

یک روش فراابتکاری ترکیبی برای مسئله مکان‌یابی-مسیریابی وسیله نقلیه ظرفیت‌دار با پنجره‌های زمانی سخت

علی‌رضا محمدی شاد^۱ و پرویز فتاحی*^۲

^۱ دانش‌آموخته کارشناسی ارشد مهندسی صنایع - دانشگاه بوعلی سینا

^۲ دانشیار و مدیر گروه مهندسی صنایع - دانشگاه بوعلی سینا

(تاریخ دریافت ۹۱/۱/۲۷، تاریخ دریافت روایت اصلاح‌شده ۹۱/۳/۲۰، تاریخ تصویب ۹۱/۴/۱۸)

چکیده

مسئله مکان‌یابی-مسیریابی وسیله نقلیه ظرفیت‌دار^۱ (CLRP)، یکی از حوزه‌های جدید تحقیقاتی در مدیریت پخش است. این موضوع، دو مسئله مشکل مکان‌یابی تسهیلات و مسیریابی وسایل نقلیه را با یکدیگر ترکیب می‌کند. هدف از CLRP گشودن مجموعه‌ای از دپوها، تخصیص مشتری‌ها به دپوهای احداث‌شده و سپس طراحی تورهای وسیله نقلیه برای کمینه‌کردن هزینه کل است. محدودیت پنجره‌های زمانی کاربردهای زیادی در دنیای واقعی دارد، با این وجود در CLRP اهمیت کمی به آن داده شده است. این مقاله، مسئله مکان‌یابی-مسیریابی وسیله نقلیه ظرفیت‌دار را با پنجره‌های زمانی سخت^۲ (CLRPHTW) در نظر می‌گیرد. در این مقاله، ابتدا یک مدل برنامه‌ریزی خطی عدد صحیح مختلط برای CLRPHTW ارائه شده و سپس روشی فراابتکاری بر مبنای الگوریتم جستجوی همسایگی متغیر برای حل این مسئله پیشنهاد می‌شود. برای ارزیابی عملکرد روش پیشنهادی، این چارچوب با استفاده از یک مجموعه مثال‌های آزمایش مورد بررسی قرار می‌گیرد. آزمایش‌های محاسباتی کارآیی روش پیشنهادی را نشان می‌دهند.

واژه‌های کلیدی: مکان‌یابی، مسیریابی وسیله نقلیه، پنجره زمانی، روش فراابتکاری، جستجوی همسایگی متغیر،

بهینه‌سازی ترکیب

مقدمه

سبب افزایش هزینه‌های پشتیبانی زنجیره تأمین خواهد شد. هر دو مسئله مکان‌یابی تسهیلات و مسیریابی وسیله نقلیه پیچیدگی زمانی از نوع مسائل NP-hard دارند، بنابراین LRP نیز مسئله‌ای با پیچیدگی زمانی از نوع مسائل NP-hard است. بنابراین، حل LRP در اندازه بزرگ با استفاده از روش‌های دقیق، سخت و تقریباً ناممکن است [۱]. به همین دلیل در بیشتر تحقیقات انجام گرفته برای حل آن، به توسعه روش‌های ابتکاری و فراابتکاری پرداخته شده است.

در مسئله LRP محدودیت ظرفیت شامل دپو یا وسیله نقلیه (نه هر دو باهم) می‌شود. به تازگی تعدادی از محققان مسئله LRP را با محدودیت ظرفیت، هم در دپو و هم در وسیله نقلیه مورد مطالعه قرار داده‌اند [۲]. این موضوع با عنوان مسئله مکان‌یابی-مسیریابی وسیله نقلیه ظرفیت‌دار (CLRP) نام‌گذاری شده است. پرینس^۴ و همکاران [۲] در تحقیق خود، مدل ریاضی برای CLRP ارائه کرد. آن‌ها CLRP را با استفاده از ترکیب رویکرد جستجوی توافقی تصادفی حریصانه^۵ (GRASP) و یک

در دو دهه اخیر رقابت بین شرکت‌ها در راستای عرضه کالا و خدمات، تبدیل به واقعیتی بزرگ برای پیشرفت آن‌ها شده است. امروزه شرکت‌ها و کارخانجات نیازمند یکپارچه‌سازی و منعطف کردن همه فعالیت‌های تولیدی از تهیه مواد اولیه تا تحویل کالای نهایی به مصرف‌کننده، در فرایند عرضه و پخش محصولات خود هستند. سیستم‌های توزیع و پشتیبانی بخشی از فرآیند زنجیره عرضه‌اند، به طوری که جریان مؤثر و کارای انبارش کالاها، خدمات و اطلاعات وابسته به آن‌ها را از نقطه شروع تا نقطه مصرف به منظور برآورده کردن نیازمندی‌های مشتری برنامه‌ریزی، اجرا و کنترل می‌کنند. لجستیک جزئی مهم از مدیریت زنجیره تأمین است.

مسئله مکان‌یابی-مسیریابی وسیله نقلیه^۳ (LRP)، یکی از مسائل مهم و کاربردی در مدیریت پخش و لجستیک است. LRP ترکیبی از دو مسئله مکان‌یابی تسهیلات و مسیریابی وسیله نقلیه است که به طور همزمان هر دو این مسائل را در نظر می‌گیرد. در نظر نگرفتن همزمان هر دو مسئله مکان‌یابی و مسیریابی،

خود، همین مسئله را با استفاده از روشی مبتنی بر الگوریتم جستجوی همسایگی تکرارشونده حل کردند و جواب‌های تحقیق پیشین خود را بهبود دادند [۱۰].

با وجود کاربرد فراوان محدودیت پنجره زمانی در مسائل دنیای واقعی، در CLRP اهمیت کمی به آن داده شده است. جبل عاملی و غفاری نسب [۱۱] مسئله مکان‌یابی-مسیریابی وسیله نقلیه را در هر دو حالت پنجره‌های زمانی نرم و سخت مدل‌سازی کردند و با ارائه مثالی کوچک اعتبار مدل خود را به نمایش گذاردند. نیک‌بخش و ذگردی مسئله مکان‌یابی-مسیریابی وسیله نقلیه دو سطحی با محدودیت پنجره‌های زمانی نرم را مورد مطالعه قرار دادند. نویسندگان این تحقیق برای مسئله مدل برنامه‌ریزی ریاضی چهار - شاخصه و روشی ابتکاری برای حل ارائه کردند. همچنین آنها با استفاده از آزادسازی لاگرانژی کران پایینی برای مسئله به دست آوردند و جواب‌های حاصل از الگوریتم ابتکاری خود را با آن مقایسه کردند.

هدف اصلی در این تحقیق، افزودن محدودیت پنجره‌های زمانی به CLRP برای کاربردی‌تر کردن مسئله است. محدودیت پنجره‌های زمانی، از مسئله مسیریابی وسیله نقلیه با محدودیت پنجره‌های زمانی^{۱۲} (VRPTW) گرفته شده است. این موضوع با عنوان مسئله مکان‌یابی-مسیریابی وسیله نقلیه ظرفیت‌دار با محدودیت پنجره‌های زمانی سخت (CLRPHTW) نام‌گذاری می‌شود. در این تحقیق ابتدا به مدل‌سازی ریاضی مسئله پرداخته می‌شود. سپس با توجه به پیچیدگی حل مسئله و ناتوانی روش‌های حل دقیق در حل مسئله با اندازه‌های متوسط و بزرگ در زمانی معقول، الگوریتمی فراابتکاری بر مبنای روش جستجوی همسایگی متغیر برای حل آن ارائه می‌شود. نوآوری تحقیق حاضر و تفاوت آن با سایر تحقیق‌ها، افزودن مدت زمان انتظار به مدل ریاضی ارائه‌شده توسط جبل عاملی و غفاری نسب [۱۱] برای مسئله مکان‌یابی-مسیریابی وسیله نقلیه ظرفیت‌دار با محدودیت پنجره‌های زمانی سخت و طراحی الگوریتم فراابتکاری برای حل این مسئله است.

ادامه مقاله به صورت ذیل سازمان‌دهی شده است: در بخش دوم، به بیان تعریفی از مسئله و ارائه مدل ریاضی آن پرداخته می‌شود. در بخش بعد، الگوریتم پیشنهادی به همراه جزئیات آن ارائه می‌شود. در بخش چهارم، نتایج

فرایند یادگیری و یک مکانیزم مرتبط کردن مسیرها پیشنهاد کردند. همچنین آنها در تحقیقی دیگر از یک الگوریتم ممتیک با مدیریت جمعیت استفاده کردند. در این الگوریتم بهبود جواب‌ها با استفاده از جستجوهای محلی و جایگزینی عملگر سنتی جهش با یک تکنیک مدیریت جمعیت پویا بر مبنای فاصله تا جمعیت، انجام می‌گیرد [۳].

بارتو^۴ و همکاران [۴] از یک روش ابتکاری مبتنی بر دسته‌بندی مشتری‌ها برای حل CLRP استفاده کردند. آنها از چند روش سلسله مراتبی و غیر سلسله مراتبی برای دسته‌بندی استفاده کردند. پریز و همکاران [۵] از تکنیک سست‌سازی لاگرانژی و جستجوی ممنوعه دانه‌ای برای توسعه رویکردی دو فازی برای حل CLRP استفاده کردند. این الگوریتم به طور متناوب بین یک فاز مکان‌یابی و یک فاز مسیریابی، اطلاعات را جابه‌جا می‌کند. در فاز اول مسیرها و مشتری‌های آن‌ها به صورت یک ابرمشتری ترکیب می‌شوند تا مسئله به یک مسئله مکان‌یابی تسهیلات تبدیل شود. آزادسازی لاگرانژی در محدودیت‌های تخصیص برای حل مسئله مکان‌یابی تسهیلات به دست آمده مورد استفاده قرار می‌گیرد. در فاز دوم جستجوی ممنوعه دانه‌ای برای توسعه مسیریابی چند دپویی به دست آمده از فاز اول به کار می‌رود.

