فصلنامه علمی ـ پژوهشی مطالعات مدیریت
صنعتی سال هشتم، شماره ۱۸، پائیز ۸۹
صفحات ۱۰۵ تا ۱۲۲

Motaleat-e modiriat-e sanati
Vol.8, No.18, Autumn 2010

حل مسأله فروشنده دوره گرد متقارن با در نظر گرفتن زمان عزیمت فازی بین شهرها توسط الگوریتم فرا ابتکاری مورچگان

جمشید صالحی صدقیانی *

چکیده

مساله فروشنده دوره گرد یکی از معروفترین مسائل بهینه سازی ترکیبی است که با توجه به ویژگی‌های خاص آن، در سال های اخیر نیز بستر مناسبی برای اعتبارسنجی الگوریتم‌های مختلف ابتکاری، فراابتکاری و دقیق بوده است. کاربردهای متعدد این مساله از لحاظ نظری و عملیاتی نیز باعث توجه ویژهٔ محققان به آن شده است. الگوریتم فراابتکاری بهینه سازی توسط کلونی مورچگان در زمره روش‌های فراابتکاری موفقی است که در سال‌های اخیر به نحو موفقیت آمیزی برای حل مسائل بهینه سازی ترکیبی گسسته استفاده شده است. در این مقاله، الگوریتمی مبتنی بر بهینه سازی توسط کلونی مورچگان، برای حل مسأله فروشنده دوره گرده با داده‌های فازی ارائه شده است. الگوریتم پیشنهادی در محیط برنامه نویسی C++

---

* استاد گروه مدیریت صنعتی، دانشکده حسابداری و مدیریت، دانشگاه علامه طباطبایی

تاریخ دریافت:۸۹/۱/۲۳ تاریخ پذیرش: ۸۹/۳/۲۵

کدنویسی و اجرا گردیده و نتایج هر بار اجرای آن با نتایج الگوریتم دقیق انشعاب و تحدید که از کدنویسی در محیط LINGO8.0 به دست آمده، مقایسه شده است. الگوریتم پیشنهادی در مورد مثال‌هایی با ابعاد کوچک به جواب بهینه دست یافته و در مورد مثال‌های بزرگ در زمان‌هائی بسیار کوتاه به جواب‌های شدنی مناسبی دست می‌یابد.

واژگان کلیدی: الگوریتم کلونی مورچگان، مسئله فروشنده دوره گرد، داده‌های فازی، الگوریتم AS

مقدمه

الگوریتم کلونی مورچگان اولین بار توسط دوریگوو همکارانش در سال ۱۹۹۱ به عنوان راه حل جدیدی برای مسائل بهینه‌سازی ترکیبی ارائه گردید [۵]. از آن زمان، این ایده به مرور گسترش یافت، به نحوی که در میانه دهه ۹۰ میلادی اثر بخشی این الگوریتم به وضوح نمایان شد. در سال‌های بعد محققان گونه‌های تکامل یافته‌تری از الگوریتم‌های مورچگان را توسعه داده و برای حل مسائل مختلفی بکارگرفتند [8,14-17]. در واقع الگوریتم کلونی مورچگان الهام گرفته از مطالعات و مشاهدات روی کلونی مورچه‌های واقعی است [۹].

بهینه‌سازی به کمک الگوریتم مورچه‌ها که به اختصار ACO نامیده می‌شود، از رفتار مورچه‌های واقعی در یافتن کوتاه‌ترین مسیر بین منبع غذایی و لانه الهام گرفته است [7,16]. بر اساس این روش، مورچه‌ها در ACO از طریق حرکتی همزمان و در مسیرهایی متمایز روی یک گراف مناسب، به تولید جواب برای مسأله در حال حل اقدام می‌کنند. الگوریتم شامل سه مرحله کلی (۱) ساخت جواب، (۲) به روزآوری فرومون و (۳) برخی اعمال خارق‌العاده از جمله جستجوی موضعی و تغییر پارامترهای الگوریتم با گذشت زمان است. در شکل ۱ نمای کلی الگوریتم مورچگان برای حل TSP به صورت شبه کد نشان داده شده است [۵].

```
procedure ACO for TSP
  Initialize Data
  while (not terminate)  do
    ConstructSolutions
    LocalSearch
    UpdateStatistics
    UpdatePheremone Trails
  end – while
end – procedure
```

شکل ۱. شبه کد ACO برای حل TSP

اگرچه ACO را می‌توان در اغلب مسائلی با ساختاری گراف گونه به کار برد، ولی کارایی آن در مسائل Np-hard که الگوریتم‌های دقیق در مورد مثال های بزرگ یا متوسط آنها کارایی خوبی ندارند، مسائل کوتاهترین مسیر پویا که در آنها همزمان با فرایند بهینه‌سازی، برخی از خصوصیات گراف مسأله در طول زمان تغییر می‌کند و مسائلی که در آنها ساختار محاسبات شکلی فضایی و پیچیده دارند، اثبات شده است[5]. امروزه کاربردهای متنوعی از ACO در ادبیات موضوع جهت حل انواع مسائل بهینه سازی ترکیبی گسسته ارائه شده است [۵-۲]. از این دست می‌توان به انوع مسائل تخصیص، زمانبندی، مسیریابی و زیرمجموعه‌ها اشاره نمود [۲۳-۱۸و۴].. کاربردهای ACO به حل مسائل بهینه سازی ترکیبی گسسته محدود نمی‌شود. مسائل بهینه‌سازی پیوسته، یادگیری ماشین و تحلیل ساختار پیچیده پروتئین‌ها نیز از جمله مواردی هستند که ACO نتایج قابل قبولی را از خود به نمایش گذاشته است [۵].

