

## مسئله یکپارچه‌ی دریافت، تحویل و بازگشت وسایل نقلیه با محدودیت‌های بارگذاری سه‌بعدی و پنجره‌ی زمانی

امیرحسرتی<sup>۱</sup>، قاسم مصلحی<sup>۲\*</sup>، محمد رئیسی‌نافچی<sup>۳</sup>

۱. دانشجوی کارشناسی‌ارشد، دانشکده مهندسی صنایع و سیستم‌ها، دانشگاه صنعتی اصفهان، اصفهان، ایران

۲. استاد، دانشکده مهندسی صنایع و سیستم‌ها، دانشگاه صنعتی اصفهان، اصفهان، ایران

۳. استادیار، دانشکده مهندسی صنایع و سیستم‌ها، دانشگاه صنعتی اصفهان، اصفهان، ایران

### خلاصه

مسائل مسیریابی و بارگذاری، دو موضوع مهم برای کاهش هزینه‌های حمل‌ونقل محسوب می‌شود. در دهه‌ی اخیر به دلیل نزدیک‌سازی مسائل مسیریابی وسایل نقلیه به دنیای واقعی، این مسائل را به صورت یکپارچه با یکدیگر در نظر گرفته‌اند. رعایت نکردن محدودیت‌های بارگذاری منجر به آسیب رسیدن به کالاها و یا استفاده‌ی کمتر از فضای وسیله‌نقلیه می‌شود که در هر کدام از حالت‌ها باعث خسارت و ایجاد هزینه‌ی اضافه می‌شود. در این مقاله برای اولین بار مسئله یکپارچه‌ی مسیریابی دریافت، تحویل و بازگشت با محدودیت‌های بارگذاری سه‌بعدی و پنجره‌ی زمانی در نظر گرفته شده که محدودیت‌های انباشت، جهت‌گیری، عدم بارگذاری مجدد و شرایط چند تحویلی در این مسئله مورد بررسی قرار گرفته است. همچنین در این مطالعه، آیت‌ها و کانتینرها ناهمگون در نظر گرفته شده‌اند. با بررسی ادبیات موضوع این مسئله در ادبیات موضوع مشاهده نگردید. برای این مسئله یک مدل برنامه‌ریزی عدد صحیح مختلط، یک الگوریتم ابتکاری و دو الگوریتم فراابتکاری بر مبنای جست‌وجوی ممنوع و جست‌وجوی همسایگی متغیر ارائه شده است. الگوریتم‌های فراابتکاری در ابعاد کوچک با حل پایین حاصل از آزادسازی برخی محدودیت‌های مدل ارائه شده مورد ارزیابی قرار گرفته و در ابعاد بزرگ نیز دو الگوریتم فراابتکاری با یکدیگر مقایسه شده‌اند. نتایج نشان می‌دهد متوسط درصد خطای نسبی در الگوریتم جست‌وجوی ممنوع و جست‌وجوی همسایگی متغیر به ترتیب برابر ۰/۹۶ و ۰/۸۸ می‌باشد. همچنین الگوریتم جست‌وجوی ممنوع و جست‌وجوی همسایگی متغیر توانسته‌اند از ۵۴ نمونه به ترتیب در ۲۷ و ۲۵ نمونه جواب بهتری ارائه دهند.

### اطلاعات مقاله

تاریخچه مقاله:

دریافت ۱۳۹۸/۳/۸

پذیرش ۱۳۹۸/۷/۱۱

(مقاله پژوهشی)

کلمات کلیدی:

مسیریابی

دریافت

تحویل و بازگشت

بارگذاری سه‌بعدی

جست‌وجوی ممنوع

جست‌وجوی همسایگی متغیر

### ۱. مقدمه

چیدمان مناسب برای تقاضاهای مشتریان بوده، به طوری که بیشترین استفاده از فضای وسیله‌نقلیه شود. در اغلب موارد، استفاده بهینه از فضای وسیله نقلیه، موجب کاهش تعداد وسایل نقلیه مورد استفاده می‌شود. همچنین در نظر گرفتن محدودیت‌های بارگذاری برای تقاضاهای مشتری‌ها موجب نزدیک شدن مسئله به دنیای واقعی شده و قابلیت اجرایی آن بیشتر می‌شود.

مسائل مسیریابی و بارگذاری وسایل نقلیه دو چالش بزرگ در زمینه‌ی حمل‌ونقل و لجستیک هستند که محققین در سال‌های اخیر به ترکیب و یکپارچه در نظر گرفتن این دو مسئله توجه بسیار کرده‌اند. در مسئله مسیریابی وسیله‌نقلیه به دنبال برآورد کردن تقاضای مشتری‌ها با کمترین هزینه بوده و در مسئله بارگذاری در پی به دست آوردن

\* نویسنده مسئول: قاسم مصلحی

تلفن: ۰۳۱-۳۳۹۱۵۵۰۹؛ پست الکترونیکی: [moslehi@iut.ac.ir](mailto:moslehi@iut.ac.ir)

مسأله ارائه شده است که مجموعه‌ای از آزمایش‌ها و بررسی‌ها مربوط به سیستم برنامه‌ریزی مسأله هواپیمای بدون سرنشین می‌باشد. لی و لیم [۲] نیز مسأله دریافت و تحویل را با محدودیت پنجره‌ی زمانی در نظر گرفتند؛ آن‌ها برای حل این مسأله روش‌های فراابتکاری شبیه‌سازی تبرید ( $SA^2$ ) و جست‌وجوی ممنوع ( $TS^3$ ) استفاده کرده‌اند. تابع هدف مسأله آن‌ها شامل چهار جزء، مجموع هزینه‌ی سفر، کل زمان مشغول بودن وسیله‌نقلیه، تعداد وسایل‌نقلیه‌ی مورد استفاده و میزان زمان معطلی کانتینر برای ورود به گره می‌باشد. میزان اولویت این چهار جز متغیر بوده و اولویت‌ها به ترتیب برای تعداد وسیله‌نقلیه، مجموع هزینه‌ی سفر، کل زمان مشغول بودن وسیله‌نقلیه و میزان معطلی کانتینر برای ورود به گره می‌باشد. لی و لیم [۲] با توجه به مسائل نمونه‌ی سلومون [۳] ۳۵۴ نمونه‌ی تولید و حل کرده و ادعا نموده‌اند که بهترین نتایج را تا آن زمان به دست آورده‌اند.

در سال ۲۰۱۹ برآو و همکاران [۴] یک الگوریتم تکاملی برای مسأله دریافت و تحویل ارائه دادند که این مسأله چند تابع هدف شامل حداقل کردن هزینه و آلودگی هوای تولید شده و همچنین حداکثر کردن تعداد مشتری‌های ملاقات شده را دارد. آن‌ها با وسایل‌نقلیه‌ی ناهمگون، نقاط دریافت و تحویل را طی کردند. برای ابعاد کوچک مدل ریاضی و برای ابعاد بزرگ نیز الگوریتم تکاملی ارائه شده و برای توابع هدف مرز پار تو ترسیم شده است. استیک و ویکارک [۵] در سال ۲۰۱۹ مدل ریاضی برای مسأله دریافت و تحویل در نقاط مختلف ارائه دادند. این مدل صرفاً در ابعاد کوچک قادر به حل مسأله بوده و لذا برای ابعاد بزرگ یک رویکرد ترکیبی توسط آن‌ها ارائه شده است. گسترش مسأله تنها در نظر گرفتن چندین نقاط تحویل برای یک دریافت است که گسترشی از حمل‌ونقل شرکت‌های پستی می‌باشد.

رامادانکومر و راجانداران [۶] برای مسأله مسیریابی دریافت و تحویل هم‌زمان بازگشتی با وسایل‌نقلیه‌ی ناهمگون طی یک مدل ریاضی عدد صحیح ارائه دادند. تابع هدف مسأله در نظر گرفته شده شامل هزینه‌ی ثابت استفاده از وسیله‌نقلیه و طی کردن مسافت بین گره‌ها می‌باشد. اچوری و همکاران [۷] نیز برای مسأله مسیریابی بازگشتی یک مدل ریاضی ارائه داده‌اند. این مدل ریاضی با تمرکز بر به‌دست آوردن راه‌حل‌های معتبر توسط مسیرهای همپلتون تشکیل شده است؛ همچنین روش لانه‌ی زنبور عسل ( $ABC^4$ ) را برای مسأله مسیریابی بازگشت و وسایل‌نقلیه ارائه دادند که در پی به‌دست آوردن مجموعه‌ای از وسایل‌نقلیه است که مجموعه هزینه‌ی طی شده‌ی آن برای گره‌ها حداقل شود، به‌شرطی که محدودیت مسیر و پنجره‌ی زمانی برای آن رعایت شده باشد.

کراسپسم و برانادا [۸] یک الگوریتم ترکیبی TS و جست‌وجوی همسایگی متغیر ( $VNS^5$ ) برای مسیریابی بازگشتی ترکیبی و مسیریابی بازگشتی هم‌زمان ارائه دادند. در مسیریابی بازگشت هم‌زمان مشتریان می‌توانند در یک زمان هم کالا دریافت و هم تحویل دهند

در مسأله مسیریابی دریافت و تحویل<sup>۱</sup> (PDP) تقاضای  $n$  مشتری باید برآورده شود. می‌توان تقاضاهای یک مشتری از گره‌ی دریافت آن گرفته شده و به گره‌ی تحویل متناظرش داده می‌شود. گره‌های دریافت مشتریان شامل گره‌های ۱ تا  $n$  و گره‌ی تحویل مشتریان شامل گره‌های  $n+1$  تا  $2n$  می‌باشد. گره‌های  $\{0, 2n+1\}$  نیز انبارهای ابتدایی و انتهایی می‌باشد که وسیله‌نقلیه به ترتیب از آن خارج و به آن داخل می‌شود. مسیریابی دریافت و تحویل دو محدودیت بیشتر از مسیریابی کلاسیک با نام‌های پیش‌نیازی و جفت کردن دارد. محدودیت پیش‌نیازی الزام می‌کند که گره‌ی دریافت یک مشتری زودتر از گره‌ی تحویل همان مشتری طی شود. محدودیت جفت‌سازی نیز بیان می‌کند که گره دریافت و تحویل یک مشتری باید توسط یک وسیله‌نقلیه طی شود. از آنجایی که در این تعریف هر گره دریافت فقط به یک گره تحویل و هر گره تحویل فقط با یک گره دریافت، ارتباط دارد؛ به آن مسأله دریافت و تحویل یک‌به‌یک گفته می‌شود.

در مسأله مسیریابی بازگشتی بدون دریافت و تحویل، هدف تحویل آیت‌های رفت به گره‌های رفت و دریافت آیت‌های بازگشت، از گره‌های بازگشت با کمترین هزینه می‌باشد. در این مسأله، آیت‌های رفت از انبار ابتدایی دریافت و به گره‌های رفت تحویل داده می‌شوند و آیت‌های بازگشتی از گره‌های بازگشت دریافت و به انبار انتهایی تحویل داده می‌شوند. در این مسأله، فرض‌های متفاوتی در نظر گرفته می‌شود؛ به‌طور مثال، نحوه‌ی طی کردن گره‌های رفت و بازگشت را می‌توان به دو صورت مختلط و یا تفکیک شده در نظر گرفت. در حالت طی شدن گره‌ها به‌صورت مختلط، گره‌های بازگشت در میان گره‌های رفت، طی شده؛ اما در حالت تفکیک شده، گره‌های بازگشت بعد از گره‌های رفت طی می‌شوند.

در ادامه در بخش ۲ مرور بر ادبیات موضوع مسأله، مورد بررسی صورت خواهد گرفت؛ و سپس در بخش ۳ تعریف دقیقی از مسأله ارائه خواهد شد. در ادامه و در بخش ۴، یک مدل برنامه‌ریزی ریاضی برای مسأله ارائه می‌گردد؛ و پس از آن در بخش ۵، یک الگوریتم ابتکاری و دو الگوریتم فراابتکاری توسعه داده شده برای حل مسأله، با جزئیات توصیف خواهند شد. آزمایشات عددی برای بررسی عملکرد الگوریتم‌های ارائه شده در بخش ۶ آمده؛ و در نهایت، در بخش ۷ ضمن جمع‌بندی، پیشنهادهایی برای مطالعات آتی بیان شده است.

## ۲. مرور ادبیات موضوع

به‌دلیل این‌که این مسیریابی، ترکیبی از مسیریابی دریافت و تحویل به‌صورت یک‌به‌یک و مسیریابی بازگشت می‌باشد، در این بخش به مرور ادبیات موضوع هر دو نوع مسیریابی پرداخته شده است.

کاراک و عبدلقانی [۱] برای مسأله ترکیبی هواپیماهای بدون سرنشین با نقاط دریافت و تحویل یک مدل ریاضی عدد صحیح و یک جواب ابتکاری ارائه دادند؛ همچنین دو روش ابتکاری دیگر برای این

4. Artificial Bee Colony

5. Variable Neighborhood Search

1. Pickup and Delivery Problem

2. Simulated Annealing

3. Tabu Search

بارگذاری کاربرد زیادی در زنجیره‌ی تأمین دارد، اما کمتر به آن پرداخته شده است. بورتفولد و همکاران [۱۵] در سال ۲۰۱۵ مسیریابی بازگشتی خوشه‌بندی شده را با محدودیت‌های بارگذاری سه‌بعدی و هم‌چنین پینتو و همکاران [۱۶] در سال ۲۰۱۸ مسیریابی بازگشتی ترکیبی را با محدودیت‌های بارگذاری دوبعدی در نظر گرفتند. در این دو مطالعه، پنجره‌ی زمانی در نظر گرفته نشده است. تنها در مطالعه‌ی ریل و همکاران [۱۷]، مسیریابی بازگشتی با محدودیت‌های بارگذاری سه‌بعدی و پنجره‌ی زمانی به‌صورت یکپارچه در نظر گرفته شده است. در این مطالعه، چهار نوع مسیریابی بازگشتی که شامل مسیریابی بازگشتی خوشه‌بندی شده، مسیریابی بازگشتی ترکیبی، مسیریابی بازگشتی با دریافت و تحویل هم‌زمان و مسیریابی دریافت و تحویل قابل تقسیم می‌باشد، با استفاده از جست‌وجوی ممنوع (اول بسته‌بندی و سپس مسیریابی) حل شده است. در این مطالعه، رویکردهایی مانند دو قسمت کردن کانتینر و یا بارگذاری جانبی نیز مورد آزمایش قرار گرفته است. در بارگذاری جانبی، فرض وجود درب در کناره‌های کانتینر برای بارگذاری و تخلیه‌ی آسان‌تر است. این ایده با وجود این‌که در واقعیت کاربردی ندارد، اما از نظر تئوری ایده‌ی جالبی است.

در جدول (۱) خلاصه‌ای از مطالعات صورت گرفته در زمینه‌ی یکپارچه‌سازی مسیریابی و بارگذاری آورده شده و با مقاله‌ی حاضر مقایسه شده است؛ بر این اساس، مسأله مورد بررسی در این مقاله از منظر محدودیت‌های در نظر گرفته شده در مسیریابی و بارگذاری جدید است. هم‌چنین فرض بازگشت نیز علاوه بر این موارد در مسأله لحاظ شده که در کل، مسأله را کاربردی‌تر نموده و بررسی آن را جذاب می‌نماید؛ لذا به‌طور کلی می‌توان گفت مسأله دریافت و تحویل یک‌به‌یک با پنجره‌ی زمانی و در نظر گرفتن بازگشت تاکنون در ادبیات موضوع مشاهده نشده و در این مقاله برای اولین بار به این موضوع پرداخته خواهد شد.

### ۳. تعریف مسأله

در این مطالعه، مسأله مسیریابی دریافت، تحویل و بازگشت با مسأله بارگذاری با محدودیت‌های سه‌بعدی به‌صورت یکپارچه در نظر گرفته شده است. مسأله مسیریابی در نظر گرفته شده شامل دو مسیر رفت و بازگشت است. در مسیر رفت آیت‌ها از گره‌های دریافت، دریافت شده و به گره‌های تحویل متناظرش تحویل داده می‌شود و در مسیریابی بازگشت آیت‌ها از گره‌های بازگشت دریافت شده و به گره‌ی انبار انتهایی تحویل داده می‌شود. قابل ذکر است که محدودیت‌های بارگذاری و مسیریابی برای مسیر رفت و بازگشت در نظر گرفته می‌شود. به‌عبارت دیگر، در مسیر رفت  $n$  درخواست وجود دارد و تعداد گره‌ها برابر  $3n+2$  است. هر درخواست شامل گره‌ی دریافت  $k$  و گره‌ی تحویل  $n+k$  می‌باشد. در مسیر بازگشت تعداد  $n$  درخواست وجود دارد که گره‌ی بازگشت درخواست  $k$  گره  $2n+k$  می‌باشد و مختصات

برای افزایش تنوع فرضیات کمتری در نظر گرفته شده است. جست‌وجو در ساختار همسایگی‌ها با انجام کارهایی که ممنوع نیست انجام می‌گیرد. راه‌حل‌های ارائه شده در اغلب نمونه‌های صالحی و ناجی [۹] که شامل ۵۰ تا ۱۹۹ مشتری می‌باشد، نتایج بهتری به‌دست آورده است.

ترکیب مسائل مسیریابی و بارگذاری سه‌بعدی اولین بار توسط گندرو و همکاران [۱۰] مطرح شد؛ آن‌ها این مسأله را مسیریابی وسایل نقلیه با محدودیت‌های بارگذاری سه‌بعدی نامیده که در سال‌های اخیر توجه به این نوع مسائل، بسیار افزایش یافته است. در این مقاله برای اولین بار محدودیت‌های بارگذاری سه‌بعدی با مسیریابی به‌صورت توأم مطرح شد و یک روش TS برای مسأله مسیریابی و یک روش TS برای بارگذاری سه‌بعدی ارائه شد.

فگرهولت و همکاران [۱۱] مسأله مسیریابی با دریافت و تحویل و محدودیت‌های بارگذاری را با محدودیت پنجره‌ی زمانی فرموله کردند. در این مطالعه، مسأله دریافت و تحویل یک‌به‌یک در نظر گرفته شده است. محدودیت‌های پایه‌ای بارگذاری از قبیل عدم هم‌پوشانی دو آیت‌م و پوشش کامل آیت‌های بارگذاری شده در کانتینر نیز در نظر گرفته شده است. محدودیت بارگذاری وزنی، توزیع وزنی و چرخش ۹۰ درجه نیز برای نزدیک شدن به دنیای واقعی در این مطالعه گنجانده شده است. چرکسی و همکاران [۱۲] نیز مسأله فگرهولت و همکاران [۱۱] را با محدودیت شرایط چند تحویلی (LIFO) و پنجره‌ی زمانی در نظر گرفتند. الگوریتم شاخه و قیمت برای حل دقیق این مسأله تا ۷۵ درخواست جواب می‌دهد.

منل و بورتفولد [۱۳] در سال ۲۰۱۵ مسأله دریافت و تحویل یک‌به‌یک را با محدودیت‌های بارگذاری در پنج دسته تقسیم کردند. مطالعه‌ی آن‌ها به‌دلیل مخصوص بودن آیت‌ها، تابع هدف را عدم بارگذاری مجدد آیت‌م در نظر گرفته و سه دسته از پنج دسته ارائه شده را مورد بررسی قرار دادند. آنان نشان دادند که عدم بارگذاری مجدد آیت‌م تنها با محدودیت LIFO برقرار نمی‌باشد و نیاز به در نظر گرفتن محدودیت‌هایی تحت‌عنوان «محدودیت‌های مسیر جزئی» است. محدودیت مسیر جزئی الزام می‌کند اگر یک کانتینر به گره‌ی تحویل یک مشتری برود، باید بعد از آن تمام گره‌های تحویل مشتری‌هایی که آیت‌های آن‌ها در کانتینر می‌باشد را طی کند و بعد از خالی شدن کانتینر می‌تواند به گره‌ی دریافت دیگری برود. در این مطالعه نمونه‌هایی با ۵۰، ۷۵ و ۱۰۰ مشتری تولید و حل شده‌اند.

