

## مسأله مکان‌یابی چنددپویی ظرفیت‌دار با برداشت و تحویل همزمان و بارهای برش‌یافته: مدلسازی و حل ابتکاری

مصطفی ستاک<sup>۱\*</sup>، وحید عزیزی<sup>۲</sup>، حسین کریمی<sup>۳</sup>

۱. استادیار دانشکده مهندسی صنایع، دانشگاه خواجه نصیرالدین طوسی، تهران

۲. دانش‌آموخته کارشناسی‌ارشد مهندسی صنایع، دانشگاه خواجه نصیرالدین طوسی، تهران

۳. دانشجوی دکتری مهندسی صنایع، دانشگاه خواجه نصیرالدین طوسی، تهران

### خلاصه

در مدلسازی مسائل دنیای واقعی باید تا حد ممکن تمامی محدودیت‌های حاکم بر مسأله را در نظر گرفت. این رویکرد در مدلسازی سبب می‌شود تا جواب حاصل از مدل شدنی بوده و تا حد زیادی نزدیک به آنچه در عمل رخ می‌دهد، باشد. بنابراین باید به دنبال ایجاد مدل‌های قابل اطمینانی بود که پاسخگوی تعداد قابل‌قبولی از محدودیت‌های موجود در هر مسأله مورد بررسی باشند، که صرفاً جنبه تئوری نداشته و در عمل نیز پرکاربرد باشند. در این مقاله مدلی ریاضی طراحی می‌گردد که پاسخگوی شرایطی خاص با محدودیت‌های گوناگون و در عین حال کاربردی در حوزه مسائل مکانیابی - مسیریابی است. مسأله اشاره شده، مسأله مکان‌یابی مسیریابی چنددپویی ظرفیت‌دار با برداشت و تحویل همزمان و بارهای برش‌یافته است. این تحقیق بیشتر بر ویژگی برش تأکید می‌کند که در عمل نیز به وفور قابل مشاهده است. برش هنگامی رخ می‌دهد که تقاضای برخی از مشتریان از ظرفیت هر کدام از وسایل نقلیه موجود بیشتر باشد. در این مقاله یک مدل برنامه‌ریزی عدد صحیح مختلط برای این مسأله ارائه می‌شود. سپس دو روش فراابتکاری ترکیبی برای حل مسأله در ابعاد بزرگ پیشنهاد می‌شود. الگوریتم اول مبتنی بر الگوریتم ژنتیک گسسته و ژنتیک پیوسته است که به اختصار DCGA نامیده می‌شود و روش دوم براساس ژنتیک گسسته و الگوریتم بهینه‌سازی تجمعی ذرات است که DGAPSO نام دارد. در نهایت نتایج عددی حاصل از حل مسائل نمونه با استفاده از حل‌کننده سیپلکس و روش‌های پیشنهادی ارائه می‌شود. تجزیه و تحلیل نتایج آزمایشات مؤثر بودن الگوریتم‌های پیشنهادی به خصوص DGAPSO را نشان می‌دهد.

### اطلاعات مقاله

تاریخچه مقاله:

دریافت ۱۳۹۳/۲/۲۹

پذیرش ۱۳۹۳/۱۱/۲۶

کلمات کلیدی:

مکانیابی - مسیریابی

برداشت و تحویل همزمان

برش

برنامه‌ریزی عدد صحیح مختلط

الگوریتم ژنتیک

### ۱- مقدمه

کلاسیک شبکه‌های توزیع است. یکی از مسائل پرکاربرد در حوزه شبکه‌های توزیع، مسأله مکانیابی-مسیریابی می‌باشد. این مسأله به‌طور گسترده‌ای در ادبیات بررسی شده است. مسأله مکان‌یابی-مسیریابی با تعیین مکان تسهیلات و مسیرهای وسایل نقلیه برای خدمت‌دهی به مشتریان تحت شرایطی مانند ظرفیت‌های تسهیل و وسیله نقلیه، طول مسیر و غیره سروکار دارد. هدف این مسأله برآورده ساختن تقاضاهای تمام مشتریان و حداقل کردن هزینه کل شامل هزینه‌های مسیریابی، هزینه‌های ثابت وسیله نقلیه، هزینه‌های

یکی از مسائل مهم در زنجیره تأمین و مدیریت لجستیک، طراحی شبکه‌های توزیع می‌باشد که می‌تواند هزینه‌ها را تا حد زیادی کاهش دهد. یک بعد مهم از طراحی شبکه‌های توزیع در نظر گرفتن انواع محدودیت‌های موجود در دنیای واقعیت و گنجاندن آنها در مسائل

\* نویسنده مسئول. مصطفی ستاک

تلفن: ۰۶۳۳۷۳-۸۴-۰۲۱؛ پست الکترونیکی: setak@kntu.ac.ir

معرفی کرده‌اند. یک روش فراابتکاری ترکیبی مبتنی بر الگوریتم ژنتیک و الگوریتم شبیه‌سازی تبرید تدریجی نیز برای این مسئله توسط کاراواغلان و آلتی پارماک [۱۱] پیشنهاد شده است. برای ارزیابی عملکرد این روش نتایج حاصل از آن با حد بالای به‌دست آمده از فرمولبندی برنامه ریزی عدد صحیح مختلط در مجموعه‌ای از مسائل نمونه مقایسه شده است. نتایج محاسباتی بیانگر عملکرد خوب الگوریتم پیشنهادی در به‌دست آوردن جواب‌های بهینه یا نزدیک به بهینه در مدت زمان مناسب است. برای آشنایی بیشتر با این دسته از مسائل نیز می‌توان به بریگلیا و همکاران [۱۲] مراجعه کرد.

