نشـــریه بین المللـــی مهنـــدسی صنـایع و مدیـــریت تولیـــد تابستان ۱۳۸۸- صفحه ۳۰-۲۳

شمـــاره ۲، جلـــد ۲۰، تابستـــان ۱۳۸۸

نشـــریه بین المللـــی مهنـــدسی صنایـــع و مدیـــریت تولیـــد

نشریه بین المللی مهندسی صنایع و مدیریت تولید دانشگاه علم و صنعت ایران

IJIE

ISSN: 2008-4870

Journal Website: http://IJIEPM.iust.ac.ir/

Modified Ant Colony Algorithm for the Vehicle Routing Problem with Time Windows

M. Taghavifard*, K. Sheikh & A. Shahsavari,

M. Taghavifard, *Assistant Prof., Allame Tabatabaei Univ., Tehran-Iran, dr.taghavifard@gmail.com*
K. Sheikh, *Industrial Engineering Dept., Faculty of Graduate Studies, Azad Univ., Tehran South Branch,keyvansheikh@gmail.com*
A. Shahsavari, *Industrial Engineering Dept., Faculty of Graduate Studies, Azad Univ., Tehran South Branch, Sharian61@gmail.com*

Keywords

Vehicle Routing Problem with Time Windows, Ant Colony Algorithm, Solomon test-problems

ABSTRACT

Vehicle Routing Problem with Time Windows (VRPTW) is an NP-Complete Optimization Problem. Even finding an optimal solution for small size problems is too hard and time-consuming. The objective of VRPTW is to use a fleet of vehicles with specific capacity to serve a number of customers with dissimilar demands and time window constraints at minimum cost, without violating the capacity and time window constraints. This problem has been solved with a number of heuristic and meta-heuristic solution algorithms and optimal or near optimal solutions gained. In this paper, a modified Ant Colony algorithm is proposed. In this algorithm we tried to simplify the solution procedure and computational complexities of ant colony meta-heuristic. To gain this capability, we sacrificed some computational accuracy. Testing the solution procedure on the Solomon test-problems showed that this algorithm is capable of generating relatively good solutions.

© (نشریه بین المللی مهندسی صنایع و مدیریت تولید) شماره ۲، جلد ۲۰، ۱۳۸۸

ارائه روش اصلاح شده کلونی مورچگان جهت حل مسئله مسیریابی وسایل نقلیه به همراه پنجره های زمانی

محمدتقی تقوی‌فرد، کیوان شیخ و آرین شهسواری

کلمات کلیدی

مسئله مسیریابی وسایل نقلیه به همراه پنجره‌های زمانی، الگوریتم کلونی مورچگان، نمونه مسائل Solomon

چکیده:

مسئله مسیریابی وسایل نقلیه به همراه پنجره های زمانی، در زمره مسائل NP-Complete می‌باشد، بگونه‌ای که حتی یافتن یک جواب بهینه برای ابعاد کوچک آن بسیار دشوار و زمانبر است. هدف این مسئله بکارگرفتن ناوگانی از وسایل نقلیه با ظرفیت‌های معین جهت خدمت‌دهی به تعداد معینی از مشتریان با تقاضاهای متفاوت و محدودیت‌های زمانی متفاوت می‌باشد، بگونه‌ای که هزینه کمینه شده و ظرفیت‌ها و نیز پنجره زمانی نقض نگردند. این مسئله تاکنون توسط بسیاری از روشهای حل ابتکاری و فرابتکاری مورد حل واقع شده و جوابهای بهینه یا نزدیک به جواب بهینه

تاریخ وصول: ۸۶/۸/۲۲

تاریخ تصویب: ۸۷/۱۰/۱۵

دکتر محمد تقی تقوی فرد، استادیار دانشگاه علامه طباطبایی، dr.taghavifard@gmail.com

کیوان شیخ، کارشناس ارشد مهندسی صنایع، دانشکده تحصیلات تکمیلی، دانشگاه آزاد اسلامی واحد تهران جنوب، keyvansheikh@gmail.com

آرین شهسواری، کارشناس ارشد مهندسی صنایع، دانشکده تحصیلات تکمیلی، دانشگاه آزاد اسلامی واحد تهران جنوب، Sharian61@gmail.com

[Downloaded from ijiepm.iust.ac.ir on 2026-11-02]

حاصل شده است. در این مقاله نوع اصلاح شده الگوریتم کلونی مورچگان پیشنهاد گردیده و در آن سعی شده تا حد ممکن از پیچیدگی‌های محاسباتی اجتناب و سهولت روش حل فراهم گردد؛ البته درنظر گرفتن چنین قابلیتی منجر به از دست دادن مقدار کمی از دقت محاسباتی شده است. با این حال اجرای الگوریتم پیشنهادی بر روی تعدادی از نمونه مسائل Solomon، آشکار نمود که این الگوریتم توانایی تولید جواب‌های نسبتاً خوب را دارا می‌باشد.

۱. مقدمه

مسئله مسیریابی وسایل نقلیه به همراه پنجره‌های زمانی یک نوع مسئله تصمیم گیری است و کاربردهای گوناگونی از جمله تحویل نامه و روزنامه، مسیریابی اتوبوس مدرسه، مراکز توزیع و ... دارد. این مسئله متشکل از ناوگانی از وسایل نقلیه با ظرفیت‌های معین، یک دپو (ترمینال) مرکزی و مجموعه‌ای از مشتریان با تقاضاها و پنجره‌های زمانی متفاوت می‌باشد. هر یک از وسایل نقلیه از دپو مرکزی حرکت خود را آغاز و کالاها را بدون تخطی از محدودیت‌های زمانی، حمل می‌نمایند و در نهایت مجدداً به دپو مرکزی باز می‌گردند. هر یک از مشتریان می‌توانند تنها یکبار و توسط تنها یک وسیله نقلیه ملاقات شوند. هدف، کمینه نمودن تعداد کل وسایل نقلیه و نیز کل مسافت طی شده می‌باشد[۱].