منل و بورتفولد [۱۴] در سال ۲۰۱۸ دو دسته از مسائل باقی‌مانده در مطالعه‌ی منتشر شده در سال ۲۰۱۵ را با در نظر گرفتن محدودیت‌های بسته‌بندی حل کردند. در این مطالعه، عدم بارگذاری مجدد به‌صورت محدودیت نرم در نظر گرفته شده است که تعادلی بین عدم بارگذاری مجدد و مسافت طی شده به‌دست آورده است. آن‌ها نمونه‌هایی با ۵۰، ۷۵ و ۱۰۰ مشتری را تولید و حل کرده‌اند. اگرچه یکپارچه‌سازی مسائل مسیریابی بازگشتی وسایل نقلیه و

قرارگیری آن با گره‌ی تحویل مشتری  $k$  یکسان است ( $k=1,2,\dots,n$ ). انتهای می‌باشد که هر وسیله‌نقلیه، شروعش از گره 0 و مقصدش گره دو گره‌ی 0 و  $3n+1$  به ترتیب نشان‌دهنده‌ی گره‌ی انبار ابتدایی و  $3n+1$  است.

جدول (۱): خلاصه‌ای از مطالعه‌های انجام گرفته در یکپارچه‌سازی مسیریابی و بارگذاری

نویسندگان	سال	محدودیت‌ها		روش حل	حداکثر ابعاد (مشتری)
		بارگذاری	مسیریابی		
پولاریس و همکاران [۱۸]	۲۰۱۴	وزن، توزیع وزن تخصیص	LIFO	مدل ریاضی	۲۵
تائو و وانگ [۱۹]	۲۰۱۵	وزن، جهت‌گیری، انباشت، تخصیص	LIFO	TS	۱۲۵
جانکوئرا و مورابیتو [۲۰]	۲۰۱۵	پایداری، تخصیص	LIFO	SA و LS	۱۰۰
ژانگ و همکاران [۲۱]	۲۰۱۵	وزن، جهت‌گیری، انباشت و تخصیص	LIFO	ELS <sup>۱</sup>	۱۲۵
مهوش و همکاران [۲۲]	۲۰۱۶	وزن، جهت‌گیری، انباشت، تخصیص	LIFO و تعداد مسیر	تولید ستون	۱۰۰
حسن‌آبادی [۲۳]	۲۰۱۶	وزن، انباشت، جهت‌گیری	LIFO و پنجره‌ی زمانی	دقیق و فراابتکاری	۱۰۰
راچاریداس و همکاران [۲۴]	۲۰۱۷	جهت‌گیری	LIFO و تعداد مسیر	ابتکاری	۱۵۰
پینتو و همکاران [۱۶]	۲۰۱۸	وزن و جهت‌گیری	LIFO	VNS	۲۵۵
وگا و تروس [۲۵]	۲۰۱۹	وزن و توزیع وزنی	پنجره‌ی زمانی	دقیق و ابتکاری	۲۵
سانگ و همکاران [۲۶]	۲۰۱۹	محدودیت تثبیت موقعیت	پنجره زمانی	دقیق و VNS	۴۸۹
بورتفلدت و وی [۲۷]	۲۰۱۹	وزن، انباشت، جهت‌گیری	تقسیم آیتم‌ها	دقیق و ابتکاری	۱۰۰
مقاله‌ی حاضر		وزن، انباشت، جهت‌گیری، عدم بارگذاری مجدد	LIFO و پنجره‌ی زمانی	دقیق و VNS و TS	۱۰۰

شروع به کار می‌کند بعد از گره‌ی 0 به یک گره‌ی دریافت برود. در مسیر رفت آیتم‌های برخی از گره‌های دریافت مشتری‌ها در کانتینر بارگذاری می‌شوند و در گره‌ی تحویل همان مشتری، این آیتم‌ها تخلیه می‌شوند و در مسیر بازگشت نیز آیتم‌هایی از گره‌های بازگشت در کانتینر بارگذاری می‌شوند و به گره‌ی  $3n+1$  برده می‌شوند. علاوه بر محدودیت‌های عدم هم‌پوشانی و قرارگیری کامل در کانتینر، محدودیت‌های جهت‌گیری، پایداری، انباشت، LIFO، عدم بارگذاری مجدد، محدودیت تخصیص و پنجره‌ی زمانی در نظر گرفته می‌شود.

لازم به ذکر است که مسیریابی PDP با در نظر گرفتن گره‌های بازگشت به صورت مجزا و مسیریابی PDP توأم با بازگشت با هم متفاوت می‌باشد؛ به عبارت دیگر نمی‌توان جواب بهینه‌ی مسیریابی PDP بدون گره‌های بازگشت و جواب بهینه‌ی مسیریابی فقط گره‌های بازگشت را به دست آورد و ادعا نمود که جواب بهینه در حالت توأم به دست آمده است. برای نشان دادن این موضوع، به مثال ۱ توجه شود.

**مثال ۱:** با توجه به داده‌های جدول (۱) و جدول (۲) مسأله در دو حالت بررسی شده است که گره‌های P و D به ترتیب گره‌های دریافت و تحویل یک مشتری را نشان می‌دهند. در حالت اول مثال ۱، مسیریابی PDP بدون آیتم‌های بازگشتی در نظر گرفته شده که در نهایت آیتم‌های بازگشت به صورت مجزا از مسیریابی PDP به این حالت اضافه شده است. در حالت دوم، مسیریابی PDP با آیتم‌های بازگشتی به صورت توأم در نظر گرفته شده است. آیتم‌ها باید توسط

تعداد  $V$  وسیله‌نقلیه‌ی ناهمگون در زمان صفر در انبار ابتدایی وجود دارد.  $E$  مجموعه‌ای از یال‌های  $(k,l)$  می‌باشد که جفت گره‌های  $(0 \leq k, l \leq 3n+1)$  را با فرض‌هایی به یکدیگر متصل می‌سازد. هزینه‌ی سفر  $C_{kl}$  به هر یال  $(k,l)$  اختصاص دارد و متقارن است یا به عبارت دیگر  $C_{kl} = C_{lk}$  می‌باشد. هزینه‌ی سفر هر یال با مدت‌زمان طی کردن یال برابر فرض شده است. برای هر گره  $k$  پنجره‌ی زمانی  $[A_k, B_k]$  تعریف می‌شود؛ که زودتر از  $A_k$  و دیرتر از  $B_k$  نمی‌توان وارد گره  $k$  شد ( $0 \leq k \leq 3n$ ). علاوه بر این هر مشتری  $k$  دارای  $I_k$  جعبه است و هر جعبه دارای طول، عرض و ارتفاع می‌باشد که به ترتیب با  $l_{ik}$ ،  $w_{ik}$  و  $h_{ik}$  نشان داده شده و  $i$  نشان‌دهنده‌ی جعبه و  $k$  نشان‌دهنده‌ی مشتری است.

یک مسیر امکان‌پذیر  $R$  یک توالی از  $3n+1$  گره می‌باشد که از گره‌ی 0 شروع شده و به گره‌ی  $3n+1$  ختم می‌شود.  $R$  باید شامل گره‌های دریافت و تحویل و بازگشت مشتریان باشد. مختصات قرارگیری گره‌ی تحویل و بازگشت یکسان می‌باشند؛ به این معنی که همواره گره‌ی تحویل و گره‌ی بازگشت یک مشتری بر روی هم قرار گرفته و فاصله‌ی آن‌ها صفر است. گره‌ی دریافت باید مقدم بر گره‌ی تحویل همان درخواست باشد و باید هر گره تنها یک بار طی شود و صرفاً آیتم‌های بازگشتی می‌توانند با هم در یک کانتینر قرار گیرد. هم‌چنین فرض می‌شود یک وسیله‌نقلیه نمی‌تواند صرفاً برای جمع‌آوری آیتم‌های بازگشتی اختصاص داده شود و حتماً باید وسیله‌نقلیه‌ای که

#### ۴. مدل برنامه‌ریزی ریاضی

در این بخش یک مدل ریاضی ارائه شده که با عنوان (3L\_PDBP\_TW) نام‌گذاری می‌شود. ابتدا پارامترها و متغیرهای تصمیم این مدل توضیح داده شده و سپس فرمول‌بندی این مدل شرح داده می‌شود.

#### پارامترها

$n$	تعداد مشتری‌ها
$V$	مجموعه وسایل نقلیه در دسترس
$P = \{1, \dots, n\}$	مجموعه گره‌های دریافت
$D = \{n + 1, \dots, 2n\}$	مجموعه گره‌های تحویل
$R$	مجموعه گره‌های بازگشتی
$R = \{2n + 1, \dots, 2n + n\}$	
$T$	مجموعه دوره‌های زمانی
$I_k$	مجموعه آیتم‌های مشتری $k$
$V_r$	مجموعه کل گره‌ها
$V_r = P \cup D \cup \{0, 3n + 1\}$	
$c_{kl}$	هزینه‌ی سفر از گره‌ی $l$ به گره‌ی $k$
$t_{kl}$	زمان سفر از گره‌ی $k$ به گره‌ی $l$
$A_k, B_k$	پنجره‌ی زمانی برای گره‌ی $k$
$A_k, B_k$	$k \in P \cup D \cup \{0\}$
$m_{ik}$	وزن آیتم $i$ گره‌ی $k$
$s_k$	مدت‌زمان انجام خدمت گره‌ی $k$
$C_v$	ظرفیت وسیله‌نقلیه‌ی $v$
$v_v$	هزینه‌ی استفاده از وسیله‌نقلیه‌ی $v$
$l_{ik}^+$	برابر یک است اگر طول آیتم $i$ مشتری $k$ بتواند به‌عنوان ارتفاع قرار گیرد، در غیر این‌صورت برابر صفر است.
$l_{ik}^+$	$i \in I_k, k \in P \cup R$
$w_{ik}^+$	برابر یک است اگر عرض آیتم $i$ مشتری $k$ بتواند به‌عنوان ارتفاع قرار گیرد، در غیر این‌صورت برابر صفر است.
$w_{ik}^+$	$i \in I_k, k \in P \cup R$
$h_{ik}^+$	برابر یک است اگر ارتفاع آیتم $i$ مشتری $k$ بتواند به‌عنوان ارتفاع قرار گیرد، در غیر این‌صورت برابر صفر است.
$h_{ik}^+$	$i \in I_k, k \in P \cup R$
$l_{ik}$	طول آیتم $i$ مشتری $k$
$w_{ik}$	عرض آیتم $i$ مشتری $k$
$h_{ik}$	ارتفاع آیتم $i$ مشتری $k$
$L_v$	طول کانتینر $v$
$W_v$	عرض کانتینر $v$

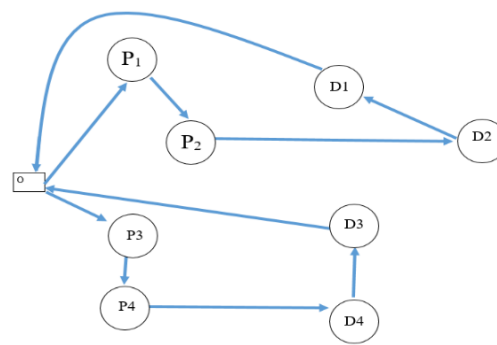
کانتینر بزرگ و کوچک که به‌ترتیب دارای هزینه‌ی ۱۲۰ و ۱۰۰ و ظرفیت به‌ترتیب ۴۵۰۰ و ۹۱۵۰۰ می‌باشد، برآورده شود. جواب بهینه‌ی حالت ۱ در شکل (۱) و برای حالت ۲ در شکل (۲) نشان داده شده است؛ همان‌طور که مشاهده می‌شود، جواب در دو حالت یکسان نمی‌باشد. مقدار هزینه برای شکل (۱) و شکل (۲) به‌ترتیب ۱۰۶۰ و ۹۴۰ است.

جدول (۱): فاصله‌های گره‌های دریافت و تحویل در مثال ۱

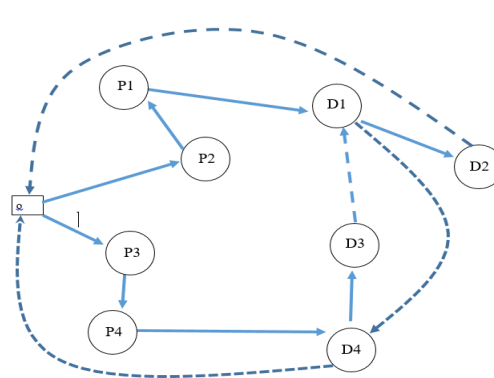
از/به	0	P1	P2	P3	P4	D1	D2	D3	D4
D4	۱۲۰	۱۱۰	۱۷۰	۱۵۰	۷۰	۴۰	۳۰	۲۰	۰
P1	۲۶۰	۲۵۰	۱۵	۳۰	۲۰۰	۳۰۰	۲۰	۰	۲۰
P2	۲۰۰	۱۹۰	۷۰	۴۰	۱۷۰	۱۸۰	۰	۲۰	۳۰
P3	۲۰۰	۴۰	۱۹۰	۱۶۰	۲۰	۰	۱۸۰	۳۰۰	۴۰
P4	۴۰	۲۰۰	۱۷۰	۱۲۰	۰	۲۰	۱۷۰	۲۰۰	۷۰
D1	۵۰	۴۰	۱۰۰	۰	۱۲۰	۱۶۰	۴۰	۳۰	۱۵۰
D2	۲۳۰	۱۸۰	۰	۱۰۰	۱۷۰	۱۹۰	۷۰	۱۵	۱۷۰
D3	۸۰	۰	۱۸۰	۴۰	۲۰۰	۴۰	۱۹۰	۲۵۰	۱۱۰
D4	۰	۸۰	۲۳۰	۵۰	۴۰	۲۰۰	۲۰۰	۲۶۰	۱۲۰

جدول (۲): ویژگی آیتم‌های مشتری در مثال ۱

گره	تعداد آیتم‌ها	مجموع وزن آیتم‌ها
P1	۲	۲۰۰۰
P2	۳	۲۵۰۰
P3	۲	۱۹۰۰
P4	۲	۱۰۰۰



شکل (۱): حل مثال ۱ در حالت اول



شکل (۲): حل مثال ۱-۳ در حالت دوم

$(XY)$  باشد مقدار یک و در غیر این صورت مقدار صفر می‌گیرد.

$$i \in I_k, k \in P, R,$$

$$a \in \{1, 2, \dots, L-1\}, b \in \{1, 2, \dots, W-1\}$$

اگر آیتم  $j$  مشتری  $l$  در دوره‌ی  $t$  در نقطه‌ی  $a, b$  صفحه‌ی  $(XY)$  جعبه‌ی  $i$  مشتری  $k$  را حمایت کند.

$$a \in \{1, 2, \dots, L-1\}, b \in \{1, 2, \dots, W-1\}$$

$$i, j \in I_k, k, l \in P \cup R, t \in T$$

اگر آیتم  $i$  مشتری  $k$  روی کف وسیله‌نقلیه نباشد مقدار یک می‌گیرد، در غیر این صورت مقدار صفر می‌گیرد.

$$i \in I_k, k \in P, R$$

در ادامه به دلیل طولانی بودن مدل، هر قسمت آن به صورت جداگانه ارائه و توضیح داده می‌شود.

#### ۴-۱. تابع هدف و محدودیت‌های وزن و ابعاد

$$\text{Min} \sum_{v=1}^V U_v V_v + \sum_{v=1}^V \sum_{k=0}^{3n+1} \sum_{l=0}^{3n+1} \sum_{t=0}^{3n} c_{kl} d_{klv}^t \quad (1)$$

$$\sum_{k=1}^n \sum_{i=1}^{I_k} m_{ik} p_{ikv}^t \leq C_v U_v \quad v \in V \quad (2)$$

$$\sum_{k=2n+1}^{3n+1} \sum_{i=1}^{I_k} m_{ik} p_{ikv}^t \leq C_v u_v \quad v \in V \quad (3)$$

$$x'_{ik} \leq \sum_{v=1}^V (l_v - L) p_{ikv}^t + L \quad i \in I_k, k \in P \cup R, v \in V \quad (4)$$

$$y'_{ik} \leq \sum_{v=1}^V (w_v - W) p_{ikv}^t + W \quad (5)$$

$$i \in I_k, k \in P \cup R, v \in V$$

$$z'_{ik} \leq \sum_{v=1}^V (h_v - H) p_{ikv}^t + H \quad (6)$$

$$i \in I_k, k \in P \cup R, v \in V$$

تابع هدف (۱) تعداد وسایل نقلیه‌ی استفاده شده و مجموع هزینه‌های مسیر را کمینه می‌کند. محدودیت (۲) و (۳) به ترتیب محدودیت ظرفیت را برای آیتم‌های رفت و بازگشت بیان می‌کند که مجموع وزن آیتم‌هایی که در هر دوره در وسیله‌نقلیه‌ی  $v$  قرار می‌گیرند، از ظرفیت کانتینر تجاوز نکند. محدودیت (۴) تا (۶) تضمین می‌کند که آیتم‌های هر مشتری از ابعاد وسیله‌نقلیه‌ای که در آن قرار می‌گیرد تجاوز نکنند. در این حالت اگر در دوره‌ی  $t$  هنوز آیتم‌های مشتری  $k$  بارگذاری نشده باشند یا پیش از دوره‌ی  $t$  تخلیه شده باشند، مختصات نقطه‌ی روبه‌روی نقطه‌ی مرجع آیتم  $i$  مشتری  $k$  کوچک‌تر از حداکثر طول، عرض و ارتفاع وسایل نقلیه است؛ به عبارت دیگر، در این حالت محدودیت مازاد می‌شود.

#### ۴-۲. محدودیت‌های جهت‌گیری

$$x'_{ik} - x_{ik} = r_{ik11} l_{ik} + r_{ik12} w_{ik} + r_{ik13} h_{ik} \quad (7)$$

$$i \in I_k, k \in P \cup R$$

$H_v$	ارتفاع کانتینر $v$	$v \in V$
$L$	بزرگ‌ترین طول کانتینر مورداستفاده	
$W$	بزرگ‌ترین عرض کانتینر مورداستفاده	
$H$	بزرگ‌ترین ارتفاع کانتینر مورداستفاده	

#### متغیرهای تصمیم پیوسته

$(z_{ik}, y_{ik}, x_{ik})$  مختصات مکان قرارگیری گوشه‌ی مرجع

آیتم  $i$  مشتری  $k$   $i \in I_k, k \in R \cup P$

$(z'_{ik}, y'_{ik}, x'_{ik})$  مختصات مکان قرارگیری گوشه‌ی روبه‌روی

مرجع آیتم  $i$  مشتری  $k$

$$i \in I_k, k \in R \cup P$$

$T_{kv}$  زمان شروع خدمت گره‌ی  $k$  توسط وسیله

نقلیه‌ی  $v$   $k \in V_r, v \in V$

#### متغیرهای تصمیم صفر و یک

اگر وسیله‌نقلیه‌ی  $v$  در دوره‌ی  $t$  از گره‌ی  $k$  به گره‌ی  $l$

سفر کند، مقدار یک و در غیر این صورت مقدار صفر

می‌گیرد.  $t \in T, k, l \in V_r, v \in V$

اگر وسیله‌نقلیه‌ی  $v$  استفاده شود مقدار یک و در غیر

این صورت مقدار صفر می‌گیرد.

$v \in V$

اگر آیتم  $i$  مشتری  $k$  در دوره‌ی  $t$  در وسیله‌نقلیه‌ی  $v$

قرار داشته باشد مقدار یک و در غیر این صورت مقدار

صفر می‌گیرد.  $t \in T, k \in V_r, v \in V, i \in I_k$

اگر بعد  $b$  آیتم  $i$  مشتری  $k$  در راستای محور  $b$  باشد

مقدار یک، در غیر این صورت مقدار صفر می‌گیرد.

$i \in I_k, k \in P \cup R, a, b \in \{1, 2, 3\}$

اگر آیتم  $i$  مشتری  $k$  در سمت راست آیتم  $j$  مشتری  $l$

قرار بگیرد مقدار یک، در غیر این صورت مقدار صفر

می‌گیرد.  $i, j \in I_k, k, l \in P \cup R$

اگر آیتم  $i$  مشتری  $k$  جلوی آیتم  $j$  مشتری  $l$  قرار بگیرد

مقدار یک، در غیر این صورت مقدار صفر می‌گیرد.

$i, j \in I_k, k, l \in P \cup R$

اگر آیتم  $i$  مشتری  $k$  بالای آیتم  $j$  مشتری  $l$  قرار بگیرد

مقدار یک، در غیر این صورت مقدار صفر می‌گیرد.

$i, j \in I_k, k, l \in P \cup R$

اگر گره‌ی  $k$  زودتر از گره‌ی  $l$  طی شود مقدار یک، در

غیر این صورت مقدار صفر می‌گیرد.