ماریناکیس و ماریناکی^۶ [۶] از روشی ترکیبی مبتنی بر الگوریتم بهینه‌سازی گروه ذرات برای حل مسئله استفاده کردند. دوهمال^۸ [۷] الگوریتم (GRASP+ELS)^۹ که ترکیبی از الگوریتم جستجوی تصادفی حریمانه توافقی با روش جستجوی محلی تکاملی برای حل مسئله CLRP استفاده کردند. یو^{۱۰} و همکاران [۸] از الگوریتم شبیه‌سازی تبرید ابتکاری برای حل CLRP استفاده کرده‌اند. آنها برای بهبود عملکرد الگوریتم شبیه‌سازی تبرید از سه ساختار همسایگی استفاده کردند. آنها ادعا کردند که استفاده از این ساختار سبب بهبود در الگوریتم شبیه‌سازی تبرید برای حل CLRP می‌شود. آنها برای بهبود عملکرد الگوریتم شبیه‌سازی تبرید از سه ساختار همسایگی با انتخاب احتمالی استفاده کردند. به تازگی نگیون^{۱۱} و همکاران [۹] در تحقیق خود از الگوریتم جستجوی تصادفی حریمانه توافقی با استفاده از فرایند یادگیری برای حل CLRP در دو سطح استفاده کردند. آنها همچنین در تحقیق دیگر

ظرفیت‌دار ارائه‌شده توسط پرینس و همکاران [۵]، مسئله مورد تحقیق مدل‌سازی می‌شود. مجموعه‌ها:

I مجموعه نقاط کاندید احداث دپو

J مجموعه مشتری‌ها

V مجموعه کل نقاط (گره‌ها) که $V = I \cup J$

K مجموعه وسایل نقلیه

پارامترهای مدل:

d_j تقاضای مشتری j .

O_i هزینه بازگشایی دپوی کاندید i

F_i هزینه ثابت استفاده از یک وسیله نقلیه در دپوی

کاندید i

C_{ij} هزینه طی مسافت از گره i به گره j

t_{ij} مدت زمان طی مسیر از گره i به گره j

Cap_i ظرفیت دپوی کاندید i

Q ظرفیت هر وسیله نقلیه

T حداکثر مدت زمانی طی یک تور

e_i زودترین زمان شروع سرویس به گره i

l_i دیرترین زمان شروع سرویس به گره i

St_j مدت زمان ارائه سرویس به مشتری j

M عددی بزرگ

متغیرهای تصمیم:

At_i^k زمان رسیدن وسیله نقلیه k به گره i

W_i میزان انتظار در گره i

y_i اگر دپوی کاندید i گشوده شود، یک و در غیر

این صورت، صفر

f_{ij} اگر تقاضای مشتری j توسط دپوی i تأمین

شود، یک و در غیر این صورت، صفر

X_{ij}^k اگر وسیله نقلیه k از گره i به طور مستقیم به

گره j برود یک و در غیر این صورت، صفر

با توجه به موارد ذکر شده در بالا، مدل برنامه‌ریزی

عدد صحیح مختلط CLRPHTW به صورت ذیل ارائه

می‌شود:

$$\text{Minimize } \sum_{i \in I} O_i y_i + \sum_{i \in V} \sum_{j \in V} \sum_{k \in K} C_{ij} X_{ij}^k + \sum_{i \in I} \sum_{j \in J} \sum_{k \in K} F_i X_{ij}^k \quad (1)$$

Subject to:

$$\sum_{i \in V} \sum_{k \in K} X_{ij}^k = 1, \quad \forall j \in J \quad (2)$$

$$\sum_{i \in V} \sum_{j \in J} d_j X_{ij}^k \leq Q, \quad \forall k \in K \quad (3)$$

آزمایشات محاسباتی آورده می‌شود و در نهایت در بخش آخر در مورد الگوریتم پیشنهادی و ارائه پیشنهاد برای آینده بحث می‌شود.

۲. تعریف مسئله و بیان مدل ریاضی آن

در مسئله مکان‌یابی-مسیریابی وسیله نقلیه ظرفیت‌دار با محدودیت پنجره‌های زمانی سخت، مجموعه‌ای از مشتری‌ها و مجموعه‌ای از نقاط کاندید احداث دپو برای عرضه محصول، در سطحی جغرافیایی با مختصات مکانی معلوم وجود دارند. تقاضای مشتری‌ها و ظرفیت دپوها از ابتدا مشخص هستند. وسایل حمل و نقل کالا هم‌جنس با ظرفیتی یکسان هستند. هر مشتری برای دریافت کالای مورد تقاضای خود فقط به یک دپو و یک وسیله نقلیه متعلق به آن دپو تخصیص داده می‌شود که در اینجا باید محدودیت‌های ظرفیت دپو و همچنین ظرفیت وسیله نقلیه مد نظر قرار گیرد. هر وسیله نقلیه نیز تور خود را از دپو آغاز و پس از سرویس‌دهی به مشتری‌های تخصیص یافته به خود، دوباره به همان دپو باز می‌شود. هر مشتری یک بازه زمانی از قبل مشخص شده $[e_i, l_i]$ دارد. در اینجا e_i زودترین زمان شروع دریافت سرویس و l_i نیز دیرترین زمان شروع دریافت سرویس از وسیله نقلیه است. در حالت پنجره‌های زمانی سخت مشتری سرویس را فقط در بازه زمانی تعریف‌شده توسط خود دریافت می‌کند و اگر وسیله نقلیه زودتر از شروع بازه به محل مشتری برسد، باید در آنجا صبر کند تا پنجره زمانی گشوده شود. هدف از این موضوع، کمینه‌کردن هزینه‌های مربوطه و تعیین مکان‌های مناسب احداث دپو و تعیین و برنامه‌ریزی تورهای وسایل نقلیه هر دپو است؛ به طوری که تقاضای همه مشتری‌ها پاسخ داده شود.

برای بیان مدل‌سازی برنامه‌ریزی عدد صحیح مختلط^{۱۳} (MILP) مسئله مورد تحقیق، ابتدا نمادها و پارامترهای به کار رفته در مدل ریاضی تشریح می‌شوند، سپس با افزودن ملاحظات مربوط به محدودیت پنجره‌های زمانی به مدل مسئله مکان‌یابی-مسیریابی وسیله نقلیه

$$\sum_{j \in J} d_j Y_{ij} \leq Cap_i y_i, \forall i \in I \quad (۴)$$

$$\sum_{i \in V} \sum_{j \in V} t_{ij} X_{ij}^k \leq T, \forall k \in K \quad (۵)$$

$$\sum_{i \in S} \sum_{j \in S} X_{ij}^k \leq |S| - 1, \forall S \subseteq J, k \in K \quad (۶)$$

$$\sum_{j \in V} X_{ij}^k - \sum_{j \in V} X_{ji}^k = 0, \forall i \in V, k \in K \quad (۷)$$

$$\sum_{i \in I} \sum_{j \in J} X_{ij}^k \leq 1, \forall k \in K \quad (۸)$$

$$\sum_{m \in V} X_{im}^k + \sum_{h \in V} X_{ih}^k \leq 1 + f_{ij}, \forall i \in I, j \in J, k \in K \quad (۹)$$

$$At_i^k + St_i + W_i + t_{ij} - At_j^k \leq (1 - X_{ij}^k)M, \forall i \in V, j \in V, k \in K \quad (۱۰)$$

$$e_j \leq At_j^k + W_j \leq l_j, \forall j \in J, k \in K \quad (۱۱)$$

$$At_j^k + W_j + St_j + t_{ij} - l_i \leq (2 - y_i - X_{ji}^k) \times M, \forall i \in I, j \in J, k \in K \quad (۱۲)$$

$$At_i^k = 0, \forall i \in I, k \in K \quad (۱۳)$$

$$W_i = 0, \forall i \in I \quad (۱۴)$$

$$At_i \geq 0, \forall i \in V \quad (۱۵)$$

$$W_i \geq 0, \forall i \in V \quad (۱۶)$$

$$f_{ij} \in \{0,1\}, \forall i \in I, j \in J \quad (۱۷)$$

$$y_i \in \{0,1\}, \forall i \in I \quad (۱۸)$$

$$X_{ij}^k \in \{0,1\}, \forall i \in V, j \in V, k \in K \quad (۱۹)$$

(۱۶) بیانگر مثبت بودن متغیرهای تصمیم زمان رسیدن و زمان انتظار است. سرانجام محدودیت‌های (۱۷)، (۱۸) و (۱۹) بیانگر دودویی بودن متغیرهای تصمیم دیگر است. پس از تعریف کامل مسئله و مدل‌سازی ریاضی آن، در ادامه به ارائه روشی فراابتکاری برای حل مسئله می‌پردازیم.

