶] میانگین ۱۰۰ اجرای
F۳۲	۸۴۱۸۰	**۸۴۴۳۲**	میانگین ۱۰۰ اجرای	۸۵۵۳۵	PSO [۲۵] میانگین ۱۰۰ اجرای
		۸۴۴۳۲	میانگین ۱۰۰ اجرای	۸۵۳۹۲	GA [۲۵] میانگین ۱۰۰ اجرای
		۸۴۴۳۲	میانگین ۱۰۰ اجرای	۱۴۶۳۱۷	TGMA-HC [۲۴] میانگین ۱۰۰ اجرای
		۸۴۴۳۲	میانگین ۱۰۰ اجرای	۱۰۸۰۴۸	SSMA-HC[۲۴] میانگین ۱۰۰ اجرای
		۸۴۴۳۲	میانگین ۱۰۰ اجرای	۱۲۴۶۴۰	IDGA[۲۶] میانگین ۱۰۰ اجرای
C۴۰	۶۲۷۶۸	**۶۲۷۶۸**	میانگین ۱۰۰ اجرای	۶۴۲۱۲	PSO [۲۵] میانگین ۱۰۰ اجرای
		۶۲۷۶۸	میانگین ۱۰۰ اجرای	۶۳۷۵۳	GA [۲۵] میانگین ۱۰۰ اجرای
		۶۲۷۶۸	میانگین ۱۰۰ اجرای	۱۹۷۸۲۹	TGMA-HC [۲۴] میانگین ۱۰۰ اجرای
		۶۲۷۶۸	میانگین ۱۰۰ اجرای	۱۲۸۱۱۷	SSMA-HC[۲۴] میانگین ۱۰۰ اجرای
		۶۲۷۶۸	میانگین ۱۰۰ اجرای	۱۲۵۷۲۱	IDGA[۲۶] میانگین ۱۰۰ اجرای
F۴۱	۶۸۱۶۸	**۶۸۱۶۸**	میانگین ۱۰۰ اجرای	۶۹۹۹۵	PSO [۲۵] میانگین ۱۰۰ اجرای
		۶۸۱۶۸	میانگین ۱۰۰ اجرای	۶۹۷۰۲	GA [۲۵] میانگین ۱۰۰ اجرای
		۶۸۱۶۸	میانگین ۱۰۰ اجرای	۱۵۸۴۶۱	TGMA-HC [۲۴] میانگین ۱۰۰ اجرای
		۶۸۱۶۸	میانگین ۱۰۰ اجرای	۱۱۵۸۶۰	SSMA-HC[۲۴] میانگین ۱۰۰ اجرای
		۶۸۱۶۸	میانگین ۱۰۰ اجرای	۱۲۵۴۶۱	IDGA[۲۶] میانگین ۱۰۰ اجرای
Eil۵۱	۴۲۶	**۴۳۴,۲۵**	میانگین ۲۰ اجرای	۴۳۶	PSO [۲۵] میانگین ۱۰۰ اجرای
		۴۳۳,۴۵	میانگین ۲۰ اجرای	۴۳۴	GA [۲۵] میانگین ۱۰۰ اجرای
		۴۲۷	بهترین جواب ۲۰ اجرای	**۴۲۶**	IDGA[۲۶] بهترین جواب ؟ اجرای
Att ۴۸	۱۰۶۲۸	**۱۰۷۶۰**	میانگین ۲۰ اجرای	۱۰۸۶۷	PSO [۲۵] میانگین ۱۰۰ اجرای
		۱۰۷۶۰	میانگین ۲۰ اجرای	۱۰۸۲۸	GA [۲۵] میانگین ۱۰۰ اجرای
KroA۱۰۰	۲۱۲۸۲	**۲۱۳۴۵**	بهترین جواب ۲۰ اجرای	۲۱۳۶۹	IDGA[۲۶] بهترین جواب ؟ اجرای

کنفرانس داده کاوی ایران



(ب) (الف)

نمودار (۴) مقایسه عملکرد اجرایی الگوریتم پیشنهادی. (الف) الگوریتم ممتیک. (ب) الگوریتم ژنتیک.