مساله فروشنده دوره گرد یکی از مسائل کلیدی در بحث بهینه سازی ترکیبی بوده و به لحاظ ساختار خاصی که دارد به عنوان مرجع معتبری جهت آزمایش اکثر روش‌های ابتکاری و فراابتکاری مورد استفاده قرار گرفته است [۶،۸،۱۰،۱۳،۱۵-۲].. در این مقاله نیز، مسأله فروشنده دوره‌گرد، در شرایطی که زمان مسافرت از شهری به شهر دیگر در آن، دارای رفتاری فازی می‌باشد، بررسی خواهد شد. در بخش دوم

مساله فروشنده دوره گرد در حالت قطعی معرفی می‌شود. بخش سوم به معرفی کلی الگوریتم کلونی مورچه‌ها برای حل مساله فروشنده دوره گرد اختصاص دارد. در بخش چهارم روش مورد استفاده برای تطبیق و تبدیل داده‌های فازی به داده‌های قطعی به عنوان ورودی‌های الگوریتم کلونی مورچگان پیشنهادی ارائه می‌شود. بخش پنجم به تشریح دقیق الگوریتم کلونی مورچگان پیشنهادی برای حل مساله فروشنده دوره گرد با داده‌های فازی اختصاص دارد. در بخش ۶ الگوریتم پیشنهادی با مثال‌های نمونه آزمایش و نتایج آزمایشات با جواب‌های بهینه‌ای که از مدل برنامه‌ریزی ریاضی توسعه داده شده در نرم افزار LINGO به دست آمده است، مقایسه می‌شوند. در نهایت مقاله در بخش ۷ با جمع بندی و نتیجه گیری خاتمه می‌یابد.

مسأله فروشنده دوره‌گرد

مسأله فروشنده دوره‌گرد (TSP)، فروشنده دوره گردی است که از شهر محل سکونت خود شروع به حرکت می‌کند و قصد دارد کوتاه‌ترین مسیری را بیابید که او را از مجموعه‌ای مفروض از شهرهای مشتری عبور می‌دهد و سرانجام به شهر محل سکونت خود باز می‌گرداند. البته فروشنده از هر شهر مشتری دقیقاً یکبار عبور می‌کند. برای بیان سازمان یافته‌تر این مسأله، TSP را می‌توان با یک گراف کامل موزون به صورت G = (N,A) نمایش داد. به صورتی که N مجموعهٔ گره‌ها را نشان می‌دهد که معرف شهرها بوده و A مجموعه کمان‌ها است.

به هر کمان $(i,j)\in A$ ارزشی برابر d_{ij} اختصاص داده می‌شود که فاصله میان شهر i و j است ($i,j \in N$). در مسأله فروشندهٔ دوره گرد نامتقارن، فاصلهٔ میان یک جفت گره مانند i و j به جهت حرکت از روی کمان وابسته است. یعنی به ازاء دو شهر i و j یک کمان (i,j) وجود دارد که در آن $d_{ij} \neq d_{ji}$ می‌باشد. در TSP متقارن، برای تمام کمان‌های مجموعهٔ A ، $d_{ij} = d_{ji}$ است. هدف TSP یافتن کوتاه‌ترین دور همیلتونی گراف مسأله است. دور همیلتونی مسیر بسته‌ای است که از تمام n گره گراف G دقیقاً یکبار عبور می‌نماید. بنابراین یک جواب بهینه برای TSP

ترتیبی مانند π از شماره‌های گره‌ها به صورت $\{1,2,...,n\}$ است، بطوری که طول $f(\pi)$ کمینه گردد. تابع $f(\pi)$ به صورت زیر محاسبه می‌شود.

$$f(\pi)=\sum_{i=1}^{n-1} d_{\pi(i)\pi(i+1)} + d_{\pi(n)\pi(1)} \qquad (۱)$$

شرح الگوریتم AS برای حل TSP

گراف G = (C,L) را در نظر بگیرید، بطوری که مجموعهٔ کمان‌های L ، تمام اجزای C را به یکدیگر متصل می‌کند. مجموعه حالت‌های مسأله متناظر با مجموعهٔ تمامی دورهای نیمه تمام ممکن روی گراف است و مجموعهٔ محدودیت‌ها تضمین می‌کنند که مورچه‌ها فقط دورهای کامل شدنی را تولید نماید. چون گراف، گرافی کامل است، هر مسیر بسته‌ای که تنها یکبار از تمام گره‌ها عبور کند، با یک دور کامل شدنی متناظر خواهد بود.

در الگوریتم‌های ACO موجود برای حل TSP، ردهای فرومون با کمان‌ها متناظرند (ردهای فرومون روی کمان‌ها باقی می‌ماند). لذا τ_{ij} دلالت بر تمایل بازدید شهر j ام بلافاصله بعد از شهر i ام دارد. اطلاعات ابتکاری به صورت $\eta_{ij} = 1/d_{ij}$ انتخاب می‌شود. یعنی، تمایل ابتکاری برای عزیمت بلادرنگ از شهر iام به شهر jام، با فاصلهٔ میان دو شهر نسبت عکس دارد.