۳. روش حل پیشنهادی

رویکرد پیشنهادی برای حل CLRPHTW روشی بر مبنای الگوریتم فراابتکاری جستجوی همسایگی متغیر است. در الگوریتم پیشنهادی ابتدا با استفاده از الگوریتم تعمیم یافته^{۱۴} PFH جواب اولیه برای شروع الگوریتم پیشنهادی به دست می‌آید. جواب اولیه (S_0) به عنوان جواب فعلی الگوریتم قرار می‌گیرد. حال الگوریتم اصلی که بر مبنای الگوریتم جستجوی همسایگی متغیر است، آغاز می‌شود. در چارچوب الگوریتم جستجوی همسایگی متغیر، برای جستجوی بهتر فضای جواب از الگوریتم شبیه‌سازی تبرید استفاده می‌شود. در ادامه، اجزای روش جدید به تفصیل تشریح می‌شوند و در پایان، ساختار الگوریتم پیشنهادی ارائه می‌شود.

در این مدل، تابع هدف (۱) مجموع هزینه‌های ثابت دپو و هزینه‌های مسیریابی که شامل هزینه ثابت وسیله نقلیه و هزینه مسافت طی شده است را کمینه می‌کند. عبارت محدودیت دوم تضمین می‌کند که هر مشتری فقط و فقط به یک مسیر متعلق است و در تور فقط یک گره پیش‌نیاز دارد.

محدودیت (۳) و (۴) به ترتیب محدودیت‌های ظرفیت مربوط به وسیله نقلیه و دپو هستند. محدودیت حداکثر زمان طی مسافت در یک تور توسط عبارت (۵) نشان داده شده است. محدودیت (۶) مربوط به حذف زیرتورها است. محدودیت (۷) تضمین‌کننده پیوستگی تورها است و محدودیت (۸) تضمین می‌کند که هر وسیله نقلیه از هر دپویی که حرکت خود را آغاز می‌کند بار دیگر باید به همان دپو بازگردد. عبارت (۹) بیان می‌دارد که یک مشتری به یک دپو تخصیص می‌یابد، اگر توری باشد که این دو را به هم مرتبط کند.

عبارت (۱۰) رابطه میان زمان رسیدن یک مشتری و رسیدن به مشتری بعدی‌اش را نشان می‌دهد. محدودیت‌های (۱۱) و (۱۲) تضمین می‌کنند که پنجره زمانی مشتری‌ها و دپوها نقض نشود. محدودیت‌های (۱۳) و (۱۴) شرایط اولیه متغیرهای زمان انتظار و زمان رسیدن را برای دپوها مشخص می‌کند. محدودیت‌های (۱۵) و

```

Input: a set of neighborhood structures  $N_l, l = 1, 2, \dots, l_{max}$ 
 $S = \text{generate initial solution } ()$ ;
Repeat
     $l = 1$ ;
    While ( $l \leq l_{max}$ )
    {
         $S' = \text{Shaking } (S, N_l)$ 
         $S^* = \text{Local search } (S')$ 
        if  $f(S^*) < f(S)$ 
             $S \leftarrow S^*$ 
             $l = 1$  ;
        else
             $l = l + 1$ ;
    }
Until stopping condition are met;
Output: The best solution;

```

شکل ۱: شبه‌کد الگوریتم جستجوی همسایگی متغیر

۲.۳. الگوریتم شبیه‌سازی تبرید

الگوریتم شبیه‌سازی تبرید^{۱۷} (SA) روشی ابتکاری بر مبنای جستجوی محلی است. این الگوریتم قادر است از گیر افتادن در بهینه محلی با قبول جواب‌های بدتر، با احتمالی کم، جلوگیری کند. کیفیت جواب‌های حاصل از الگوریتم شبیه‌سازی تبرید موجب شده تا در حل مسائل بهینه‌سازی ترکیبی پیچیده در حوزه‌های مختلف مسائل دنیای واقعی مورد استفاده قرار گیرد. شبیه‌سازی تبرید توسط کرک پاتریک^{۱۸} و همکاران [۱۵] عمومیت پیدا کرد. مفهوم اصلی شبیه‌سازی تبرید از فرایند تبرید فلزات در صنعت متالورژی گرفته شده است.

فرایند بهینه‌سازی در شبیه‌سازی تبرید، جستجو برای یک جواب (نزدیک به) کمینه سراسری است. الگوریتم از یک جواب تصادفی به عنوان جواب اولیه حرکت خود را آغاز می‌کند و دمای سیستم برابر دمای اولیه قرار می‌گیرد ($T = T_0$). در هر تکرار یک جواب همسایه جواب فعلی به دست می‌آید. مقدار تابع هدف جواب جدید با جواب فعلی مقایسه می‌شود. اگر جواب بهتر بود، جایگزین جواب فعلی می‌شود و اگر جواب بدتر بود با یک احتمال که از تابع بالترنم $exp(-\Delta/KT)$ به دست می‌آید، جایگزین جواب فعلی می‌شود. این مکانیزم منجر به این می‌شود که الگوریتم در بهینه محلی گیر نیافتد. در این رابطه Δ برابر میزان اختلاف مقدار تابع هدف جواب فعلی با جواب جدید است، K ضریب بالترنم است که از قبل تعیین می‌شود و T نیز دمای فعلی است. فرایند جستجوی همسایگی ادامه می‌یابد تا زمانی که تعداد تکرارها به مقداری از پیش تعیین شده برسد. پس از اینکه تعداد

۱.۳. الگوریتم جستجوی همسایگی متغیر

الگوریتم جستجوی همسایگی متغیر^{۱۵} (VNS) که شبه‌کد آن در شکل ۱ آمده است، نخستین بار توسط ملادنویک و هانسن^{۱۶} [۱۳] در سال ۱۹۹۷ مطرح شد. ایده اصلی این الگوریتم تعویض سیستماتیک ساختار همسایگی در طول جستجو برای جلوگیری از افتادن در بهینه محلی است. سادگی در پیاده‌سازی و کیفیت جواب‌های حاصل از الگوریتم VNS، این الگوریتم را به سرعت تبدیل به روشی خوب برای حل مسائل بهینه‌سازی کرد. الگوریتم VNS با تولید جواب اولیه و تعریف ساختارهای همسایگی و استفاده از روشی برای جستجوی همسایگی کار خود را آغاز می‌کند. ساختارهای همسایگی الگوریتم $N_l, l = \{1, 2, \dots, l_{max}\}$ هستند، که ساختار همسایگی l ام است. پس از تعیین ساختارهای همسایگی ممکن ترتیب آن‌ها تعیین می‌شود. دو نکته مهم در اینجا، انتخاب ساختارهای همسایگی مناسب و تعیین ترتیبی مناسب (به عنوان مثال ترتیب بر مبنای بزرگ شدن ساختارهای همسایگی) است. تولید جواب اولیه با کیفیت، تعریف ساختارهای همسایگی و تعیین ترتیب آن‌ها و استفاده از روش مناسب برای جستجوی محلی، عوامل تعیین‌کننده در کیفیت جواب‌های حاصل از الگوریتم است. الگوریتم با استفاده از جواب اولیه تولید شده (S_0) کار خود را آغاز می‌کند و تا زمانی که معیار خاتمه برقرار شود، حلقه اصلی الگوریتم تکرار می‌شود. حلقه اصلی الگوریتم VNS شامل دو فاز اصلی تکان‌دهنده و جستجوی محلی است.

در فاز تکان‌دهنده، الگوریتم از جواب فعلی با استفاده از l امین ساختار همسایگی به جواب همسایه (S') حرکت می‌کند. در فاز جستجوی محلی نیز به روی جواب S' با استفاده از روش‌های جستجوی محلی، جستجو انجام می‌گیرد تا به بهینه محلی S'^* برسد. حال در بخش حرکت یا عدم حرکت اگر بهینه محلی به دست آمده از جواب فعلی S بهتر بود، جایگزین آن می‌شود و جستجو به N_1 برمی‌گردد، در غیر این صورت از ساختار همسایگی بعد (N_{l+1}) برای ادامه جستجو استفاده می‌شود. این جستجو تا زمانی که $l \leq l_{max}$ باشد، ادامه می‌یابد. در شکل (۱)، شبه‌کد الگوریتم جستجوی همسایگی متغیر به نمایش گذارده می‌شود [۱۴].

گام ۳: از لیست UC با توجه به ترتیب نزدیکی فاصله به دپوی گشوده شده مشتری‌ها به دپوی مورد نظر تخصیص می‌یابند تا زمانی که محدودیت ظرفیت دپو نقض نگردد. گام ۴: پس از تخصیص، این مشتری‌ها از لیست UC حذف می‌شوند و دپو نیز از لیست دپوهای کاندید حذف می‌شود.

گام ۵: اگر هنوز مشتری‌ای در لیست مشتری‌های تخصیص نداده شده بود، به گام ۲ باز می‌گردیم، در غیر این صورت به گام آخر می‌رویم.

گام ۶: در این گام برای هر دپوی احداث شده با توجه به مشتری‌های تخصیصی به آن از الگوریتم PFIH برای حل مسئله مسیریابی وسیله نقلیه با پنجره‌های زمانی استفاده می‌شود.

در ۵ گام آغازین روش ذکر شده، از روش تولید جواب اولیه برای مسئله مکان‌یابی-مسیریابی وسیله نقلیه ارائه شده توسط یو و همکاران [۸] استفاده می‌شود. جواب تولیدشده از الگوریتم فوق به عنوان جواب اولیه برای شروع الگوریتم پیشنهادی به امید رسیدن به جواب‌هایی با کیفیت در زمانی معقول مورد استفاده قرار می‌گیرد.