مسئله مورد نظر در زمره مسائل بهینه‌سازی NP-Complete طبقه بندی می‌گردد بگونه‌ای که حتی یافتن یک جواب بهینه برای ابعاد کوچک آن بسیار دشوار و زمانبر است[۲]. تعدادی از محققان روشهای حل ابتکاری برای حل این مسئله ارائه نموده اما بعلت پیچیدگی این مسئله، معمولاً جوابهای آنها رضایت بخش نبوده‌اند. علاوه بر آن روشهای حل فراابتکاری از جمله الگوریتم ژنتیک، جستجوی ممنوع، شبیه سازی تبرید و نیز برای حل این مسئله بکارگرفته شده اند[۱]. این روشهای فراابتکاری توانایی یافتن جوابهای بهینه و یا نزدیک به بهینه را در زمان معقول دارا می‌باشند.

الگوریتم بهینه سازی کلونی مورچگان که برای اولین بار توسط مارکو دوریگو در تز دکتری ایشان معرفی شد، یکی از جدیدترین روشهای فراابتکاری بوده و مبتنی بر بررسی رفتار مورچه‌های واقعی در جستجوی غذا می‌باشد. این روش اولین بار بر روی مسئله فروشنده دوره‌گرد بکار گرفته شد [۳]. در این مقاله نوع اصلاح شده الگوریتم کلونی مورچگان جهت حل مسئله مسیریابی وسائل نقلیه به همراه پنجره‌های زمانی ارائه شده است. از آنجاکه الگوریتم کلونی مورچگان، الگوریتمی بسیار پیچیده می‌باشد، هدف این الگوریتم جدید پیشنهادی، کاهش پیچیدگی های محاسباتی الگوریتم کلونی مورچگان و ارائه الگوریتمی ساده و قابل فهم می باشد.

۲. بیان ریاضی مسئله

مسئله مسیریابی وسائل نقلیه به همراه پنجره های زمانی، متشکل از ناوگانی از وسایل نقلیه با ظرفیت‌های معین، یک دپو مرکزی، تعداد معینی از مشتریان با تقاضاها و محدودیت‌های زمانی متفاوت، و شبکه‌ای که دپو مرکزی را به تمامی این مشتریان ارتباط می‌دهد، می‌باشد. برای سادگی، دپو مرکزی را به عنوان "مشتری شماره صفر" درنظر گرفته و با این حساب تعداد کل مشتریان برابر با $N+1$ می‌باشد. رویکرد مسیریابی هر وسیله نقلیه بدین صورت است: از دپو حرکت را آغاز می‌نماید، مشتریان را ملاقات می‌کند و در نهایت به دپو مرکزی باز می‌گردد.

هر مشتری i دارای تقاضای $m(i)$ می‌باشد و تنها می‌تواند یکبار و توسط تنها یک وسیله نقلیه ملاقات شود. وسیله نقلیه k دارای ظرفیت $q(k)$ می‌باشد و $q(k)$ باید بزرگتر و یا مساوی با مجموع کل تقاضای تمامی مشتریانی باشد که این وسیله آنها را ملاقات می‌نماید. اضافه بار ممنوع می‌باشد. پنجره زمانی بدین معنا است که هر مشتری دارای یک بازه زمانی از پیش تعریف شده است که شامل زودترین زمان ورود $e(i)$ و دیرترین زمان ورود $l(i)$ برای وسایل نقلیه، جهت ملاقات آن مشتری می‌باشد. وسایل نقلیه می‌بایست قبل از دیرترین زمان ورود $l(i)$ به محل مشتری برسند و اگر قبل از زودترین زمان ورود $e(i)$ به محل مشتری برسند، آنگاه باید منتظر بمانند و در اینجا زمان انتظار $w(i)$ بوجود می‌آید. هر مشتری دارای زمان سرویس $f(i)$ می‌باشد که دربرگیرنده زمان تخلیه بار/بارگیری کالاها می‌باشد. پنجره زمانی دپو بدین معناست که هر وسیله که زمان $e(0)$ از دپو خارج می‌شود، باید قبل از زمان $l(0)$ به آن بازگردد.

مسافت بین تمامی مشتریان براساس فاصله مستقیم الخط اقلیدوسی محاسبه می‌شود و سرعت حرکت هر وسیله برابر با یک واحد مسافت در هر واحد زمان می‌باشد. این فرض باعث سهولت حل مسئله می‌شود. مدل ریاضی مسئله مسیریابی وسائل نقلیه به همراه پنجره های زمانی ذیلاً ارائه شده است[۲،۴]:

$$\text{Minimize} \quad \sum_{i=0}^{N}\sum_{j=0}^{N}\sum_{k=1}^{K} c(i,j) * X_{ijk}$$

Subject to:

$$\sum_{j=1}^{N} X_{0jk} \le 1, \quad for \quad k \in (1,2,...,K) \qquad (۱)$$

$$\sum_{i=1}^{N} X_{i0k} \le 1, \quad for \quad k \in (1,2,...,K) \qquad (۲)$$

[Downloaded from ijiepm.iust.ac.ir on 2026-11-02]

$$\sum_{k=1}^{K}\sum_{i=0}^{N} X_{ijk} = 1 \quad for \quad j \in \{1,2,...N\} \qquad (۳)$$

$$\sum_{k=1}^{K}\sum_{j=0}^{N} X_{ijk} = 1 \quad for \quad i \in \{1,2,...N\} \qquad (۴)$$

$$\sum_{i=0}^{N}\left(m_i \sum_{j=0}^{N} X_{ijk} \right) \le q(k) \quad for \quad k \in \{1,2,...,K\} \qquad (۵)$$

$$\sum_{k=1}^{K}\sum_{j=0}^{N} X_{ijk}\left(t(i)+t(i,j)+ f(i)+w(i)\right)= t(j) \qquad (۶)$$
$$for \quad j \in \{1,2,...N\}$$