ساخت دور کامل

در AS، m مورچه (مصنوعی) برای ساختن یک دور کامل در مسأله TSP مشارکت دارند. ابتدا مورچه‌ها بطور تصادفی شهرها را انتخاب می‌کنند. در هر گام تولید جواب، مورچه kام یک قانون انتخاب احتمالی را به کار می‌برد (قانون نسبت تصادفی). مورچه kام این قانون را برای تصمیم‌گیری درباره شهر بعدی که باید به آن برود، به کار می‌برد. احتمال اینکه مورچه kام که هم اکنون در شهر i می‌باشد شهر j را برای انتقال بعدی انتخاب کند، برابر است با:

$$P_{ij}^{k} = \frac{\left[\tau_{ij}\right]^{\alpha}\left[\eta_{ij}\right]^{\beta}}{\sum_{l\in N_i^k}\left[\tau_{il}\right]^{\alpha}\left[\eta_{il}\right]^{\beta}} \ , \ if \ \ j \in N_i^k \ , \qquad (۲)$$

به طوری که در آن:

- $\eta_{ij} = 1/d_{ij}$ مقدار ابتکاری الویت شهرها برای عزیمت است.
- β و α به ترتیب تأثیر نسبی اطلاعات ابتکاری و ردهای فرومون را معین می‌کنند.
- N_i^k همسایگی قابل قبول مورچه kام را وقتی در شهر i ام است، نشان می‌دهد. یعنی N_i^k مجموعه شهرهایی را نشان می‌دهد که مورچه kام هنوز به آنها نرفته است (احتمال انتخاب شهری که خارج از N_i^k قرار دارد، صفر می‌باشد).

به روزآوری ردهای فرومون

پس از اینکه تمام مورچه‌ها مسیرهای خود را ساختند، ردهای فرومون به روز می‌شوند. معمولاً به روزآوری رد فرومون ابتدا با کاهش مقدار فرومون روی تمام کمان‌ها با نرخی ثابت انجام می‌شود (فرایند تبخیر فرومون). سپس طبق قواعد خاصی به کمان‌هایی که مورچه‌ها از آنها عبور کرده‌اند، فرومون اضافه می‌شود. تبخیر فرومون طبق معادلهٔ زیر انجام می‌شود.

$$\tau_{ij} \leftarrow (1-\rho)\tau_{ij} \ , \ \forall (i,j) \in L \qquad (۳)$$

بطوری که $0 < \rho \leq 1$ نرخ تبخیر فرومون است. از پارامتر ρ برای جلوگیری از تجمع بی‌رویه و نامحدود ردهای فرومون استفاده می‌شود. در واقع ρ الگوریتم را قادر به فراموشی تصمیمات نامناسب پیشین و اجتناب از دام بهینه‌های محلی می‌سازد. در حقیقت اگر کمانی توسط مورچه‌ها انتخاب نشود، مقدار فرومون متناظر با آن کمان با افزایش تعداد تکرارهای الگوریتم طبق رابطه (۳) کاهش می‌یابد. پس از فرایند تبخیر فرومون، تمام یا برخی ازمورچه‌ها روی کمان‌هایی که در مسیرشان از آنها عبور کرده‌اند، طبق رابطه زیر فرومون باقی می‌گذارند.

$$\tau_{ij} \leftarrow \tau_{ij} + \sum_{k=1}^{m} \Delta\tau_{ij}^{k} \ , \ \forall (i,j) \in L \qquad (۴)$$

به طوری که $\Delta\tau_{ij}^{k}$ مقدار فرومونی است که مورچهٔ kام روی کمان‌هایی که از آنها عبور کرده باقی می‌گذارد. این مقدار به صورت زیر تعریف می‌شود:

$$\Delta\tau_{ij}^{k} = \begin{cases} 1/C^{k}, if\ are (i,j) belongs\ to\ T^{k}; \\ 0, \qquad\qquad otherwise; \end{cases} \qquad (۵)$$

به طوری که C^k طول دور کامل (T^k) است که مورچه kام می‌سازد و با جمع‌کردن طول تمام کمان‌هایی که به T^k تعلق دارند، محاسبه می‌شود. از رابطه ۵ چنین برداشت می‌شود که هر قدر یک دور کامل ساخته شده توسط یک مورچه بهتر (کوتاه‌تر) باشد، کمان‌هایی که به آن دور تعلق دارند، فرومون بیشتری دریافت خواهند کرد. به طور کلی، کمان‌هایی که تعداد زیادی مورچه از آنها استفاده می‌کنند، فرومون بیشتری دریافت خواهند نمود. بنابراین با احتمال بیشتری در تکرارهای بعدی الگوریتم توسط مورچه‌ها انتخاب خواهند شد.

داده‌های فازی

در این مقاله فرض می‌شود که اطلاعات دقیقی در مورد مقدار قطعی $d_{ij} = d_{ji}$ در دست نیست. در مساله فروشنده دوره گرد جنس مقادیر $d_{ij} = d_{ji}$ از نوع هزینه است. یعنی کمال مطلوب آنها در کاهش مقادیرشان نهفته است. این مقادیر می‌توانند واحد هزینه، فاصله و یا زمان سفر بین دو شهر باشند. در دنیای واقعی معمولاً تخمین این پارامترها به صورت قطعی میسر نمی‌باشد و این مقادیر همواره توام با عدم قطعیت و ابهام است. بدون ایجاد خللی در منطق الگوریتم پیشنهادی، فرض می‌شود که d_{ij} برای تمامی شهرها به عنوان زمان سفر بین دو شهر مفروض بوده و برای نمایش ابهام و نادقیق بودن زمان سفر این فواصل به صورت اعداد فازی ذوزنقه‌ای در نظر گرفته می‌شوند. نمایش یک عدد فازی ذوزنقه‌ای به صورت شکل ۱ خواهد بود. برخی تعاریف ابتدائی در مورد مجموعه‌های فازی به شرح زیر می‌باشند.