۴.۳. نمایش جواب و محاسبه تابع هزینه

برای نوشتن برنامه کامپیوتری الگوریتم، ابتدا به نحوه ساختار نمایش جواب‌ها برای مسئله نیاز است. ساختار نمایش جواب، مسئله را به صورتی قابل فهم برای کامپیوتر تبدیل می‌کند. این ساختار باید به گونه‌ای باشد که همه اجزای یک جواب به خوبی قابل مشاهده و بررسی باشد. طراحی ساختار مناسب نقش زیادی در عملکرد یک الگوریتم دارد. ساختار نمایش جواب مورد استفاده برای CLRPTW همانند ساختار معمول LRP است. اگر تعداد n مشتری، m مکان کاندید احداث دپو و k وسیله نقلیه داشته باشیم، آنگاه نمایش جواب شامل آرایه‌ای با ترتیبی از n مشتری که با اعداد $1, 2, \dots, n$ مشخص می‌شود، و m دپو که با اعداد $n+1, n+2, \dots, n+m$ نمایش داده می‌شود و $k-2$ درایه صفر است. در شکل (۳) مسئله کوچکی از CLRP با ۱۰ مشتری و ۳ مکان احداث کاندید دپو را نشان می‌دهد.

تکرارها به اندازه مقداری معین و از پیش تعیین شده گشت، دمای سیستم کاهش می‌یابد. این فرایند تا زمانی ادامه پیدا می‌کند که الگوریتم به معیار خاتمه برسد. شبه کد این الگوریتم در شکل ۲ نمایش داده شده است.

۳.۳. تولید جواب اولیه

برای شروع کار الگوریتم نیاز به جواب اولیه مناسب و با کیفیت است، به همین دلیل برای تولید جواب اولیه برای الگوریتم پیشنهادی از ساختار تعمیم‌یافته الگوریتم PFIH برای VRPTW در تحقیق سولومون^{۱۹} [۱۶] استفاده می‌شود. الگوریتم PFIH از الگوریتم‌های ابتکاری و متداول کسب جواب اولیه مناسب و شدنی برای حل مسئله مسیریابی وسیله نقلیه با محدودیت پنجره‌های زمانی است. این الگوریتم با توجه به محدودیت‌های پنجره زمانی و محدودیت ظرفیت وسیله نقلیه جوابی شدنی و مناسب ارائه می‌کند. گام‌های تولید جواب اولیه برای مسئله مورد تحقیق به شرح ذیل است:

```

Initialize parameters;
S=generate initial solution ();
T = T0;
While (T < Tfinal)
{
    Until (N ≤ IIter)
    {
        Generate solution S' in the neighborhood of S
        if f(S') < f(S)
            S ← S'
        else
            Δ = f(S') - f(S);
            r = random();
            if (r < exp(-Δ/K * T))
                S ← S'
    }
    T = α × T;
}
Return the best solution found;

```

شکل ۲: شبه کد الگوریتم شبیه سازی تبرید

گام ۱: ابتدا همه مشتری‌ها در لیست مشتری‌های تخصیص داده نشده (UC) قرار می‌گیرند. گام ۲: هر مشتری داخل لیست UC به نزدیک‌ترین دپو تخصیص می‌یابد. حال دپویی که بیشترین مشتری‌های نزدیک به خود را دارد، به عنوان دپوی کاندید احداث گشوده می‌شود.

۱۱	3	5	7	0	2	1	12	13	4	8	6	0	10	9	0	0
----	---	---	---	---	---	---	----	----	---	---	---	---	----	---	---	---

شکل ۳: نمایش جواب استفاده شده برای الگوریتم پیشنهادی

از عنصر j در آرایه نمایش جواب قرار می‌گیرد. در عملگر حرکتی Swap پس از انتخاب دو عنصر جای آن‌ها در نمایش جواب با یکدیگر عوض می‌شود. عملگر حرکتی Or-opt یک تور را برداشت سه مشتری متوالی و قرار دادن این زنجیره در مکانی دیگر از تور بهبود می‌دهد. عملگر حرکتی $opt^* - 2$ دو تور از نمایش جواب به طور تصادفی انتخاب می‌کند، سپس هر تور را از مکانی به طور تصادفی در طول آن به دو قسمت تقسیم می‌کند. حال جای مشتری‌های قسمت‌های اول و دوم از تور اول را به ترتیب با قسمت‌های دوم و اول از تور دوم عوض در نمایش جواب فعلی جابه‌جا می‌کند.

۶.۳. الگوریتم پیشنهادی برای حل CLRPHTW

پس از تشریح اجزای اصلی، در این بخش به معرفی کامل الگوریتم پیشنهاد شده در این تحقیق برای حل مسئله مکان‌یابی-مسیریابی وسیله نقلیه با محدودیت پنجره‌های زمانی سخت می‌پردازیم. در ابتدا الگوریتم کار خود را با به دست آوردن جواب اولیه S_0 آغاز می‌کند. جواب اولیه به عنوان جواب فعلی سیستم قرار می‌گیرد ($S = S_0$). برای حل مسئله از چهار ساختار همسایگی استفاده می‌شود. شروع الگوریتم با ساختار همسایگی اول، یعنی $l = 1$ ، است. ابتدا در فاز تکان‌دهنده (Shaking) بر اساس ساختار همسایگی l ام الگوریتم از جواب فعلی S به جواب همسایه S' حرکت می‌کند. حال در فاز جستجوی همسایگی به روی جواب همسایگی S' جستجوی همسایگی صورت می‌گیرد تا بهینه محلی S'^* به دست آید. جستجوی همسایگی در الگوریتم پیشنهادی با استفاده از روش شبیه‌سازی تبرید با استفاده از ساختار همسایگی l ام به عنوان عملگر حرکتی برای حرکت به نقاط همسایه صورت می‌گیرد. در اینجا طول زنجیره مارکوف برای جستجوی همسایگی با یک ساختار همسایگی مشخص برابر تعداد N تکرار متوالی بدون بهبود در تابع هدف در نظر گرفته می‌شود. در بخش حرکت یا عدم حرکت، اگر مقدار تابع هدف بهینه محلی S'^* بهتر از S باشد، $S = S'^*$ و $l = 1$ می‌شود، در غیر این صورت

در این شکل دو دپوی ۱۱ و ۱۳ گشوده شده‌اند. در این نمایش جواب زمانی یک دپو باز است که بین این دپو تا دپوی دیگر در نمایش جواب حداقل یک مشتری باشد. همان طور که از این شکل بر می‌آید، مشتری‌های ۳، ۵، ۷، ۲ و ۱ به دپوی ۱۱ تخصیص داده شده‌اند و سایر مشتری‌ها از دپوی شماره ۱۳ سرویس دریافت می‌کنند. همچنین در این نمایش جواب هر دو دپو دارای دو تور وسیله نقلیه هستند. به طور مثال در تور اول از دپوی ۱۱، حرکت از دپو ۱۱ آغاز می‌شود و پس از سرویس دادن به مشتری‌های ۳، ۵ و ۷، به ترتیب، دوباره به دپوی ۱۱ باز می‌گردد.

نکته مهم، نقض محدودیت‌های ظرفیت دپو، ظرفیت وسیله نقلیه و پنجره‌های زمانی است که ممکن است در این نمایش جواب رخ دهد. بدین منظور، در این مقاله از استراتژی جریمه‌گذاری برای برخورد با جواب‌های نشدنی استفاده می‌شود. در این استراتژی میزان جریمه‌ای متناسب با میزان نقض محدودیت‌ها به تابع هدف افزوده می‌شود. در این مقاله پارامترهای $P_vehicle$ ، P_depot و P_tw به ترتیب برابر میزان جریمه به ازای نقض هر واحد محدودیت ظرفیت وسیله نقلیه، ظرفیت دپو و پنجره‌های زمانی تعریف می‌شوند.

۵.۳. ساختارهای همسایگی

در الگوریتم پیشنهادی از چهار ساختار همسایگی استفاده می‌شود. ساختارهای استفاده شده در این تحقیق از ساختارهای همسایگی متداول برای حل مسئله مسیریابی وسیله نقلیه با پنجره‌های زمانی هستند. این چهار ساختار همسایگی شامل عملگر حرکتی relocation، عملگر حرکتی Swap، عملگر حرکتی Or-opt و عملگر حرکتی $opt^* - 2$ هستند. ترتیب ساختارهای همسایگی استفاده شده در الگوریتم پیشنهادی به ترتیب ذکر شده است.