مسئله مسیریابی وسیله نقلیه با برش در تقاضای تحویل (SDVRP)<sup>۴</sup> نیز عمدتاً برای کاهشی که در هزینه‌ها ایجاد می‌کند، مورد علاقه پژوهشگران بوده است. در این نوع مسئله در صورتی که تقاضای یک مشتری بیش از اندازه ظرفیت وسیله نقلیه باشد، به چند بخش تقسیم شده (برش می‌یابد) و هر بخش توسط یک وسیله نقلیه حمل می‌شود. SDVRP برای نخستین بار توسط درور و ترودیو [۱۳] معرفی شد. در سال‌های اخیر تمرکز بر روی این دسته از مسائل رو به افزایش بوده است. هو و هاگلند [۱۴] یک روش جستجوی ممنوع را برای مسئله مسیریابی وسایل نقلیه با پنجره‌های زمانی و برش در تقاضا پیشنهاد کرده‌اند. توکلی مقدم و همکاران [۱۵] از الگوریتم فراابتکاری شبیه‌سازی تبرید تدریجی برای حل مسئله مسیریابی وسیله نقلیه با برش در تقاضا و در شرایط ناوگان ناهمگون، استفاده کرده است. میترا [۱۶] خوشه‌بندی و مسیریابی را برای مسئله، در حالتی که شامل حمل در بازگشت است، به کار برد. شاهین و همکاران [۱۷] نیز یک روش ابتکاری برای مسئله‌ای با دارا بودن محدودیت برداشت و تحویل و برش در تقاضای تحویل ارائه کرده است. در چند سال اخیر الگوریتم‌های ترکیبی نیز برای حل این مسئله پیشنهاد شده است. سه الگوریتم ترکیبی مرتبط با این مسئله، در ادبیات یافت شد. اولین الگوریتم توسط چن و همکاران [۱۸] پیشنهاد شد. در این روش یک جواب اولیه توسط الگوریتم استاندارد کلرک و رایت تولید می‌شود. سپس این جواب توسط یک برنامه‌ریزی عدد صحیح به کار گرفته می‌شود تا نقاط ابتدا و انتهای هر مسیر (به جز دپوها) به صورت بهینه تخصیص داده شوند. در ادامه این جواب نیز توسط الگوریتمی به نام VRTR بهبود داده می‌شود. الگوریتم ترکیبی دوم توسط آرچتی و همکاران [۱۹] پیشنهاد شد. که بر خلاف روش قبلی از یک جستجوی ممنوعه در تولید جواب ابتدایی استفاده می‌کنند. سومین الگوریتم ترکیبی را جین و همکاران پیشنهاد کردند [۲۰]. نویسندگان در این الگوریتم از یک رویکرد تولید ستون مشابه آنچه در سیرکسما و تیجسن [۲۱] معرفی شده، استفاده کرده‌اند. آرچتی و اسپرانزا [۲۲] یک مطالعه مرور ادبیات در زمینه SDVRP انجام داده‌اند.

براساس اطلاعات نویسندگان در ادبیات این دسته مسائل، مقاله‌ای

ثابت تسهیل، هزینه‌های عملیاتی تسهیل می‌باشد. مسئله مکانیابی- مسیریابی، در شکل عمومی خود، فرض می‌کند که مشتری‌ها یا کالا را تحویل می‌گیرند و یا تحویل می‌دهند و موضوع مورد علاقه چگونگی توزیع محموله‌ها توسط ناوگانی از وسایل نقلیه مستقر در یک مجموعه دپوی باز بین مشتری‌ها است. حال که مسئله مکانیابی- مسیریابی به‌طور خلاصه معرفی شد، با توجه به نکاتی که قبلاً به آنها اشاره شد، می‌توان با افزودن محدودیت‌های مسائل دنیای واقعی مسئله مکانیابی- مسیریابی را کاربردی‌تر نمود. تقاضاهای برداشت و تحویل و امکان برش در تقاضاها دو نمونه از محدودیت‌هایی هستند که ممکن است در واقعیت برای مشتری‌ها در یک سیستم توزیع وجود داشته باشند.

اگر چه مفهوم مسئله مکانیابی مسیریابی (LRP)<sup>۱</sup> در دهه ۱۹۶۰ توسط وب [۱] معرفی شد اما این مسئله از مسائلی است که در سال‌های اخیر به‌طور گسترده‌ای در ادبیات مطالعه و بررسی شده است. در ادامه به برخی از مطالعات با ارزش در این حوزه اشاره می‌شود.

چندین مدل ریاضی و رویه حل دقیق برای مسایل LRP در اندازه‌های کوچک و متوسط ارائه شده است که می‌توان به عنوان نمونه به کار بلنگوئر و همکاران [۲] اشاره کرد. به‌خاطر پیچیدگی مسئله چندین روش ابتکاری برای حل LRP در اندازه‌های بزرگ ارائه شده است. روش فراابتکاری که ملچوسکی و همکاران [۳] پیشنهاد کرده‌اند را نیز می‌توان از نمونه رویکردهای فراابتکاری موفق برای این مسئله به‌شمار آورد. برای آشنایی بیشتر با این دسته از مسائل می‌توان به مقاله نجی و سالحی [۴] مراجعه کرد.

برای مسئله مسیریابی وسیله نقلیه با برداشت و تحویل (VRPPD)<sup>۲</sup> نیز کارهای ارزشمند فراوانی در طی سال‌های اخیر انجام شده است [۵].

در این دسته مسائل مبدا و مقصد مسئله برداشت و تحویل می‌توانند مشترک یا غیرمشترک باشند. در هر دو دسته مقالات فراوانی در ادبیات موجود است. در گروه دوم یا همان دسته مسائلی که مبدا و مقصد مشترک نیست، می‌توان به کارهای مینیس و تاتاراکیس [۶] و هاف و همکاران [۷] اشاره کرد. کاراواغلان و همکاران [۸] و سابرامانیا و همکاران [۹] نیز از جمله مقالاتی هستند که مبدا و مقصد مشترک و یا همان برداشت و تحویل همزمان را بررسی کرده‌اند. مسئله مکانیابی مسیریابی با برداشت و تحویل همزمان (LRSPD)<sup>۳</sup> ترکیبی از مسئله LRP و VRPPD است که برای اولین بار توسط کاراواغلان و همکاران [۸] معرفی شد. نویسندگان یک گراف جهت دار با ناوگانی همگون از وسایل نقلیه را در نظر گرفته و مسئله را فرمول‌بندی کرده‌اند. کاراواغلان و همکاران [۱۰] در مقاله‌ای دیگر، یک روش شاخه و برش برای LRSPD

1. Location-Routing Problem

2. Vehicle Routing Problem with Pickup and Delivery

3. Location Routing Problem with Simultaneous Pickup and Delivery

## ۲-۱- فرمول‌بندی مسأله

شبکه جهت‌دار  $G = (N, A)$  را در نظر می‌گیریم به طوری که  $N = N_D \cup N_C$  که  $N_D$  گره‌های دپو و  $N_C$  گره‌های مشتری را نمایش می‌دهند و  $A = \{(i, j) : i, j \in N\}$  مجموعه‌ای از یال‌هاست که هر یال  $(i, j) \in N$  هزینه غیرمنفی  $c_{ij}$  را دارد.  $FD_k$  و  $CD_k$  به ترتیب نشان‌دهنده ظرفیت دپو و هزینه ثابت دپو هستند که  $k \in N_D$ . در هر دپو یک ناوگان از وسایل نقلیه با ظرفیت  $CV$  و هزینه ثابت عملیاتی  $FV$  برای خدمت‌دهی به مشتری‌ها در دسترس است. هر مشتری  $i \in N_C$  تقاضای برداشت  $p_i$  و تقاضای تحویل  $d_i$  را دارد.