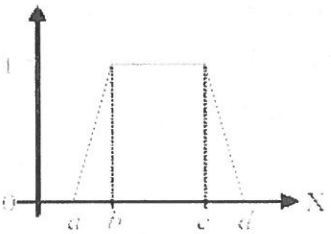
تعریف ۱-۴ مجموعه فازی: یک مجموعه فازی روی یک مجموعه مبدا X

مجموعه‌ای از جفت‌های $(x,\mu_A(x))$ است که به صورت رابطه ۶ نمایش داده می‌شود.

$$A=\{\mu_A(x)/x : x\in X, \mu_A(x)\in[0,1]\in\Re\} \qquad (۶)$$

به طوری که $\mu_A(x)$ تابع درجه عضویت عضو فازی x مجموعه A نامیده می‌شود. تابع درجه عضویت مقداری حقیقی بین ۰ و ۱ را می‌پذیرد. برای توابع عضویت انتخاب‌های متفاوتی وجود دارد. در یک تقسیم بندی کلی که توسط زاده ارائه شده، می‌توان توابع فازی را به دو دسته خطی و غیر خطی (منحنی) تقسیم بندی کرد. توابع مثلثی، یکه، L، گاما، ذوزنقه، S، گاوسی و شبه نمایی از جمله معروفترین توابعی هستند که برای مدل کردن درجه عضویت در مجموعه‌های فازی برای کاربردهای متفاوت مورد استفاده قرار می‌گیرند. در شکل ۱ تابع ذوزنقه‌ای نشان داده شده است.

تعریف ۲-۴ عدد فازی: اعداد فازی مجموعه‌های فازی هستند که برای نمایش ضمنی عدم قطعیت داده‌های عددی مورد نیاز از آنها استفاده می‌شود. در واقع اعداد فازی واژه‌های کلامی را در کنار مقادیر عددی لحاظ می‌کنند. اعداد فازی روی مجموعه‌ای از اعداد حقیقی و زیر مجموعه‌های آن تعریف و تابع عضویت آنها باید نرمال و محدب باشد.



شکل ۱. عدد فازی ذوزنقه ای

در شکل ۱ محور عمودی نشان دهنده درجه عضویت و محور افقی نماینده مقادیر حقیقی است. عدد فازی ذوزنقه‌ای را می‌توان به صورت $\widetilde{A}=(a,b,c,d)$ نیز نمایش داد. البته عدد A به صورت $\widetilde{A}=(b,c,\alpha,\beta)$ نیز قابل نمایش است که در آن $\alpha=b-a$ پهنای سمت چپ و $\beta=d-c$ پهنای سمت راست عدد مورد نظر

را نمایش می‌دهد.

با این مفروضات مساله فروشنده دوره گرد با زمان عزیمت فازی بین دو شهر، تبدیل به یک برنامه ریزی ریاضی فازی می‌شود. برای حل اینگونه مسائل چندین روش وجود دارد که متداول‌ترین آنها بر مفهوم مقایسه اعداد فازی با استفاده از تابع رتبه بندی پایه گذاری شده‌اند [۱]. در حقیقت، یک روش کارا برای مرتب کردن اعضای $F(R)$ تعریف تابع رتبه بندی $\Re : F(R) \to R$ است که هر عدد فازی را به یک عدد حقیقی تصویر می‌کند [۱].

فرض کنید $\tilde{a}$ و $\tilde{b}$ در $F(R)$ هستند و $\Re$ یک تابع رتبه بندی خطی باشد. رتبه‌ها روی $F(R)$ به صورت زیر تعریف می‌شوند:

(۱) $\tilde{a} \geq_{\Re} \tilde{b}$ اگر و تنها اگر $\Re(\tilde{a}) \geq \Re(\tilde{b})$ ،

(۲) $\tilde{a} >_{\Re} \tilde{b}$ اگر و تنها اگر $\Re(\tilde{a}) > \Re(\tilde{b})$ ،

(۳) $\tilde{a} =_{\Re} \tilde{b}$ اگر و تنها اگر $\Re(\tilde{a}) = \Re(\tilde{b})$ ،

(۴) $\tilde{a} \geq_{\Re} \tilde{b}$ اگر و تنها اگر $\tilde{a} - \tilde{b} \geq_{\Re} 0$ یا $-\tilde{b} \geq_{\Re} -\tilde{a}$ ،

(۵) اگر $\tilde{a} \geq_{\Re} \tilde{b}$ و $\tilde{c} \geq_{\Re} \tilde{d}$ آن گاه $\tilde{a} + \tilde{c} \geq_{\Re} \tilde{b} + \tilde{d}$ ،

(۶) $\Re(k\tilde{a} + \tilde{b}) = k\Re(\tilde{a}) + R(\tilde{b})$.

برای عدد فازی ذوزنقه‌ای $\tilde{a} = (a^L, a^U, \alpha, \beta)$ تابع رتبه بندی خطی به صورت زیر می‌شود:

$$\Re(\tilde{a}) = \int_0^1 (\inf \tilde{a}_\alpha + \sup \tilde{a}_\alpha) d\alpha \qquad (۷)$$

که به عبارت زیر ساده می‌شود:

$$\Re(\tilde{a}) = a^L + a^U + \frac{1}{2}(\beta - \alpha) \qquad (۸)$$

برای اعداد فازی ذوزنقه‌ای $\tilde{a}$ و $\tilde{b}$:

$\tilde{a} \geq_{\Re} \tilde{b}$ اگر و تنها اگر

$$a^L + a^U + \frac{1}{2}(\beta - \alpha) \geq b^L + b^U + \frac{1}{2}(\theta - \gamma) \qquad (۹)$$

اجرای الگوریتم

عمده‌ترین گام‌های در یک الگوریتم ACO، شامل ساخت جواب، مدیریت ردهای فرومون و روش‌های دیگری مانند جستجوی محلی است. به علاوه، ساختارهای داده و پارامترها باید تنظیم (تنظیمات اولیه) و آماره‌هایی در مورد اجرای الگوریتم (مانند زمان اجرا، میانگین حافظهٔ مصرفی، تعداد تکرارها، شرایط توقف، درصد رسیدن به بهینگی و ...) نیز محاسبه و نگهداری شوند. در ادامه جزئیات الگوریتم پیشنهادی برای حل مساله فروشنده دوره گرد ارائه شده است.