در عملگر حرکتی relocation، ابتدا دو عنصر i و j در طول آرایه نمایش جواب به طور تصادفی انتخاب می‌شوند، سپس عنصر i از جای خود برداشته و در مکان پس

```

Input: a set of neighborhood structures  $N_l, l = 1, 2, \dots, l_{max}$ 
 $S = \text{generate initial solution} ();$ 
 $T = T_0;$ 
While ( $T < T_{final}$ )
{
     $l = 1;$ 
    While ( $l \leq l_{max}$ )
    {
         $S' = \text{Shaking} (S, N_l)$ 
         $S^* = \text{Local search} (S')$ 
        if  $f(S^*) < f(S)$ 
        {
             $S \leftarrow S^*, l = 1;$ 
        }
        else
             $l = l + 1;$ 
        }
    }
     $T = \alpha \times T;$ 
    Update the best obtained solution
}
Output: The best solution;

```

شکل ۴: شبه کد الگوریتم پیشنهادی برای حل CLRPHTW

۱.۴. ارزیابی عملکرد الگوریتم پیشنهادی در مثال‌های کوچک

برای ارزیابی صحت مدل ریاضی ارائه شده و همچنین کارایی الگوریتم پیشنهادی در مسائل کوچک ابتدا باید تعدادی مثال تولید شوند. در این مقاله تعداد ۱۰ مثال کوچک، با حداکثر ۱۵ مشتری و ۵ مکان کاندید دپو، تولید می‌شوند. برای تولید مثال‌ها بدین صورت عمل می‌شود: ۲۰ نقطه به طور تصادفی در سطحی به اندازه 50×50 تولید می‌شود. ۱۵ نقطه اول مشتری‌ها و ۵ نقطه دیگر مکان‌های کاندید احداث دپو هستند. برای تولید تقاضای هر مشتری عددی به طور تصادفی و با توزیع یکنواخت در بازه $[10, 20]$ تولید می‌شود. برای ۱۵ مشتری در ابتدا عددی در فاصله $[0, 20]$ به عنوان زودترین زمان سرویس و سپس عددی به طور تصادفی در بازه $[30, 50]$ به عنوان دیرترین زمان انجام سرویس تولید می‌شوند. پنجره زمانی مربوط به هر دپو به صورت $[0, 100]$ در نظر گرفته می‌شود. وسایل نقلیه هم‌جنس با ظرفیت ۸۰ واحد کالا و به تعداد ۱۰ عدد در نظر گرفته می‌شود. ظرفیت هر مکان کاندید احداث دپو ۲۰۰ واحد است. هزینه ثابت استفاده از هر دپو ۲۰۰ واحد و هزینه ثابت هر وسیله نقلیه ۰ واحد قرار می‌گیرد. حداکثر زمان طی مسیر توسط یک وسیله نقلیه معادل ۲۰۰ واحد زمانی در نظر گرفته می‌شود. هزینه طی مسیر به ازای هر واحد زمانی معادل ۱ واحد پولی در نظر گرفته می‌شود. جدول

از ساختار همسایگی بعدی استفاده می‌شود ($l = l + 1$)
 ۱. حلقه الگوریتم جستجوی همسایگی متغیر در یک دما تا زمانی که $l \leq l_{max}$ باشد ادامه می‌یابد، در اینجا $l_{max} = 4$ است، و پس از آن دما با استفاده از قانون کاهش هندسی ($T = \alpha \times T$) به روزرسانی می‌شود. الگوریتم پیشنهادی تا رسیدن به دمای پایانی ادامه می‌یابد. فلوجارت الگوریتم پیشنهادی در شکل (۴) به - نمایش گذارده شده است.

۴. آزمایشات محاسباتی

الگوریتم پیشنهادی به زبان C++ برنامه‌نویسی شده و در سیستم رایانه همراه با مشخصات پردازش‌گر Core i5 2.53 GHz و حافظه جانبی 4 GB اجرا شده است. مقداردهی مناسب پارامترهای الگوریتم پیشنهادی نقش بسزایی در کیفیت جواب‌های حاصله دارد، بدین منظور برای هر پارامتر تعدادی مقادیر پیشنهادی در نظر گرفته می‌شود سپس مقدار هر پارامتر به ترتیب جدول (۱)، با ثابت نگاه‌داشتن بقیه پارامترها در کمترین مقدار خود، بر اساس میزان تابع هدف و زمان رسیدن به جواب تنظیم می‌شود. پس از انجام آزمایش‌های بالا به روی یکی از مثال‌های تولیدشده، مقادیر پارامترهای الگوریتم پیشنهادی به صورت ذیل تنظیم شد: همه مقادیر جریمه‌ها (هر واحد نقض محدودیت ظرفیت دپو، ظرفیت وسیله نقلیه و پنجره زمانی) برابر ۲۰۰ واحد در نظر گرفته شد. دمای اولیه (T_0) برابر ۵۰ درجه و دمای پایانی (T_{final}) برابر ۰,۰۰۱ در نظر گرفته شد. ضریب کاهش دما (α) برابر ۰,۹۵ و ضریب بالترمن (K) برابر ۰,۲ در نظر گرفته شد. تعداد تکرار متوالی بدون بهبود در جستجوهای محلی (N) برابر $L(L - 1)$ در نظر گرفته شد که در اینجا، L برابر طول آرایه نمایش جواب است.

از آنجا که هیچ‌گونه مثال معیاری برای مسئله مورد تحقیق وجود ندارد، بنابراین ابتدا به تولید تعدادی مثال نیاز است. در تحقیق حاضر ارزیابی عملکرد الگوریتم پیشنهادی در دو بخش انجام می‌گیرد: در بخش اول به ارزیابی درستی مدل و عملکرد الگوریتم در دسته‌ای از مثال‌های با اندازه کوچک اختصاص می‌یابد و در بخش دوم عملکرد الگوریتم در مثال‌های با اندازه متوسط و بزرگ مورد ارزیابی قرار می‌گیرد.

جواب و $time$ نشان‌دهنده محدودیت زمان در رسیدن به جواب است. در دو ستون بعدی جواب و زمان رسیدن به آن توسط الگوریتم پیشنهادی در یک تک اجرا آورده شده است. ستون آخر نیز درصد انحراف جواب حاصل از الگوریتم نسبت به جواب بهینه را نشان می‌دهد.

همان‌گونه که مشاهده می‌شود، در ۷ مثال اول الگوریتم به جواب بهینه رسیده است. در مثال‌های ۸ و ۹ الگوریتم پیشنهادی به جواب‌های نرم‌افزار GAMS رسیده است، ولی جواب‌ها در GAMS با استفاده از محدودیت زمان ۵ ساعت به دست آمده است؛ با این وجود تضمینی بر بهینه بودن جواب‌ها نیست. در مثال آخر در محدودیت زمان تعریف شده نرم‌افزار GAMS حتی به جوابی برابر با جواب الگوریتم پیشنهاد شده نیز نرسیده است. با توجه به جدول (۳)، می‌توان نتیجه گرفت که با بزرگ شدن اندازه مسئله زمان رسیدن به جواب بهینه به شدت افزایش می‌یابد، حال آنکه الگوریتم پیشنهادی توانایی دارد در زمانی بسیار کم به جواب خوبی برسد. مقایسه‌های بالا نشان از عملکرد مناسب الگوریتم پیشنهادی در مثال‌های کوچک دارد و جواب‌های حاصل نشان‌دهنده صحت مدل ریاضی ارائه شده است. در ادامه عملکرد الگوریتم برای مثال‌های متوسط و بزرگ مورد بررسی قرار می‌گیرد.

(۲)، مختصات مکانی، پنجره‌های زمانی و تقاضا مثال‌های تولیدشده بالا را نمایش می‌دهد.

دسته مثال تولیدشده در این بخش در تعداد مشتری و تعداد مکان احداث دپو متفاوت هستند. به طور مثال در مسئله با n مشتری و m دپو کاندید (در اینجا n حداکثر برابر ۱۵ و m نیز حداکثر برابر ۵ است)، مشخصات n مشتری اول، از لیست مشتری‌های تولیدشده و m دپو اول از لیست دپوهای کاندید مدنظر قرار می‌گیرند. بقیه پارامترها برای همه مثال‌ها برابر در نظر گرفته می‌شوند. مدل ریاضی هر یک از ده مثال توسط نرم‌افزار بهینه‌سازی GAMS 23.6 با حل‌کننده CPLEX در سیستم بارانه همراه به مشخصات پردازشگر Core i5 2.53 GHz و حافظه داخلی 4 GB با محدودیت زمانی ۱۸۰۰۰ ثانیه (۵ ساعت) حل شدند.

در جدول (۳)، مقایسه‌ای میان جواب‌های حاصل از الگوریتم پیشنهادی در یک اجرا و جواب‌های حاصل از حل مدل ریاضی با استفاده از نرم‌افزار نمایش داده شده است. در جدول فوق مشخصات ۱۰ مثال کوچک تولید شده از نظر تعداد مشتری و تعداد مکان کاندید احداث دپو در سه ستون اول به نمایش گذارده شده است. در سه ستون بعد جواب، زمان رسیدن به جواب و وضعیت جواب به دست آمده توسط نرم‌افزار GAMS مشخص شده است. در ستون وضعیت، opt نشان‌دهنده بهینه‌بودن

جدول ۱: مقادیر پیشنهادی برای تنظیم پارامترهای الگوریتم

شماره	پارامتر	مقادیر پیشنهادی
1	جریمه	50,100, 200, 500
2	N	$0.5L(L-1), L(L-1), 2L(L-1), 3L(L-1)$
3	T_0	30, 50, 70, 100
4	α	0.9,0.93,0.95,0.97,0.99
5	T_{final}	0.001, 0.01,0.1,1
6	K	0.1,0.2,0.5,1