مسأله تعیین مکان دپوها و تخصیص مشتری‌ها به دپوهای باز و مسیره‌های متناظر با هر دپو با کمترین هزینه است به طوری که شرایط زیر نیز صادق باشد:

- هر وسیله حداکثر در یک مسیر مورد استفاده قرار گیرد.
  - هر مسیر از یک دپو شروع شده و در همان دپو خاتمه می‌یابد.
  - کل بار وسیله نقلیه در هر نقطه از مسیر نباید از ظرفیت آن تجاوز کند.
  - کل بار برداشتی و بار تحویلی به مشتری‌های اختصاص داده شده به یک دپوی باز نباید از ظرفیت آن دپو تجاوز کند.
  - تقاضای مشتری‌هایی که بیش از ظرفیت وسایل نقلیه است، برش داده شود.
- سایر پارامترها و متغیرهای مورد استفاده در مسأله در ادامه معرفی می‌گردند.

$V$ : مجموعه تعداد وسایل نقلیه موجود در هر دپو

$M$ : یک عدد بزرگ

متغیرهای تصمیم در این مسأله نیز شامل موارد زیر است:

$$x_{ijp} = \begin{cases} 1 & \text{اگر وسیله نقلیه } p \text{ م به طور مستقیم از گره } i \text{ به} \\ & \text{گره } j \text{ سفر کند.} \\ 0 & \text{در غیر این صورت} \end{cases}$$

$$y_k = \begin{cases} 1 & \text{اگر دپوی } k \text{ ام احداث شود.} \\ 0 & \text{در غیر این صورت} \end{cases}$$

$$z_{ik} = \begin{cases} 1 & \text{اگر مشتری } i \text{ ام به دپوی } k \text{ ام اختصاص یابد.} \\ 0 & \text{در غیر این صورت} \end{cases}$$

$u_{ijp}$ : مقدار بار تحویلی موجود در وسیله نقلیه  $p$  ام در طول مسیر  $i$  به  $j$

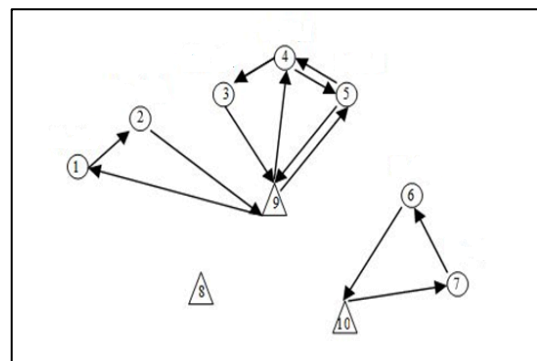
$v_{ijp}$ : مقدار بار برداشتی موجود در وسیله نقلیه  $p$  ام در طول مسیر  $i$  به  $j$

$f_{ijp}$ : کسری از تقاضای تحویل مشتری  $i$  ام که توسط وسیله نقلیه  $p$  ام پس از طی مسیر  $i$  به  $j$ ، تحویل می‌گردد.

وجود ندارد که خواسته‌های مکانیابی، مسیریابی، برداشت و تحویل و برش در تقاضای مشتریان را به صورت همزمان برآورده نماید. هدف این مقاله ارائه یک مدل ریاضی جدید و روش حل برای مسأله‌ای با ویژگی‌های اشاره شده است.

## ۲- شرح مسأله

در مسأله مکانیابی- مسیریابی با برداشت و تحویل همزمان و بارهای برش‌یافته در ارتباط با موضوع دپوها، با توجه به کاربرد مسأله، می‌توان به دو روش از مسأله بهره برد. یا فرض می‌شود که تعدادی نقطه کاندید برای احداث دپو موجود است و در عمل می‌خواهیم کمترین تعداد و مناسب‌ترین این نقاط برای احداث دپو انتخاب شود و یا اینکه تعداد مشخصی دپو موجود است که در این صورت هدف، استفاده از کمترین تعداد دپوها و مناسب‌ترین آنها است. همچنین این مسأله شامل مجموعه‌ای از مشتری‌ها می‌باشد که هر دو نوع تقاضای تحویل و برداشت مرتبط با این مشتری‌ها باید به صورت همزمان برآورده گردند. همان‌طور که قبلاً هم اشاره شد، هدف از حل این مسأله تعیین تعداد مناسبی از نقاط کاندید برای احداث دپوی جدید و یا استفاده از دپوهای موجود است به طوری که مسیره‌های ایجاد شده از دپوهای احداث شده به مشتری‌ها تمامی تقاضاهای آنها را با رعایت شرایط برش در تقاضاها، برآورده نمایند و کمترین هزینه را بر جای گذارند. برای درک بهتر مسأله مثال تصویری در شکل (۱) ارائه شده است. این مثال شمایی کلی از یک جواب‌شدنی برای مسأله مطرح شده در این مقاله را نشان می‌دهد. این مثال شامل ۳ دپو و ۷ مشتری است. شماره‌های ۱ تا ۷ مشتری‌ها را نمایش می‌دهند. شماره‌های ۸ تا ۱۰ نیز بیانگر دپوها هستند. توالی کمان‌ها مسیره‌های وسایل نقلیه را نمایش می‌دهند. همان‌گونه که مشاهده می‌شود در این جواب‌شدنی دپوهای شماره ۹ و ۱۰ برای خدمت‌رسانی باز می‌باشند و نیازی به استفاده از دپوی شماره ۸ نمی‌باشد. علاوه بر این از این شکل به وضوح می‌توان دریافت که تقاضای مشتری‌های ۴ و ۵ بیش از ظرفیت وسیله نقلیه بوده و در نتیجه تقاضای آنها برش‌یافته و با بیش از یک وسیله نقلیه (در این جا با دو وسیله نقلیه) خدمت‌دهی شده‌اند.



شکل (۱): شمایی کلی از یک جواب‌شدنی برای مسأله پیشنهادی

$$\sum_{j \in N_C} \sum_{p \in V} v_{jkp} = \sum_{j \in N_C} z_{jk} p_j \quad \forall k \in N_D \quad (14)$$

$$\sum_{j \in N_C} \sum_{p \in V} v_{jkp} = 0 \quad \forall k \in N_D \quad (15)$$

$$\sum_{i \in N} f_{ijp} d_j \leq CV \quad \forall j \in N_C, \forall p \in V \quad (16)$$

$$\sum_{i \in N} w_{ijp} p_j \leq CV \quad \forall j \in N_C, \forall p \in V \quad (17)$$

$$\sum_{i \in N, i \neq j} \sum_{p \in V} f_{ijp} = 1 \quad \forall j \in N_C \quad (18)$$

$$\sum_{i \in N, i \neq j} \sum_{p \in V} w_{ijp} = 1 \quad \forall j \in N_C \quad (19)$$

$$f_{ijp} \leq x_{ijp} \quad \forall j \in N_C, \forall i \in N, i \neq j, \forall p \in V \quad (20)$$

$$w_{ijp} \leq x_{ijp} \quad \forall j \in N_C, \forall i \in N, i \neq j, \forall p \in V \quad (21)$$

$$f_{ijp} d_j \leq u_{ijp} \quad \forall j \in N_C, \forall i \in N, i \neq j, \forall p \in V \quad (22)$$