$k, l \in P \cup D$

اگر آیتم  $j$  مشتری  $l$  ارتفاع مناسبی برای حمایت آیتم

$i$  مشتری  $k$  داشته باشد مقدار یک، در غیر این صورت

مقدار صفر می‌گیرد.  $i, j \in I_k, k, l \in P \cup R$

اگر آیتم  $i$  مشتری  $k$  شامل نقطه‌ی  $a, b$  در صفحه‌ی

$q_{abik}$

$he_{ikjl}$

$LS_{kl}$

$x_{ikjl}$

$y_{ikjl}$

$z_{ikjl}$

$$i, j \in I_k \cup I_l, k, l \in P \cup R$$

$$x'_{ik} - 1 \leq x'_{jl} + x'_{ikjl} L \quad (17)$$

$$i, j \in I_k \cup I_l, k, l \in P \cup R$$

$$y'_{jl} \leq y_{ik} + (1 - y'_{ikjl}) W \quad (18)$$

$$i, j \in I_k \cup I_l, k, l \in P \cup R$$

$$y_{ik} - 1 \leq y'_{jl} + y'_{ikjl} W \quad (19)$$

$$i, j \in I_k \cup I_l, k, l \in P \cup R$$

$$z'_{jl} \leq z_{ik} + (1 - z'_{ikjl}) H \quad (20)$$

$$i, j \in I_k \cup I_l, k, l \in P \cup R$$

زمانی که متغیرهای  $x'_{ikjl}, x'_{jlik}, y'_{ikjl}, y'_{jlik}, z'_{ikjl}, z'_{jlik}$  برابر یک هستند، دو آیت  $i$  و  $k$  در طول هیچ محوری هم‌پوشانی ندارند. برای جلوگیری از اشغال بخشی از فضا توسط دو آیت کافی است که هم‌پوشانی حداقل در طول یکی از محورها وجود نداشته باشد؛ به عبارت دیگر، در صورتی که تنها یکی از متغیرهای  $x'_{ikjl}, x'_{jlik}, y'_{ikjl}, y'_{jlik}, z'_{ikjl}, z'_{jlik}$ ،  $i$  و  $k$  برابر یک شود، عدم هم‌پوشانی دو آیت  $i$  و  $k$  تضمین می‌شود. سمت راست محدودیت (۱۵) تنها زمانی که دو آیت  $i$  و  $k$  در یک دوره و در یک وسیله‌نقلیه با هم حضور داشته باشند برابر یک می‌شود. در این حالت این محدودیت الزام می‌کند که یکی از متغیرهای  $x'_{ikjl}, x'_{jlik}, y'_{ikjl}, y'_{jlik}, z'_{ikjl}, z'_{jlik}$  برابر یک شود تا عدم هم‌پوشانی دو آیت  $i$  و  $k$  تضمین شود. هم‌پوشانی زمانی اتفاق می‌افتد که دو آیت در یک وسیله‌نقلیه باشند که این امر توسط سمت راست محدودیت (۱۵) بیان شده است. نحوه‌ی تعریف متغیرهای  $x'_{ikjl}, y'_{ikjl}, z'_{ikjl}$  توسط محدودیت‌های (۱۶) تا بیان شده است؛ برای مثال، محدودیت‌های (۱۶) و (۱۷) تضمین می‌کند که  $x'_{ikjl} = 1$  است اگر و تنها اگر  $x'_{ik} \geq x'_{ik}$  باشد. پارامترهای  $L, W, H$  در این محدودیت‌ها استفاده شده‌اند، زیرا معلوم نیست آیت‌های  $i$  و  $k$  درون کدام وسیله‌نقلیه قرار گرفته است.

برای محدودیت (۴) تا (۲۰) از مدل ارائه شده توسط پاکوی و همکاران [۲۸] الهام گرفته شده و اندیس دوره به متغیر  $P'_{ikv}$  اضافه شده است.

#### ۴-۴. محدودیت پایداری و شکندگی

$$z_{ik} + (1 - g_{ik}) H \geq 1 \quad i \in I_k, k \in P \cup R \quad (21)$$

$$z_{ik} \leq +g_{ik} H \quad i \in I_k, k \in P \cup R \quad (22)$$

$$x'_{ik} \leq a + (1 - q_{abik}) L \quad (23)$$

$$i \in I_k, a \in \{1, 2, \dots, L - 1\},$$

$$b \in \{1, 2, \dots, W - 1\}, k \in P \cup R$$

$$x'_{ik} \geq (a + 1) q_{abik} \quad (24)$$

$$i \in I_k, a \in \{1, 2, \dots, L - 1\},$$

$$b \in \{1, 2, \dots, W - 1\}, k \in P \cup R$$

$$y_{ik} \leq b + (1 - q_{abik}) W \quad (25)$$

$$y'_{ik} - y_{ik} = r_{ik21} l_{ik} + r_{ik22} w_{ik} + r_{ik23} h_{ik} \quad (8)$$

$$i \in I_k, k \in P \cup R$$

$$z'_{ik} \leq \sum_{v=1}^V (H_v - h) P'_{ikv} + H \quad (9)$$

$$i \in I_k, k \in P \cup R$$

$$\sum_{a=1}^3 r_{ikab} = 1 \quad (10)$$

$$i \in I_k, k \in P \cup R, b \in \{1, 2, 3\}$$

$$\sum_{b=1}^3 r_{ikab} = 1 \quad (11)$$

$$i \in I_k, k \in P \cup R, a \in \{1, 2, 3\}$$

$$r_{ik31} \leq l_{ik}^+ \quad i \in I_k, k \in P \cup R \quad (12)$$

$$r_{ik32} \leq w_{ik}^+ \quad i \in I_k, k \in P \cup R \quad (13)$$

$$r_{ik33} \leq h_{ik}^+ \quad i \in I_k, k \in P \cup R \quad (14)$$

محدودیت‌های (۷) تا (۹) مقدار متغیرهای  $x'_{ik}, y'_{ik}, z'_{ik}$  را با توجه به متغیرهای  $x_{ik}, y_{ik}, z_{ik}$  و متغیرهای  $r_{ikab}$  که جهت‌گیری آیت  $i$  مشتری  $k$  را نشان می‌دهند، تعیین می‌کنند. محدودیت (۱۰) الزام می‌کند که هر بعد از آیت  $i$  مشتری  $k$  تنها در راستای یک محور مختصات باید قرار گیرد. محدودیت (۱۱) الزام می‌کند که در راستای هر محور مختصات تنها یک بعد از آیت  $i$  مشتری  $k$  باید قرار بگیرد. پارامترهای  $l_{ik}^+, w_{ik}^+, h_{ik}^+$  جهت‌گیری آیت  $i$  مشتری  $k$  را نشان می‌دهند و محدودیت‌های (۱۲) تا (۱۴) بیان می‌کنند که متغیرهای جهت‌گیری  $r_{ikab}$  که جهت‌گیری آیت  $i$  مشتری  $k$  را در راستای محور  $z$  نشان می‌دهد، باید از مقادیر این پارامتر کوچک‌تر باشند؛ به عبارت دیگر، اگر برای مثال  $l_{ik}^+$  برابر با صفر بود آن‌گاه طول آیت  $i$  مشتری  $k$  نمی‌تواند به‌عنوان ارتفاع قرار بگیرد. در نتیجه متغیر  $r_{ik31}$  که نشان‌دهنده‌ی قرارگیری طول آیت  $i$  مشتری  $k$  در راستای محور  $z$  است باید برابر صفر شود. با تعریف متغیرهای  $r_{ikab}$  هر شش جهت‌گیری را می‌توان برای آیت‌ها تعریف نمود. از آن‌جاکه در فرضیات در نظر گرفته شده در این مقاله تنها دو جهت‌گیری مجاز است، پارامترهای  $l_{ik}^+$  و  $w_{ik}^+$  که در صورت یک شدن نشان‌دهنده‌ی این هستند که بعد طول و عرض آیت می‌توانند به‌عنوان ارتفاع قرار بگیرند، برابر با صفر قرار داده می‌شود. در این حالت تنها بعد ارتفاع آیت‌ها می‌توان به‌عنوان ارتفاع آیت‌ها قرار گیرد و آیت‌ها می‌توانند روی قاعده خود بچرخند.

#### ۴-۳. محدودیت هم‌پوشانی

$$x'_{ikjl} + x'_{jlik} + y'_{ikjl} + y'_{jlik} + z'_{ikjl} + z'_{jlik} \quad (15)$$

$$\geq (p'_{ikv} + p'_{jlv}) - 1$$

$$t \in T, i, k \in I_k \cup I_l, k, l \in P \cup R$$

$$x'_{jl} \leq x'_{ik} + (1 - x'_{ikjl}) L \quad (16)$$

به این مدل اضافه شده‌اند. که محدودیت‌های (۲۱) و (۲۲) تعریف متغیرهای  $g_{ik}$  را نشان می‌دهند. بدین صورت که اگر آیتم  $i$  مشتری  $k$  روی کف وسیله‌نقلیه قرار بگیرد ( $z_{ik} > 0$ )،  $g_{ik}$  باید یک شود. محدودیت‌های (۲۳) تا (۲۶) تعریف متغیر  $q_{abik}$  را نشان می‌دهد؛ به عبارت دیگر، این محدودیت الزام می‌کند که تنها در صورتی متغیر  $q_{abik} < b < y'_{ik}$  و  $x_{ik} < a < x'_{ik}$  که  $q_{abik}$  برابر یک خواهد شد که  $x_{ik} < a < x'_{ik}$  و  $y_{ik} < b < y'_{ik}$  باشد. با یک شدن متغیر  $q_{abik}$ ، آیتم  $i$  مشتری  $k$  شامل نقطه  $(a, b)$  در صفحه‌ی  $XY$  می‌شود. محدودیت (۲۷) نیز الزام می‌کند، مجموع نقاطی که سطح هر آیتم شامل آن می‌شود باید دقیقاً برابر با مساحت سطح زیرین آن باشد. عبارت سمت راست مساوی، مساحت سطح زیرین را با توجه به جهت‌گیری آیتم نشان می‌دهد. شکل (۲) متغیرهای  $q_{abjl}$  را برای آیتم  $j$  مشتری  $l$  نشان می‌دهد. همان‌طور پکه در شکل مشخص است، نقاط پیرنگ روی سطح بالایی آیتم  $j$  مشتری  $l$  نقاطی هستند که این آیتم در صفحه‌ی  $XY$  شامل آن‌ها می‌شود.

محدودیت (۲۸) تا (۳۱) مقدار قدر مطلق  $|z'_{jl} - z_{ik}|$  را با استفاده از متغیر  $e_{ijkl}$  تعریف می‌کنند. اگر مقدار این قدر مطلق صفر باشد، بدین معنی است که آیتم  $j$  مشتری  $l$  ارتفاع مناسبی برای حمایت آیتم  $i$  مشتری  $k$  دارد. در این حالت محدودیت (۳۲) الزام می‌کند که متغیر  $he_{ijkl}$  برابر یک شود. در واقع محدودیت (۳۱) متغیر  $he_{ijkl}$  را با استفاده از متغیر  $e_{ijkl}$  تعریف می‌کند. محدودیت (۳۳) و (۳۴) تعریف متغیر  $SV_{klv}$  را نشان می‌دهند؛ بدین صورت که این متغیر تنها در صورتی برابر یک می‌شود که گره‌ی  $k$  و  $l$  توسط یک وسیله‌نقلیه بازدید شوند.

محدودیت‌های (۳۵) و (۳۶) تعریف متغیرهای  $\lambda'_{abikjl}$  را نشان می‌دهد. متغیرهای  $\lambda'_{abikjl}$  تنها در صورتی برابر یک می‌شود که آیتم  $i$  مشتری  $k$  و آیتم  $j$  مشتری  $l$  شامل نقطه‌ی  $(a, b)$  در صفحه‌ی  $XY$  شوند و آیتم  $j$  مشتری  $l$  ارتفاع مناسبی برای حمایت آیتم  $i$  مشتری  $k$  داشته باشد و هر دو مشتری  $l$  و  $k$  توسط یک وسیله‌نقلیه بازدید شوند و در دوره‌ی  $t$  هر دو در وسیله‌نقلیه حضور داشته باشند. محدودیت (۳۷) محدودیت پایداری را نشان می‌دهد. این محدودیت بیان می‌کند که در هر دوره اگر آیتم  $i$  مشتری  $k$  در وسیله‌نقلیه‌ی  $v$  وجود داشته باشد و روی سطح وسیله‌نقلیه قرار نداشته باشد، مجموعه نقاطی که از سطح زیرین این آیتم توسط آیتم‌های دیگر حمایت می‌شوند باید از  $\alpha$  درصد از مساحت سطح زیرین آن بیشتر باشد. در شکل (۳) این محدودیت نشان داده شده است. همان‌طور که در شکل (۳) مشخص است، آیتم  $i$  مشتری  $k$  دقیقاً روی آیتم  $j$  مشتری  $l$  قرار گرفته و فصل اشتراک آن‌ها در صفحه‌ی  $XY$  شامل شش نقطه می‌شود؛ به عبارت دیگر، آیتم  $j$  مشتری  $l$  به اندازه‌ی شش واحد سطح از آیتم  $i$  مشتری  $k$  حمایت می‌کند.

محدودیت (۳۸) محدودیت شکنندگی را نشان می‌دهد. زمانی که آیتم  $j$  مشتری  $l$  شکننده باشد ( $f_{jl} = 1$ ) تنها باید آیتم‌های شکننده

$$i \in I_k, a \in \{1, 2, \dots, L-1\}, \\ b \in \{1, 2, \dots, W-1\}, k \in P \cup R$$

$$y'_{ik} \geq (b+1)q_{abik} \quad (26)$$

$$i \in I_k, a \in \{1, 2, \dots, L-1\}, \\ b \in \{1, 2, \dots, W-1\}, k \in P \cup R$$

$$\sum_{b=0}^{W-1} \sum_{a=0}^{L-1} q_{abik} = l_{ik} w_{ik} r_{ik33} + l_{ik} h_{ik} r_{ik32} + w_{ik} h_{ik} r_{ik31} \quad (27)$$

$$i \in I_k, k \in P \cup R$$

$$z_{ik} - z'_{jl} \leq e_{ijkl} \quad i, j \in I_k \cup I_l, k, l \in P \cup R \quad (28)$$

$$z'_{jl} - z_{ik} \leq e_{ijkl} \quad i, j \in I_k \cup I_l, k, l \in P \cup R \quad (29)$$

$$z'_{jl} - z_{ik} \geq e_{ijkl} - 2H z_{ijkl}^p \quad (30)$$

$$i, j \in I_k \cup I_l, k, l \in P \cup R$$

$$z'_{jl} - z_{ik} \geq e_{ijkl} - 2H(1 - z_{ijkl}^p) \quad (31)$$

$$i, j \in I_k \cup I_l, k, l \in P \cup R$$

$$e_{ijkl} \leq (1 - he_{ijkl})H \quad (32)$$

$$i, j \in I_k \cup I_l, k, l \in P \cup R$$

$$SV_{klv} \geq \sum_{t=0}^{2n} \sum_{g=1}^{2n+1} d_{kgv}^t + \sum_{t=0}^{2n} \sum_{g=0}^{2n} d_{glv}^t - 1 \quad (33)$$

$$k, l \in P \cup R$$

$$2SV_{klv} \leq \sum_{t=0}^{2n} \sum_{g=1}^{2n+1} d_{kgv}^t + \sum_{t=0}^{2n} \sum_{g=0}^{2n} d_{glv}^t \quad (34)$$

$$k, l \in P \cup R$$

$$\lambda'_{abikjl} \geq q_{abik} + q_{abjl} + he_{ijkl} \quad (35)$$

$$+ \sum_{v=1}^V (SV_{klv} + p'_{ikv} + p'_{jlv}) - 5$$

$$i, j \in I_k \cup I_l, k, l \in P \cup R$$

$$a \in \{1, 2, \dots, L-1\}, t \in T,$$

$$b \in \{1, 2, \dots, W-1\}$$

$$6\lambda'_{abikjl} \leq q_{abik} + q_{abjl} + he_{ijkl} \quad (36)$$

$$+ \sum_{v=1}^V SV_{klv} + \sum_{v=1}^V p'_{ikv} + \sum_{v=1}^V p'_{jlv}$$

$$i, j \in I_k \cup I_l, k, l \in P \cup R$$

$$a \in \{1, 2, \dots, L-1\}, t \in T,$$

$$b \in \{1, 2, \dots, W-1\}$$

$$\sum_{a=0}^{L-1} \sum_{b=0}^{W-1} \sum_{l=1}^n \sum_{j=1}^{I_j} \lambda'_{abikjl} (g_{ik} + p'_{ikv} - 2)LW \quad (37)$$

$$\geq \alpha (l_{ik} w_{ik} r_{ik33} + l_{ik} h_{ik} r_{ik32} + w_{ik} h_{ik} r_{ik31})$$

$$i \in I_k, k \in P \cup R$$

$$q_{abik} + q_{abjl} \leq 5 - z_{ijkl}^p + f_{ik} - f_{jl} - p'_{ikv} - p'_{jlv} \quad (38)$$

$$i, j \in I_k \cup I_l, k, l \in P \cup R$$

$$a \in \{1, 2, \dots, L-1\}, t \in T,$$

$$b \in \{1, 2, \dots, W-1\}$$

محدودیت‌های (۲۱) تا (۳۷) برای تعریف محدودیت‌های پایداری



$$\sum_{v=1}^V \sum_{t=0}^{2n+1} \sum_{l=1}^{3n+1} t d_{n+k,l,v}^t - \sum_{v=1}^V \sum_{t=0}^{2n} \sum_{l=0}^{2n} t d_{k,l,v}^t \geq 1 \quad (47)$$

$$k \in P$$

$$\sum_{t=0}^{3n} \sum_{l=1}^{3n+1} d_{klv}^t - \sum_{t=0}^{3n} \sum_{l=0}^{3n+1} d_{k+n,l,v}^t = 0 \quad (48)$$

$$k \in P, v \in V$$

$$T_{lv} \geq T_{lv} + S_k + t_{kl} - (B_{2n+1} - A_0 + 1)(1 - d_{klv}^t) \quad (49)$$

$$t \in T, v \in V, k, l \in P \cup R \cup D$$

$$A_k \leq T_{kv} \leq B_k \quad (50)$$

$$k \in P \cup D \cup \{3n+1\}, v \in V$$

$$(3n+1)I_k \left( \sum_{l=1}^{3n+1} d_{k+n,l,v}^{t''} + \sum_{l=1}^{3n+1} d_{k,l,v}^{t'} - 2 \right) \quad (51)$$

$$+ I_k(t'' - t') \leq \sum_{t=t'}^{t''-1} \sum_{i=1}^{I_k} p_{ikv}^t$$

$$t, t', t'' \in T, t'' > t', i \in I_k, v \in V, k \in P$$

$$(3n+1)I_k \left( 2 - \sum_{l=1}^{3n+1} d_{k+n,l,v}^{t''} - \sum_{l=1}^{2n+1} d_{k,l,v}^{t'} \right) \geq \quad (52)$$

$$\sum_{t=0}^{t'-1} \sum_{i=1}^{I_k} p_{ikv}^t + \sum_{t=t''}^{2n+1} \sum_{i=1}^{I_k} p_{ikv}^t$$

$$t, t', t'' \in T, t'' > t', i \in I_k, v \in V, k \in P$$

$$(3n+1)I_k \left( \sum_{l=1}^{2n+1} d_{k,l,v}^{t''} + \sum_{l=1}^{2n+1} d_{l,3n+1,v}^{t'} - 2 \right) \quad (53)$$

$$+ I_k(t'' - t') \leq \sum_{t=t'}^{t''-1} \sum_{i=1}^{I_k} p_{ikv}^t$$

$$t, t', t'' \in T, t'' > t', i \in I_k, v \in V, k \in R$$

$$(3n+1)I_k \left( 2 - \sum_{l=1}^{2n+1} d_{l,2n+R+1,v}^{t''} - \sum_{l=1}^{2n+1} d_{k,l,v}^{t'} \right) \geq \quad (54)$$

$$\sum_{t=0}^{t'-1} \sum_{i=1}^{I_k} p_{ikv}^t + \sum_{t=t''}^{2n+1} \sum_{i=1}^{I_k} p_{ikv}^t$$

$$t, t', t'' \in T, t'' > t', i \in I_k, v \in V, k \in R$$

$$\sum_{t=0}^{3n} \sum_{k=n+1}^{3n} \sum_{v=1}^V d_{klv}^t = 1 \quad (55)$$

$$l \in R$$

$$\sum_{t=0}^{3n} \sum_{l=0}^{2n} \sum_{v=1}^V d_{klv}^t = 0 \quad (56)$$

$$k \in R$$

$$T_{kv}, e_{ijkl} \geq 0$$

$$x_{ik}, y_{ik}, z_{ik}, x'_{ik}, y'_{ik}, z'_{ik} \in Z^+ \cup \{0\}$$

$$u_v, p_{ikv}^t, x_{ikjl}^p, y_{ikjl}^p, z_{ikjl}^p, r_{ikrs}, d_{klv}^t \in \{0, 1\}$$

$$h_{e_{ijkl}}, g_{ik}, SV_{klv}, LS_{kl}, q_{abikl}, \lambda'_{abikjl} \in \{0, 1\} \quad (57)$$

$$\forall v \in V, \forall k, l \in P \cup R$$

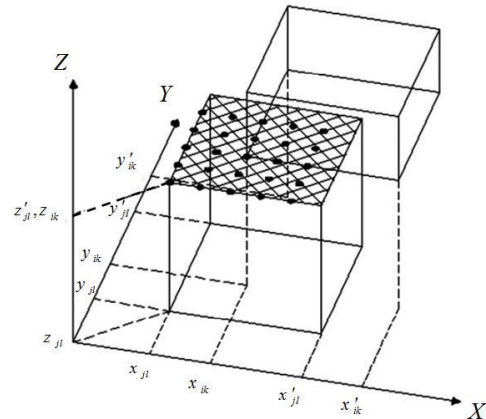
$$\forall t \in \{1, 2, \dots, 3n+1\}$$

$$\forall i \in \{1, \dots, I_k\} \forall j \in \{1, \dots, I_l\}$$

$$\forall r, s \in \{1, 2, 3\}$$

$$\forall a \in \{1, \dots, L\} \forall b \in \{1, \dots, W\}$$

روی آن قرار بگیرد. محدودیت (۳۸) بیان می‌کند زمانی که هر دو آیتیم  $i$  مشتری  $k$  و آیتیم  $j$  مشتری  $l$  در یک دوره و در یک وسیله نقلیه وجود داشته باشد و در صورتی که آیتیم  $j$  مشتری  $l$  شکننده باشد و آیتیم  $i$  مشتری  $k$  شکننده نباشد ( $f_{jl} = 0$ ) و آیتیم  $i$  مشتری  $k$  بالای آیتیم  $j$  مشتری  $l$  باشد، در این حالت آیتیم  $i$  مشتری  $k$  نمی‌تواند روی آیتیم  $j$  مشتری  $l$  قرار گیرد و لذا در هیچ نقطه  $(a, b)$  در صفحه‌ی  $XY$  نباید اشتراک داشته باشند.