جدول ۲: مشخصات مثال‌های کوچک تولیدشده

نقاط	مختصات مکانی		تقاضا	پنجره زمانی		نقاط	مختصات مکانی		تقاضا	پنجره زمانی	
	X	Y		E	L		X	Y		E	L
1	27	4	18	1	37	11	22	33	11	18	49
2	16	30	19	8	48	12	26	20	20	12	50
3	5	24	11	5	37	13	23	41	20	11	42
4	31	35	20	16	32	14	44	36	15	3	31
5	39	35	16	9	46	15	26	49	18	17	34
6	21	32	11	19	38	16	36	9	0	0	100
7	4	1	13	3	35	17	24	37	0	0	100
8	13	3	16	5	38	18	30	13	0	0	100
9	7	16	20	3	32	19	18	21	0	0	100
10	14	27	20	2	32	20	31	46	0	0	100

جدول ۳: مقایسه نتایج نرم افزار GAMS و الگوریتم پیشنهادی در مثال‌های کوچک

شماره مسئله	تعداد مشتری	تعداد دیو	GAMS			الگوریتم پیشنهادی		GAP (%)
			جواب	زمان (ثانیه)	وضعیت	جواب	زمان (ثانیه)	
1	5	2	342.867	1.016	opt	342.867	0	0
2	7	2	457.845	16.782	opt	457.845	0	0
3	10	2	520.602	1801.34	opt	520.602	1	0
4	10	3	425.199	824.43	opt	425.199	1	0
5	12	2	524.147	12400	opt	524.147	1	0
6	12	3	428.981	14812.4	opt	428.981	2	0
7	12	5	۳۸۷,۱۲۷	۱۷۵۶۳	opt	387.127	2	0
8	15	2	631.029	18000	time	631.029	2	-----
9	15	3	612.264	18000	time	612.264	2	-----
10	15	5	628.015	18000	time	605.089	2	-----

جدول ۴: جزئیات نتایج آزمایشات عملکرد الگوریتم در مثال‌های با ۲۵ مشتری و ۵ دیو کاندید

کد مثال	VNS		VNS+SA		GAP (%)	کد مثال	VNS		VNS+SA		GAP (%)
	مقدار تابع هدف	زمان	مقدار تابع هدف	زمان			مقدار تابع هدف	زمان	مقدار تابع هدف	زمان	
MC101	1117.73	6	1077.92	7	-3.69	MR201	1160.73	6	1112.35	6	-4.35
MC102	1100.34	6	1076.85	6	-2.18	MR202	1038.93	6	1019.93	6	-1.86
MC103	1092.24	5	1069.68	7	-2.11	MR203	980.86	5	973.05	6	-0.8
MC201	1159.75	6	1136.11	7	-2.08	MRC101	1351.62	6	1306.92	6	-3.42
MC202	1152.89	5	1149.33	7	-0.31	MRC102	1211.55	6	1116.41	7	-8.52
MC203	1148.35	5	1140.49	6	-0.69	MRC103	1100.46	5	1100.46	7	0
MR101	1727.12	6	1727.12	6	0	MRC201	1124.05	6	1124.05	7	0
MR102	1573.8	4	1573.79	6	0	MRC202	1101.85	6	1101.85	6	0
MR103	1317.24	6	1317.24	6	0	MRC203	1091.17	6	1091.17	6	0

جدول ۵: جزئیات نتایج آزمایشات عملکرد الگوریتم در مثال‌های با ۵۰ مشتری و ۱۰ دیو کاندید

کد مثال	VNS		VNS+SA		GAP (%)	کد مثال	VNS		VNS+SA		GAP (%)
	مقدار تابع هدف	زمان	مقدار تابع هدف	زمان			مقدار تابع هدف	زمان	مقدار تابع هدف	زمان	
MC101	1364.28	32	1361.5	39	-0.2	MR201	1713.34	36	1612.33	40	-6.26
MC102	1365.73	33	1360.2	40	-0.41	MR202	1507.21	36	1460.5	43	-3.2
MC103	1357.67	38	1356.75	50	-0.07	MR203	1381.57	45	1320.8	45	-4.6
MC201	1288.09	53	1282.81	52	-0.41	MRC101	2221.74	28	2132.6	58	-4.18
MC202	1293.48	51	1259.63	50	-2.69	MRC102	1951.77	42	1795.27	63	-8.72
MC203	1263.36	32	1263.92	48	0.04	MRC103	1718.82	32	1651.16	60	-4.1
MR101	2584.13	35	2519.69	65	-2.56	MRC201	1581.32	35	1552.68	34	-1.84
MR102	2304.86	32	2216.56	50	-3.98	MRC202	1517.77	34	1517.03	33	-0.05
MR103	2043.92	34	1973.19	35	-3.58	MRC203	1441.66	35	1417.38	36	-1.71

یکدیگر مقایسه می‌شوند. الگوریتم اول که VNS نامیده می‌شود، الگوریتم جستجوی همسایگی متغیر بدون استفاده از ساختار الگوریتم شبیه‌سازی تبرید در ساختار جستجوی همسایگی است. الگوریتم دوم الگوریتم VNS+SA نامگذاری می‌شود که از ساختار الگوریتم جستجوی همسایگی متغیر پیشنهادی با استفاده از الگوریتم شبیه‌سازی تبرید در ساختار جستجوی محلی استفاده می‌کند.

با توجه به دانش ما هیچ مثال معیاری برای مسئله

۲.۴. ارزیابی عملکرد الگوریتم پیشنهادی در مثال‌های متوسط و بزرگ

حل دقیق و به دست آوردن جواب‌های بهینه جز برای دسته‌ای از مسائل کوچک برای سایر مسائل مقدور نیست. به علاوه، با توجه به اینکه در پیشینه موضوع، روشی برای حل مسئله مکان‌یابی-مسیریابی وسیله نقلیه ظرفیت‌دار با محدودیت پنجره‌های زمانی سخت وجود ندارد، بنابراین مقایسه با روش‌های قبل نیز قابل انجام نیست. به همین دلیل، در تحقیق حاضر دو الگوریتم پیشنهاد شده با

جدول ۶: جزئیات نتایج آزمایشات عملکرد الگوریتم در مثال‌های با ۱۰۰ مشتری و ۱۰ دپو کاندید

کد مثال	VNS		VNS+SA		GAP(%)	P-value	نتیجه آزمون
	مقدار تابع هدف	زمان	مقدار تابع هدف	زمان			
MC۱۰۱	۲۸۸۹,۶۵	۱۶۲	۲۷۰۹,۲۵	۱۴۸	-6.66	0.006	قبول فرض مقابل
MC۱۰۲	۲۷۲۷,۹۱	۱۳۱	۲۶۶۴,۹۸	۱۵۰	-2.36	0.004	قبول فرض مقابل
MC۱۰۳	۲۷۳۲,۶۳	۱۰۴	۲۶۳۵,۳۲	۱۵۵	-3.69	0.012	قبول فرض مقابل
MC۲۰۱	۲۰۴۵,۲	۱۳۹	۱۹۹۹,۶۴	۱۵۵	-2.28	0.001	قبول فرض مقابل
MC۲۰۲	۲۰۴۱,۴۴	۱۳۰	۱۹۲۷,۰۳	۱۴۴	-5.94	0.014	قبول فرض مقابل
MC۲۰۳	۲۰۵۳,۴	۱۳۷	۱۹۵۸,۱۴	۱۳۹	-4.86	0.000	قبول فرض مقابل
MR۱۰۱	۴۲۵۷,۲۵	۱۱۲	۴۰۸۳,۸۴	۱۷۶	-4.25	0.005	قبول فرض مقابل
MR۱۰۲	۳۹۰۵,۸۱	۱۵۳	۳۸۲۹,۷۳	۱۹۷	-1.99	0.072	رد فرض صفر
MR۱۰۳	۳۳۹۱,۶۴	۱۴۲	۳۲۹۷,۵۳	۱۶۰	-2.85	0.042	قبول فرض مقابل
MR۲۰۱	۲۶۱۶,۲۱	۱۴۶	۲۴۵۸,۹۶	۱۵۰	-6.39	0.000	قبول فرض مقابل
MR۲۰۲	۲۴۳۰,۳۱	۱۳۱	۲۳۸۴,۶۳	۱۵۵	-1.92	0.132	رد فرض صفر
MR۲۰۳	۲۲۱۰,۸۳	۱۲۷	۲۰۵۱,۵۹	۱۵۴	-7.76	0.000	قبول فرض مقابل
MRC۱۰۱	۴۰۳۳,۴۷	۱۲۴	۳۹۰۷,۲۷	۱۵۷	-3.23	0.031	قبول فرض مقابل
MRC۱۰۲	۳۸۰۰,۹	۱۲۶	۳۵۱۹,۶۱	۱۴۹	-7.99	0.013	قبول فرض مقابل
MRC۱۰۳	۳۳۱۳,۴	۱۲۴	۳۲۹۰,۷۲	۱۶۸	-0.69	0.124	رد فرض صفر
MRC۲۰۱	۲۹۴۶,۹۷	۹۵	۲۹۱۸,۴۹	۱۳۳	-0.98	0.072	رد فرض صفر
MRC۲۰۲	۲۷۶۴,۳۶	۱۲۳	۲۷۸۴,۵۲	۱۴۷	0.72	0.365	رد فرض صفر
MRC۲۰۳	۲۴۳۵,۳۱	۱۰۲	۲۴۵۴,۵	۱۳۸	0.78	0.616	رد فرض صفر

جدول ۷: جواب‌های حاصل از الگوریتم پیشنهادی در مقایسه با سایر الگوریتم‌های موجود در پیشینه موضوع