$$\sum_{r \in N} w_{rip} p_i - M(1 - x_{ijp}) \leq v_{ijp} \quad \forall j \in N, \forall i \in N_C, i \neq j, \forall p \in V \quad (23)$$

$$u_{ijp} \leq CV x_{ijp} - \sum_{r \in N} f_{rip} d_i + M(1 - x_{ijp}) \quad \forall j \in N, \forall i \in N_C, i \neq j, \forall p \in V \quad (24)$$

$$u_{ijp} \leq CV x_{ijp} \quad \forall j \in N, \forall i \in N_C, i \neq j, \forall p \in V \quad (25)$$

$$v_{ijp} \leq CV x_{ijp} - w_{ijp} p_j \quad \forall j \in N, \forall i \in N_C, i \neq j, \forall p \in V \quad (26)$$

$$f_{ikp} = 0 \quad \forall i \in N_C, \forall k \in N_D, \forall p \in V \quad (27)$$

$$w_{ikp} = 0 \quad \forall i \in N_C, \forall k \in N_D, \forall p \in V \quad (28)$$

$$\sum_{i \in N_C} x_{kip} \leq 1 \quad \forall k \in N_D, \forall p \in V \quad (29)$$

$$x_{ijp} \in \{0,1\} \quad \forall i, j \in N, \forall p \in V \quad (30)$$

$$z_{ik} \in \{0,1\} \quad \forall i \in N_C, \forall k \in N_D \quad (31)$$

$$y_k \in \{0,1\} \quad \forall k \in N_D \quad (32)$$

رابطه (۱) تابع هدف مدل را نشان می‌دهد. این تابع مجموع سه عبارت است که شامل هزینه مسیره‌های طی شده، هزینه احداث دپوها و هزینه ثابت وسایل نقلیه به‌کار گرفته شده می‌باشد. رابطه (۲) بیان می‌کند که هر وسیله از یک گره مشتری فقط به یک گره دیگر می‌تواند برود. رابطه (۳) تضمین می‌کند که یال‌های ورودی و

$w_{ijp}$ : کسری از تقاضای برداشت مشتری  $z$  ام که توسط وسیله نقلیه  $p$  ام پس از طی مسیر  $i$  به  $j$ ، برداشت می‌گردد. مدل طراحی شده برای مسأله مورد بررسی در ادامه ارائه می‌شود:

$$\min \sum_{i \in N} \sum_{j \in N} \sum_{p \in V} c_{ij} x_{ijp} + \sum_{k \in N_D} FD_k y_k + \sum_{k \in N_D} \sum_{i \in N_C} \sum_{p \in V} FV x_{kip} \quad (1)$$

Subject to:

$$\sum_{j \in N} x_{ijp} \leq 1 \quad \forall i \in N_C, \forall p \in V \quad (2)$$

$$\sum_{j \in N} x_{jip} - \sum_{j \in N} x_{ijp} = 0 \quad \forall i \in N, \forall p \in V \quad (3)$$

$$\sum_{k \in N_D} z_{ik} = 1 \quad \forall i \in N_C \quad (4)$$

$$x_{ikp} \leq z_{ik} \quad \forall i \in N_C, \forall k \in N_D, \forall p \in V \quad (5)$$

$$x_{kip} \leq z_{ik} \quad \forall i \in N_C, \forall k \in N_D, \forall p \in V \quad (6)$$

$$x_{ijp} + z_{ik} + \sum_{r \in N_D, r \neq k} z_{jr} \leq 2 \quad \forall i, j \in N_C, i \neq j, \forall k \in N_D, \forall p \in V \quad (7)$$

$$\sum_{i \in N_C} d_i z_{ik} \leq CD_k y_k \quad \forall k \in N_D \quad (8)$$

$$\sum_{i \in N_C} p_i z_{ik} \leq CD_k y_k \quad \forall k \in N_D \quad (9)$$

$$\sum_{i \in N} u_{lip} - \sum_{j \in N} u_{ijp} = \sum_{j \in N} f_{ijp} d_i \quad \forall i \in N_C, \forall p \in V \quad (10)$$

$$\sum_{j \in N} v_{ijp} - \sum_{j \in N} v_{jip} = \sum_{j \in N} w_{jip} p_i \quad \forall i \in N_C, \forall p \in V \quad (11)$$

$$\sum_{j \in N_C} \sum_{p \in V} u_{kjp} = \sum_{j \in N_C} z_{jk} d_j \quad \forall k \in N_D \quad (12)$$

$$\sum_{j \in N_C} \sum_{p \in V} u_{jkp} = 0 \quad \forall k \in N_D \quad (13)$$

تابع هزینه را حداقل کند). این روش توسط جان هالند طی دهه‌های ۱۹۶۰ و ۱۹۷۰ میلادی ارائه شد و توسط یکی از شاگردانش به نام دیوید گلدبرگ تعمیم داده شد [۲۴].

الگوریتم حل ارائه شده در این قسمت الگوریتمی دو مرحله‌ای می‌باشد. این الگوریتم از ساختار الگوریتم‌های ژنتیک گسسته و ژنتیک پیوسته به صورت همزمان بهره می‌برد. الگوریتم ژنتیک گسسته برای یافتن بهترین توالی (جایگشت) از مشتری‌ها و دپوها مورد استفاده قرار می‌گیرد و مسیرهای مناسب را برای وسایل نقلیه تعیین می‌کند. الگوریتم ژنتیک پیوسته به دنبال یافتن بهترین ترکیب برش برای تقاضاهایی برداشت و تحویل مشتریانی است که از ظرفیت وسیله نقلیه بیشتر هستند و به ناچار باید تفکیک شده و توسط چندین وسیله خدمت‌دهی شوند. در ادامه به بررسی جزئیات مراحل اجرای الگوریتم پیشنهادی پرداخته می‌شود.

### کروموزوم و جمعیت اولیه

در روش DCGA دو نوع کروموزوم مورد استفاده قرار می‌گیرد. یک نوع برای قسمت گسسته الگوریتم و نوع دیگر برای قسمت پیوسته الگوریتم استفاده می‌شود. در ادامه برای درک بهتر نحوه نمایش کروموزوم‌های معرفی شده از یک مثال استفاده می‌کنیم. این مثال شامل ۷ مشتری و ۳ دپو است. در هر دپو ۳ وسیله نقلیه موجود است که ظرفیت هر کدام ۱۵۰ واحد می‌باشد. در روند ایجاد کروموزوم بخش گسسته ابتدا مقادیر تقاضای تمامی مشتری‌ها بررسی می‌گردد و تعداد ژن‌های مربوط به مشتری‌هایی که تقاضای آنها بیش از ظرفیت وسیله نقلیه باشد، به تعداد وسایل نقلیه موجود در هر کروموزوم در نظر گرفته می‌شود. مثلاً در مثالی که اینجا بررسی می‌گردد تقاضای مشتری‌های شماره ۴ و ۷ باید برش یابند به این منظور برای هر کدام از این مشتری‌ها در کروموزوم گسسته ۳ (تعداد وسایل نقلیه موجود) ژن در نظر گرفته می‌شود. به این ترتیب در کروموزوم گسسته‌ای که تشکیل می‌گردد ژن‌های شماره ۱، ۲، ۳، ۵ و ۶ مربوط به مشتری‌های شماره ۱، ۲، ۳، ۵ و ۶ می‌باشند. ژن‌های شماره ۴، ۸ و ۹ مربوط به مشتری شماره ۴ می‌باشند و ژن‌های شماره ۷، ۱۰ و ۱۱ مربوط به مشتری شماره ۷ می‌باشند. و ژن‌های شماره ۱۲، ۱۳ و ۱۴ هم بیانگر دپوها می‌باشد. به این ترتیب یک جایگشت تصادفی از اعداد ۱ تا ۱۴ می‌تواند ساختار یک کروموزوم گسسته را مانند شکل ۲ نمایش دهد.