که در آن $t(0)= f(0)= w(0)=0$

$$\sum_{i=0}^{N}\sum_{j=0}^{N} X_{ijk}\left(t(i,j)+ f(i)+w(i)\right) \le l(0) \qquad (۷)$$
$$for \quad k \in \{1,2,...K\}$$

$$e(i) \le \left(t(i)+w(i)\right) \le l(i) \quad for \quad i \in \{1,2,...,N\} \qquad (۸)$$

بطوری که

k : وسیله نقلیه kام و $k \in \{1,2,...,K\}$

$C(i)$: مشتری iام و $i \in \{1,2,...N\}$

$C(0)$: دپو مرکزی

$d(i,j)$: فاصله مستقیم الخط اقلیدسی بین $C(i)$ و $C(j)$ که $i,j \in \{1,2,...N\}$ و $i \neq j$

$c(i,j)$: هزینه جابجایی از $C(i)$به $C(j)$ که $i,j \in \{1,2,...N\}$ و $i \neq j$

$t(i,j)$: زمان سفر بین $C(i)$و$C(j)$ که $i,j \in \{1,2,...N\}$ و $i \neq j$

$m(i)$: تقاضای مشتری iام

$q(k)$: ظرفیت وسیله نقلیه kام

$e(i)$: زودترین زمان ورود به محل مشتری iام

$l(i)$: دیرترین زمان ورود به محل مشتری iام

$f(i)$: زمان سرویس در محل مشتری iام

$t(i)$: زمان ورود به محل مشتری iام

$w(i)$: زمان انتظار در محل مشتری iام

$X_{ijk}=1 or 0$ ، یک برای زمانی که یک وسیله نقلیه از گره i به گره j حرکت کرده باشد و صفر در غیر اینصورت.

هدف کمینه نمودن هزینه کل می‌باشد. معادلات ۱ و ۲ محدودیت‌هایی می‌باشند که اجازه می‌دهند در بیشترین حالت، k وسیله نقلیه دپو مرکزی را ترک نموده و در نهایت مجدداً به همان دپو باز گردند. معادلات ۳ و ۴ هر مشتری را به ملاقات شدن توسط تنها یک وسیله نقلیه محدود می‌کنند. معادله ۵ محدودیت ظرفیت وسیله نقلیه می‌باشد. معادلات ۶ الی ۸ محدودیت‌های پنجره زمانی برای مشتریان و دپو مرکزی می‌باشد.

۳. الگوریتم کلونی مورچگان

الگوریتم کلونی مورچگان (ACO) یک روش فراابتکاری جدید برای حل مسائل بهینه سازی می‌باشد و برای اولین بار توسط مارکو دوریگو مطرح شد[۵]. این الگوریتم فراابتکاری مبتنی بر رفتارهای کلونی مورچه‌های واقعی است. در واقع مورچه‌های واقعی نابینا هستند اما توانایی یافتن کوتاهترین مسیر بسمت منبع غذایی را دارا می‌باشند و اینکار را به کمک برجای گذاری اثر فرمون انجام می‌دهند. مورچه‌های مصنوعی از قوانین انتقال (معادلات ۹ و ۱۰) استفاده نموده و این رفتار مورچه‌های طبیعی را تقلید و مقصد بعدی را می‌یابند. همچنین مورچه‌های مصنوعی از قوانین بروزرسانی اثر فرمون (معادلات ۱۱ و ۱۲) استفاده می‌نمایند و از آن برای بروزرسانی اثر فرمون روی مسیرها استفاده می‌کنند.

$$j = \begin{cases} \arg\max_{r \in T_K(i)} \left\{ [\lambda(i,r)] \cdot [v(i,r)]^{\beta} \right\} & if \quad q \le q_0 \\ J & o.w. \end{cases} \qquad (۹)$$

فرض کنید k یک مورچه مصنوعی باشد که وظیفه آن ایجاد یک مسیر است. این مورچه تمامی مشتریان را ملاقات می‌نماید و به نقطه اولیه باز می‌گردد. همراه با مورچه kام لیستی تحت عنوان $T_k(i)$شامل تمامی مشتریانی که هنوز ملاقات نشده‌اند، وجود دارد. مورچه kام که در محل مشتری iام قرار دارد براساس معادله ۱-۳ به محل مشتری j ام می‌رود. $\lambda(i,r)$ نشان دهنده مقدار اثر فرمون بر روی کمان بین دو مشتری i و r می‌باشد. $v(i,r)$یک مقدار ابتکاری است و از معکوس فاصله بین دو مشتری i و r حاصل می‌شود و β یک پارامتر می‌باشد که اهمیت نسبی $v(i,r)$ را نشان می‌دهد. q مقداری است که بطور تصادفی بین ۰ و ۱ تولید می‌شود و q_0 نیز پارامتری است که توسط کاربر تعیین می‌شود و بین ۰ و ۱ می‌باشد. J نیز یک متغیر تصادفی است و براساس تابع توزیع احتمال ارائه شده در معادله ۱۰ تولید می‌شود. در معادله ۱۰ ، $P_k(i,j)$احتمال انتخاب مشتری j ام می‌باشد.