کد مثال	BKS	GRASP	MAPM	LRGTS	GRASP+ELS	SALRP	روش پیشنهادی	CPU (sec)	GAP (%)
20-5-1a	54,793	55,021	54,793	55,131	54,793	54,793	54,793	25	0.00
20-5-1b	39,104	39,104	39,104	39,104	39,104	39,104	39,104	22	0.00
20-5-2a	48,908	48,908	48,908	48,908	48,908	48,908	48,908	33	0.00
20-5-2b	37,542	37,542	37,542	37,542	37,542	37,542	37,542	38	0.00
50-5-1a	90,111	90,632	90,160	90,160	90,111	90,111	90,111	39	0.00
50-5-1b	63,242	64,741	63,242	63,256	63,242	63,242	63,242	40	0.00
50-5-2a	88,298	88,786	88,298	88,715	88,643	88,298	88,298	46	0.00
50-5-2b	67,308	68,042	67,893	67,698	67,308	67,340	67,308	37	0.00
50-5-2bis	84,055	84,055	84,055	84,181	84,055	84,055	84,055	33	0.00
50-5-2Bbis	51,822	52,059	51,822	51,992	51,822	51,822	51,822	48	0.00
50-5-3a	86,203	87,380	86,203	86,203	86,203	86,456	86,203	54	0.00
50-5-3b	61,830	61,890	61,830	61,830	61,830	62,700	61,830	31	0.00
100-5-1a	276,960	279,437	281,944	277,935	276,960	277,035	275,919	245	-0.38
100-5-1b	214,885	216,159	216,656	214,885	215,854	216,002	214,646	289	-0.11
100-5-2a	194,124	199,520	195,568	196,548	194,267	194,124	194,677	381	0.28
100-5-2b	157,150	159,550	157,325	157,792	157,375	157,150	157,265	245	0.07
100-5-3a	200,242	203,999	201,749	201,952	200,345	200,242	200,247	218	0.00
100-5-3b	152,467	154,596	153,322	154,709	152,528	152,467	152,503	291	0.02
100-10-1a	290,429	323,171	316,575	291,887	301,418	291,043	290,919	348	0.17
100-10-1b	234,210	271,477	270,251	235,532	269,594	234,210	233,503	281	-0.30
100-10-2a	244,265	254,087	245,123	246,708	243,778	245,813	244,253	312	0.00
100-10-2b	203,988	206,555	205,052	204,435	203,988	205,312	203,988	237	0.00
100-10-3a	250,882	270,826	253,669	258,656	253,511	250,882	251,120	388	0.09
100-10-3b	204,597	216,173	204,815	205,883	205,087	205,009	205,578	295	0.48

معیار سولومون برای مسئله مسیریابی وسیله نقلیه با محدودیت پنجره‌های زمانی در لینک <http://web.cba.neu.edu/~msolomon> قابل دسترسی است. سولومون [۱۶] تعدادی مثال معیار نیز برای مسئله VRPTW ارائه کردند. پایگاه داده ایشان معروف‌ترین و پراستفاده‌ترین پایگاه برای سنجش کیفیت

مکان‌یابی-مسیریابی وسیله نقلیه ظرفیت‌دار با محدودیت پنجره‌های زمانی سخت وجود ندارد. بنابراین در این مقاله، تعدادی از مثال‌های معیار سولومون [۱۶]، برای مسئله مسیریابی وسیله نقلیه با محدودیت پنجره‌های زمانی با انجام تغییراتی به عنوان مثال‌های معیار مورد استفاده قرار می‌گیرند. مثال‌های

بهتری نسبت به الگوریتم VNS به دست آورده است. در همه مثال‌های با ۵۰ مشتری و ۱۰ مکان کاندید احداث دپو جز مثال MC203، جواب‌های الگوریتم VNS+SA از جواب‌های الگوریتم VNS بهتر شده است.

در جهت بالا بردن اعتبار الگوریتم پیشنهادی، در مثال‌های با ۱۰۰ مشتری و ۱۰ مکان کاندید دپو، علاوه بر مقایسه جواب‌های الگوریتم‌های پیشنهادی از نظر بهترین جواب کسب‌شده، از آنالیز آماری آزمون تی زوجی استفاده می‌کنیم. از آنجا که برای هر مثال و برای هر الگوریتم ۱۰ اجرا (۱۰ جواب) در اختیار است، بنابراین اندازه نمونه برای هر مثال و هر الگوریتم نیز برابر ۱۰ است. در اینجا فرض بهبود عملکرد الگوریتم HVNS را در مقابل الگوریتم الگوریتم VNS پیشنهادی آزمون می‌کنیم. بدین منظور آزمون فرض ذیل را در نظر می‌گیریم:

فرض صفر: میانگین جواب‌های الگوریتم پیشنهادی VNS+SA با میانگین جواب‌های حاصل از الگوریتم پیشنهادی VNS برابر است.

فرض مقابل: میانگین جواب‌های الگوریتم پیشنهادی VNS+SA کمتر از میانگین جواب‌های حاصل از الگوریتم پیشنهادی VNS است.

از آنجا که آنالیز آماری در سطح معنی‌داری ۰,۰۵ آزمون می‌شود، چنانچه مقدار P-Value کمتر از مقدار $\alpha = 0.05$ باشد، دلیلی برای قبول فرض صفر وجود ندارد. در جدول (۶)، نتایج حاصل از بررسی عملکرد دو الگوریتم پیشنهادی به همراه نتایج حاصل از تحلیل آماری آزمون تی زوجی به نمایش گذارده می‌شود. در این جدول در ستون Sig. (2-tailed) مقدار P-value، و ستون آخر نیز نتیجه آزمون تی زوجی را گزارش می‌کند.

همان گونه که در جدول (۶) مشاهده می‌شود، جواب‌های الگوریتم VNS+SA جز در ۶ مسئله از ۱۸ مسئله با اطمینان ۰,۹۵ بهتر از جواب‌های حاصل از الگوریتم VNS است؛ در حالی که با مقایسه درصد اختلاف میان بهترین جواب کسب شده توسط الگوریتم VNS+SA در مقایسه با VNS این برتری در همه مثال‌ها جز ۲ مثال مشاهده می‌شود.

۳,۴. ارزیابی عملکرد الگوریتم پیشنهادی در

حل مسئله CLRP

الگوریتم‌های ارائه شده برای VRPTW و سایر مسائل مشابه آن است. در مثال‌های ارائه شده توسط ایشان، تعداد مشتری‌ها برابر ۲۵، ۵۰ و ۱۰۰ مشتری است. ایشان برای تولید مثال‌ها، ۱۰۰ مشتری و یک دپو را مدنظر قرار دادند و با در نظر گرفتن ۲۵ و یا ۵۰ مشتری اول مسائل با اندازه‌های کوچک‌تر را تولید کردند. ایشان مثال‌های خود را در سه دسته اصلی C، R و RC در نظر گرفتند. مکان‌های جغرافیایی در نظر گرفته شده برای دسته مثال‌های C به صورت دسته‌بندی شده، برای دسته R به صورت تصادفی و برای دسته RC به صورت ترکیبی از دو حالت دسته‌بندی و تصادفی در سطح دو بعدی تولید شده‌اند. برای هر دسته مثال، مکان‌های مشتری‌ها و دپو و تقاضای مشتری‌ها یکی در نظر گرفته شده است و فقط در پنجره‌های زمانی با یکدیگر متفاوتند. هر دسته از دو زیردسته ۱ و ۲ تشکیل شده است در زیردسته ۱ مثال‌ها دارای پنجره‌های زمانی پهن هستند. وسایل نقلیه هم‌جنس و با ظرفیت برابرند و تعداد آنها نیز محدود و معین است.

برای اصلاح مثال‌های معیار سولومون، به مثال‌های با ۲۵ مشتری ۴ و به مثال‌های دیگر ۹ دپو اضافه می‌شود. همه دپوها ظرفیت و هزینه‌های مشخص دارند. مثال‌های تولیدشده برای CLRPHTW توسط افزودن کلمه M به اول کد مثال‌های سولومون نام‌گذاری می‌شود.

برای انجام آزمایش‌ها، هر الگوریتم برای هر مثال ۱۰ بار اجرا می‌شود و بهترین جواب کسب‌شده توسط آن در طی این ۱۰ اجرا به همراه زمان آن به نمایش گذارده می‌شود. در این جداول کد مثال به همراه بهترین جواب و زمان کسب شده توسط هر دو الگوریتم آورده شده است. ستون آخر درصد اختلاف میان بهترین جواب کسب شده توسط الگوریتم VNS+SA در مقایسه با VNS است. اگر جواب حاصل از الگوریتم VNS برابر S و جواب حاصل از VNS+SA برابر S' باشد، آنگاه درصد اختلاف میان آن‌ها از رابطه $GAP = \frac{S'-S}{S} \times 100$ ذیل حساب می‌شود.

همان طور که در جدول (۴) مشاهده می‌شود، در ۷ مثال از ۱۸ مثال با ۲۵ مشتری و ۵ مکان کاندید احداث دپو جواب‌های الگوریتم پیشنهادی با جواب‌های حاصل از الگوریتم جستجوی همسایگی متغیر پیشنهادی برابر شده است و در سایر موارد الگوریتم VNS+SA جواب‌های

۵. نتیجه‌گیری

مسئله مکان‌یابی-مسیریابی وسیله نقلیه ظرفیت‌دار یکی از مسائل با اهمیت و پر کاربرد در مدیریت زنجیره تأمین و لجستیک است. در تحقیق حاضر برای نزدیک سازی بیشتر مسئله به دنیای واقعی و در جهت کاربردی‌تر کردن آن، محدودیت پنجره‌های زمانی به مسئله افزوده می‌شود. در این مقاله، پس از تعیین مفروضات، مسئله با همه محدودیت‌ها و فرضیه‌های در نظر گرفته شده مدل‌سازی شد.