۱۴	۵	۳	۹	۷	۱۲	۲	۸	۴	۶	۱۳	۱	۱۰	۱۱
----	---	---	---	---	----	---	---	---	---	----	---	----	----

شکل (۲): ساختار یک کروموزوم گسسته مورد استفاده در الگوریتم DCGA پیشنهادی

کروموزوم نوع دوم برای بخش پیوسته الگوریتم استفاده می‌شود. نحوه تشکیل این کروموزوم به این ترتیب است که برای هر کدام از مشتری‌هایی که تقاضای آنها بیش از ظرفیت نقلیه است به صورت تصادفی یک ترکیب برش به تعداد وسایل نقلیه موجود ایجاد

خروجی به یک گره، ناشی از یک وسیله نقلیه، برابر باشند. رابطه (۴) بیان می‌کند که هر مشتری می‌تواند تنها به یکی از دپوها اختصاص یابد. رابطه (۵) بیان می‌کند که ایجاد یال از یک دپو به یک مشتری زمانی می‌تواند رخ دهد که مشتری ذکر شده به آن دپو اختصاص یافته باشد. رابطه (۶) نیز بیان می‌کند که ایجاد یال از یک مشتری به یک دپو زمانی می‌تواند صورت بگیرد که مشتری ذکر شده به آن دپو اختصاص یافته باشد رابطه (۷) از ایجاد مسیرهای غیرمجاز جلوگیری می‌کند. رابطه‌های (۸) و (۹) به ترتیب بررسی می‌کنند که محموله‌های تحویلی و محموله‌های برداشتی از ظرفیت دپو تجاوز نکنند. رابطه‌های (۱۰) تا (۱۵) محدودیت‌های مربوط به جریان در شبکه هستند. رابطه (۱۶) تضمین می‌کند که مجموع محموله‌هایی که توسط یک وسیله نقلیه برای تحویل حمل می‌شود از ظرفیت وسیله نقلیه تجاوز ننماید. رابطه (۱۷) تضمین می‌کند که مجموع محموله‌هایی که توسط یک وسیله نقلیه برداشت می‌شود از ظرفیت وسیله نقلیه تجاوز ننماید. رابطه‌های (۱۸) و (۱۹) به ترتیب نشان‌دهنده یک بودن مجموع کسره‌های برش در تقاضای تحویل و تقاضای برداشت هستند. رابطه‌های (۲۰) و (۲۱) تضمین می‌کنند که برش در تقاضای تحویل و تقاضای برداشت یک مشتری تنها زمانی امکان رخ دادن دارد که مسیری به آن مشتری ایجاد شده باشد. رابطه‌های (۲۲) تا (۲۶) حدود بالا و پایین متغیرهای  $V_{ijp}$  و  $U_{ijp}$  را نشان می‌دهند. رابطه‌های (۲۷) و (۲۸) بیان می‌کنند که کسره‌های برش در دپوها صفر می‌باشد زیرا در گره‌های دپو به مانند مشتری‌ها تقاضای تحویل یا برداشت وجود ندارد. رابطه (۲۹) تضمین می‌کند که هر وسیله نقلیه‌ای که از یک دپو خارج می‌شود تنها اجازه ورود به یکی از مشتری‌ها را دارد، یعنی نمی‌تواند به طور همزمان به بیش از یک مشتری خدمت‌رسانی نماید. و در نهایت رابطه‌های (۳۰) تا (۳۲) متغیرهای باینری مسأله را نمایش می‌دهند.

### ۳- الگوریتم‌های فرا ابتکاری پیشنهادی

با توجه به این که مسأله مکانیابی-مسیریابی بدون در نظر داشتن محدودیت‌های مسأله پیشنهادی در این مقاله مسأله ای NP-hard است بنابراین مسأله معرفی شده نیز NP-hard بوده و حل دقیق آن در ابعاد بزرگ، سخت و تقریباً غیرممکن است [۲۳]. بنابراین در مواجهه با این مشکل دو الگوریتم فرا ابتکاری پیشنهاد می‌شود که در ادامه با جزئیات تشریح می‌شوند.

#### ۳-۱- الگوریتم DCGA برای MDCLRPSPDS<sup>۲</sup>

الگوریتم ژنتیک (GA) یک تکنیک جستجو و بهینه‌سازی مبتنی بر اصول ژنتیک و انتخاب طبیعی است. GA این امکان را می‌دهد که جمعیتی متشکل از تعداد زیادی از افراد طبق قوانین خاص مربوط به انتخاب به وضعیتی درآیند که «شایستگی» را حداکثر کند (یعنی

1. Discrete and Continuous Genetic Algorithm
2. Multi depots Capacitated Location-Routing Problem with Simultaneous Pickup and Delivery and Split Loads

$$\alpha = [random * N_{var}]$$

که در این رابطه  $N_{var}$  تعداد ژن‌های هر کروموزوم است. فرض کنید والدین به صورت زیر باشند:

که  $m$  و  $d$  به ترتیب بیانگر والد مادر و پدر هستند. سپس

$$P_1 = [p_{m1}p_{m2} \dots p_{ma} \dots p_{mN_{var}}]$$

$$P_2 = [p_{d1}p_{d2} \dots p_{da} \dots p_{dN_{var}}]$$

متغیرهای انتخاب شده برای تشکیل متغیرهای جدیدی که در فرزندان ظاهر می‌شوند، با یکدیگر ترکیب می‌گردند.

$$p_{new1} = p_{ma} - \beta[p_{ma} - p_{da}]$$

$$p_{new2} = p_{da} + \beta[p_{ma} - p_{da}]$$

که  $\beta$  نیز یک مقدار تصادفی بین صفر و یک است. آخرین گام کامل کردن ادغام به وسیله بقیه کروموزوم است:

اگر اولین متغیر کروموزوم انتخاب شود، آن‌گاه فقط متغیرهای سمت راست متغیر انتخاب شده معاوضه می‌شوند. اگر آخرین متغیر کروموزوم انتخاب شود، آن‌گاه تنها متغیرهای سمت چپ متغیر انتخاب شده رد و بدل می‌گردند.

$$Child_1 = [p_{m1}p_{m2} \dots p_{new1} \dots p_{dN_{var}}]$$

$$Child_2 = [p_{d1}p_{d2} \dots p_{new2} \dots p_{mN_{var}}]$$

### عملگر جهش در بخش ژنتیک پیوسته

ابتدا با ضرب نرخ جهش در تعداد کل ژن‌هایی که می‌توانند در جمعیت جهش یابند، تعداد جهش‌ها به دست می‌آید. سپس اعدادی تصادفی جهت انتخاب سطر و ستون متغیرهایی که باید جهش یابند انتخاب می‌شوند. متغیر جهش یافته با یک متغیر تصادفی جایگزین می‌گردد. اگر اعداد تصادفی انتخاب شده به صورت (۴، ۴، ۷) برای ردیف‌ها و (۱، ۲، ۱) برای ستون‌ها باشند، به این معنی است که اولین زوج تصادفی (۴، ۱) است. بنابراین، مقدار قرار گرفته در سطر ۴ و ستون ۱ ماتریس جمعیت (اولین ژن در کروموزوم چهارم) با یک عدد تصادفی یکنواخت بین صفر و یک جایگزین می‌شود.

### تولید جواب شدنی

برای ایجاد مسیره‌ها و تولید جواب شدنی، عناصر توالی (کروموزوم) ایجاد شده در بخش ۳-۱ وارد فرآیندی می‌شود که طبق آن برای جلوگیری از تجاوز از حداکثر ظرفیت و وسیله نقلیه و حداکثر ظرفیت هر دپو دسته‌بندی می‌شوند. برای این کار، ابتدا توالی دپوها از توالی ابتدایی جدا می‌گردد (شکل (۵))، سپس با شروع از مشتری اول در توالی، تقاضاهای برداشت و تحویل مشتری‌ها برآورده می‌گردند. این کار تا زمانی که ظرفیت یک وسیله نقلیه از حداکثر میزان خود تجاوز ننماید و همچنین ظرفیت دپوی مربوطه برای برآورده ساختن نیازهای مربوط به مشتری‌ها نقض نگردد، ادامه می‌یابد. به این ترتیب مسیره‌های مجاز و دپوهای باز و دپوهای بسته ایجاد می‌شوند.

می‌گردد (شکل (۳)).

از به هم پیوستن تمامی ترکیب‌های برش، کروموزوم مورد نظر ما برای بخش پیوسته الگوریتم به منظور یافتن بهترین ترکیب برش تشکیل می‌گردد (شکل (۴)).

در این الگوریتم برای هر کروموزوم گسسته، الگوریتم ژنتیک پیوسته اجرا می‌گردد تا مناسب‌ترین ترکیب برش برای آن کروموزوم به دست آید. پس برای اجرای الگوریتم ابتدا یک جمعیت اولیه از کروموزوم‌های گسسته به صورت تصادفی ایجاد می‌گردد و سپس

مشتری‌ها	۱	۲	۳	۴	۵	۶	۷
تقاضای تحویل	۷۰	۱۰	۱۶۰	۱۹۰	۱۵	۳۰	۱۸۵
تقاضای برداشت	۲۰	۵	۳۰	۱۰	۱۰	۱۵	۱۵



شکل (۳): نحوه ایجاد ترکیب‌های برش

۰.۳۵	۰.۳	۰.۳۵	۰.۲۵	۰.۶	۰.۱۵
------	-----	------	------	-----	------

شکل (۴): ساختار یک کروموزوم پیوسته مورد استفاده در

### الگوریتم DCGA

برای هر کروموزوم گسسته یک جمعیت اولیه از کروموزوم‌های پیوسته (ترکیب برش) به صورت تصادفی تشکیل می‌شود. در ادامه برای هر کروموزوم گسسته فرآیند الگوریتم ژنتیک پیوسته اجرا می‌گردد. بعد از این فرآیند، نسل بعدی کروموزوم‌های گسسته ایجاد می‌شود و روند قبلی مجدداً تکرار می‌شود. در بخش بعد عناصر اصلی ژنتیک پیوسته و ژنتیک گسسته مورد بحث و بررسی قرار می‌گیرد.

### عملگرها در ژنتیک گسسته و ژنتیک پیوسته

ساختار کلی و عملکرد ژنتیک پیوسته و ژنتیک گسسته مشابه است و تفاوت‌چندانی با یکدیگر ندارند. تفاوت‌های موجود بین آنها در عملگرهای تقاطع و عملگر جهش می‌باشد که در ادامه شرح داده می‌شوند.

### عملگرهای تقاطع و جهش در بخش ژنتیک گسسته

عملگر تقاطع مورد استفاده در بخش ژنتیک گسسته الگوریتم پیشنهادی، عملگر تقاطع تک‌نقطه‌ای می‌باشد.

همچنین در این بخش از الگوریتم در هر مرتبه از اجرای عملگر جهش به طور تصادفی از یکی از سه عملگر جهش تعویض، جایگذاری و معکوس‌سازی استفاده شده است.

### عملگر تقاطع در بخش ژنتیک پیوسته

در این بخش روشی که در DCGA برای فرایند تقاطع برای کروموزوم پیوسته استفاده می‌شود را شرح می‌دهیم. این روش با انتخاب تصادفی یک متغیر در اولین جفت والدین به عنوان نقطه تقاطع آغاز می‌شود:

شد [۲۵]. این الگوریتم یک الگوریتم محاسبه تکاملی الهام گرفته از طبیعت و مبتنی بر تکرار می‌باشد. در گام نخست الگوریتم، ذرات با موقعیت‌ها و سرعت‌های تصادفی تولید می‌شوند. در طی اجرای الگوریتم، موقعیت ذره و سرعت هر ذره در مرحله  $t+1$  از  $x_j^i[t]$  و  $v_j^i[t+1]$  از الگوریتم از روی اطلاعات مرحله قبلی ایجاد می‌شود. اگر منظور از  $z_j$  مولفه  $z$ ام از بردار  $z$  باشد، و  $i$  نشان‌دهنده ذره خاص، آنگاه روابطی که سرعت و موقعیت ذرات را تغییر می‌دهند، عبارتند از:

$$v_j^i[t+1] = wv_j^i[t] + c_1r_1(x_j^{i,best}[t] - x_j^i[t]) + c_2r_2(x_j^{gbest}[t] - x_j^i[t])$$

$$x_j^i[t+1] = x_j^i[t] + v_j^i[t+1]$$

در این روابط  $x_j^{i,best}[t]$  بهترین موقعیت ذره  $i$  در  $x_j^{gbest}[t]$  بهترین موقعیت تمام ذرات،  $w$  ضریب اینرسی،  $r_1$  و  $r_2$  اعداد تصادفی در بازه  $[0,1]$  با توزیع یکنواخت، و همچنین  $c_1$  و  $c_2$  ضرایب یادگیری هستند.  $r_1$  و  $r_2$  موجب می‌شوند تا نوعی گوناگونی در جواب‌ها به وجود آید و به این نحو جستجوی کامل‌تری در فضای جواب انجام گیرد. همچنین  $c_1$  ضریب یادگیری مربوط به تجارب شخصی هر ذره و  $c_2$  ضریب یادگیری مربوط به تجارب کل ذرات می‌باشد.