تعیین مقادیر اولیه داده‌ها

در تعیین مقدار اولیه داده‌ها ابتدا مثال مورد نظر خوانده می‌شود. سپس ماتریس زمان طی مسافت بین شهرها محاسبه می‌شود که در این مقاله به صورت اعداد فازی ذوزنقه‌ای در نظر گرفته شده و سپس در الگوریتم طراحی شده با استفاده از روش پیشنهادی در [۱] داده‌های دقیق تولید می‌شود. به عبارت بهتر داده‌های فازی با استفاده از این روش فازی زدایی می شوند. در گام بعدی لیست‌های نزدیک‌ترین همسایه‌ها برای تمام شهرها محاسبه و ماتریس فرومون و ماتریس انتخاب ـ اطلاعات تنظیم می‌شود. سپس مورچه‌ها تعریف و پارامترهای الگوریتم تنظیم می‌شوند. در نهایت برخی از متغیرهایی که اطلاعات آماری را در خود نگهداری می‌کنند، مانند زمان صرف شده توسط CPU، تعداد تکرارها و بهترین جواب یافت شده تاکنون تعریف می‌شوند.

شرط توقف

در صورت تحقق حداقل یکی از شرایط توقف، برنامه متوقف می‌شود:

۱) جواب‌های الگوریتم در فاصلهٔ معینی از حد پایین کیفیت جواب بهینه قرار داشته باشد.

۲) دورهای کامل ساخته شده یا تکرارهای الگوریتم به حداکثر تعداد خود رسیده باشد.

ساخت جواب

برای ساخت جواب در ابتدای هر تکرار، حافظه هر مورچه شامل یک آرایه $1\times n$ خالی می‌گردد. این عمل با قرار دادن تمام شهرها در لیست شهرهای ملاقات نشده انجام می‌شود. هر مورچه باید به یک شهر اولیه اختصاص یابد (تا از آن شهر شروع به تولید جواب نماید). به طور تصادفی هر مورچه به شهری اختصاص داده می‌شود. تابع تصادفی، مقداری تصادفی را طبق توزیع یکنواخت از مجموعهٔ $\{1,\ldots,n\}$ ارائه می‌دهد. در این مرحله هر مورچه یک دور کامل را تولید می‌کند. در هر گام از ساخت جواب، مورچه‌ها قانون انتخاب AS را (رابطه (۲)) برای انتخاب شهر بعدی به کار می‌برند. نهایتاً مورچه‌ها به شهر اولیه باز می‌گردند و طول دور کامل مربوط به هر مورچه محاسبه می‌شود. برای سهولت نمایش یک دور کامل، شمارنده اولین شهر مجدداً در موقعیت $n+1$ تکرار می‌شود. این فرایند به نحوی همزمان و هماهنگ است که مورچه‌ها به طور موازی جواب‌ها را می‌سازند.

به روز آوری فرومون

آخرین گام در یک تکرار از AS به روزآوری فرومون است. رویه به روزآوری فرومون در AS از دو قسمت تبخیر فرومون و باقی گذاردن فرومون تشکیل می‌شود. در اولین قسمت، یعنی رویهٔ تبخیر مقدار ردهای فرومون موجود روی کمان‌های (i,j) با یک فاکتور ثابت ρ کاهش می‌یابد. در دومین قسمت، یعنی رویهٔ باقی گذاردن فرومون به کمان‌هایی فرومون اضافه می‌شود که به دورهای کامل ساخته شده توسط مورچه‌ها تعلق دارند. بررسی کدها در حل مثال‌های بزرگ TSP نشان می‌دهد که تبخیر فرومون و سایر محاسبات به زمان قابل توجهی نیاز دارد. یکی از راهکارهای مقابله با این حجم بالای محاسبات استفاده از فهرست‌های منتخب در ساخت جواب است. در این صورت تنها قسمت کوچکی از درایه‌های ماتریس‌های متناظر در محاسبات مورد توجه قرار می‌گیرند.

بنابراین استفاده از فهرست‌های منتخب به روزآوری فرومون را نیز تسریع می‌کند. در حقیقت، استفاده از فهرست‌های منتخب به همراه تعداد ثابتی از نزدیک‌ترین

همسایه‌ها، درجهٔ پیچیدگی محاسباتی دو رویهٔ ذکر شده را به O(n) کاهش می‌دهد.
پایان الگوریتم

آخرین گام در اجرای AS، ذخیره کردن داده‌های آماری مربوط به رفتار الگوریتم است. جواب مثال‌ها در واقع مربوط به بهترین جواب یافت شده از زمان شروع اجرای الگوریتم یا تعداد تکرارهایی است که در آن بهترین جواب یافت شده است.