سپس با توجه به پیچیدگی زمانی نمایی حل مسئله و ناتوانی روش‌های بهینه‌سازی دقیق در حل مسئله در ابعاد بزرگ، رویکردی فراابتکاری برای حل مسئله ارائه شد. روش مورد استفاده برای حل مسئله رویکردی بر مبنای الگوریتم نزول همسایگی متغیر است. از آنجا که هیچ نمونه مثالی برای ارزیابی عملکرد الگوریتم وجود نداشت تعدادی مثال تولید شد.

پس از آن عملکرد الگوریتم ترکیبی پیشنهاد شده برای مسئله مکان‌یابی-مسیریابی وسیله نقلیه ظرفیت‌دار با پنجره‌های زمانی سخت و درستی مدل ارائه شده برای این مسئله مورد ارزیابی قرار گرفت. در انتها عملکرد الگوریتم برای حل مسئله مکان‌یابی-مسیریابی وسیله نقلیه ظرفیت‌دار بدون در نظر گرفتن پنجره‌های زمانی مورد آزمایش قرار گرفت. نتایج حاصل، کیفیت الگوریتم را در یافتن جوابی مناسب برای مسئله مورد مطالعه در زمانی معقول نشان می‌دهد.

تحقیقات انجام شده در زمینه مسئله مکان‌یابی-مسیریابی با محدودیت پنجره‌های زمانی بسیار اندک است، بنابراین مطالعه بیشتر روی مسئله و افزودن فرضیه‌های دیگر می‌تواند به کاربردی‌تر کردن مسئله کمک کرد. توسعه روش‌های ابتکاری و فراابتکاری دیگر برای بهبود جواب‌های حاصل، مطالعه مسئله با پنجره‌های زمانی نرم، استفاده از الگوریتم پیشنهادی برای حل سایر مسائل بهینه‌سازی ترکیبی و افزودن سایر فرضیه‌های مسئله مسیریابی وسیله نقلیه و ... از جمله پیشنهادات برای ادامه تحقیق حاضر در آینده است.

برای آزمایش عملکرد الگوریتم پیشنهادی از مسائل نمونه Prodhon که امکان دسترسی به آن از طریق لینک <http://prodhon.free.fr/homepage> فراهم است، استفاده می‌شود. در این مثال‌ها هم مسیره‌ها و هم دیوها دارای ظرفیت هستند. الگوریتم برای هر مثال ۱۰ بار اجرا شد و بهترین جواب به دست آمده در جدول (۸) با بعضی از الگوریتم‌های موجود در پیشینه موضوع مقایسه شد. بهترین الگوریتم‌های موجود در پیشینه موضوع برای مسئله CLRP، به ترتیب الگوریتم GRASP [۲]، الگوریتم MAPM [۳]، الگوریتم LRGTS [۵]، الگوریتم GRASP+ELS [۷] و الگوریتم SALRP [۸] می‌باشند.

در جدول (۸)، در ستون اول کد هر مثال آمده است. عدد اول و دوم مشخصه مثال به ترتیب نشان‌دهنده تعداد مشتری و دیو کاندید است. در ستون دوم بهترین جواب به دست آمده تا کنون (BKS) آمده و در ستون‌های بعد نیز مقدار جواب حاصل از روش‌های انتخاب شده در پیشینه موضوع و بهترین جواب کسب شده توسط الگوریتم پیشنهادی آمده است. در انتها مدت زمان اجرا (CPU) بر حسب ثانیه و فاصله از بهترین جواب کسب شده تا به حال بر حسب درصد (GAP) آمده است. همان طور که از جدول (۷) بر می‌آید، الگوریتم پیشنهاد شده در این مقاله از ۲۴ مثال معیار در ۱۵ مثال بهترین جواب کسب شده تا به حال را کسب می‌کند یعنی میزان انحراف از بهترین جواب کسب شده برابر ۰٪ است. همچنین الگوریتم پیشنهادی در ۳ مثال جواب‌هایی بهتر از بهترین جواب کسب شده تا به حال به دست می‌آورد. الگوریتم پیشنهاد شده در ۶ مثال باقی‌مانده نیز دارای میزان انحراف مابین ۰٫۰۲٪-۰٫۴۸٪ است. در مجموع میانگین میزان انحراف از بهترین جواب کسب شده تا به حال برای ۲۴ مثال معیار انتخابی برابر ۰٫۰۱٪ است. نتایج بالا نشان می‌دهد، با توجه به کیفیت جواب‌های حاصل از الگوریتم پیشنهادی و زمان‌های منطقی رسیدن به آن جواب‌ها توسط الگوریتم، عملکرد آن با سایر الگوریتم‌های موجود برای حل مسئله مکان‌یابی-مسیریابی وسیله نقلیه ظرفیت‌دار بدون در نظر گرفتن پنجره‌های زمانی نیز قابل رقابت است.

مراجع

- 1- Nagy, G. and Salhi, S. (2007). "Location-routing: issues, models and methods." *European Journal of Operational Research*, 177: 649–72.
- 2- Prins, C., Prodhon, C. and Wolfler-Calvo, R. (2006). "Solving the capacitated location- routing problem by a GRASP complemented by a learning process and a path relinking." *4OR—A Quarterly Journal of Operations Research*, 4: 221–38.
- 3- Prins, C., Prodhon, C. and Wolfler-Calvo, R. (2006). "A memetic algorithm with population management (MA|PM) for the capacitated location- routing problem." *Lecture notes in computer science. Berlin: Springer*, 3906:183–94.
- 4- Barreto, S. S., Ferreira, C., Paixa, J. and Santos, B. S. (2007). "Using clustering analysis in capacitated location-routing problem." *European Journal of Operational Research*, 179: 968–977.
- 5- Prins, C., Prodhon, C., Ruiz, A., Soriano, P. and Wolfler Calvo, R. (2007). "Solving the capacitated location-routing problem by a cooperative Lagrangean relaxation granular tabu search heuristic." *Transportation Science*, 41:470–483.
- 6- Marinakis, Y. and Marinaki, M. (2008). "A particle swarm optimization algorithm with path relinking for the location routing problem." *Journal of Mathematical modeling and algorithms*, 7: 59–78.
- 7- Duhamel, C., Lacomme, P., Prins, C. and Prodhon, C. (2010). "A GRASP×ELS approach for the capacitated location-routing problem." *Computers & Operations Research*, 37: 1912-1923.
- 8- Yu, V. F., Lin, S-W., Lee, W. and Ting, C.-J. (2010). "A simulated annealing heuristic for the capacitated location-routing problem." *Computers & Industrial Engineering*, 58: 288-299.
- 9- Nguyen, V.-P., Prins, C. and Prodhon, C. (2012). "Solving the two-echelon location routing problem by a GRASP reinforced by a learning process and path relinking." *European Journal of Operational Research*, 216: 113-126.
- 10- Nguyen, V.-P., Prins, C. and Prodhon, C. (2012). "A multi-start iterated local search with tabu list and path relinking for the two-echelon location-routing problem." *Engineering Applications of Artificial Intelligence*, 25: 56-71.
- 11- Jabal-Ameli, M.S. and Ghaffari-Nasab, N. (2010). "Location-routing problem with time windows: Novel mathematical programming formulations." *7th International Industrial Engineering Conference*, Isfahan, Iran.
- 12- Nikbakhsh, E. and Zegordi, S.H. (2010). "A Heuristic Algorithm and a Lower Bound for the Two-Echelon Location-Routing Problem with Soft Time Window Constraints." *Scientia Iranica*, 17: 36-47.
- 13- Mladenović, N. and Hansen, P. (1997). "Variable neighborhood search." *Computers and Operations Research*, 24: 1097–1100.
- 14- Hansen, P., Mladenovic, N. and Pérez-Brito, D. (2001). "Variable neighborhood decomposition search." *Journal of Heuristics*, 7: 335-350.
- 15- Kirkpatrick, S., Gelatti, C.D. and Vecchi, M.P. (1994). *Optimization by simulated annealing*. In H. Gutfreund & G. Toulouse (Eds.), *Advanced series in neurosciences; biology and computation: A physicist's choice*, 3: 671–680.
- 16- Solomon, M.M. (1987). "Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints." *Operations Research*, 35: 254-265.

واژه های انگلیسی به ترتیب استفاده در متن

- 1- Capacitated Location-Routing Problem
 - 2- Capacitated Location-Routing Problem with Hard Time Windows
 - 3- Location-Routing Problem
 - 4- Prins
 - 5- Greedy Randomized Adaptive Search Procedure
 - 6- Barreto
 - 7- Marinakis & Marinaki
 - 8- Duhamel
 - 9- GRASP & Evolutionary Local Search
 - 10- Yu
 - 11- Nguyen
 - 12- Vehicle Routing Problem with Time Windows
 - 13- Mixed-Integer Linear Programming
 - 14- Push-Forward Insertion Heuristic
 - 15- Variable Neighborhood Search
 - 16- Mladenovi'c & Hansen
 - 17- Simulated Annealing
 - 18- Kirkpatrick
 - 19- Solomon
-