#### تنظیم پارامتر الگوریتم DGAPSO

در این بخش نیز برای تنظیم پارامترها از آنالیز تاگوچی استفاده شده است که مقادیر نهایی در جدول (۲) نشان داده شده است. شکل (۷) این نتایج را برای دسته مسائل کوچک نشان می‌دهد.

جدول (۲): تنظیم پارامترهای الگوریتم DGAPSO

پارامتر	نوع نمونه مسائل	مقادیر پیشنهادی	مقدار انتخاب شده
تعداد تکرار ژنتیک	کوچک	۳۰، ۲۰، ۱۰	۳۰
گسسته	متوسط	۷۰، ۶۰، ۵۰	۷۰
اندازه جامعه ژنتیک	کوچک	۲۰، ۱۵، ۱۰	۲۰
گسسته	متوسط	۲۰، ۱۵، ۱۰	۱۵
نرخ عملگر تقاطع	کوچک	۰.۸، ۰.۷، ۰.۶	۰.۶
ژنتیک گسسته	متوسط	۰.۸، ۰.۷، ۰.۶	۰.۷
نرخ عملگر جهش	کوچک	۰.۳، ۰.۲، ۰.۱	۰.۳
ژنتیک گسسته	متوسط	۰.۳، ۰.۲، ۰.۱	۰.۲
تعداد تکرار PSO	کوچک	۶، ۵، ۴	۴
متوسط	متوسط	۲۰، ۱۵، ۱۰	۱۵
تعداد ذرات PSO	کوچک	۵، ۴، ۳	۵
متوسط	متوسط	۲۰، ۱۵، ۱۰	۱۵
وزن اینرسی PSO	کوچک	۱.۵، ۱، ۰.۵	۱
متوسط	متوسط	۱.۵، ۱، ۰.۵	۱
ضریب کاهش وزن اینرسی PSO	کوچک	۰.۹۹، ۰.۹۵، ۰.۹	۰.۹۹
متوسط	متوسط	۰.۹۹، ۰.۹۵، ۰.۹	۰.۹۹
ضریب یادگیری شخصی PSO	کوچک	۲.۵، ۲، ۱.۵	۲
متوسط	متوسط	۲.۵، ۲، ۱.۵	۲.۵
ضریب یادگیری کل PSO	کوچک	۲، ۱.۵، ۱	۲
متوسط	متوسط	۲، ۱.۵، ۱	۲

#### تنظیم پارامتر الگوریتم DCGA

برای تنظیم پارامترهای الگوریتم پیشنهادی از آنالیز تاگوچی بهره گرفته شده است.

در این الگوریتم هفت پارامتر برای تعیین مقادیر مناسب برای اجرای الگوریتم مورد بررسی قرار گرفتند. مقادیر مناسب پیشنهادی، به لحاظ تجربی و مناسب‌ترین مقادیر برای این پارامترها در جدول (۱) ارائه شده است. شکل (۶) این نتایج را برای دسته مسائل کوچک نشان می‌دهد.

۱۴	۵	۳	۹	۷	۱۲	۲	۸	۴	۶	۱۳	۱	۱۰	۱۱
----	---	---	---	---	----	---	---	---	---	----	---	----	----



توالی دیوها

۱۴	۱۲	۱۳
----	----	----

توالی مشتری‌ها

۵	۳	۹	۷	۲	۸	۴	۶	۱	۱۰	۱۱
---	---	---	---	---	---	---	---	---	----	----

شکل (۵): جداسازی ترتیب دیوها از ترتیب مشتری‌ها

جدول (۱): تنظیم پارامترهای الگوریتم DCGA

مقدار انتخاب شده	مقادیر پیشنهادی	نوع نمونه مسائل	پارامتر
۲۰	۳۰، ۲۰، ۱۰	کوچک	تعداد تکرار ژنتیک
۷۰	۷۰، ۶۰، ۵۰	متوسط	گسسته
۲۰	۲۰، ۱۵، ۱۰	کوچک	اندازه جامعه ژنتیک
۲۰	۲۰، ۱۵، ۱۰	متوسط	گسسته
۰.۶	۰.۸، ۰.۷، ۰.۶	کوچک	نرخ عملگر تقاطع
۰.۷	۰.۸، ۰.۷، ۰.۶	متوسط	ژنتیک گسسته
۰.۳	۰.۳، ۰.۲، ۰.۱	کوچک	نرخ عملگر جهش
۰.۳	۰.۳، ۰.۲، ۰.۱	متوسط	ژنتیک گسسته
۶	۶، ۵، ۴	کوچک	تعداد تکرار ژنتیک
۱۵	۲۰، ۱۵، ۱۰	متوسط	پیوسته
۶	۶، ۵، ۴	کوچک	اندازه جامعه ژنتیک
۱۰	۲۰، ۱۵، ۱۰	متوسط	پیوسته
۰.۳	۰.۳، ۰.۲، ۰.۱	کوچک	نرخ عملگر جهش
۰.۳	۰.۳، ۰.۲، ۰.۱	متوسط	ژنتیک پیوسته