ویژگی‌های الگوریتم

۱) استفاده از فهرست همسایگی هنگام عزیمت از شهری به شهر دیگر
۲) استفاده از یک روش فازی زدایی مناسب برای برای تبدیل ورودی‌های فازی به قطعی
۳) قابلیت بکارگیری در مسأله فروشنده دوره گرد نا متقارن با حداقل تغییرات مورد نیاز

انجام آزمایشات

به منظور سنجش کارایی مدل پیشنهادی، ابتدا مدل برنامه‌ریزی ریاضی مساله فروشنده دوره گرد در نرم افزار LINGO 8.0 کد نویسی شد. سپس الگوریتم مورچگان پیشنهادی برای حل مساله فروشنده دوره گرد با فرض زمان‌های عزیمت فازی در بستر نرم افزاری C++ پیاده سازی و کد نویسی گردید. ۵ مثال فروشنده دوره گرد در ابعاد ۶، ۲۰، ۳۰، ۵۰ و ۱۰۰ شهر به صورت کاملاً تصادفی تولید و برای آزمایش الگوریتم انتخاب شدند. هر سه مثال با استفاده از مدل برنامه ریزی ریاضی توسعه داده شده در محیط نرم افزار LINGO 8.0 حل شدند. شرایط سخت افزاری و نرم افزاری آزمایش‌های انجام شده در جدول ۱ خلاصه شده‌اند. جدول ۲ نتایج بهینه مثال‌های یاد شده که خروجی نرم افزار LINGO 8.0 می‌باشند را نشان می‌دهد.

جدول ۱. شرایط سخت افزاری و نرم افزاری آزمایش‌ها

نوع الگوریتم	محیط برنامه‌نویسی	ابعاد مثال‌ها	پردازشگر رایانه	حافظه	سیستم عامل
فراابتکاری AS	C++	۶، ۲۰، ۳۰، ۵۰، ۱۰۰	Athlon 5400+	2 GB	Win XP Pro.
انشعاب و تحدید	LINGO 8.0	۶، ۲۰، ۳۰، ۵۰، ۱۰۰	Athlon 5400+	2 GB	Win XP Pro.

جدول ۲. جواب‌های بهینه نرم افزار LINGO 8.0 برای مثال‌های آزمایشی

	۶ شهر	۲۰ شهر	۳۰ شهر	۵۰ شهر	۱۰۰ شهر
جواب بهینه	۶۶۱۰	۲۷۱	۲۰۰	۳۰۰	۲۴۴
تعداد تکرارها	۳۳	۲۴۵	۵۷۷۸۲	۱۲۹۳۳	۱۳۱۹۵۱۲
زمان(ثانیه)	۰,۱	۱	۱۳	۵۰	۸۶۸۹

جدول ۳. آزمایشات طراحی شده

مثال انتخابی	α	β	ρ	تعداد آزمایش	تعداد اجرا
۶ شهری	۱	۲و۲,۵و۳و۴	۰,۱ و ۰,۲	۸	۱۰
۲۰ شهری	۱	۲و۲,۵و۳و۴	۰,۱ و ۰,۲	۸	۱۰
۳۰ شهری	۱	۲و۲,۵و۳و۴	۰,۱ و ۰,۲	۸	۱۰
۵۰ شهری	۱	۲و۲,۵و۳و۴	۰,۱ و ۰,۲	۸	۱۰
۱۰۰ شهری	۱	۲و۲,۵و۳و۴	۰,۱ و ۰,۲	۸	۱۰

جدول ٤. آزمایشات مربوط به مثال ٦ شهری ($\alpha = 1.0 , \beta = 2.5 , \rho = 0.2$)

تعداد تکرار	زمان (ثانیه)	بهترین جواب	فاصله از بهینه	اجرا
۸	۰	۶۶۱۰	٪۰	۱
۷	۰	۶۶۱۰	٪۰	۲
۲	۰	۶۶۱۰	٪۰	۳
۶	۰	۶۶۱۰	٪۰	۴
۳	۰	۶۶۱۰	٪۰	۵
۳	۰	۶۶۱۰	٪۰	۶
۳	۰	۶۶۱۰	٪۰	۷
۲	۰	۶۶۱۰	٪۰	۸
۳	۰	۶۶۱۰	٪۰	۹
۶	۰	۶۶۱۰	٪۰	۱۰
-------	۰	۶۶۱۰	٪۰	میانگین

آزمایشات مستقلی برای سنجش کارائی الگوریتم و تنظیمات مناسب پارامترهای آن ترتیب داده شدند. نتیجه‌گیری در خصوص تنظیمات مناسب پارامترهای الگوریتم نیازمند آزمایش‌های کنترل شده‌ای است که بتوان با تحلیل واریانس و

انجام آزمون‌های آماری مناسب برتری ترکیب مشخصی از پارامترها را اثبات نمود. در این مقاله آزمایش‌های متعددی با شرایط کنترل شده مطابق اطلاعات جدول ۳ ترتیب داده شده است. نتایج تمامی آزمایش‌ها مورد تحلیل قرار گرفته و بهترین تنظیمات برای پارامترهای هر مثال‌ها گزینش شده‌اند (جداول ۴ الی ۸).

جدول ۵ . آزمایشات مربوط به مثال ۲۰ شهری ($\alpha = 1.0 , \beta = 3 , \rho = 0.5$)

تعداد تکرار	زمان (ثانیه)	بهترین جواب	فاصله از بهینه	اجرا
۱۸۳	۱۳	۲۸۱	٪۳,۶۹	۱
۵۱۹	۳۱	۲۷۷	٪۲,۲۱	۲
۹۳۸	۵۱	۲۷۸	٪۲,۵۸	۳
۴۳۵	۲۷	۲۷۲	٪۰,۳۷	۴
۱۶۲	۱۲	۲۷۱	٪.	۵
۲۸۲	۱۹	۲۷۸	٪۲,۵۸	۶
۹۶	۷	۲۸۱	٪۳,۶۹	۷
۲۱۶	۱۶	۲۷۶	٪۱,۸۵	۸
۳۱۲	۱۹	۲۷۷	٪۲,۲۱	۹
۱۵۳	۱۳	۲۷۱	٪.	۱۰
-------	۲۰,۸	۲۷۶,۲	٪۱,۹۲	میانگین