$$P_k(i,j) = \begin{cases} \dfrac{[\lambda(i,j)] \cdot [v(i,j)]^{\beta}}{\sum_{r \in T_K(i)} [\lambda(i,r)] \cdot [v(i,r)]^{\beta}} & if \quad j \in T_k(i) \\ 0 & o.w. \end{cases} \qquad (۱۰)$$

معادله ۱۱ برای بروزرسانی محلی اثر فرمون استفاده می‌شود. این بروزرسانی محلی زمانی اتفاق می‌افتد که مورچه kام از محل مشتری i به محل مشتری j می‌رود. در این معادله ρ پارامتر محو

[Downloaded from ijiepm.iust.ac.ir on 2026-11-02]

شدن اثر فرمون محلی روی مسیر می‌باشد و مقدار آن بین ۰ و ۱ تعریف می‌شود و λ_0 مقدار فرمون اولیه بر روی مسیرها می‌باشد.

$$\lambda(i,j)=(1-\rho)\lambda(i,j)+\rho\Delta\lambda(i,j),\quad \Delta\lambda(i,j)=\lambda_0 \qquad (۱۱)$$

پس از آنکه مورچه‌ها مسیرهای خود را تولید نمودند، بروزرسانی کلی اثر فرمون[1] انجام می‌گیرد. اینکار براساس معادله ۱۲ صورت می‌پذیرد که در آن α پارامتر محو شدن اثر فرمون کلی. روی مسیر می‌باشد و مقدار آن بین ۰ و ۱ تعریف می‌شود. همچنین L_{gb} طول بهترین مسیر از ابتدای حل است.

$$\lambda(i,j)=(1-\alpha)\lambda(i,j)+\alpha\frac{1}{L_{gb}} \qquad (۱۲)$$

۴. الگوریتم کلونی مورچگان اصلاح شده

محاسباتی در این الگوریتم برگرفته از بهینه سازی کلونی مورچگان می‌باشد، درواقع نوع اصلاح شده الگوریتم بهینه سازی کلونی مورچگان مطرح می‌شود. در این الگوریتم نوع ناوگان بصورت ناهمگن[2] درنظر گرفته شده است[۶-۷]. در این رویکرد چون مورچه‌ها بعنوان نمایندگانی برای وسائل نقلیه قرار داده شده‌اند، پس با داشتن k نوع وسیله نقلیه، k نوع مورچه خواهیم داشت. یعنی در کل n وسیله (nمورچه) از k نوع خواهیم داشت. هر یک از این k نوع مورچه (وسیله) دارای یک ظرفیت معین می‌باشند و مسئله را حل خواهند نمود. در شکل ۱ (پیوست مقاله) فلوچارت الگوریتم تولید جواب ارائه شده است.

در ابتدا باید براساس صورت مسئله، ماتریس فاصله تشکیل شود، این ماتریس از فاصله مستقیم‌الخط اقلیدسی استفاده می‌کند، یعنی فاصله بین دو گره i و j که دارای مختصات‌های بترتیب (x_i,y_i) و (x_j,y_j) می‌باشند، عبارت است از $d_{ij}=\sqrt{(x_i-x_j)^2+(y_i-y_j)^2}$ و ماتریس مربعی با بعدی برابر تعداد مشتریان + ۱ حاصل می‌نماید.

پس از تشکیل ماتریس فاصله، نوبت به یافتن یک حل اولیه می‌باشد. علت یافتن حل اولیه این است که از طول مسافت طی شده در آن، جهت محاسبه مقدار فرمون اولیه بر روی مسیرها استفاده می‌شود. حل اولیه در اینجا با استفاده از روش ابتکاری نزدیکترین همسایه[3] حاصل می‌شود. البته این حل ممکن است موجه یا غیر موجه باشد. (حل موجه به حلی گفته می‌شود که در آن تمامی گره‌ها تنها یکبار ویزیت شده و مقدار بار در کل مسیر از ظرفیت وسیله نقلیه فراتر نرفته و پنجره‌های زمانی نقض نشده باشند).

گام‌های الگوریتم ابتکاری نزدیکترین همسایه:

مشتریان (گره‌ها) و مختصات (x,y) آنها بصورت ستونی یادداشت کنید. از گره دپو آغاز کنید. فاصله تمامی گره‌ها را از این گره محاسبه نمائید. کمترین فاصله را در نظر بگیرید و نام گره را در یک لیست بنویسید. فاصله تمامی گره‌ها را از گره مرحله قبل پیدا کنید. مجدداً کمترین فاصله را یافته و به لیست اضافه کنید. اینکار را ادامه داده تا تمامی گره‌های موجود، در لیست وارد شوند و دیگر گره‌ای باقی نماند. به ترتیب از گره‌ای که از آن شروع نموده‌اید تا آخرین گره، کمترین فاصله‌ها را با هم جمع نمائید.این مقدار همان کوتاهترین مسیر (موجه یا غیر موجه) اولیه می‌باشد. پس از یافتن کوتاهترین مسیر اولیه، طول آنرا در نظر گرفته و مقدار فرمون اولیه برای تمامی مسیرهای مسئله برابر با رابطه زیر قرار می‌گیرد، که در آن L طول کوتاهترین مسیر و n تعداد مشتریان (گره‌های) موجود در حل می‌باشد. λ_0 نیز مقدار فرمون اولیه روی مسیر است.

$$\lambda_0=\frac{1}{nL}$$

این الگوریتم به دفعات تکرار می‌شود. در هر تکرار الگوریتم تمامی k نوع مورچه (وسیله) مسئله مورد نظر را حل می‌کنند. منظور از حل کردن مسئله این است که مثلاً مورچه اول پس از قرار گرفتن در دپو، با محاسبه مطلوبیت رفتن به گره‌های مجاور، بهترین گره را برای حرکت بعدی یافته و بدین ترتیب گره‌ها را یکی پس از دیگری پشت سر می‌گذارد، تا زمانی که دیگر گره‌ای باقی نماند و مورچه به دپو باز گردد (البته ممکن است مورچه بخاطر پر شدن ظرفیت یا نقض پنجره‌های زمانی، بارها وبارها به گره دپو بازگردد). پس از آنکه تمامی k نوع مورچه (وسیله) مسیرهای خود را تشکیل دادند، این مسیرها براساس معیارهایی مورد ارزیابی قرار می‌گیرند و بهترین مسیر برای بروزرسانی مقدار فرمون مسیرهای مسئله مورد استفاده قرار می‌گیرد. یعنی در واقع در این الگوریتم تنها یکبار بروزرسانی انجام می گیرد و آن هم در پایان هر تکرار از الگوریتم است.