شکل (۳): نمایش نقطه مشترک بین دو آیتیم  $i$  مشتری  $k$  و  $j$  مشتری  $l$  [۲۹]

۴-۵. محدودیت‌های LIFO، پنجره‌ی زمانی، مسیریابی و حدود متغیرها

$$LS_{kl} \geq (2n+2) \left( \sum_{g=0}^{3n+1} d_{kgv}^t + \sum_{g=0}^{3n+1} d_{kgv}^t - 2 \right) + 1 \quad (39)$$

$$t \in T, k, l \in P \cup D$$

$$\sum_{g=0}^{3n+1} \sum_{t=1}^{2n} t d_{k+n,g,v}^t \leq \sum_{g=0}^{3n+1} \sum_{t=1}^{3n} t d_{l+n,g,v}^t \quad (40)$$

$$+ (2n+2)(2 - LS_{lk} - LS_{k,l+n})$$

$$t \in T, k, l \in P \cup D$$

$$(2n+2)(LS_{kl} + LS_{l,k+n} - 2) + 1 \leq x_{jlik}^p + z_{jlik}^p + y_{jlik}^p + y_{ikjl}^p \quad (41)$$

$$i, j \in I_k \cup I_l, k, l \in P \cup R$$

$$\sum_{v=1}^V \sum_{t=0}^{3n} \sum_{l=1}^{3n+1} d_{klv}^t = 1 \quad (42)$$

$$k \in P \cup D \cup R$$

$$\sum_{v=1}^V \sum_{t=0}^{3n} \sum_{l=1}^{3n+1} t d_{klv}^t - \sum_{v=1}^V \sum_{t=0}^{3n-1} \sum_{l=0}^{3n} t d_{lkv}^t = 1 \quad (43)$$

$$k \in P \cup D \cup R$$

$$\sum_{l=1}^{3n+1} d_{klv}^{t+1} - \sum_{l=0}^{3n} d_{lkv}^t = 0 \quad (44)$$

$$K \in V_r / R$$

$$\sum_{l=0}^n d_{0lv}^0 = u_v \quad (45)$$

$$v \in V$$

$$\sum_{t=1}^{3n} \sum_{k=2n+1}^{3n} d_{k,3n+1,v}^t = u_v \quad (46)$$

$$v \in V$$

در این صورت مجموع  $\sum_{l=0}^{3n+1} d_{k+n,l}^i$  و  $\sum_{l=1}^{3n+1} d_{klv}^i$  برابر صفر خواهد بود؛ آنگاه تمام متغیرهای  $p_{ikv}^i$  که  $t$  آن‌ها در بازه‌ی  $[t^-, t^+]$  وجود دارد، برابر با یک شوند.  $t''$  جزئی از بازه نیست، زیرا در این دوره از گره‌ی تحویل  $k+n$  به گره‌ی دیگری سفر می‌شود؛ در این حالت دیگر آیتم‌های مربوط به مشتری  $k$  در وسیله‌نقلیه‌ی  $v$  وجود نخواهند داشت و در گره‌ی  $k+n$  تخلیه شده‌اند. محدودیت (۵۲) نیز تضمین می‌کند که در صورت بازدید گره تقاضای  $k$ ام در دوره‌ی  $t''$  و گره‌ی دریافت مربوط به آن در دوره‌ی  $t'$  تمام متغیرهای  $p_{ikv}^i$  که  $t$  آن‌ها قبل از  $t'$  و یا بعد از  $t''$  وجود دارند، برابر صفر شوند. در این دو محدودیت عبارت  $(2n+1)I_k$  نقش عدد به اندازه‌ی کافی بزرگ مثبت را بازی می‌کند؛ در واقع، این عبارت تعداد متغیرهای  $p_{ikv}^i$  را به ازای هر مشتری و هر وسیله‌نقلیه نشان می‌دهد. محدودیت‌های (۵۱) و (۵۲) برای یکپارچه‌سازی مدل بارگذاری و مسیر رفت استفاده شده است. محدودیت (۵۳) مانند محدودیت (۵۱) و محدودیت (۵۴) مانند محدودیت (۵۲) می‌باشد که برای آیتم‌های بازگشتی نوشته شده است. محدودیت (۵۵) الزام می‌کند که تنها از گره‌ی تحویل و یا بازگشت می‌توان به گره‌ی بازگشت رفت و محدودیت (۵۶) نیز الزام می‌کند که تنها از گره‌های بازگشت می‌توان به گره‌ی بازگشت دیگر و یا گره‌ی  $3n+1$  رفت.

#### ۴-۶. بهبود فرمول‌بندی مدل ریاضی براساس متغیر نشان‌دهنده‌ی دوره‌ی بازدید هر گره

در تعریف متغیرهای مربوط به مسیریابی از اندیس  $t$  که نشان‌دهنده‌ی دوره است، استفاده شده است. اندیس  $t$  از این جهت به این متغیرها افزوده شده که مشخص شود در هر دوره چه گره‌ای بازدید شده و آیتم‌های هر مشتری برای چه دوره‌هایی درون وسیله‌نقلیه هستند. با دانستن این مطلب، مسأله بارگذاری نحوه‌ی چیدمان آیتم‌هایی که در هر دوره درون وسیله‌نقلیه هستند را مشخص می‌کند.

در این حالت با تعریف  $n$  مشتری، تعداد  $3n+2$  گره وجود دارد که  $n$ تای آن‌ها مربوط به گره‌های دریافت،  $n$ تای آن‌ها مربوط به گره‌ی تحویل و  $n$ تای آن‌ها گره‌های بازگشت و دو گره‌ی باقی‌مانده نشان‌دهنده‌ی انبار ابتدایی و پایانی هستند. از طرفی تعداد دوره‌ها حداکثر برابر با تعداد گره‌ها منهای یک  $3n+1$  است. با این تعاریف، تعداد متغیرهای  $d_{klv}^i$  که برای مسیریابی تعریف شده، برابر با  $(3n+1)V(3n+2)^2$  خواهد بود که در آن  $V$  تعداد کل وسایل نقلیه است. از طرفی در مسأله مسیریابی برای هر گره‌ی  $k$ ،  $(3n+1)V(3n+2)$  متغیر تعریف می‌شود که با توجه به محدودیت (۴۲) در میان آن‌ها تنها یکی برابر یک است؛ لذا  $(3n+1)V - 1$  متغیر باقی‌مانده برابر صفر خواهد بود. در این بخش سعی می‌شود، با تعریف متغیر جدیدی، اندیس  $t$  از متغیر  $d_{klv}^i$  حذف شده و در ازای اضافه کردن متغیرهای جدید، متغیرهای بیشتری از مدل حذف شوند. از طرفی با حذف اندیس  $t$  از متغیر  $d_{klv}^i$  تعداد متغیرها برای هر گره‌ی  $k$  به  $(3n+2)^2V$  کاهش یابد.

محدودیت‌های (۳۹) تا (۴۱) محدودیت  $LIFO$  را نشان می‌دهد. محدودیت (۳۹) تعریف متغیر  $LS_{kl}$  را نشان می‌دهد. متغیر  $LS_{kl}$  برابر یک می‌شود اگر دوره‌ی که در آن گره‌ی  $l$  بازدید می‌شود بزرگ‌تر از دوره‌ای باشد که در آن گره‌ی  $k$  بازدید می‌شود. محدودیت (۴۰) بیان می‌کند که اگر دوره‌ای که در آن گره‌ی  $k$  بازدید می‌شود بزرگ‌تر از دوره‌ای باشد که در آن گره‌ی  $l$  بازدید شده و همچنین کوچک‌تر از دوره‌ای باشد که در آن گره‌ی  $l+n$  بازدید شده، در این صورت گره‌ی تحویل  $k$ ، یعنی گره‌ی  $k+n$  نیز باید قبل از گره‌ی  $l+n$  بازدید شود. محدودیت (۴۱) بیان می‌کند که اگر گره‌ی  $l$  بعد از گره‌ی  $k$  و قبل از گره‌ی  $k+n$  در مسیر قرار داشت، آنگاه آیتم‌های مشتری  $k$  نباید بالای آیتم‌های مشتری  $l$  و جلوی آن‌ها باشند؛ بدین منظور باید حداقل یکی از متغیرهای  $y_{jlk}, z_{jlk}, x_{ikjl}, y_{jlk}$  برابر یک شوند.

محدودیت (۴۲) بیان می‌کند که هر گره تنها توسط یک وسیله‌نقلیه و تنها در یک دوره بازدید شود. محدودیت (۴۳) الزام می‌کند که اگر در دوره‌ای کانتینر به گره‌ای وارد شود باید در دوره‌ی بعد تنها یک‌بار از همان گره خارج شود. محدودیت (۴۴) پیوستگی تور را بیان می‌کند؛ به این صورت که اگر در دوره‌ی  $t$  از گره‌ی  $l$  به گره‌ی  $k$  سفر شود باید در دوره‌ی  $t+1$  از گره‌ی  $k$  به گره‌ی دیگری مانند  $l$  سفر شود. محدودیت (۴۵) بیان می‌کند که در دوره‌ی صفر هر وسیله‌نقلیه سفر خود را از انبار شروع می‌کند و تنها وسیله‌نقلیه‌ای سفر خود را از انبار شروع می‌کند که در مدل از آن‌ها استفاده شده باشد؛ به عبارت دیگر، اگر یک وسیله‌نقلیه استفاده شود آنگاه  $u_p = 1$  خواهد بود و طرف راست محدودیت (۴۵) برابر یک می‌شود در این حالت محدودیت الزام می‌کند که این وسیله‌نقلیه سفر خود را از انبار شروع می‌کند. محدودیت (۴۶) مانند محدودیت (۴۵) الزام می‌کند که هر وسیله‌نقلیه که مورد استفاده قرار گرفته است در انتهای مسیر خود به انبار برگردد. محدودیت (۴۷) الزام می‌کند که گره تحویل یک تقاضا بعد از گره‌ی دریافت آن بازدید شوند. محدودیت (۴۸) الزام می‌کنند که گره‌های دریافت و تحویل یک تقاضا توسط یک وسیله‌نقلیه بازدید شوند. محدودیت (۴۹) الزام می‌کند که اگر  $l$  بعد از گره‌ی  $k$  توسط وسیله‌نقلیه‌ی  $v$  بازدید شد آنگاه زمان شروع خدمت گره‌ی  $l$  باید بعد از زمان اتمام خدمت گره‌ی  $k$  و زمان سفر از گره‌ی  $k$  به گره‌ی  $l$  باشد. در این محدودیت عبارت  $(B_{3n+1} - A_0 + 1)$  نقش عدد بزرگ مثبت را بازی می‌کند. در این جا فرض شده است که پنجره‌ی زمانی مربوط به گره‌ی انبار که د واقع گره‌ی صفر و گره‌ی  $3n+1$  نشان‌دهنده‌ی آن هستند، بزرگ‌ترین بازه را دارا هستند؛ لذا همواره اختلاف زمان شروع دو گره از این مقدار باید کمتر باشد. محدودیت (۵۰) الزام می‌کند که زمان شروع خدمت هر گره در پنجره‌ی زمانی مربوط به آن است. محدودیت (۴۲) تا (۴۹) از مدل ارائه شده توسط جانکوئرا و همکاران [۳۰] الهام گرفته شده است.

محدودیت (۵۱) بیان می‌کند که اگر گره‌ی تحویل تقاضای  $k$ ام در دوره‌ی  $t''$  و گره‌ی دریافت مربوط به آن در دوره‌ی  $t'$  بازدید شد که

هر تور فراخوانی کنید. اولین باری که بارگذاری امکان‌پذیر پیدا شد، موقعیت قرارگیری گره‌ی  $P$  و  $D$  مشتری مورد بررسی را در موقعیت امکان‌پذیر ثابت کنید و به قدم ۷ بروید. در غیر این صورت یک عدد کانتینر اضافه کنید و گره‌ی  $P$  و  $D$  مشتری  $k$ ام گام ۲ را در آن قرار دهید.

قدم ۷: قرار دهید  $k=k+I$ . اگر مقدار  $k > n$  بود توقف کنید و در غیر این صورت لیست  $m$  را تهی و به قدم ۴ بروید.

قدم ۸: در تمام تورهای موجود بعد از آخرین گره‌ی تحویل، گره‌ی بازگشت همان مشتری را قرار دهید و گره‌های بازگشت طی نشده را به ترتیب کران پایین پنجره‌ی زمانی آن‌ها به صورت غیر نزولی در لیست  $r$  مرتب کنید و طول آن‌ها را  $R$  بنامید و قرار دهید  $k=I$ .

قدم ۹: گره‌ی بازگشت  $k$ ام لیست  $r$  را در تمام موقعیت‌های امکان‌پذیر از نظر محدودیت‌های مسیریابی قرار دهید. هر تور امکان‌پذیر از نظر محدودیت‌های مسیریابی را به همراه میزان تابع هدف کل تورها در لیست  $m$  قرار دهید.

قدم ۱۰: لیست  $m$  را برای گره‌های بازگشت مشتری‌ها براساس میزان تابع هدف به صورت غیر نزولی مرتب کنید.

قدم ۱۱: تابع بارگذاری را از اول لیست  $m$  برای هر تور فراخوانی کنید. اولین باری که بارگذاری امکان‌پذیر پیدا شد، موقعیت قرارگیری گره‌ی  $R$  را ثابت کنید. در غیر این صورت یک عدد کانتینر اضافه کنید. گره‌های  $P$  و  $D$  مشتری  $k$ ام را از جواب فعلی حذف کنید و هر سه گره‌ی مشتری  $k$ ام را در کانتینر جدید قرار دهید. اگر محدودیت ظرفیت برای گره‌ها در کانتینر جدید برقرار بود از کانتینر کوچک و در غیر این صورت از کانتینر بزرگ استفاده کنید.

قدم ۱۲: قرار دهید  $k=k+I$  اگر  $k$  بزرگ‌تر از  $R$  باشد توقف و در غیر این صورت به قدم ۹ بروید.

#### ۵-۲. الگوریتم ابتکاری بارگذاری

برای آیتیم‌های دریافت و آیتیم‌های بازگشت باید محدودیت‌های بارگذاری سه‌بعدی بررسی شده و برقراری این محدودیت‌ها برای همه‌ی آیتیم‌ها تأیید شود؛ بدین منظور از روش ابتکاری مبتنی بر نقاط مرجع استفاده شده است. نقاط مرجع  $EP^2$  اولین مرتبه توسط کرانیک و همکاران [۳۱] ارائه شد که این ایده در شکل (۴) نشان داده شده است.

برای این منظور متغیرهای صفر و یک  $d_{klv}$  و  $R_{kvt}$  به صورت زیر تعریف می‌شود که در این حالت مرتبه‌ی تعداد متغیرها از  $n^3$  به  $n^2$  کاهش می‌یابد.

اگر وسیله‌نقلیه‌ی  $v$  در دوره‌ی  $t$  گره‌ی  $k$  را بازدید کند مقدار یک و در غیر این صورت مقدار صفر می‌گیرد.

$$k \in P \cup D \cup R, v \in V, t \in T$$

اگر وسیله‌نقلیه‌ی  $v$  از گره‌ی  $k$  به گره‌ی  $l$  سفر کند مقدار یک و در غیر این صورت مقدار صفر می‌گیرد.

$$k, l \in P, D, R \quad v \in V$$

#### ۵. روش‌های حل ابتکاری و فراابتکاری ارائه شده

تجربه‌ی نویسندگان مقاله نشان داده که در این مسأله مدل‌های ریاضی تنها قادر به حل نمونه‌های با ابعاد کوچک هستند؛ لذا در این بخش برای حل مسائل در ابعاد واقعی و بزرگ یک الگوریتم ابتکاری و دو الگوریتم فراابتکاری بر پایه‌ی جست‌وجوی ممنوع و جست‌وجوی همسایگی متغیر ارائه شده است. در ادامه، ابتدا اجزاء الگوریتم‌ها معرفی شده و سپس دو الگوریتم فراابتکاری ارائه خواهند شد.

#### ۵-۱. الگوریتم ابتکاری مسیریابی

در این مقاله، یک روش ابتکاری به نام «یک مشتری در هر زمان»<sup>۱</sup> (OCT) ارائه شده که یک جواب خوب در مدت‌زمان کم ارائه می‌دهد؛ همچنین از این جواب به عنوان جواب اولیه‌ی روش‌های فراابتکاری استفاده شده است. قدم‌های این الگوریتم در ادامه آمده است.

قدم ۱: کران پایین تعداد وسایل نقلیه را از تقسیم مجموع وزن کل آیتیم‌های مشتری‌ها بر ظرفیت وسیله‌نقلیه‌ی بزرگ به دست آورید.

قدم ۲:  $n$  مشتری را براساس کران پایین پنجره‌ی زمانی گره‌ی دریافتشان به صورت غیر نزولی مرتب و در لیست  $C$  قرار داده و قرار دهید  $k=I$ ؛ هم‌چنین گره‌های دریافت و تحویل هر مشتری را به ترتیب با  $P$  و  $D$  نشان دهید.

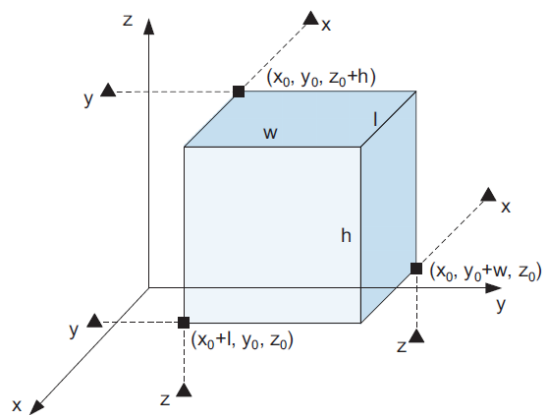
قدم ۳: آیتیم‌های گره‌های  $P$  و  $D$  مشتری اول لیست  $C$  را در اولین کانتینر قرار دهید و  $k$  را یک واحد افزایش دهید. اگر با کوچک در نظر گرفتن کانتینر محدودیت ظرفیت برای گره‌ی  $P$  برقرار بود کانتینر را کوچک و در غیر این صورت کانتینر را بزرگ در نظر بگیرید.