جدول ٦. آزمایشات مربوط به مثال ۳۰ شهری ($\alpha = 1.0 , \beta = 2,5 , \rho = 0.5$)

تعداد تکرار	زمان (ثانیه)	بهترین جواب	فاصله از بهینه	اجرا
۱۰۰۰	۴۶۲	۲۵۵	٪۲۰,۳	۱
۲۳	۶	۲۰۷	٪۲,۵۸	۲
۱۶۳	۳۴	۲۰۰	٪۰	۳
۲۶۱	۱۲۹	۲۱۰	٪۳,۶۹	۴
۴	۱	۲۰۵	٪۱,۸۵	۵
۱۰۳	۵۶	۲۰۸	٪۲,۹۵	۶
۵	۱	۲۰۶	٪۲,۲۱	۷
۴	۱	۲۰۴	٪۱,۴۸	۸
۵	۱	۲۰۴	٪۱,۴۸	۹
۱۳	۳	۲۱۴	٪۵,۱۷	۱۰
-------	۶۹,۴	۲۱۱,۳	٪۴,۱۷	میانگین

جدول ۷. آزمایشات مربوط به مثال ۵۰ شهری $(\alpha = 1.0 , \beta = 2 , \rho = 0.1)$

تعداد تکرار	زمان (ثانیه)	بهترین جواب	فاصله از بهینه	اجرا
۵	۹	۳۵۱	٪۱۷	۱
۱۲	۱۱	۳۴۹	٪۱۶,۳	۲
۴	۵	۳۴۴	٪۱۴,۶	۳
۶	۴	۳۵۵	٪۱۸,۳	۴
۷	۹	۳۵۰	٪۱۶,۶	۵
۴	۵	۳۴۶	٪۱۵,۳	۶
۹	۱۱	۳۴۱	٪۱۳,۶	۷
۴	۵	۳۴۹	٪۱۶,۶	۸
۷	۱۰	۳۳۸	٪۱۲,۶	۹
۷	۱۲	۳۴۱	٪۱۳,۶	۱۰
--------	۸	۳۴۶,۴	٪۱۵,۴	میانگین

جدول ۸ . آزمایشات مربوط به مثال ۱۰۰ شهری $(\alpha = 1.0 , \beta = 2,5 , \rho = 0.5)$

تعداد تکرار	زمان (ثانیه)	بهترین جواب	فاصله از بهینه	اجرا
۳۹	۹۵۶	۲۸۷	٪۱۷,۶	۱
۱۹	۷۱۲	۲۸۴	٪۱۶,۳	۲
۷	۴۰۱	۲۹۸	٪۲۲,۱	۳
۲۴	۸۰۹	۲۸۶	٪۱۷,۲	۴
۵۶	۱۳۸۲	۲۸۱	٪۱۵,۱	۵
۴۷	۹۱۱	۲۸۲	٪۱۵,۵	۶
۱۴	۵۵۶	۲۹۱	۱۹,۲	۷
۱۳	۸۸۸	۲۸۴	۱۶,۳	۸
۴	۱۸۳	۲۹۰	٪۱۸,۸	۹
۴۵	۹۹۴	۲۸۳	٪۱۵,۹	۱۰
-------	۷۷۹,۲	۲۸۶,۶	۱۷,۴	میانگین

جمع بندی و نتیجه گیری

مسئله مورد بررسی در تحقیق عناصر موضوع جالب و جدیدی است که بستر مناسبی برای تحقیقات آتی به شمار می‌رود. الگوریتم پیشنهادی این مقاله پایه و

اساس اکثر الگوریتم‌های ACO می‌باشد و با اعمال تغییرات اندکی در آن می‌توان دامنه کاربرد آن را به نحو مطلوبی گسترش داد.

در این مقاله مسأله NP-Hard فروشنده دوره گرد با فرض فازی بودن زمان‌های عزیمت بین شهرها توسط الگوریتم فراابتکاری مورچگان مورد بررسی قرار گرفت. الگوریتمی مبتنی بر سیستم مورچه با ورودی‌های فازی طراحی گردید. الگوریتم مورچگان پیشنهادی برای حل مساله فروشنده دوره گرد با فرض فازی بودن زمان‌های عزیمت بین شهرها با استفاده از زبان برنامه نویسی C++ کد گردید و نتایج آن با نتایج الگوریتم دقیق انشعاب و تحدید در بستر نرم افزاری LINGO 8.0 مقایسه قرار گرفت بررسی مثال‌های مختلف با ابعاد گوناگون نشان داد که الگوریتم پیشنهادی از کارایی مناسبی برخوردار است. همچنین مشخص گردید که تغییر پارامترها عامل مهمی در نزدیکی جواب‌های حاصل از الگوریتم پیشنهادی به جواب‌های بهینه است و طی آزمایشات طراحی شده تنظیمات مناسب پارامترهای الگوریتم مشخص گردید. الگوریتم پیشنهادی در مورد مثال‌هائی با ابعاد کوچک در کمترین زمان ممکن به جواب بهینه دست یافته و در مورد مثال‌های بزرگتر در زمان‌هایی کوتاه، به جواب‌های با کیفیتی بسیار نزدیک به بهینه دست یافت.

منابع و مآخذ

[۱] جهانشاهلو، غلامرضا، فرهاد حسین زاده لطفی، توفیق الهویرانلو، (۱۳۸۵)، **تحقیق در عملیات فازی**.