۶- جمع بندی

در این مقاله روش جدیدی بر مبنای الگوریتم ابرابتکاری ارائه شد. این روش توانایی قابل توجهی در جستجو در فضای مساله دارد. روش پیشنهادی می تواند کاوش و بهره گیری را به طور مناسب انجام دهد. این روش بهره گیری را با توجه به ناحیه جستجو انجام می دهد که باعث می گردد که این الگوریتم مستقل از نوع مساله باشد. الگوریتم پیشنهادی برای حل مساله TSP به کار بسته شد که نتایج نشان داده شده از اعمال این الگوریتم برای حل این مساله نشان دهنده کارایی قابل توجه این الگوریتم است.

مراجع

[۱] R. Neapolitan and K. Naimipour, "*foundations of algorithms using C++ Pseudocode*", third edition, Jones and Bartlett Publisher ۲۰۰۴ (۶۱۸ pages).

[۲] R. F. J. O'Brien." *Ant Algorithm Hyperheuristic Approaches for Scheduling Problems*". Thesis submitted to The University of Nottingham for the degree of Master of Philosophy, August ۲۰۰۷

[۳] B. F. Al-Dulaimi, and H. A. Ali," *Enhanced Traveling Salesman Problem Solving by Genetic Algorithm Technique (TSPGA)*", World Academy of Science, Engineering and Technology ۳۸ (۲۰۰۸).

[٤] M. Dorigo, L. M. Gambardella, "*Ant colonies for the traveling salesman problem*", Accepted for publication in BioSystems, ۱۹۹۷. In press.

[٥] M. Bakhouya and J. Gaber, *"An Immune Inspired-based Optimization Algorithm: Application to the Traveling Salesman Problem"* AMO - Advanced Modeling and Optimization, Volume ۹, Number ۱, ۲۰۰۷.

[٦] D. Wolpert and W. G. MacReady. *"No free lunch theorems for optimization*". IEEE Transactions on Evolutionary Computation, ۱۶۷ ۸۲, ۱۹۹۷.

[۷] J.H. Ang, K.C. Tan, A.A. Mamun ,"*An evolutionary memetic algorithm for rule extraction*" Expert Systems with Applications ۳۷ (۲۰۱۰) ۱۳۰۲–۱۳۱۵

[۸] Ong, Y. S., & Keane, A. J.. "*Meta-Lamarckian learning in memetic algorithms*." IEEE Transactions on Evolutionary Computation, ۸(۲), ۹۹–۱۱۰. (۲۰۰۴)

[۹] Merz, P. & Freisleben, B. "*A comparison of memetic algorithms, Tabu search, and ant colonies for the quadratic assignment problem"*. In Proceedings of the congress on evolutionary computation (Vol. ۱, pp. ۲۰۶۳–۲۰۷۰). (۱۹۹۹)

[۱۰] P. Cowling, G. Kendall, and E. Soubeiga. "*A hyperheuristic approach to scheduling a sales summit"*. In Selected papers of Proceedings of the ۳rd Inter-national Conference on the Practice And Theory of Automated Timetabling, LNCS ۲۰۷۹, pages ۱۷۶ ۱۹۰. Springer, ۲۰۰۱.

[۱۱] Kendall, G. and Mohd Hussin, N., "*An Investigation of a Tabu Search Based Hyper-heuristic for Examination Timetabling*. Selected Papers from MISTA, Kluwer Academic Publisher, ۳۰۹-۳۲۸.(۲۰۰۵)

[۱۲] Bai,R , Burke, E. K. , Kendall, G. and McCollum, B., "*A Simulated Annealing Hyper-heuristic for University Course Timetabling*", E. K. Burke, H. Rudová (Eds.): PATAT ۲۰۰۶, pp. ۳۴۵–۳۵۰. ISBN ۸۰-۲۱۰-۳۷۲۶-۱.

[۱۳] Bai, R. and Kendall, G., *"An Investigation of Automated Planograms Using a Simulated Annealing Based Hyper-heuristics,"*, Progress as Real ProblemSolvers-(OperationsResearch/Computer Science Interfaces Series, Vol. ۳۲) Berlin, Heidelberg, New York: Springer, pp. ۸۷-۱۰۸, ۲۰۰۵.