بمنظور اتمام الگوریتم باید معیاری برای توقف آن تعریف نمود. نمونه‌هایی از معیارهای قابل تعریف در زیر ارائه شده اند:

۱. تعداد تکرارهای معین سپری شود (بطور مثال ۱۰۰۰ تکرار الگوریتم)

۲. زمان محاسباتی معینی سپری شود (بطور مثال ۲ ساعت اجرای الگوریتم)

۳. پس از گذشت تعداد معینی از تکرارها، هیچ بهبودی در حل حاصل نشود.

هر یک از k نوع مورچه پس از قرار گرفتن در گره دپو، برای یافتن گره‌های موجهی که می‌توانند در مرحله بعد به آنها بروند از الگوریتم فیلترینگ استفاده می‌کنند. مراحل این الگوریتم ذیلاً ارائه شده است:

گام های الگوریتم فیلترینگ (جهت انتخاب مشتری بعدی):

[1] *Global Pheromone Trail Update*
[2] *Heterogeneous*
[3] *Nearest Neighbor Heuristic*

[Downloaded from ijiepm.iust.ac.ir on 2026-11-02]

۱. لیست تمامی مشتریانی که هنوز ملاقات نشده‌اند را تهیه کنید. $T_k(i)$ لیست تمامی مشتریان ملاقات نشده.

۲. برای تمامی مشتریان لیست، محاسبات زیر را انجام دهید:

۲-۱. Load := Load ام k مورچه + (تقاضای مشتری فعلی (در این بند، بار تجمیعی هر مورچه (وسیله) با اضافه نمودن تقاضای مشتری n ام محاسبه می شود)

۲-۲. Arrival time ام k زمان مورچه( ) + زمان سفر از مشتری فعلی به مشتری جدید

(در این بند، زمانی که مورچه (وسیله) k ام به مشتری n ام می‌رسد، محاسبه می‌شود)

۳. اگر مقدار Load وسیله k ام کوچکتر یا مساوی ظرفیت آن باشد و مقدار Arrival time نیز کوچکتر و یا مساوی با دیرترین زمان ورود به مشتری جدید باشد؛ (در این بند محدودیت‌های مربوط به ظرفیت مورچه (وسیله) k ام و پنجره زمانی مشتری n ام بررسی می‌شود)

۴. اگر مجموع چهار مقدار Arrival time، زمان انتظار در محل مشتری جدید، زمان سرویس به مشتری جدید و زمان سفر از مشتری جدید به دپو، کوچکتر و یا مساوی با دیرترین زمان ورود به دپو باشد، (در این بند بررسی می‌شود که آیا مورچه (وسیله) k ام می‌تواند دیرترین زمان ورود به دپو را پس از اضافه نمودن مشتری n ام، ارضاء نماید) آنگاه:

۵. مشتری جدید به لیست $R_k(i)$ مورچه k ام اضافه می‌شود. (لیست مشتریان موجهی که دو محدودیت فوق را ارضاء می‌کنند، مربوط به مورچه k ام)

۶. اگر لیست خالی بود، گره دپو را به آن اضافه کن (بدین معنی که اگر مشتری دیگری برای اضافه نمودن به لیست $R_k(i)$ وجود ندارد، آنگاه گره دپو (C_0) را به لیست $R_k(i)$ اضافه نماید)

در نتیجه این الگوریتم، لیست تمامی مشتریانی که محدودیت‌ها را ارضاء می‌کنند حاصل می‌شود. حال باید از روی این لیست و محاسبه و تحلیل روابط زیر، مشتری دارای احتمال بالاتر را برای حرکت بعدی انتخاب نمود.

$$j=\begin{cases}\arg\max\limits_{r\in R_k(i)}\left\{[\lambda(i,r)]\cdot[v(i,r)]^{\beta}\cdot[WC]^{\gamma}\right\} & if\ \ q\le q_0\\ J & o.w.\end{cases} \qquad (۱۳)$$

$$P_k(i,j)=\begin{cases}\dfrac{[\lambda(i,j)]\cdot[v(i,j)]^{\beta}\cdot[WC]^{\gamma}}{\sum\limits_{r\in R_k(i)}[\lambda(i,r)]\cdot[v(i,r)]^{\beta}\cdot[WC]^{\gamma}} & if\ \ j\in R_k(i)\\ 0 & o.w.\end{cases} \qquad (۱۴)$$