قدم ۴: گره‌ی  $P$  مشتری در موقعیت  $k$ ام از لیست  $C$  را در تمام موقعیت‌های امکان‌پذیر قرار دهید و به ازای هر موقعیت از  $P$ ، گره‌ی  $D$  مشتری  $k$ ام را در تمام موقعیت‌های امکان‌پذیر از نظر محدودیت‌های مسیریابی در تمام تورها قرار دهید. هر تور امکان‌پذیر از نظر محدودیت‌های مسیریابی را به همراه میزان تابع هدف کل تورها در لیست  $m$  قرار دهید.

قدم ۵: لیست  $m$  را به صورت غیر نزولی براساس تابع هدف مرتب کنید.

قدم ۶: تابع بارگذاری که با استفاده از آن بارگذاری امکان‌پذیری تورها از نظر محدودیت‌های بارگذاری بررسی می‌شود را از اول لیست  $m$  برای

شکل (۴): نقاط EP حاصل از بارگذاری یک آیتم [۳۱]  
 حسن‌آبادی [۲۳] این روش ابتکاری بارگذاری را برای مسیریابی دریافت و تحویل یک‌به‌یک در نظر گرفت که شبه‌کد آن در شکل (۵) نشان داده شده است. در این مقاله نیز از این روش استفاده شده و برای مسیریابی دریافت، تحویل و بازگشت تعمیم داده شده است.



**Loading\_Heuristic** (in: *Customers*, Loading-Plan:*L-P*, first customer to be loaded: *j*, out: *Status*)

Set *Status* to False.

Sort *Customers* based on visiting sequence.

**For**  $i+1 \leq rq \leq |Customers|$  **Do**

Sort boxes of *Customers*[*rq*] by decreasing volume.

Set *NumLoad*[*rq*] to 0.

**If**  $2 \leq I[Customers[*rq*]] \leq 4$  **then**

Set *NumChaOrd*[*rq*] = factorial(*I*[*Customers*[*rq*]]).

**If** *I*[*Customers*[*rq*]] = 1 **then**

Set *NumChaOrd*[*rq*] = 0.

**If** *I*[*Customers*[*rq*]] = 5 **then**

Set *NumChaOrd*[*rq*] = 36.

**EndIf**

**EndFor**

Set *rq* to *j* and Set *i* to *j*.

Set *L-P1* to *L-P*.

**If** *L-P* =  $\emptyset$  **then**

Add (0,0,0) to *EpSet*.

**EndIf**

**While** *Status* is False **And**  $i \leq |Customers|$  **Do**

**While**  $rq \leq |Customers|$

**For** all  $1 \leq b \leq I[Customers[*rq*]]$  **Do**

*Status* := **LoadInFirstEp**(*EpSet*, *b*, *rq*, *L-P*).

**If** *Status* is False **then**

**breakFor**

**Else**

Update *EpSet*.

**EndIf**

**EndFor**

**If** *Status* is False **then**

**If** *NumLoad*[*i*]  $\leq$  *NumChaOrd*[*i*] **then**

Change order of boxes of *Customers*[*i*].

Set *NumLoad*[*i*] = *NumLoad*[*i*] + 1.

**Else**

Sort boxes of *Customers*[*i*] by decreasing volume.

Set  $i = i + 1$ .

**EndIf**

Set *rq* to *j*.

Set *L-P* to *L-P1*.

Update *EpSet*.

**breakWhile**

**EndIf**

Set  $rq = rq + 1$ .

**EndWhile**

**EndWhile**

شکل (۵): شبه‌کد الگوریتم ابتکاری بارگذاری [۲۳]

از روش‌های تخریب و سازنده استفاده می‌شود. در روش‌های تخریب برای به‌دست آوردن جواب جدید، تعداد مشخصی مشتری از جواب فعلی و یا به‌عبارت دیگر، از تورهای فعلی حذف شده و به لیست تخریب اضافه می‌گردد؛ سپس با استفاده از روش‌های سازنده، مجدداً

### ۳-۵. عملگرهای ساخت همسایه

در این مقاله مسیر از انبار شروع و به انبار پایان می‌یابد و توسط یک کانتینر طی می‌شود که به آن «تور» گفته می‌شود؛ بنابراین یک جواب امکان‌پذیر شامل تعدادی تور (کانتینر) می‌باشد. برای ایجاد همسایگی

قرارگیری گره‌های دریافت و تحویل این مشتری‌ها تغییر نمی‌کند. روش تخریب تصادفی تور: ( $randT\_wrck^5$ ) در این روش، سعی می‌شود تمام  $k$  مشتری از یک تور حذف شود؛ بنابراین اگر تعداد مشتری‌های  $P, D, R$  برابر  $k$  و یا کمتر باشد گره‌های دریافت و تحویل این مشتری‌ها با گره‌های بازگشتشان در کلیه توره‌ها حذف می‌شوند؛ همچنین اگر تعدادی گره‌ی بازگشت هنوز در این تور باشد آن‌ها نیز حذف خواهند شد؛ بنابراین در این حالت، کل این تور حذف می‌گردد. اگر تعداد مشتری با  $P, D, R$  بیشتر از  $k$  باشد به اندازه‌ی  $k$  مشتری و به‌صورت تصادفی کلیه‌ی گره‌های این مشتری‌ها از توره‌ها حذف شده و به لیست تخریب اضافه می‌گردد.

روش تخریب بدترین هزینه: ( $WC\_wrck^6$ ) در این روش هزینه‌ی عدم وجود هر مشتری در جواب فعلی که اختلاف هزینه‌ی حضور و عدم حضور تمام گره‌های آن مشتری در جواب فعلی هست، حساب می‌شود. تمام گره‌های  $k$  مشتری که هزینه‌ی عدم وجود آن‌ها از بقیه‌ی مشتری‌ها بیشتر است از توره‌ها حذف شده و به لیست تخریب اضافه می‌شوند [۲۳].

روش تخریب مشتری کانتینرهای بزرگ: ( $CBC\_wrck^7$ ) از آنجایی که کانتینر بزرگ هزینه‌ی بیشتری دارد، لذا این روش تخریب با هدف تبدیل کانتینرهای بزرگ به کوچک به دنبال کاهش تابع هدف بوده است؛ لذا در این روش سعی می‌شود گره‌های مربوط به  $k$  مشتری از تورهایی که با کانتینرهای بزرگ انجام می‌شوند به‌صورت تصادفی حذف شده و به لیست تخریب اضافه شوند.

بعد از حذف گره‌های مشتری‌ها و اضافه شدن آن‌ها به لیست تخریب باید آن‌ها را با استفاده از روش‌های سازنده به توره‌ها و در موقعیت‌های امکان‌پذیر برای به‌دست آوردن جواب جدید اضافه کرد. در بعضی روش‌های سازنده، مفهومی تحت عنوان شناوری مطرح شده است که قبل از بیان روش‌های سازنده به بیان آن پرداخته می‌شود.

#### ۵-۵. شناوری

به‌مدت زمان انتظار و بیکاری کانتینر و یا عدم استفاده از تمام زمان در اختیار به هر دلیل، شناوری گفته می‌شود. هر کانتینر اگر زودتر از بازه‌ی زمانی هر گره به آن برسد باید تا فرارسیدن زمان مجاز منتظر بماند. این انتظار یکی از دلایل بیکاری کانتینر است. برای این انتظارها در این مقاله سه نوع مختلف شناوری تعریف شده است.

شناوری تور: ( $tour\_float^8$ ) این شناوری صرفاً برای گره‌ی انبار در نظر گرفته شده است. این شناوری در واقع مدت‌زمانی است که یک کانتینر زودتر از دیرترین زمان ممکن ورود به انبار به آن وارد می‌شود. این شناوری از رابطه‌ی ۵۸ برای هر تور به‌دست می‌آید.

(۵۸) زمان رسیدن کانتینر به گره‌ی انبار - دیرترین زمان ورود به

گره‌ی انبار = شناوری تور

موقعیت جدیدی برای همین مشتری‌های حذف شده در توره‌ها در نظر گرفته می‌شود. در صورت عدم وجود موقعیت امکان‌پذیر برای هر کدام از گره‌ها با اضافه کردن یک کانتینر، تور جدیدی به جواب اضافه می‌شود. کانتینر جدید ابتدا کوچک در نظر گرفته شده و اگر محدودیت‌های بارگذاری و مسیریابی در آن برقرار نباشد از کانتینر بزرگ استفاده می‌شود. تعداد مشتری‌های حذف شده عددی تصادفی در بازه‌ی  $[4a, nb]$  است که با  $k$  نشان داده می‌شود [۲۳]. در این رابطه  $n$  تعداد مشتری‌ها و  $a, b$  پارامترهای ورودی مسأله هستند که از طریق تنظیم پارامتر تاگوچی به‌دست می‌آیند.

#### ۵-۴. روش‌های تخریب

برای ایجاد همسایگی، گره‌های مربوط به  $k$  مشتری از جواب فعلی حذف می‌شوند. از آنجایی که نباید در هیچ توری تنها گره‌ی بازگشتی وجود داشته باشد، در تمام روش‌های تخریب اگر بعد از حذف مشتری‌ها از یک تور، تنها گره‌ی بازگشتی در آن تور باقی‌مانده باشد، تمام گره‌های بازگشت در این تور نیز به لیست تخریب اضافه شده و تور به‌طور کامل حذف می‌شود؛ در ادامه به توضیح هر کدام از روش‌های تخریب پرداخته شده است.

روش تخریب تصادفی: ( $rand\_wrck^1$ ) در این روش  $k$  مشتری به‌صورت تصادفی انتخاب شده و تمام گره‌های مربوط به آن‌ها اعم از دریافت، تحویل و بازگشت حذف شده و به لیست تخریب اضافه می‌شوند. تفاوت این روش تخریب با ادبیات موضوع، حذف شدن گره‌های بازگشتی از داخل توره‌ها می‌باشد که در این روش تخریب این گره‌ها نیز مانند گره‌های دریافت و تحویل به‌صورت تصادفی حذف می‌شوند [۲۳].

روش تخریب تور با هزینه‌ی بالا: ( $MCT\_wrck^2$ ) در این روش ابتدا هزینه‌ی هر تور محاسبه شده و این هزینه بر مجموع تعداد گره‌های دریافت، تحویل و بازگشت موجود در این تور تقسیم می‌شود. بدین ترتیب برای هر تور مقدار متوسط هزینه به ازای هر گره به‌دست می‌آید. سپس  $k$  مشتری با بالاترین هزینه‌ی متوسط، حذف شده و تمام گره‌های مربوط به آن‌ها به لیست تخریب اضافه می‌شود [۲۳].

روش تخریب تور تک مشتری: ( $TOC\_wrck^3$ ) هدف از این روش کمینه کردن تعداد کانتینرهای مورد استفاده است. برای این منظور در صورت وجود تور تک‌مشتری، از بین آن‌ها یک تور به‌صورت تصادفی انتخاب شده و تمام گره‌های مشتری این تور حذف شده و به لیست تخریب اضافه می‌شوند [۳۲]. مابقی  $k-I$  مشتری نیز به‌صورت تصادفی از سایر توره‌ها حذف شده و به لیست تخریب اضافه می‌شوند. روش تخریب گره‌های بازگشت: ( $RN\_wrck^4$ ) در این روش تنها گره‌ی بازگشت  $k$  مشتری به‌صورت تصادفی از جواب فعلی حذف شده و این گره‌ها به لیست تخریب اضافه می‌شوند. در این روش، موقعیت

4. Recursion Node wreckage

5. Random Tour wreckage

6. Worst Cost wreckage

7. Customer in Big Container wreckage

8. Tour floating

1. Random Wreckage

2. Max Cost in Tour wreckage

3. Tour Only Customer wreckage

شناوری چند بخشی: ( $MP\_f1ot^1$ ) مقدار تفاوت بین زودترین زمان ورود به انبار و دیرترین زمان ورود به سه بازه‌ی مساوی تقسیم می‌شود. برای هر گره زودترین زمان ورود هر گره در یکی از این سه بازه قرار می‌گیرد و می‌توان برای هر گره زمان شناوری مربوط به آن بازه را تعریف نمود. مدت‌زمان شناوری چند بخشی براساس اختلاف زمان رسیدن کانتینر به گره و زودترین زمان ورود به آن گره محاسبه می‌شود.

شناوری مجموع: ( $sum\_f1ot^2$ ) اگر بخش‌های مختلف شناوری چند بخشی یک تور با هم جمع شود، شناوری مجموع برای آن تور به‌دست می‌آید. درواقع شناوری مجموع، کل زمان انتظار کانتینر از زمان خروج آن از انبار تا زمان ورود آن به انبار در گره‌های مختلف می‌باشد.

#### ۵-۶. روش‌های سازنده

با استفاده از روش‌های سازنده گره‌های موجود در لیست تخریب به تورها برای تولید جواب جدید اضافه می‌شوند. این روند تا پیدا کردن موقعیت‌های امکان‌پذیر برای تمام گره‌های موجود در لیست تخریب ادامه دارد. ضمناً برای موقعیت هر گره‌ی یک مشتری باید تمام محدودیت‌ها از جمله محدودیت‌های بارگذاری برقرار باشد. اگر برای گره‌ی دریافت و تحویل مشتری موقعیت امکان‌پذیر پیدا نشود، با اضافه کردن یک کانتینر تور جدیدی به جواب اضافه می‌شود. اگر برای گره‌ی بازگشت مشتری موقعیت امکان‌پذیر پیدا نشود، از آنجایی که نباید در یک تور تنها گره‌ی بازگشت وجود داشته باشد، در این حالت گره‌ی دریافت و تحویل یک مشتری به‌صورت تصادفی از یک تور حذف شده و در تور جدید قبل از گره‌ی بازگشت قرار داده می‌شود؛ در ادامه به توضیح روش‌های سازنده پرداخته خواهد شد.

روش سازنده‌ی حریصانه: ( $grd\_mak^3$ ) در این روش، یک گره به‌صورت تصادفی از لیست تخریب انتخاب شده و بین تمام گره‌های امکان‌پذیر در تمام تورها جایگذاری شده و موقعیت با کم‌ترین هزینه انتخاب می‌شود [۳۳]. این روند تا زمانی که تمام گره‌های لیست تخریب به تورها اضافه شوند، ادامه دارد. لازم به ذکر است در ادبیات موضوع گره‌ی بازگشت وجود نداشته، ولی در این مقاله باید موقعیت گره‌های بازگشتی را هم پیدا نمود.

روش حریصانه‌ی تصادفی: ( $rand\_grd\_mak^4$ ) این روش مشابه روش سازنده‌ی حریصانه است و تفاوت آن در این است که مقدار تابع هدف جواب جدید در یک عدد تصادفی در بازه‌ی  $[0.8, 1]$  ضرب می‌شود تا جواب به‌دست آمده در مراحل بعدی الگوریتم بیشتر مورد توجه قرار گیرد و از آن برای ایجاد همسایگی استفاده شود [۲۳].

روش سازنده‌ی گره‌ی بازگشت: ( $RN\_mak^5$ ) همواره گره‌های بازگشت در انتهای تورها قرار داشته و هم‌چنین این گره‌ها آخرین گره‌های طی شده قبل از انبار هستند. برای قرار دادن مجدد گره‌های بازگشت مشتری در تورها از رویکرد شناوری تورها برای گره‌ی انبار

روش سازنده‌ی شناوری مجموع: ( $sum\_flt\_mak^7$ ) به‌دلیل این‌که احتمال قرارگیری مشتری‌ها در تورهایی با مجموع شناوری بیشتر زیادتار است، در این روش، گره‌ها به‌صورت تصادفی انتخاب می‌شوند و در تمام موقعیت‌های تورهایی با مجموع شناوری بالاتر قرار داده می‌شوند تا در موقعیت‌های امکان‌پذیر از نظر محدودیت‌های مسیریابی و بارگذاری قرار گرفته شوند.

#### ۵-۷. روش فراابتکاری جست‌وجوی ممنوع

روش فراابتکاری TS برای اولین بار در سال ۱۹۸۶ توسط گلوور [۳۴] ارائه شد. این الگوریتم در چندین مقاله‌ی مسیریابی استفاده شده که نتایج خوبی داشته است [۱۰، ۹]؛ لذا برای حل مسأله مسیریابی دریافت، تحویل و بازگشت از این روش استفاده شده است. در این مقاله نیز یک الگوریتم جست‌وجوی ممنوع با نام PDBP\_TS ارائه می‌شود که پارامترهای و شبه‌کد آن به‌ترتیب در جدول (۳) و شکل (۶) نشان داده شده است.

جدول (۳): پارامترهای الگوریتم PDBP\_TS

پارامترهای PDBP_TS	توضیحات
<i>I</i> ter	تعداد کل تکرار
<i>ConsI</i> ter	تعداد تکرار بدون بهبود
<i>MaxConsI</i> ter	حداکثر تکرار بدون بهبود
<i>Tabu_Leng</i> ht	طول لیست ممنوع
<i>Tabu_Leng</i> ht_Num	تعداد ردیف پر شده از لیست ممنوع
<i>iter_adaptive</i>	تعداد تکرار با رویکرد تطابق
<i>iter_adaptive_start</i>	شروع تکرار روند تطابق
$f_{new}$	جواب جدید
$f_{current}$	جواب در دست

5. Recursion Node Maker

6. Multi Part floating Maker

7. Sum floating maker

1. Multi Part floating

2. Sum floating

3. Greedy Maker

4. Random Greedy Maker

محدودیت‌های بارگذاری فقط برای تورهایی که در آن‌ها موقعیت گره‌ها تغییر کرده است، بررسی شود که به این تورها Tour Changed گفته می‌شود.

اگر جواب جدید در لیست ممنوع قرار داشت الگوریتم، معیار تنفس<sup>۱</sup> را بررسی خواهد کرد. بر اساس این معیار اگر جواب جدید ( $f_{new}$ ) از بهترین جواب ( $f_{best}$ ) بهتر باشد، آنگاه بهترین جواب ( $f_{best}$ ) مساوی جواب جدید ( $f_{new}$ ) قرار داده می‌شود. پس از به‌روزرسانی بهترین جواب، لیست ممنوع نیز به‌روزرسانی می‌شود؛ به این معنا که ویژگی‌هایی از جواب جدید که اکنون برابر با بهترین جواب است در لیست ممنوع قرار داده می‌شود تا حد ممکن از بازگشت مجدد الگوریتم به آن جواب و ایجاد حلقه جلوگیری شود. در واقع، لیست ممنوع ابزاری در الگوریتم جست‌وجوی ممنوع است که توسط آن از قرار گرفتن الگوریتم در بهینه‌ی محلی تا حدود زیادی جلوگیری می‌شود. تکرار الگوریتم تا جایی ادامه می‌یابد که شرط خاتمه دیده شود.