[۲] خلیلی دامغانی. کاوه، (۱۳۸٤)، **بهینه سازی توسط کلونی مورچگان (معرفی سیستم مورچگان برای حل مسائل بهینه سازی ترکیبی)**، سمینار کارشناسی ارشد مهندسی صنایع- صنایع دانشکده تحصیلات تکمیلی تهران جنوب.

[۳] خلیلی دامغانی. کاوه (۱۳۸٤)، **حل مساله زمانبندی پروژه با منابع محدود درحالت احتمالی با استفاده از الگوریتم مورچگان**، پایان نامه کارشناسی ارشد مهندسی صنایع - صنایع، دانشکده تحصیلات تکمیلی تهران جنوب.

[٤] توکلی مقدم، رضا، محمد شاهعلیزاده کلخوران، کاوه خلیلی دامغانی، (۱۳۸۵)، **حل مساله زمانبندی پروژه با منابع محدود با استفاده از الگوریتم مورچگان اصلاح شده**، تهران: سومین کنفرانس بین المللی مدیریت پروژه.

[۵] دوریگو، مارکو، توماس اشتوسل، (۱۳۸۷)، **بهینه‌سازی توسط کلونی مورچگان**، ترجمه محمدتقی تقوی فرد، ، میربهادر آریانژاد، کاوه خلیلی دامغانی، چاپ اول، تهران: انتشارات دانشگاه آزاد اسلامی واحد علوم تحقیقات تهران.

[٦] تقوی فرد، محمدتقی، کاوه خلیلی دامغانی، (۱۳۸۸)، **جانمایی پایگاه های خدماتی در کلان شهرها توسط روش‌های فراابتکاری**، تهران: هفتمین کنفرانس بین المللی مدیریت.

[7] Applegate, D., R. Bixby, v. chvatal, w. Cook, (1990), ***Finding cuts in the TSP***. Technical report 95-0.5, DiMACS center, Rutgers University, Piscataway, NJ.

[8] Deneubourg, L., S. Aron, S. Goss, M. Pasteels, (1990), ***the self organizing exploratory pattern of the Argentine ant***, I. Insect Behav., 3159.

[9] Dorigo, M., V. Maniezzo, A. Colorni, (1996), ***the ant system: Optimization by a colony of cooperating agents***, IEEE Trans. Syst, Man, Cybern. B, Vol. 26, No. 2, pp. 29–41.

[10] Dorigo, M., Maniezzo, v. Colorni, A, (1991), ***The Ant system: An autocatalytic optimizing process***, Technical Report 91-016 revised.

[11] Dorigo M. L.M. Gambardella, (1997), ***Ant Colony system: A. Cooperative learning Approach to the traveling salesman problem***, IEEE Transactions on Evolutionary computations, 1, 53-66.

[12] Mahdavi-Amiri, N., s.h. Nasseri, (2006), ***Duality in fuzzy Number linear programming by use of certain linear ranking function***. Elsevier, Applied Mathematics and Computation 180, 206–216.

[13] Klir, G. J. , B. Yuan, (1995), ***Fuzzy sets and Fuzzy Logic, Theory and applications***, Prentice Hall PTR.

[14] Colorni, A., M. Dorigo, V. Maniezzo, (1992), ***An Investigation of Some Properties of an Ant Algorithm***, Proceedings of the Parallel Problem Solving from Nature Conference (PPSN 92), Brussels, Belgium, Elsevier Publishing, Amsterdam, The Netherlands, 509–520.

[15] Stutzle, T., H. Hoos, (2000), ***The Max-Min Ant System***, Future Generation Computer Systems,16 (8): 889-914.

[16] Dorigo, M., G.Di Caro, (1999), ***The ant colony optimization meta-heuristic***, in New Ideas in Optimization, edited by D. Corne, M.Dorigo, and F. Glover, McGraw-Hill, pp. 11-32.

[17] Goss, S., S.Aron, J. L. Deneubourg, J. M. Pasteels. (1990), ***Self-organized shortcuts in the Argentine Ant***. Naturwissenschaften, 76:579-581.

[18] Dorigo, M., G. Di Caro, L. M. Gambardella. (1999), ***Ant Algorithms for Discrete Optimization***, Artificial Life, 5(2):137-172.

[19] Stutzle, T., (1998), ***An ant approach for the flow shop problem***, In Fifth European Congress on Intelligent Techniques Soft Computing (EUFIT'98), vol.3, Verlag Mainz, Aachen, pp. 1560-1564.

[20] Merkle, D., M. Middendorf, H. Schmeck, (2002), ***Ant Colony Optimization for Resource Constrained Project Scheduling***, IEEE Transactions on Evolutionary Computation, 6(4):333-346.

[21] Gambardella, L. M., E. Taillard, G. Agazzi, (1999), ***MACS-VRPTW: A Multiple Ant Colony System for Vehicle Routing Problems with Time Window, In David Corne***, Marco Dorigo, and Fred Glover, editors, New Ideas in Optimization, pp.63-76, McGraw-Hill, London.

[22] Colorni, A., M. Dorigo, V. Maniezzo, M. Trubian, (1994), ***Ant system for Job-shop Scheduling***, JORBEL -Belgian Journal of Operations Research, Statistics and Computer Science, 34(1):39-53.

[23] van der Zwaan, S., C. Marques, (1999), ***Ant Colony Optimization for Job Shop Scheduling***, In proceedings of Third workshop on Genetic Algorithms and Artificial Life.

[24] Blum, C., M. Sampels, (2002), ***An ant Colony Optimization Algorithm for FOP Shop Scheduling: A case study on different pheromone representations***, Proceedings of the 2002 Congress on Evolutionary Computation, CEC'02, pp.1558-1563.