[۱٤] Burke, E. K., Petrovic, S. and Qu, R., "*Case Based Heuristic Selection for Timetabling Problems.Journal of Scheduling*", ۹(۲): ۹۹-۱۱۳, ۲۰۰۶b.

[۱٥] P. Cowling, G. Kendell and L. Han, *"An Investigation of a Hyperheuristic Genetic Algorithm Applied to a Trainer Scheduling Problem"*, Proceeding of Congress on Evolutionary Computation (CEC۲۰۰۲), IEEE computer Society Press, Honolulu, USA, ۱۱۵۸-۱۱۹۰. ۲۰۰۲.

[١٦] L. Han, G. Kendell, and P. Cowlling, *"An adaptive length chromosome hyperheuristic genetic algorithm for a trainer scheduling problem"*, Proceeding of ٤th Asiapacific Congress on simulated Evolution and learning (SEAL '٠٢), Singapore, ٢٦٧-٢٧١. (٢٠٠٢)

[١٧] Cowling, P.I., Kendall, G., Soubeiga, E., "*Hyperheuristic Approach to Scheduling a Sales Summit"*, Selected papers of Proceedings of the Third International Conference of Practice And Theory of Automated Timetabling (PATAT), Springer LNCS vol ٢٠٧٩, pp. ١٧٦-١٩٠.(٢٠٠٠)

[١٨] P. Hansen and N. Mladenovic, "*Variable neighbourhood search: Principles and applications"*, European Journal of Operational Research ١٣٠.٤٤٩-٤٦٧. (٢٠٠١)

[١٩] R. E. Keller R. Poli, "*Self-adaptive Hyperheuristic and Greedy Search*",IEEE computer and information science,,(٢٠٠٨)

[٢٠] Ch. Nilsson, "*Heuristics for the Traveling Salesman Problem*", international conference on heuristic(٢٠٠٣)

[٢١] http://www.iwr.uniheidelberg.de/groups/comopt/software/tsplib٩٥/tsp/

[٢٢] M. Montazeri, Supervised by A. Bahrololoum, "*Investigation of effectiveness of Hyper-Heuristic in NP-Hard Problems*", B.Sc. Thesis, Shahid Bahonar University of Kerman, Kerman, Iran, ٢٠٠٩ (in Farsi).

[٢٣] M. Montazeri, A. Bahrololoum, H. Nezamabadi-pour, M. Soleymani Baghshah, M. Montazeri, "*Cooperating Of Local Searches Based Hyper-Heuristic Approach For Solving Travelling Salesman Problem*", International Conference on Evolutionary Computation Theory and Application (ECTA), Paris, France, ٢٠١١.

[٢٤] Ozcan E., Erenturk M., ٢٠٠٤. "*A brief review of memetic algorithms for solving Euclidean* ٣*D traveling* salesrep *problem"*. Proc. of the ١٣th Turkish Symposium on Artificial Intelligence and Neural Networks ٩٩–١٠٨.

[٢٥] Çunka, M., Özsalam M.Y., ٢٠٠٩. "*A comparative study on particle swarm optimization and genetic algorithm for traveling salesmen problem*", Taylor & Francis, Cybernetics and Systems.

[٢٦] Lau, H. C., Xiao, F., ٢٠٠٨. "*The oil drilling model and iterative deepening genetic annealing algorithm for the TSP"*, In A. Fink and F. Rothlauf(eds), Advanced in Computational Intelligence in Transportation and Logistics, Stuies in Computational Intelligence. Springer, ١٦٩-١٨٤.

زیرنویس ها

[1] Traveling Salesman Problem
[2] Dynamic programming
[3] Integer Programming
[4] Explore
[5] Exploit
[6] Meta-Heuristic
[7] Crossover
[8] Mutation
[9] Order Crossover
[10] Partially Matched Crossover
[11] Cycle Crossover
[12] Global Search
[13] Solution Space
[14] Low Level Heuristic (LLH)
[15] Choice Function
[16] Case Base Reasoning
[17] Project Presentation Scheduling Problem
[18] Travelling Tournament Problem
[19] Variable Neighborhood Search
[20] Permutation

کنفرانس داده کاوی ایران