بهترین جواب

$f_{best}$

تعداد همسایگی در هر رویکرد همسایه‌سازی

Num\_nigh

برای رسیدن به جواب‌های خوب، الگوریتم PDBP\_TS ابتدا از یک جواب اولیه که توسط الگوریتم OCT تولید شده شروع به حرکت می‌کند؛ سپس با استفاده از روش‌های تخریب و سازنده، تعدادی همسایه برای جواب فعلی تولید می‌کند. الگوریتم، بهترین جواب را از میان همسایه‌های جواب فعلی تحت عنوان «جواب جدید» ( $f_{new}$ ) انتخاب می‌کند. در  $f_{new}$  باید تمام محدودیت‌ها از جمله محدودیت‌های بارگذاری رعایت شده باشد. در صورتی که جواب جدید در لیست ممنوع قرار نداشته باشد، با بهترین جواب ( $f_{best}$ ) مقایسه می‌شود. اگر جواب جدید از بهترین جواب کمتر بود آنگاه بهترین جواب مساوی جواب جدید قرار داده می‌شود. در هر تکرار همه‌ی موقعیت‌های هر گره دستخوش تغییر نمی‌شوند و تنها موقعیت قرارگیری چند گره عوض می‌شود؛ بنابراین منطقی است که

```

1 Input the data and paramtrs
2 PDBP_TS(input OCT , output  $f_{best}$  )
3 sets heuristic solution OCT as  $f_{current}$  and  $f_{best}$  and Tabu_Lenght_Num=0 and status =false and ConsIter; iter=0;
4 While ( ConsIter < MaxConsIter )
5     Generate Num_nigh neighborgoods for  $f_{current}$  by method Wreckage and maker and Loading(inputTour
   changed,output Status)
6     If  $f_{new} < f_{current}$  and  $f_{new}$  not tabu and Status is true then
7         , update the tabu list and  $Iter++$ ;  $f_{current} = f_{new}$ 
8     If (Tabu_Lenght_Num < Tabu_Lenght) then
9         Insert Features of the answer in row Tabu_Lenght_Num of tabu_list and Tabu_Lenght_Num++
10    If (Tabu_Lenght_Num >= Tabu_Lenght) then
11        Insert Features of the answer in row random(0, Tabu_Lenght) of tabu_list
12    If  $f_{new} < f_{best}$  then
13         $f_{new} = f_{best}$  , ConsIter=0
14    otherwise, ConsIter++
15    If  $f_{new} < f_{best}$  and  $f_{new}$  tabu and Status is true
16        ,  $f_{best} = f_{new}$  , ConsIter =0 , Update the corresponding tabu list;  $f_{current} = f_{new}$ 
17    If (Tabu_Lenght_Num < Tabu_Lenght) then
18        Insert Features of the answer in row Tabu_Lenght_Num of tabu_list and Tabu_Lenght_Num++
19    If (Tabu_Lenght_Num >= Tabu_Lenght) then
20        A= random(0, Tabu_Lenght) , Insert Features of the answer in row A of tabu_list
21    End.
```

شکل (۶): شبه‌کد الگوریتم PDBP\_TS

نوع همسایگی	گره (ها) مشتری	موقعیت گره (ها)	گره (ها) مشتری	موقعیت گره (ها)	موقعیت گره (ها)	کانتینر
۱	R1	۱	R3	۱	۲	۲
۲	P1, D1	۱ و ۲	P3, D3	۱ و ۲	۱ و ۳	۲

شکل (۷): قسمتی از لیست ممنوع در الگوریتم PDBP\_TS

1. aspiration

بهترین جواب	$\pi_{best}$
جواب به‌دست آمده از فاز همسایه‌سازی	$\pi_{nei}$
جواب به‌دست آمده از فاز جست‌وجوی محلی	$\pi_{loc}$
شماره‌ی همسایه‌ی استفاده شده در فاز همسایه‌سازی	$K$
همسایه‌ی تولید شده از شماره‌ی همسایگی $k$	$N_k$
تکرارهای بدون بهبود	$ConsIter. iter$
حداکثر تکرارهای بدون بهبود مجاز	$MaxConsIter$

جواب اولیه در الگوریتم PDBP\_VNS مانند الگوریتم PDBP\_TS توسط الگوریتم ابتکاری OCT تولید می‌شود. این جواب به‌عنوان جواب جاری ( $\pi_{cur}$ ) در نظر گرفته می‌شود؛ سپس تا رسیدن به شرط توقف تعدادی تکرار انجام می‌گیرد. در هر تکرار این الگوریتم  $\pi_{cur}$  وارد سه فاز می‌شود. در فاز اول که فاز همسایه‌سازی نام دارد، برای جواب جاری یک همسایه با عنوان  $\pi_{nei}$  با روش همسایه‌سازی  $k$ ام ( $N_k$ ) تولید می‌شود. جواب این فاز نیز باید از نظر تمام محدودیت‌ها، از جمله محدودیت‌های بارگذاری امکان‌پذیر باشد که برای بررسی این محدودیت‌ها تابع بارگذاری فراخوانی شده است. همان‌طور که در شکل (۸) نشان داده شده است، ورودی این تابع تورهایی است که موقعیت قرارگیری گره‌های آن‌ها تغییر کرده و خروجی آن‌ها شدنی و یا نشدنی بودن محدودیت‌های بارگذاری برای ورودی مسأله می‌باشد. سپس در فاز دوم، که جست‌وجوی محلی نامیده می‌شود، تلاش می‌شود که با استفاده از یک جست‌وجوی محلی همسایگی ایجاد شده بهبود داده شود و باید جواب‌های به‌دست آمده مانند فاز اول محدودیت‌های بارگذاری برای آن‌ها بررسی شود. در نهایت بهترین جواب که از نظر محدودیت‌های بارگذاری امکان‌پذیر باشد  $\pi_{loc}$  نامیده می‌شود، اگر  $\pi_{loc} < \pi_{best}$  باشد  $\pi_{loc} = \pi_{best}$  قرار داده می‌شود؛ هم‌چنین در فاز سوم که تغییر روش همسایه‌سازی است، اگر مقدار تابع هدف  $\pi_{loc}$  بهتر از  $\pi_{cur}$  باشد  $k = 1$  و  $\pi_{loc} = \pi_{cur}$  و در غیر این صورت  $k = k + 1$  در نظر گرفته می‌شود و یا اگر مقدار  $k$  بزرگ‌تر از  $k_{max}$  شود نیز مقدار آن برابر یک در نظر گرفته می‌شود؛ در ادامه، این فازها بیشتر توضیح داده می‌شوند.

```

1 t heuristic solution OCT namely  $\pi_{cur}$  and  $\pi_{best}$  and  $K=1$  and
   $ConsIter. iter=0$ 
2 PDBP_VNS(input OCT, output  $\pi_{best}$  )
3 While (  $ConsIter < MaxConsIter$  )
4   %shaking
5    $\pi_{nei} = N_k(\pi_{cur})$ 
6   Loading(inputTour changed ,outputStatus),
   If  $f(\pi_{nei}) < f(\pi_{cur})$  then

```

در الگوریتم PDBP\_TS برای قرار نگرفتن در بهینه‌ی محلی از لیست ممنوع استفاده می‌شود. برای این‌که این لیست به‌صورت صددرصد از بهینگی محلی جلوگیری کند باید تمام خصوصیات یک جواب را در خود داشته باشد که این کار از نظر زمانی و اشغال فضای جست‌وجو به‌صرفه نمی‌باشد؛ لذا سعی می‌شود که لیست ممنوع حاوی ویژگی‌های مهم یک جواب را شامل شود. در این الگوریتم لیست ممنوع شامل چندین سطر و ستون می‌باشد. در هر سطر مشخصات یک جواب که انتخاب می‌شود، در آن نگهداری می‌شود. تعداد سطرها این لیست با عنوان  $Tabu\_Lenght$  از آن یاد می‌شود که مقدار آن توسط آزمایش تاگوچی تعیین می‌شود.

در شکل (۷) قسمتی از یک لیست ممنوع نمایش داده شده است. اگر تولید همسایگی با استفاده از روش تخریب گره‌های بازگشتی انجام شده باشد عدد ۱ و در غیر این‌صورت عدد ۲ در ستون اول شکل (۷) نوشته می‌شود. اگر نوع همسایگی ۱ باشد در لیست ممنوع گره‌های بازگشت که از جواب اولیه حذف شده‌اند، به‌علاوه موقعیت‌ها و شماره‌ی کانتینر آن‌ها قبل از حذف شدن در لیست ممنوع قرار داده می‌شود. اگر نوع همسایگی از نوعی به‌غیر از روش تخریب گره‌های بازگشتی باشد، در لیست ممنوع، ابتدا شماره‌ی گره، موقعیت و شماره‌ی کانتینر گره‌ی دریافت و سپس همین اطلاعات در مورد گره‌ی دریافت مشتری حذف شده از جواب قبلی نوشته می‌شود. درج یک سطر جدید در شکل (۷) از حذف تصادفی یک سطر و قرار دادن سطر جدید حاصل می‌شود. در هر تکرار از الگوریتم PDBP\_TS موقعیت قرارگیری گره‌ها در تورها متغیر است و باعث تغییر مقدار تابع هدف مسأله نیز می‌شود. محاسبه‌ی تابع هدف در تکرارهای مختلف از جمله مواردی است که مدت‌زمان قابل‌توجهی از اجرای الگوریتم را به‌خود اختصاص می‌دهد؛ لذا برای کاهش این مدت‌زمان، راهکارهایی استفاده شده است. برای این‌منظور هزینه‌های هر کدام از تورها به‌صورت جداگانه نگه‌داشته شده و تنها در صورتی که موقعیت گره‌های یک تور تغییر کند، صرفاً هزینه‌ی آن تور مجدداً محاسبه می‌شود و از محاسبه‌ی کل هزینه خودداری شود.

#### ۵-۸. روش فراابتکاری جست‌وجوی همسایگی متغیر

برای حل مسأله PDBP علاوه‌بر الگوریتم PDBP\_TS، یک الگوریتم فراابتکاری دیگر بر مبنای الگوریتم VNS ارائه شده است. ایده‌ی اصلی الگوریتم VNS که در زبان فارسی از آن با نام «جست‌وجوی همسایگی متغیر» یاد می‌شود، تغییر سیستماتیک روش همسایه‌سازی می‌باشد [۳۵]. الگوریتم VNS ارائه شده در این مقاله با PDBP\_VNS نام‌گذاری شده است. جدول (۴) پارامترها و شکل (۷) شبه‌کد الگوریتم PDBP\_VNS را نشان می‌دهد.

#### جدول (۴): پارامترهای مورد استفاده در شبه‌کد PDBP\_VNS

پارامتر	توضیح پارامتر
$\pi_{cur}$	جواب در دست



مقایسه شده‌اند، قابل ذکر است که روش‌های فراابتکاری بر روی یک دستگاه رایانه با مشخصات Corei7, RAM 8Gb, 3.99Hz در محیط ویندوز ۸ اجرا شده است.

#### ۶-۱. تولید داده

برای تولید داده برای مسیر رفت و بارگذاری جعبه‌ها در داخل کانتینر از روش موجود در مطالعه‌ی حسن‌آبادی [۲۳] و همچنین منل و بورتفلدت [۱۳، ۲۳] الهام گرفته شده است و برای تولید داده برای مسیر بازگشت از روش کوخ و همکاران [۳۶] استفاده شده است که در ادامه توضیح داده خواهند شد.

برای موقعیت جغرافیایی و مختصات قرارگیری گره‌ها سه حالت مختلف، تحت‌عنوان: تصادفی، خوشه‌بندی مرکب و خوشه‌بندی مطلق، در نظر گرفته شده است؛ هم‌چنین لازم به ذکر است که مختصات قرارگیری گره‌های دریافت و بازگشتی یکسان می‌باشد. در حالت تصادفی مکان قرارگیری گره‌ها به‌صورت یکنواخت و تصادفی بوده و در دو حالت دیگر گره‌های دریافت و تحویل به دو صورت خوشه‌بندی مرکب و مطلق قرار گرفته‌اند. در خوشه‌بندی مرکب، ممکن است در هر خوشه هم گره‌ی دریافت و هم گره‌ی تحویل وجود داشته باشد، اما در خوشه‌بندی مطلق فقط در هر خوشه یا گره‌ی دریافت و یا گره‌ی تحویل وجود دارد [۱۳].

تعداد جعبه‌های بارگذاری برای نیمی از گره‌های دریافت عدد تصادفی در بازه‌ی [1,3] و برای نیمی دیگر در بازه‌ی [1,5] می‌باشد. تعداد جعبه‌های گره‌های بازگشتی نیز برابر با تعداد گره‌های دریافت می‌باشد [۱۳]. تعداد گره‌های بازگشتی به‌صورت تصادفی ۳۰ درصد، ۶۰ درصد و ۱۰۰ درصد در نظر گرفته شده است [۳۶]. برای هر گره زمان خدمت یکسان و برابر ۴۰ در نظر گرفته شده است [۱۳].

وسایل نقلیه، مانند حسن‌آبادی [۲۳] ناممگون در نظر گرفته شده و تعداد وسایل نقلیه نامحدود است. وسایل نقلیه، در دو نوع با ابعاد [60,25,30] و [122,25,30] است که هزینه‌ی استفاده از کانتینر کوچک و بزرگ به‌ترتیب ۱۰۰ و ۱۲۰ واحد پولی می‌باشد [۱۳].

طول، عرض و ارتفاع جعبه‌های مشتری در بازه‌های یکنواخت تولید می‌شوند. طول آیت‌ها در بازه‌های [0.2L<sub>min</sub>, 0.6L<sub>max</sub>] و عرض جعبه‌ها نیز اعدادی تصادفی در بازه‌ی [0.2W<sub>min</sub>, 0.6W<sub>max</sub>] می‌باشد. در نهایت ارتفاع جعبه‌ها در بازه‌ی [0.2H<sub>min</sub>, 0.6H<sub>max</sub>] تولید می‌شوند. W<sub>max</sub>, L<sub>max</sub> و H<sub>max</sub> به‌ترتیب نشان‌دهنده‌ی بیشترین عرض، طول و ارتفاع کانتینر می‌باشد که مورد استفاده قرار گرفته شده است و L<sub>min</sub>, W<sub>min</sub> و H<sub>min</sub> نیز به‌ترتیب نشان‌دهنده‌ی کمترین عرض و طول و ارتفاع کانتینر مورد استفاده می‌باشد. حداقل سطح پشتیبانی برای جعبه‌ها نیز ۰/۷۵ در نظر گرفته شده است [۱۳].

ظرفیت کانتینرهای کوچک برابر با ۴۵۰۰۰ و ظرفیت کانتینرهای بزرگ ۹۱۵۰۰ می‌باشد. وزن آیت‌ها برای دو سوم مشتری‌ها برابر حجم آیت‌ها و برای مشتری‌های باقی‌مانده نسبت حجم به وزن سه‌به‌یک در نظر گرفته شده است. وزن آیت‌های بازگشت نیز یک‌سوم

```

۷ %Local Search
۸ Generate  $\pi_{loc}$  by a local search for  $\pi_{nei}$  ,
Loading(inputTour changed ,outputStatus),
۹ % Neighborhood Change
۱۰ If ( $f(\pi_{loc}) < f(\pi_{cur})$ ) and Status is true then
۱۱  $f(\pi_{loc}) = f(\pi_{cur})$ 
۱۲ Else  $k = k + 1$ ,  $Conslter++$ 
۱۳ Endif
۱۴ If ( $k > k_{max}$ ) then
۱۵  $K = 1$ 
۱۶ If ( $f(\pi_{loc}) < f(\pi_{best})$ ) then
۱۷  $\pi_{best} = \pi_{loc}$   $Conslter = 0$ 
۱۸ End

```

شکل (۸): شبه‌کد الگوریتم PDBP\_VNS

در فاز همسایه‌سازی، جواب اولیه‌ی مسأله به‌عنوان جواب جاری در نظر گرفته می‌شود و یک همسایه با استفاده از یکی از روش‌های همسایه‌سازی تولید می‌شود. برای روش همسایه‌سازی یک روش تخریب به‌صورت تصادفی با یکی از روش‌های درج در نظر گرفته می‌شود و با افزایش تکرارها هر چقدر که ترکیب یک روش تخریب و درج بهبود بیشتری را به‌دنبال داشته باشد، احتمال انتخاب این ترکیب افزایش می‌یابد؛ همان‌طور که قبلاً ذکر شد اگر یک روش همسایه‌سازی نتواند جواب را بهبود دهد در تکرار بعدی روش همسایه‌سازی دیگری که شامل یک روش تخریب و یک روش درج که به‌صورت تصادفی انتخاب می‌شود، به‌کار برده می‌شود.

در فاز جست‌وجوی محلی با استفاده از یکی از روش‌های جست‌وجو، همسایه ایجاد شده در فاز همسایه‌سازی بهبود داده می‌شود. در این روش نیز از تمام روش‌های تخریب به‌غیر از روش تخریب به‌کار برده شده در فاز همسایه‌سازی، انتخاب می‌شود و برای هر کدام از روش‌های تخریب به‌صورت تصادفی یک روش سازنده انتخاب می‌شود که مانند فاز همسایه‌سازی هر چقدر یک ترکیب، جواب‌های بهتری را تولید کند احتمال انتخاب این ترکیب افزایش می‌یابد.

همان‌طور که اشاره شد در فاز تغییر همسایه‌سازی در الگوریتم PDBP\_VNS روش‌های همسایه‌سازی متفاوتی در نظر گرفته شده است. اگر جوابی که از فاز جست‌وجوی محلی به‌دست می‌آید بهتر از جواب در دست نباشد، وارد فاز تغییر همسایه‌سازی می‌شود. در این فاز، روش درج و حذفی که برای ایجاد همسایه‌ی جدید در فاز همسایه‌سازی تکرار بعد باید استفاده شود، تغییر می‌کند.

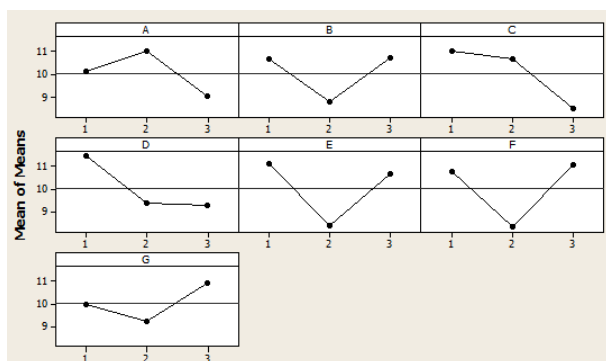
#### ۶. آزمایش‌های عددی

در این قسمت ابتدا نمونه‌هایی برای مقایسه‌ی الگوریتم‌ها تولید شده و سپس تنظیم پارامترهای دو الگوریتم فراابتکاری توسط آزمایش تاگوچی تعیین شده و در نهایت الگوریتم‌های پیشنهادی با یکدیگر

80% maxConstler	50% maxConstler	30% maxConstler		
۰/۵	۰/۴	۰/۳	E	$\alpha$
۲	۱	۰/۵	F	$\beta$
۶	۵	۴	G	Num_nigh

در شکل (۹) نمودار میانگین پارامترها نشان داده شده است. برای هر نمودار محور عمودی نشان‌دهنده‌ی مقدار متغیر پاسخ و محور افقی نشان‌دهنده‌ی سطوح پارامتر هستند. به دلیل این که تابع هدف حداقل‌سازی می‌باشد، لذا مقدار تابع هدف کمتر بهتر است؛ بنابراین، سطحی انتخاب می‌شود که متغیر پاسخ کمتری داشته باشد؛ برای مثال، برای نمودار B در شکل (۹) که مربوط به  $N\_List$  است، سطح ۲ انتخاب می‌شود. همچنین در جدول (۶) مقدار پارامترهای منتخب روش PDBP\_TS نشان داده شده است.

همان‌طور که بیان شد پارامتر iter\_adaptive تعداد تکرار برای شروع فرایند تطابق را نشان می‌دهد. در فرایند تطابق، سعی بر آن می‌شود که اگر تعداد بهبود از یک رویکرد همسایگی بیشتر باشد در تکرارهای بعدی از آن روش همسایگی، تعداد بیشتری تولید کرد. سطح ۳ این پارامتر یک عدد بزرگ در نظر گرفته شده که آزمایش تاگوچی نشان‌دهنده‌ی بهتر بودن این مقدار است؛ همچنین پارامتر iter\_adaptive\_start که شروع فرایند بعد از تعداد مشخصی تکرار بدون بهبود را نشان می‌دهد به این برداشت که فرایند سازگاری برای جواب بهتر نیست صحنه می‌گذارد.



شکل (۹): نمودار میانگین پارامترهای PDBP\_TS

جدول (۶): مقدار پارامترهای منتخب PDBP\_TS

مقدار پارامترها	پارامترها
۴۰۰	maxConstler
۱۴	N_list
۱۰۰۰۰۰۰	iter_adaptive

آیتم‌های رفت می‌باشد [۱۳].

برای هر گره ابتدا مرکز پنجره‌ی زمانی تولید می‌شود که با  $C_j$  نشان داده می‌شوند و سپس طول پنجره‌ی زمانی برای گره‌ها را که با  $l_j$  نشان داده می‌شود، تولید می‌شود. پنجره‌ی زمانی گره‌ی  $j$  را نیز با  $[A_j, B_j]$  نشان می‌دهند که از رابطه‌ی (۵۹) و (۶۰) به دست می‌آید [۳].

$$A_j = C_j - l_j \quad (59)$$

$$B_j = C_j + l_j \quad (60)$$

$C_j$  به صورت تصادفی در بازه‌ی  $(A_0 + t_{0j}, B_0 - t_{0j})$  تولید می‌شود که  $A_0$  و  $B_0$  کران بالا و پایین گره‌ی انبار می‌باشد.  $S_j$  و  $t_{0j}$  به ترتیب فاصله‌ی گره‌ی صفر تا گره‌ی  $j$  و زمان خدمت‌دهی گره‌ی  $j$  می‌باشد. در مطالعه‌ی حسن‌آبادی [۲۳]  $A_0$  را برابر صفر و  $B_0$  را برابر با حداکثر زمان سفر که در داده‌های منل و بورتفلد [۱۵] وجود دارد، در نظر گرفته شده است.  $l_j$  به صورت تصادفی توسط توزیع نرمال با میانگین ۳۰ و انحراف معیار ۱۵ تولید شده است.