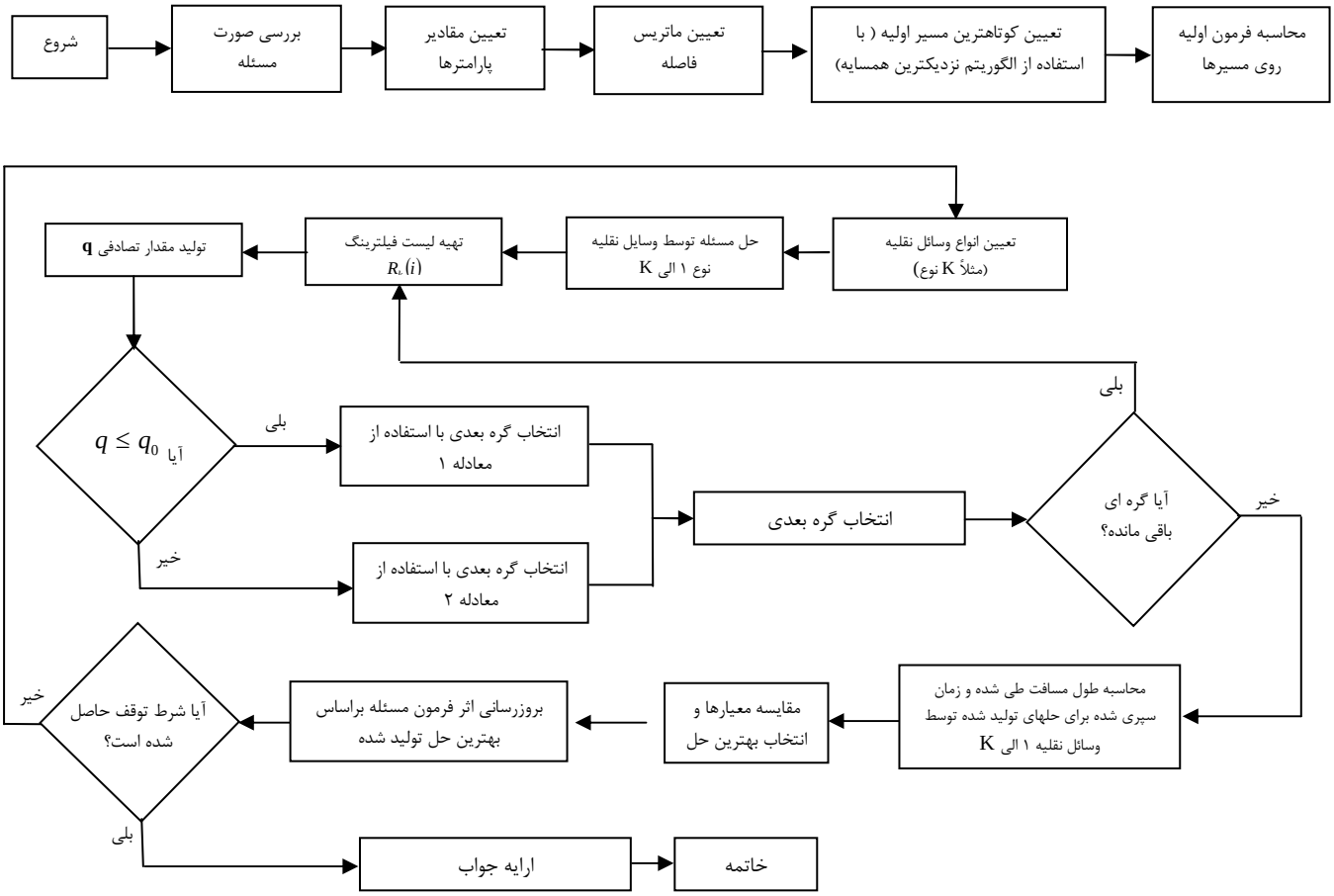
با تعریف یک مقدار $q_0\ (0\le q_0\le 1)$ و تولید یک مقدار تصادفی q، بررسی می‌کنیم که آیا $q\le q_0$ می‌باشد یا خیر؟ اگر $q\le q_0$ باشد از معادله ۱۳ و در غیر اینصورت از تابع احتمال موجود در معادله ۱۴ برای یافتن گره بعدی استفاده می‌کنیم. یعنی در واقع در هر تکرار برای هر نوع از مورچه‌ها (وسیله‌ها) بطور مجزا مقدار تصادفی q تولید شده و با مقدار q_0 مقایسه می‌گردد؛ البته باید توجه داشت که برای هر نوع مورچه (وسیله) به تعداد گام هایی که در هر تکرار طی می‌نماید تا حل کامل حاصل شود، مقدار تصادفی q تولید می‌گردد. در معادلات مذکور $\lambda(i,r)$ نشانگر مقدار فرمون موجود بر روی کمان (i,r) می‌باشد. $v(i,r)$ یک تابع ابتکاری است و برابر با معکوس فاصله بین دو گره i و r بوده و از آن تحت عنوان نزدیکی (Closeness) نیز یاد می‌شود و اولویت بالاتری به مشتریانی که نزدیکتر هستند اختصاص می‌دهد. WC نیز یک تابع ابتکاری است و از معکوس طول پنجره زمانی موجود در محل مشتری (گره) r حاصل می‌شود و سبب می‌گردد اولویت بالاتری به مشتریانی که طول پنجره زمانی کوتاهتری دارند، داده شود. γ پارامتری است که سطح اهمیت نسبی WC و β نیز سطح اهمیت نسبی $v(i,r)$ را نشان می‌دهد. پس از آنکه تمامی k نوع مورچه مسیرهای خود را تولید نمودند، حال نوبت بروزرسانی مقدار فرمون مسیرها می‌باشد. بمنظور بروزرسانی اثر فرمون دو رویکرد وجود دارد. یکی اینکه براساس تمامی مسیرهای تولید شده توسط تمامی k نوع مورچه بروزرسانی انجام شود و دیگر اینکه براساس بهترین مسیر تولید شده توسط تمامی k نوع مورچه، بروزرسانی انجام گیرد. رویکردی که در اینجا مورد استفاده قرار گرفته رویکرد دوم است. یعنی برای بروزرسانی اثر فرمون بر روی مسیرها، در انتهای هر تکرار از بهترین حل تولید شده در آن تکرار استفاده شود. البته فرض شده که تمامی k نوع مورچه به یک میزان فرمون مسیر را تغییر می دهند. یعنی مثلاً مقدار تغییر اثر فرمون در اثر حرکت مورچه نوع ۱ از یک گره به گره دیگر با اثر فرمون مورچه نوع ۲ برابر است. بعنوان مثال فرض می‌کنیم سه نوع وسیله (مورچه) داشته باشیم: سواری، مینی بوس و اتوبوس. در انتهای تکرار اول، هر کدام از این سه نوع وسیله نقلیه مسیری را تولید می‌کنند. حال باید این مسیرهای تولید شده را با هم مقایسه نمود. برای مقایسه می‌توان از یکسری معیار استفاده نمود. معیاری که در اینجا مورد استفاده قرار گرفته کل طول مسافت طی شده است. پس از یافتن بهترین مسیر تولید شده، از آن برای بروزرسانی اثر فرمون استفاده می‌کنیم و طبق رابطه زیر مقدار اثر فرمون بروز می‌گردد:

$$\lambda(i,j)=(1-\alpha)\lambda(i,j)+\alpha\frac{1}{L_{gb}} \qquad (۱۵)$$

L_{gb} طول بهترین مسیر تولید شده در هر تکرار بوده و α هم یک پارامتر است و میزان زوال اثر فرمون را تعیین می‌کند.

در این الگوریتم بجای دو بار بروزرسانی اثر فرمون (بروزرسانی محلی و بروزرسانی کلی)، بروزرسانی اثر فرمون تنها یکبار و توسط بهترین حل تولید شده در هر تکرار صورت می‌گیرد و این خود حجم زیادی از محاسبات را کاهش داده و سرعت پاسخگویی الگوریتم را بالا می‌برد.

[Downloaded from ijiepm.iust.ac.ir on 2026-11-02]



شکل ۱. فلوچارت الگوریتم پیشنهادی

۵. نتایج محاسباتی

کدنویسی در نرم افزار Visual Basic 6.0 انجام شده است. در این کد می‌توان ورودیهای ابتدایی مسئله از جمله: مختصات، میزان تقاضا، پنجره‌های زمانی و مدت زمان سرویس برای هر مشتری، را وارد نموده و جوابها شامل تعداد وسائل نقلیه، بهترین مسافت و مورچه‌ای که بهترین مسافت را تولید نموده حاصل می‌شود. برنامه تهیه شده بر روی یک دستگاه رایانه شخصی اینتل با قدرت CPU ۸۰۰ مگاهرتز و ۲۵۶ مگابایت حافظه رم اجرا گردید. البته سیستم مذکور حداقل سیستم مورد نیاز برای اجرای برنامه می‌باشد و علت استفاده از آن، شبیه سازی فضای محاسباتی مطابق با شرایطی است که در مقالات دیگر به آن اشاره شده است. جهت تست این نرم افزار از نمونه مسائل معروف Solomon که توسط Marius Solomon تولید شده، استفاده شده است[۸]. این مسائل براساس ویژگی‌ها و خصوصیاتی که دارا می‌باشند دسته بندی شده‌اند. بطورکلی این مسائل در سه گروه کلی قرار گرفته اند. هر گروه شامل ۸ تا ۱۲ مسئله و شامل حداکثر ۱۰۰ مشتری می‌باشد. این ۸ تا ۱۲ مسئله دارای اختلافات جزئی در۴ خصوصیت مختصات، میزان تقاضا، پنجره های زمانی و زمان سرویس می‌باشند.

جدول ۱. جدول مقایسه‌ای جوابهای الگوریتم پیشنهادی و جوابهای بهینه

	جوابهای الگوریتم جدید			جواب های بهینه		
٪ خطا	بهترین مسافت	تعداد انواع وسایل نقلیه	مسافت بهینه	تعداد انواع وسایل نقلیه	تعداد مشتریان	مسئله
0/27%	191/81	3	191/3	3	25	C101
6/42%	387/25	5	362/4	5	50	C101
3/01%	852/94	10	827/3	10	100	C101
5/95%	656/11	8	617/1	8	25	R101
11/28%	1176/67	13	1044	12	50	R101
6/69%	494/14	4	461/1	4	25	RC101
10/47%	1054/44	9	944	8	50	RC101

[Downloaded from ijiepm.iust.ac.ir on 2026-11-02]

معنای اسامی این ۳ دسته بصورت زیر است:

- مجموعه C شامل مشتریانی است که براساس موقعیت جغرافیایی و یا پنجره زمانیشان دسته بندی شده اند
- مجموعه R شامل مشتریانی است که بطور تصادفی و یا بطور یکنواخت توزیع شده اند
- مجموعه RC شامل ترکیبی از مشتریان دو مجموعه R و C می باشند

هر یک از این سه نوع کلی از مسائل به نوبه خود به دو دسته فرعی نیز تقسیم می‌شوند. مسائل نوع ۱ و نوع ۲. مسائل نوع ۱ دارای پنجره‌های زمانی کوچک می‌باشند و ظرفیت وسائل نقلیه در آنها کم است اما در مسائل نوع ۲، پنجره‌های زمانی بزرگ و ظرفیت وسائل نقلیه نیز زیاد می باشد. ترکیب‌هایی که از این سه دسته کلی و دو دسته فرعی حاصل می‌شوند عبارتند از:

$$(R_1, R_2, C_1, C_2, RC_1, RC_2)$$

از دسته C و نوع ۱، سه مسئله با ابعاد ۲۵، ۵۰ و ۱۰۰، از دسته R و نوع ۱، دو مسئله با ابعاد ۲۵ و ۵۰ و از دسته RC و نوع ۱، دو مسئله با ابعاد ۲۵ و ۵۰ انتخاب شد. پس از بررسی و تخصیص مقادیر گوناگون به پارامترهای موثر بر مسئله، بهترین جوابهای حاصل شده در جدول ۱ ارائه شده‌اند. الگوریتم پیشنهادی جوابهایی نزدیک به بهینه را در زمان نسبتاً کوتاهی تولید می نماید. همانگونه که در جدول ملاحظه می شود، در خصوص بهترین مسافت حاصل شده میزان خطای محاسباتی حداکثر به میزان 10% می باشد. درخصوص تعداد وسائل نقلیه نیز که یکی دیگر از معیارهای ارزیابی جواب‌ها می‌باشد، بجز مسئله R101 و RC101 با ۵۰ مشتری، که تعداد وسائل نقلیه تنها یک واحد متفاوت است، در سایر موارد تعداد وسائل نقلیه الگوریتم پیشنهادی با تعداد وسائل نقلیه حل بهینه برابر است.

۶. نتیجه گیری

در این مقاله نوع اصلاح شده الگوریتم کلونی مورچگان معرفی و برای مسئله مسیریابی وسائل نقلیه به همراه پنجره های زمانی بکار گرفته شد. این الگوریتم توانست برای مسائل نمونه‌ای با ابعاد حداکثر ۱۰۰ مشتری، جوابهای قابل قبول و نزدیک به بهینه تولید نماید. در مورد برخی از این مسائل، درصد خطا به حدود 10% نسبت به حل بهینه می رسد که این امر به علت بزرگ بودن فضای حل مسئله، حجم محاسبات زیاد و گیر افتادن الگوریتم در بهینه‌های محلی می‌باشد. البته با افزودن ملاحظات دیگر، می‌توان این میزان خطا را کاهش داده و جوابهایی نزدیک‌تر به جوابهای بهینه ایجاد نمود که این موضوع به محققین آتی واگذار می شود.

نوآوری که در این مقاله صورت گرفته، در نظر گرفتن ناوگانی ناهمگن از وسائل نقلیه می‌باشد، یعنی ناوگانی متشکل از انواع متفاوتی از وسائل نقلیه با ظرفیت‌های متفاوت، که این خود بر پیچیدگی محاسبات افزوده و نیز سبب افزایش زمان محاسباتی می‌گردد. در این الگوریتم جدید، هر یک از انواع وسائل نقلیه موجود در ناوگان وسائل نقلیه، مسئله را بطور مجزا حل نموده و از بهترین و کوتاهترین حل حاصل شده توسط بهترین نوع از وسایل، برای بروزرسانی اثر فرمون استفاده می‌شود، یعنی تنها یکبار بروزرسانی اثر فرمون صورت‌می‌گیرد؛ البته می‌توان با در نظر گرفتن آمیخته ای از وسائل نقلیه بعنوان ناوگان وسائل نقلیه، بطور همزمان از تمامی آنها برای تولید جواب استفاده نمود.

نتایج حاصله نشان داد که الگوریتم ارائه شده از روایی و پایایی مناسبی برخوردار می باشد. گام بعدی در تکمیل این الگوریتم حل، می‌تواند بکارگیری الگوریتم‌های جستجوی محلی جهت بهبود جوابهای حاصل شده باشد.

مراجع

[1] Rizzoli, A.E., Oliverio, F., Montemanni, R., Gambardella, L.M., *Ant Colony Optimisation for Vehicle Routing Problems: From Theory to Applications*, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA) Galleria 2, CH-6928 Manno, Switzerland, AntOptima, via Fusoni 4, CH-6900 Lugano, Switzerland.

[2] Wesley Changchien, S., Chih-Sheng Wu, *An Ant Colony System for Vehicle Routing Problems with Time Window*, Department of Information Management, Chaoyang University of Technology, 2001.

[3] Luca Maria Gambardella, Éric Taillard and Giovanni Agazzi, *MACS-VRPTW A Multiple Ant Colony System For Vehicle Routing Problems With Time Windows*, IDSIA, Corso Elvezia 36, 6900 Lugano, Switzerland, New Ideas in Optimization McGraw-Hill, London, UK, 1999, pp. 63-76.

[4] Chia-Ho CHEN, Ching-Jung TING, *A Hybrid Ant Colony System for Vehicle Routing Problem with Time Windows*, Department of Industrial Engineering and Management, Journal of the Eastern Asia Society for Transportation Studies, Vol. 6, 2005, pp. 2822 - 2836.

[5] Dorigo, M., Maniezzo, V., Colorni, A., *The Ant System: Optimization by a Colony of Cooperating Agent*, *IEEE* Transactions on Systems, Man and Cybernetics, Part B 26 (1), 1996, pp. 29–41.

[6] Toth, P., Vigo, D., *The Vehicle Routing Problem*, Universita delgi Studi di Bologna, Blogna, Italy, 2001.

[7] Cordeau, J.F.,Guy Desaulniers, Guy Desaulniers, Marius M., Solomon, Fran¸cois Soumis, *The VRP with Time Windows*, February, 1999, Revised: June, 2000.

[8] Alberto, V., Donati, Roberto Montemanni, Norman, Casagrande, Andrea, E., Rizzoli, Luca, M., Gambardella, *Time Dependent Vehicle Routing Problem*

[Downloaded from ijiepm.iust.ac.ir on 2026-11-02]

with a Multi Ant Colony System, Istituto Dalle Molle di Studi sull'Intelligenza Artificiale (IDSIA), Galleria 2, 6928 Manno, Switzerland, Received 1 June 2005; accepted 1 June 2006, European Journal of Operational Research 185, 2008, pp. 1174–1191.

[Downloaded from ijiepm.iust.ac.ir on 2026-11-02]