برای نام‌گذاری نمونه‌ها، ابتدا تعداد مشتریان آورده می‌شود و سپس ویژگی موقعیت جغرافیایی گره‌ها آورده شده است؛ به طور کلی  $RAND^1$  نشان‌دهنده‌ی موقعیت تصادفی گره‌ها،  $CPCD^2$  نشان‌دهنده‌ی خوشه‌ی مرکب و  $CLUS^3$  نیز نشان‌دهنده‌ی خوشه‌ی مطلق می‌باشد. در ادامه نام‌گذاری، تعداد متوسط آیتم‌های مربوط به گره‌های دریافت مشتری آورده شده است و در نهایت شماره‌ی نمونه در آن دسته آورده شده است؛ برای مثال، عبارت 50.CPCD.2.4 نشان‌دهنده‌ی گروه با ۵۰ مشتری با خوشه‌بندی مرکب که متوسط آیتم‌های گره‌ی دریافت مشتریان ۲ می‌باشد. به علاوه این نمونه چهارمین نمونه در این گروه است.

## ۲-۶. تنظیم پارامتر الگوریتم‌های فراابتکاری

برای به دست آوردن مقادیر مناسب برای پارامترهای الگوریتم PDBP\_TS با استفاده از نرم‌افزار Minitab برای هر هفت پارامتر سه‌سطحی با استفاده از طرح  $L_{27}$  تاگوچی، تعداد ۲۷ آزمایش مطرح شده است. آزمایش‌ها برای ۵۴ نمونه، انجام شده و میانگین تابع هدف به عنوان متغیر پاسخ برای روش تاگوچی در نظر گرفته شده است. هفت پارامتر مورد استفاده در روش تاگوچی برای الگوریتم PDBP\_TS در جدول (۵) نشان داده شده است.

جدول (۵): پارامترهای الگوریتم PDBP\_TS و سطوح پارامترها

نام پارامتر	نماد	سطح ۱	سطح ۲	سطح ۳
maxConstler	A	۲۰۰	۳۰۰	۴۰۰
N_list	B	۷	۱۴	۲۱
iter_adaptive	C	۳۰	۵۰	۱۰۰۰۰۰۰

شکل (۱۰): نمودار میانگین پارامترهای الگوریتم PDBP\_VNS

80% maxConsIter	iter_adaptive_start
۰/۴	$\alpha$
۱	$\beta$
۵	Num_nigh

### ۳-۶. اعتبارسنجی الگوریتم‌های ارائه شده در ابعاد کوچک

مدل ریاضی مسأله مسیریابی دریافت، تحویل و بازگشت با محدودیت‌های بارگذاری سه‌بعدی در نهایت قادر به حل نمونه‌ای با سه مشتری و سه آیتم بوده است. این مدل برای چهار مشتری و دو آیتم به دلیل افزایش نمایی تعداد متغیرها و محدودیت‌ها، در مدت‌زمان یک ساعت قادر به حل نمونه‌ای با این ابعاد نمی‌باشد؛ بنابراین سه محدودیت پایداری، انباشت و عدم بارگذاری مجدد که متغیرهای زیادی در مسأله تولید می‌کنند در مدل ریاضی در نظر گرفته نمی‌شوند. به دلیل آزادسازی محدودیت‌های ذکر شده، می‌توان مدل ریاضی را به عنوان کران پایین مسأله برای ارزیابی الگوریتم‌های ابتکاری و فراابتکاری در نظر گرفت.

جدول (۸): مقدار پارامترهای نهایی PDBP\_VNS

پارامترها	مقدار پارامترها
maxConsIter	۴۰۰
Num_nigh	۵
$\alpha$	۰/۴
$\beta$	۱

در جدول (۹) نمونه‌ها براساس تعداد مشتری، میانگین آیتم و خوشه‌بندی گره‌های دریافت و تحویل مشتری‌ها دسته‌بندی و برای آن‌ها درصد خطای الگوریتم‌های PDBP\_VNS و PDBP\_TS نسبت به کران پایین بیان شده است. برای هر دسته با تعداد مشتری، خوشه و متوسط تعداد آیتم یکسان، پنج نمونه در نظر گرفته شده است؛ هم‌چنین برای محاسبه درصد خطا از رابطه‌ی ۶۱ استفاده شده که در آن  $slu_{Lowerbound}$  و  $slu_{algorithm}$  به ترتیب مقدار کران پایین و جواب الگوریتم فراابتکاری است.

$$\text{درصد خطا} = \frac{f(slu_{algorithm}) - f(slu_{Lowerbound})}{f(slu_{Lowerbound})} \times 100 \quad (61)$$

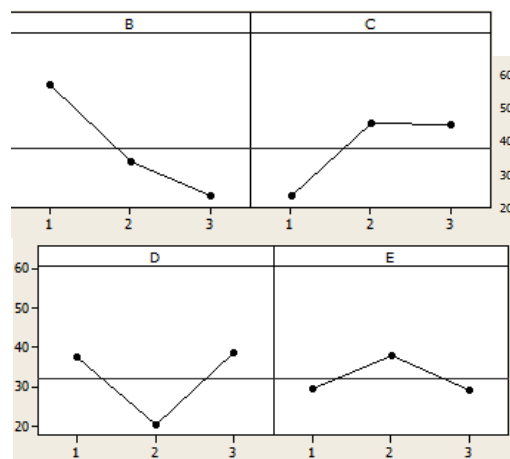
در جدول (۹) برای دسته‌هایی که مدل ریاضی نتوانسته در مدت زمان ۳۶۰۰ ثانیه حد پایین ارائه دهد، برای آن و الگوریتم‌های دیگر در جدول (۹) از علامت "-" استفاده شده است؛ لذا همان‌طور که از این جدول مشخص است، کران پایین برای ۳۶ دسته از نمونه‌ها به دست آمده است. هر دو روش فراابتکاری تنها در پنج دسته از نمونه‌ها نتوانسته‌اند نمونه‌ها را به صورت بهینه حل کنند و در ۳۱ دسته‌ی دیگر هر دو روش فراابتکاری به جواب بهینه رسیده‌اند.

پارامترهایی که در الگوریتم PDBP\_VNS توسط آزمایش‌های تاگوچی بررسی شده در جدول (۷) آورده شده است. برای این الگوریتم دو مرتبه آزمایش تاگوچی انجام شده است؛ در مرتبه اول، توجه بیشتر بر پیدا کردن ترتیبی برای عوض کردن فاز همسایگی در این الگوریتم بوده است. در این الگوریتم شش نوع همسایه‌سازی در نظر گرفته شده است که در آزمایش تاگوچی اول به دلیل زمان‌بر بودن آزمایش‌ها ترتیب قرارگیری سه رویکرد اثر بخش‌تر همسایه‌سازی را در نظر گرفته و ترتیب سه همسایه‌ی دیگر را به صورت درک شهودی بعد از آن‌ها قرار داده شده است. بعد از مشخص شدن ترتیب همسایه‌ها در فاز همسایه‌سازی بار دیگر آزمایش تاگوچی برای چهار پارامتر که در جدول (۷) آورده شده است، انجام داده می‌شود. با استفاده از نرم‌افزار Minitab برای هر چهار پارامتر سه سطحی با استفاده از طراحی آزمایش L9 تاگوچی، تعداد ۹ آزمایش انجام شده است. تنظیم پارامترهای الگوریتم PDBP\_TS بر روی ۵۴ نمونه‌ی تولید شده انجام گرفته شده و میانگین تابع هدف به عنوان متغیر پاسخ برای روش تاگوچی در نظر گرفته شده است.

جدول (۷): پارامترهای الگوریتم PDBP\_VNS و سطوح پارامترها

نام پارامتر	نماد	سطح ۱	سطح ۲	سطح ۳
maxConsIter	B	۲۰۰	۳۰۰	۴۰۰
Num_nigh	C	۴	۵	۶
$\alpha$	D	۰/۳	۰/۴	۰/۵
$\beta$	E	۰/۵	۱	۲

در شکل (۱۰) نمودار میانگین پارامترهای الگوریتم‌های PDBP\_VNS ارائه شده است؛ به‌طور مثال، در شکل (۱۰) برای پارامتر B که تعداد کل تکرارهای بدون بهبود می‌باشد سطح ۳ انتخاب می‌شود. در جدول (۸) نیز مقادیر منتخب پارامترهای الگوریتم PDBP\_VNS آورده شده است.



جدول (۹): درصد خطای PDBP\_TS و PDBP\_VNS نسبت به کران پایین

PDBP درصد خطا		تعداد نمونه	مشتری/آیتم	PDBP درصد خطا		تعداد نمونه	خوشه‌ی نمونه	مشتری/آیتم
VNS	TS			VNS	TS			
۰/۰۰	۰/۰۰	۵		۰/۰۰	۰/۰۰	۵	RAND	
۰/۰۰	۰/۰۰	۵	۳/۲	۰/۰۰	۰/۰۰	۵	CLUS	۲/۲
۰/۰۰	۰/۰۰	۵		۰/۰۰	۰/۰۰	۵	CPCD	
۰/۰۰	۰/۰۰	۵		۰/۰۰	۰/۰۰	۵	RAND	
۰/۰۰	۰/۰۰	۵	۳/۳	۰/۰۰	۰/۰۰	۵	CLUS	۲/۳
۰/۰۰	۰/۰۰	۵		۰/۰۰	۰/۰۰	۵	CPCD	
۰/۰۰	۰/۰۰	۵		۰/۰۰	۰/۰۰	۵	RAND	
۰/۰۰	۰/۰۰	۵	۳/۴	۰/۰۰	۰/۰۰	۵	CLUS	۲/۴
۰/۰۰	۰/۰۰	۵		۰/۰۰	۰/۰۰	۵	CPCD	
۰/۰۸	۰/۰۰	۵		۰/۰۰	۰/۰۰	۵	RAND	
۰/۰۰	۰/۰۰	۵	۳/۵	۰/۰۰	۰/۰۰	۵	CLUS	۲/۵
۰/۰۰	۰/۰۸	۵		۰/۰۰	۰/۰۰	۵	CPCD	
۰/۰۰	۰/۰۰	۴		۰/۰۰	۰/۰۰	۵	RAND	
۰/۰۰	۰/۰۰	۵	۳/۶	۰/۰۰	۰/۲۰	۵	CLUS	۲/۶
۰/۰۱	۰/۵۰	۳		۰/۰۲	۰/۰۰	۵	CPCD	
-	-	-		۰/۰۰	۰/۰۲	۲	RAND	
۰/۰۰	۰/۰۸	۲	۳/۷	۰/۰۰	۰/۰۰	۵	CLUS	۲/۷
-	-	-		۰/۰۰	۰/۰۰	۵	CPCD	
-	-	-		-	-	-	RAND	
-	-	-	۳/۸	۰/۱۷	۰/۰۱	۲	CLUS	۲/۸
-	-	-		۰/۱۱	۰/۰۰	۱	CPCD	

است. در ستون درصد خطا از رابطه‌ی (۶۲) استفاده شده است که در آن  $f(slu_{mean})$  میانگین تابع هدف‌های هر نمونه در پنج بار اجرا و  $f(slu_{best})$  نیز بهترین تابع هدف بعد از ۱۰ بار اجرا شدن توسط هر دو الگوریتم می‌باشند.

$$\text{درصد خطا} = \frac{f(slu_{mean}) - f(slu_{best})}{f(slu_{best})} \times 100 \quad (62)$$

جدول (۱۰): درصد خطا و درصد حل بهینه‌ی مسائل

الگوریتم	درصد بهینه‌ی حل شده	درصد خطا
PDBP_TS	۹۵	۰/۰۲
PDBP_VNS	۹۷	۰/۰۱

جدول (۱۱): مدت‌زمان اجرای الگوریتم‌های فراابتکاری (ثانیه)

تعداد مشتری	میانگین دو جعبه		میانگین سه جعبه	
	PDBP_VNS	PDBP_TS	PDBP_VNS	PDBP_TS
۵۰	۱۷۵	۲۰۰	۳۵۰	۳۱۰
۷۵	۳۲۰	۴۰۰	۷۰۰	۶۰۰
۱۰۰	۵۵۰	۸۰۰	۱۴۰۰	۱۰۰۰

در جدول (۱۰) خلاصه‌ای از درصد نمونه‌هایی که به‌صورت بهینه حل شده‌اند و همچنین درصد خطای هر کدام از الگوریتم‌ها نشان داده است. روش فراابتکاری PDBP\_TS، ۹۷ درصد نمونه‌ها و روش فراابتکاری PDBP\_VNS نیز ۹۵ درصد از مسائل را به‌صورت بهینه حل کرده است. متوسط درصد خطا برای نمونه‌ها در الگوریتم PDBP\_TS برابر ۰/۰۲ و برای الگوریتم PDBP\_VNS برابر ۰/۰۱ می‌باشد.

#### ۴-۶. مقایسه‌ی الگوریتم‌های فراابتکاری در ابعاد بزرگ

۵۴ نمونه توسط الگوریتم‌های PDBP\_TS و PDBP\_VNS هر کدام پنج‌مرتب‌ه اجرا شده است. جدول (۱۱) میانگین مدت‌زمان اجرای هر الگوریتم را برای پنج‌بار اجرا در حالت‌های مختلف با میانگین آیتم و تعداد مشتری تا رسیدن به شرط توقف نشان داده است. همان‌طور که از جدول (۱۱) مشاهده می‌شود میانگین مدت‌زمان اجرای الگوریتم PDBP\_VNS در تمام قسمت‌ها بیشتر از PDBP\_TS است. اما برای مقایسه‌ی منطقی بین این دو روش، برای هر دو روش شرط توقف هر دو الگوریتم مدت‌زمان مساوی و برابر مدت‌زمان روش PDBP\_VNS در نظر گرفته شده است و نتایج حل آن‌ها در جدول (۱۳) ارائه شده است. برای ارزیابی این دو الگوریتم، بهترین جواب به‌دست آمده از پنج‌بار اجرای هر الگوریتم به‌عنوان مبنا قرار گرفته

جدول (۱۲): نتایج محاسباتی برای پنج‌بار اجرای PDBP\_VNS و PDBP\_TS

درصد خطا		بهترین روش	تابع هدف	متوسط مقدار تابع		نام نمونه
PBPB_VNS	PDBP_TS			PBPB_VNS	PDBP_TS	
۰/۸۳	۰/۵۸	PDBP_TS	۴۴۰۶/۸۱	۴۴۴۳/۵۹	۴۴۳۲/۵۱	50.RAND.2.1
۱/۹۷	۰/۹۲	PDBP_TS	۳۷۱۰/۳۱	۳۷۸۳/۲۶	۳۷۴۴/۳۳	50.RAND.2.2
۱/۰۹	۰/۹۲	PDBP_VNS	۴۳۱۱/۹۱	۴۳۵۸/۷۰	۴۳۵۱/۶۹	50.RAND.2.3
۰/۸۷	۱/۲۰	PDBP_TS	۴۰۰۵/۵۴	۴۰۴۰/۲۴	۴۰۵۳/۷۷	50.RAND.2.4
۰/۵۸	۰/۸۱	PDBP_VNS	۴۵۰۸/۹۷	۴۵۳۵/۳۲	۴۵۴۵/۳۸	50.RAND.2.5
۱/۹۴	۱/۸۲	PDBP_TS	۵۰۳۴/۵۱	۵۱۳۱/۹۸	۵۱۲۶/۰۹	50.CLUS.2.1
۰/۹۵	۱/۲۱	PDBP_VNS	۴۹۹۷/۰۵	۵۰۴۴/۶۷	۵۰۵۷/۵۱	50.CLUS.2.2
۱/۱۲	۷/۱۲	PDBP_VNS	۵۲۳۳/۶۶	۵۲۹۲/۰۴	۵۶۰۶/۲۳	50.CLUS.2.3
۰/۵۱	۰/۷۹	PDBP_TS	۴۹۸۳/۴۳	۵۰۰۸/۶۰	۵۰۲۳/۰۴	50.CLUS.2.4
۰/۹۴	۰/۰۰	PDBP_VNS, PDBP_TS	۴۵۰۰/۲۹	۴۵۴۲/۶۴	۴۵۰۰/۲۹	50.CLUS.2.5
۰/۷۱	۰/۹۷	PDBP_VNS	۴۴۵۸/۰۵	۴۴۸۹/۷۳	۴۵۰۱/۳۴	50.CPCD.2.1
۰/۹۸	۰/۶۶	PDBP_TS	۳۹۹۶/۵۱	۴۰۳۵/۵۴	۴۰۲۲/۹۹	50.CPCD.2.2
۰/۷۴	۰/۴۵	PDBP_VNS, PDBP_TS	۴۱۰۸/۰۷	۴۱۳۸/۶۰	۴۱۲۶/۵۲	50.CPCD.2.3
۰/۴۷	۰/۸۰	PDBP_VNS	۴۶۰۴/۵۰	۴۶۲۶/۲۳	۴۶۴۱/۵۱	50.CPCD.2.4
۱/۶۲	۱/۴۲	PDBP_TS	۴۴۲۳/۰۱	۴۴۹۴/۸۸	۴۴۸۵/۶۲	50.CPCD.2.5
۰/۷۲	۱/۷۶	PDBP_VNS	۵۵۴۳/۹۹	۵۵۸۳/۸۱	۵۶۴۱/۳۴	75.RAND.2.1
۱/۵۹	۰/۱۱	PDBP_VNS	۵۵۱۸/۲۶	۵۶۰۶/۲۵	۵۵۲۴/۰۷	75.RAND.2.2
۰/۶۷	۰/۵۳	PDBP_TS	۵۸۱۵/۰۲	۵۸۵۳/۷۹	۵۸۴۵/۷۴	75.RAND.2.3
۰/۷۲	۱/۰۰	PDBP_VNS	۶۷۶۴/۹۹	۶۸۱۳/۸۷	۶۸۳۲/۴۲	75.CLUS.2.1
۰/۸۵	۱/۱۰	PDBP_TS	۶۹۵۳/۸۶	۷۰۱۲/۶۸	۷۰۳۰/۰۶	75.CLUS.2.2
۰/۳۶	۰/۲۸	PDBP_VNS	۷۲۲۵/۰۸	۷۲۵۱/۰۹	۷۲۴۵/۴۴	75.CLUS.2.3
۰/۹۷	۱/۳۹	PDBP_VNS	۶۹۰۷/۶۰	۶۹۷۴/۵۲	۷۰۰۳/۳۴	75.CPCD.2.1
۱/۴۳	۱/۳۷	PDBP_TS	۷۰۰۲/۷۰	۷۱۰۳/۸۸	۷۰۹۹/۵۵	75.CPCD.2.2
۰/۹۱	۰/۱۲	PDBP_TS	۷۲۵۷/۷۷	۷۳۲۳/۷۶	۷۲۶۶/۱۴	75.CPCD.2.3
۱/۶۵	۰/۷۲	PDBP_TS	۱۲۳۰۶/۶۵	۱۲۵۰۹/۸۷	۱۲۳۹۴/۷	100.RAND.2.1
۰/۸۴	۱/۹۶	PDBP_VNS	۱۲۱۴۶/۴۴	۱۲۲۴۸/۲۴	۱۲۳۸۴/۹	100.CLUS.2.1
۱/۳۳	۱/۷۵	PDBP_VNS	۱۲۸۳۷/۹۷	۱۳۰۰۹/۳۴	۱۳۰۶۳/۰۰	100.CPCD.2.1
۱/۰۷	۰/۰۰	PDBP_TS	۴۲۰۶/۷۰	۴۲۵۱/۷۹	۴۲۰۶/۷۰	50.RAND.3.1
۰/۷۴	۰/۹۱	PDBP_VNS	۴۳۶۱/۸۱	۴۳۹۴/۲۱	۴۴۰۱/۳۷	50.RAND.3.2
۰/۲۴	۰/۷۱	PDBP_VNS	۴۳۱۰/۳۸	۴۳۲۰/۵۸	۴۳۴۰/۹	50.RAND.3.3
۰/۸۱	۰/۹۴	PDBP_VNS	۴۱۷۴/۹۹	۴۲۰۹/۰۰	۴۲۱۴/۳	50.RAND.3.4
۰/۹۵	۱/۲۴	PDBP_VNS	۴۲۱۸/۹۹	۴۲۵۹/۱۷	۴۲۷۱/۳۳	50.RAND.3.5
۱/۰۳	۱/۳۲	PDBP_TS	۴۲۴۱/۴۲	۴۲۸۴/۹۸	۴۲۹۷/۵۳	50.CLUS.3.1
۰/۶۳	۰/۲۸	PDBP_VNS	۴۰۰۴/۲۷	۴۰۲۹/۵۳	۴۰۱۵/۵۷	50.CLUS.3.2
۰/۷۰	۰/۴۶	PDBP_TS	۴۲۵۳/۲۱	۴۲۸۲/۹۴	۴۲۷۲/۷۶	50.CLUS.3.3
۱/۰۲	۱/۸۱	PDBP_VNS	۳۹۹۵/۲۲	۴۰۳۵/۸۷	۴۰۶۷/۶۸	50.CLUS.3.4
۰/۶۱	۱/۰۰	PDBP_TS	۴۱۸۵/۶۸	۴۲۱۱/۱۷	۴۲۲۷/۴۵	50.CLUS.3.5
۰/۲۷	۰/۱۲	PDBP_VNS	۵۴۰۱/۷۰	۵۴۱۶/۵۴	۵۴۰۸/۱۴	50.CPCD.3.1
۰/۸۳	۰/۷۸	PDBP_VNS	۴۴۰۶/۸۱	۵۰۶۶/۴۰	۵۰۶۳/۵۸	50.CPCD.3.2
۰/۷۳	۱/۰۸	PDBP_VNS	۳۷۱۰/۳۱	۴۹۹۳/۲۵	۵۰۱۰/۳۴	50.CPCD.3.3
۰/۵۴	۰/۳۴	PDBP_TS	۴۳۱۱/۹۱	۵۰۳۹/۱۶	۵۰۲۹/۴۹	50.CPCD.3.4
۰/۹۵	۰/۳۹	PDBP_TS	۴۰۰۵/۵۴	۵۲۷۳/۰۱	۵۲۴۳/۹	50.CPCD.3.5
۰/۵۶	۰/۴۹	PDBP_TS	۴۵۰۸/۹۷	۷۰۶۰/۷۵	۷۰۵۵/۵۴	75.RAND.3.1
۰/۳۳	۰/۵۵	PDBP_VNS	۵۰۳۴/۵۱	۷۱۳۲/۴۸	۷۱۴۸/۲۹	75.RAND.3.2
۰/۶۹	۱/۰۹	PDBP_VNS	۴۹۹۷/۰۵	۷۳۳۲/۴۴	۷۳۶۱/۷۶	75.RAND.3.3
۱/۰۴	۰/۳۷	PDBP_TS	۵۲۳۳/۶۶	۵۴۸۲/۸۸	۵۴۴۶/۱	75.CLUS.3.1
۰/۲۱	۰/۴۵	PDBP_VNS	۴۹۸۳/۴۳	۵۲۳۲/۱۹	۵۲۴۴/۹۲	75.CLUS.3.2
۰/۸۲	۰/۵۹	PDBP_TS	۴۵۰۰/۲۹	۵۳۶۶/۱۶	۵۳۵۴/۲۲	75.CLUS.3.3
۰/۵۸	۰/۱۴	PDBP_TS	۴۴۵۸/۰۵	۷۳۵۰/۰۸	۷۳۱۷/۶۶	75.CPCD.3.1
۱/۸۰	۱/۵۷	PDBP_TS	۳۹۹۶/۵۱	۷۲۴۲/۴۵	۷۲۲۶/۴	75.CPCD.3.2
۰/۷۴	۰/۸۸	PDBP_TS	۴۱۰۸/۰۷	۶۹۷۱/۰۷	۶۹۸۰/۳۱	75.CPCD.3.3
۰/۶۲	۰/۶۱	PDBP_TS	۴۶۰۴/۵۰	۱۲۱۹۹/۵۶	۱۲۱۹۸/۵	100.RAND.3.1
۱/۰۳	۱/۰۱	PDBP_TS	۴۴۲۳/۰۱	۱۲۳۷۴/۱۵	۱۲۳۷۱/۳۲	100.CLUS.3.2

جدول (۱۲): نتایج محاسباتی برای پنج‌بار اجرای PDBP\_VNS و PDBP\_TS

نام نمونه	متوسط مقدار تابع		بهترین روش	درصد خطا	
	PDBP_TS	PBPB_VNS		PDBP_TS	PBPB_VNS
100.CPCD.3.3	۱۲۶۱۶/۰۷	۱۲۶۶۲/۸۰	PDBP_TS	۰/۷۵	۱/۱۲

شکل (۱۱): مقایسه‌ی درصد خطای سه خوشه‌بندی برای دو

الگوریتم PDBP\_VNS و PDBP\_TS

#### ۷. نتیجه‌گیری و پیشنهادات

در این مقاله، مسأله مسیریابی دریافت، تحویل و بارگذاری سه‌بعدی با پنجره‌ی زمانی ارائه شد. یک الگوریتم ابتکاری با عنوان OTC برای به‌دست آوردن یک جواب خوب در مدت‌زمان اندک ارائه گردید و از این الگوریتم به‌عنوان جواب اولیه‌ی الگوریتم‌های فراابتکاری استفاده شد؛ هم‌چنین دو الگوریتم فراابتکاری PDBP\_TS و PDBP\_VNS برای حل نمونه‌ها ارائه شد و برای تولید همسایگی در این روش‌ها چندین روش تخریب و سازنده‌ی جدید ارائه گردید. هرکدام از الگوریتم‌های فراابتکاری پنج‌بار برای هر نمونه اجرا شده است و بهترین جواب به‌دست آمده از این ده‌بار اجرا برای هر نمونه به‌عنوان جواب خوب مبنای مقایسه قرار گرفته و درصدخطای میانگین هر پنج اجرا به‌دست آورده شده است. درصد متوسط خطای الگوریتم PDBP\_TS و PDBP\_VNS به‌ترتیب برابر با ۰/۹۶ و ۰/۸۸ نسبت به بهترین جواب می‌باشد؛ هم‌چنین از ۵۴ نمونه‌ی حل شده، ۲۷ مورد بهترین جواب از الگوریتم PDBP\_TS و در ۲۵ مورد نیز از الگوریتم PDBP\_VNS و در دو مورد نیز از هر دو الگوریتم به‌دست می‌آید. در این مقاله پنجره‌ی زمانی به‌صورت سخت در نظر گرفته شده است. در مطالعه‌های آتی می‌توان با در نظر گرفتن جریمه مناسب با میزان نقض شدن این محدودیت تنوع فضای جست‌وجو را افزایش داد و جواب‌های بهتری به‌دست آورد؛ هم‌چنین در مسأله مورد بررسی تنها یک انبار ابتدایی و انتهایی در نظر گرفته شده است. در مطالعه‌های آتی می‌توان چند انبار ابتدایی و انتهایی برای مسأله مورد بررسی در نظر گرفت. محدودیت‌های دیگری مانند تثبیت موقعیت و جداسازی آیتم‌های ناسازگار که در ادبیات موضوع به آن‌ها کمتر پرداخته شده است را نیز می‌توان در مطالعات آتی در نظر گرفت. هم‌چنین محدودیت شکنندگی را می‌توان به‌صورت دیگری تعریف نمود. در وضعیت فعلی، امکان قرارگیری یک آیتم غیرشکننده بر روی آیتم شکننده وجود ندارد. ولی می‌توان فرض کرد که تعداد محدود و یا وزن محدودی می‌تواند جعبه‌ی غیرشکننده بر روی جعبه‌ی شکننده قرار گیرد. در این‌صورت، این محدودیت به دنیای واقعی برخی از مسائل نزدیک‌تر می‌شود.

#### مراجع

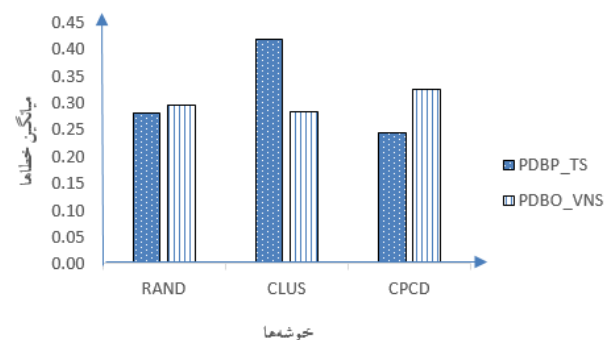
- [1] Karak, A., and Abdelghany, K., (2019). The hybrid vehicle-drone routing problem for pick-up and delivery services, Transportation Research Part C: Emerging Technologies, Vol. 102, pp. 427-449.
- [2] Li, H., and Lim, A., (2003). A metaheuristic for the

در جدول (۱۳) میانگین درصد خطا برای دو الگوریتم PDBP\_TS و PDBP\_VNS و هم‌چنین تعداد بهترین جواب‌های به‌دست آمده توسط هر کدام از این دو روش آورده شده است. این جدول نشان می‌دهد که درصد خطای الگوریتم PDBP\_VNS از الگوریتم PDBP\_TS کمتر است و در ۵۴ نمونه، بهترین جواب در بین ده اجرا در ۲۷ نمونه متعلق به الگوریتم PDBP\_TS و ۲۵ نمونه متعلق به الگوریتم PDBP\_VNS بوده و دو نمونه نیز بهترین جواب از هر دو الگوریتم به‌دست می‌آید. با توجه به نزدیک بودن درصد خطای دو الگوریتم و دارا بودن تعداد نسبتاً مساوی بهترین جواب در هر روش فراابتکاری می‌توان نتیجه گرفت که هر دو الگوریتم میزان اختلاف کمی نسبت به یکدیگر داشته و با یکدیگر در یک‌راستا حرکت می‌کنند؛ به‌عبارت دیگر، اگرچه میانگین درصد خطای الگوریتم PDBP\_TS از الگوریتم PDBP\_VNS بیشتر بوده، ولی این مقدار قابل توجه نیست و می‌توان بیان کرد که عملکرد هر دو الگوریتم تقریباً مشابه است.

در شکل (۱۱) نیز میانگین درصد خطای دو الگوریتم PDBP\_TS و PDBP\_VNS برای هر سه خوشه‌بندی با هم مقایسه شده‌اند. همان‌طور که از شکل (۱۱) مشخص است در خوشه‌ی RAND دو الگوریتم عملکرد یکسانی از نظر درصد خطا دارند. در خوشه‌ی CLUS که گره‌های دریافت و تحویل در خوشه‌های جداگانه قرار دارند. الگوریتم PDBP\_VNS عملکرد بهتری نسبت به PDBP\_TS دارد و در خوشه‌ی CPCD برتری دو الگوریتم جابه‌جا می‌شود.

جدول (۱۳): میانگین درصد خطا و تعداد بهتر بودن جواب‌ها در هر

الگوریتم	الگوریتم	
	درصدخطا	تعداد بهتر بودن
PDBP_TS	۰/۹۶	۲۷
PDBP_VNS	۰/۸۸	۲۵



- (2015). Hybrid algorithms for the vehicle routing problem with clustered backhauls and 3D loading constraints, *European Journal of Operational Research*, Vol. 243, pp. 82-96.
- [16] Pinto, T., Alves, C., and Valério de Carvalho, J., (2020). Variable neighborhood search algorithms for the vehicle routing problem with two-dimensional loading constraints and mixed linehauls and backhauls, *International Transactions in Operational Research*, Vol. 27, pp. 549-572.
- [17] Reil, S., Bortfeldt, A., and Mönch, L., (2018). Heuristics for vehicle routing problems with backhauls, time windows, and 3D loading constraints, *European Journal of Operational Research*, Vol. 266, pp. 877-894.
- [18] Pollaris, H., Braekers, K., Caris, A., Janssens, G.K., and Limbourg, S., (2016). Capacitated vehicle routing problem with sequence-based pallet loading and axle weight constraints, *EURO Journal on Transportation and Logistics*, Vol. 5, pp. 231-255.
- [19] Tao, Y., and Wang, F., (2015). An effective tabu search approach with improved loading algorithms for the 3L-CVRP, *Computers & Operations Research*, Vol. 55, pp. 127-140.
- [20] Junqueira, L., and Morabito, R., (2015). Heuristic algorithms for a three-dimensional loading capacitated vehicle routing problem in a carrier, *Computers & Industrial Engineering*, Vol. 88, pp. 110-130.
- [21] Zhang, Z., Wei, L., and Lim, A., (2015). An evolutionary local search for the capacitated vehicle routing problem minimizing fuel consumption under three-dimensional loading constraints, *Transportation Research Part B: Methodological*, Vol. 82, pp. 20-35.
- [22] Mahvash, B., Awasthi, A., and Chauhan, S., (2017). A column generation based heuristic for the capacitated vehicle routing problem with three-dimensional loading constraints, *International Journal of Production Research*, Vol. 55, pp. 1730-1747.
- [۲۳] حسن‌آبادی، س.، (۱۳۹۵). یکپارچه‌سازی مسأله مسیریابی دریافت و تحویل با محدودیت‌های بارگذاری سه‌بعدی و پنجره زمانی، دانشکده مهندسی صنایع و سیستم‌ها، دانشگاه صنعتی اصفهان.
- [24] Zachariadis, E.E., Tarantilis, C.D., and Kiranoudis, C.T., (2017). Vehicle routing strategies for pick-up and delivery service under two dimensional loading constraints, *Operational Research*, Vol. 17, pp. 115-143.
- [25] Vega-Mejía, C.A., Montoya-Torres, J.R., and Islam, S.M., (2019). A nonlinear optimization model for the balanced vehicle routing problem with loading constraints, *International Transactions in Operational Research*, Vol. 26, pp. 794-835.
- [26] Song, X., Jones, D., Asgari, N., and Pigden, T., (2019). Multi-objective vehicle routing and loading with time window constraints: a real-life application, *Annals of Operations Research*, Vol., pickup and delivery problem with time windows, *International Journal on Artificial Intelligence Tools*, Vol. 12, pp. 173-186.
- [3] Solomon, M.M., (1987). Algorithms for the vehicle routing and scheduling problems with time window constraints, *Operations research*, Vol. 35, pp. 254-265.
- [4] Bravo, M., Rojas, L.P., and Parada, V., (2019). An evolutionary algorithm for the multi-objective pick-up and delivery pollution-routing problem, *International Transactions in Operational Research*, Vol. 26, pp. 302-317.
- [5] Sitek, P., and Wikarek, J., (2019). Capacitated vehicle routing problem with pick-up and alternative delivery (CVRPPAD): model and implementation using hybrid approach, *Annals of Operations Research*, Vol. 273, pp. 257-277.
- [6] Madankumar, S., and Rajendran, C., (2019). A mixed integer linear programming model for the vehicle routing problem with simultaneous delivery and pickup by heterogeneous vehicles, and constrained by time windows, *Sādhanā*, Vol. 44, pp. 39.
- [7] Granada-Echeverri, M., Toro, E., and Santa, J., (2019). A mixed integer linear programming formulation for the vehicle routing problem with backhauls, *International Journal of Industrial Engineering Computations*, Vol. 10, pp. 295-308.
- [8] Chávez, J., Escobar, J., Echeverri, M., and Meneses, C., (2018). A heuristic algorithm based on tabu search for vehicle routing problems with backhauls, *Decision Science Letters*, Vol. 7, pp. 171-180.
- [9] Salhi, S., and Nagy, G., (1999). A cluster insertion heuristic for single and multiple depot vehicle routing problems with backhauling, *Journal of the operational Research Society*, Vol. 50, pp. 1034-1042.
- [10] Gendreau, M., Iori, M., Laporte, G., and Martello, S., (2006). A tabu search algorithm for a routing and container loading problem, *Transportation Science*, Vol. 40, pp. 342-350.
- [11] Fagerholt, K., Hvattum, L.M., Johnsen, T.A., and Korsvik, J.E., (2013). Routing and scheduling in project shipping, *Annals of Operations Research*, Vol. 207, pp. 67-81.
- [12] Cherkesly, M., Desaulniers, G., and Laporte, G., (2015). A population-based metaheuristic for the pickup and delivery problem with time windows and LIFO loading, *Computers & Operations Research*, Vol. 62, pp. 23-35.
- [13] Männel, D., and Bortfeldt, A., (2016). A hybrid algorithm for the vehicle routing problem with pickup and delivery and three-dimensional loading constraints, *European Journal of Operational Research*, Vol. 254, pp. 840-858.
- [14] Männel, D., and Bortfeldt, A., (2018). Solving the pickup and delivery problem with three-dimensional loading constraints and reloading ban, *European Journal of Operational Research*, Vol. 264, pp. 119-137.
- [15] Bortfeldt, A., Hahn, T., Männel, D., and Mönch, L.,

- computing, Vol. 20, pp. 368-384.
- [32] Hemmelmayr, V.C., Cordeau, J.-F., and Crainic, T.G., (2012). An adaptive large neighborhood search heuristic for two-echelon vehicle routing problems arising in city logistics, *Computers & Operations Research*, Vol. 39, pp. 3215-3228.
- [33] Ghilas, V., Demir, E., and Van Woensel, T., (2016). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows and scheduled lines, *Computers & Operations Research*, Vol. 72, pp. 12-30.
- [34] Glover, F., (1986). Future paths for integer programming and links to artificial intelligence, *Computers & operations research*, Vol. 13, pp. 533-549.
- [35] Mladenović, N., and Hansen, P., (1997). Variable neighborhood search, *Computers & Operations Research*, Vol. 24, pp. 1097-1100.
- [36] Koch, H., Bortfeldt, A., and Wäscher, G., (2018). A hybrid algorithm for the vehicle routing problem with backhauls, time windows and three-dimensional loading constraints, *OR Spectrum*, Vol., pp. 1-47.
- pp. 1-27.
- [27] Bortfeldt, A., and Yi, J., (2020). The split delivery vehicle routing problem with three-dimensional loading constraints, *European Journal of Operational Research*, Vol. 282, pp. 545-558.
- [28] Paquay, C., Schyns, M., and Limbourg, S., (2016). A mixed integer programming formulation for the three-dimensional bin packing problem deriving from an air cargo application, *International Transactions in Operational Research*, Vol. 23, pp. 187-213.
- [29] Junqueira, L., Morabito, R., and Yamashita, D.S., (2012). Three-dimensional container loading models with cargo stability and load bearing constraints, *Computers & Operations Research*, Vol. 39, pp. 74-85.
- [30] Junqueira, L., Oliveira, J.F., Carravilla, M.A., and Morabito, R., (2013). An optimization model for the vehicle routing problem with practical three-dimensional loading constraints, *International Transactions in Operational Research*, Vol. 20, pp. 645-666.
- [31] Crainic, T.G., Perboli, G., and Tadei, R., (2008). Extreme point-based heuristics for three-dimensional bin packing, *Inform Journal on*





DOI: 10.22084/ier.2020.19256.1855

## Integrated problem of pick up, delivery and backhaul with three-dimensional loading constraints and time window

A. Hazrati<sup>1</sup>, G. Moslehi<sup>\*2</sup>, M. Reisi-Nafchi<sup>3</sup>

1. M.A. Student, Department of Industrial and Systems Engineering, Isfahan University of Technology, Isfahan, Iran  
2. Professor, Department of Industrial and Systems Engineering, Isfahan University of Technology, Isfahan, Iran  
3. Assistant Professor, Department of Industrial and Systems Engineering, Isfahan University of Technology, Isfahan, Iran

### ARTICLE INFO

#### *Article history:*

Received 29 May 2019  
Accepted 3 October 2019

#### *Keywords:*

Routing  
Pickup  
Delivery and backhaul  
3-dimensional loading  
Tabu search  
Variable neighborhood search

### ABSTRACT

The routing and loading problems are two essential issues to reduce transportation costs. In the recent decade, these problems have been integrated to realize the vehicle routing problem. Failure to comply with the loading constraints may result in damage to the goods or less use of the vehicle space, which in each case will result in additional damage and cost. In this paper, for the first time, the integrated routing problem of pickup, delivery, and backhaul with three-dimensional loading constraints and time window is considered, where the constraints of accumulation, orientation, non-reloading, and multi-delivery conditions are examined. In this study, items and containers are considered heterogeneous. By examining the subject literature, this problem was not observed in the literature. A mixed-integer programming model, a heuristic algorithm, and two metaheuristic algorithms based on tabu search and variable neighborhood search are proposed for this problem. For small instances the proposed metaheuristics were compared to the lower bound obtained from relaxing some constraints of the model. For large instances, the two metaheuristic algorithms are compared together. The results show that the average percentage of relative error in the tabu search and variable neighborhood search algorithms is 0.96 and 0.88, respectively. Also, the tabu search algorithm and variable neighborhood search were able to give better results out of 54 instances in 27 and 25 instances, respectively.

\* Corresponding author. G. Moslehi  
Tel.: 031-33915509; E-mail address: moslehi@iut.ac.ir