#### ۲-۳ الگوریتم DGAPSO<sup>۱</sup> برای MDCLRPSPDS

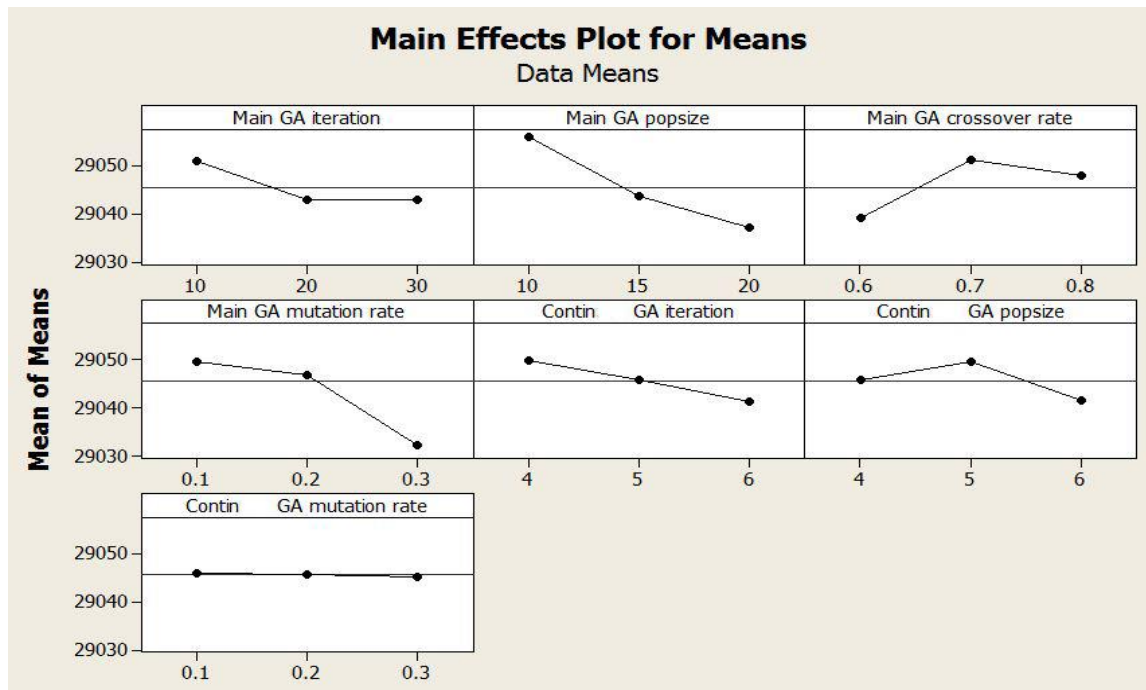
این الگوریتم نیز مشابه با الگوریتم پیشنهادی DCGA شامل ترکیب دو الگوریتم است. در این الگوریتم GA به دنبال یافتن بهترین توالی از مشتری‌ها و دیوها است. یعنی قسمت گسسته الگوریتم DGAPSO به عهده الگوریتم GA است و الگوریتم PSO به دنبال یافتن بهترین ترکیب برش است که قسمت پیوسته الگوریتم ترکیبی است. ساختار الگوریتم GA گسسته در قسمت پیش شرح داده شد. بنابراین در اینجا تنها به معرفی الگوریتم PSO یا بهینه‌سازی انبوه می‌پردازیم.

#### الگوریتم بهینه‌سازی انبوه ذرات (PSO)

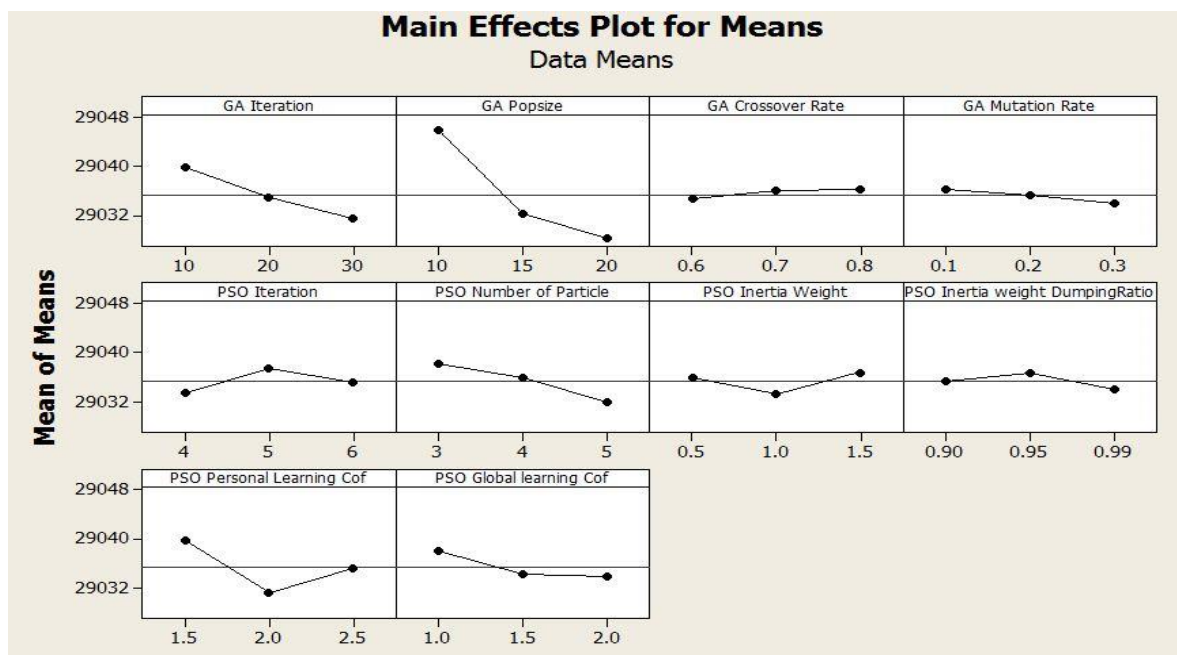
ایده اصلی این الگوریتم برای اولین بار توسط کندی و ایبره‌ارت ارائه

<sup>۱</sup> Discrete Genetic Algorithm Particle Swarm Optimization





شکل (۶): تنظیم پارامتر برای مسائل کوچک در الگوریتم DCGA



شکل (۷): تنظیم پارامتر برای مسائل کوچک در الگوریتم DGAPSO

می‌شود. ویژگی‌های این مسائل در جدول (۳) و جدول (۴) ارائه شده است. نتایج حاصل از حل این نمونه مسائل با استفاده از حل‌کننده سیپلکس (CPLEX) در نرم‌افزار گمز و الگوریتم‌های DCGA و DGAPSO (کد شده در نرم‌افزار متلب)، که بر روی یک کامپیوتر با مشخصات CPU Corei5 2.3 GHz و RAM 4G اجرا گردید، در جدول (۵) برای مسائل کوچک و جدول (۶) برای مسائل متوسط ارائه شده است.

#### ۴- آزمایشات عددی

مسائل نمونه استفاده شده در این بخش با استفاده از داده‌های موجود در مقاله کارااوغلان و همکاران [۹] طراحی شده است. با این توضیح که به‌منظور ایجاد ویژگی اجتناب‌ناپذیری برش در تقاضاها، تقاضای برخی از مشتری‌ها افزایش داده شد به نحوی که از ظرفیت وسیله نقلیه تجاوز کند. این مسائل در ۴ پارامتر از یکدیگر متمایز می‌گردند که عبارتند از: تعداد مشتری‌ها، تعداد دپوها، تعداد وسایل نقلیه موجود در هر دپو و تعداد مشتریانی که تقاضای آنها با برش مواجه



جدول (۳): ویژگی‌های اصلی مسائل نمونه کوچک طراحی شده

شماره	تعداد	تعداد	تعداد وسایل	تعداد مشتریانی
مثال	مشتری	دیو	نقلیه در هر دیو	که تقاضای آنها برش می‌یابد
۱	۱۰	۲	۳	۲
۲	۱۰	۲	۳	۳
۳	۱۰	۲	۴	۲
۴	۱۰	۲	۴	۳
۵	۱۰	۳	۳	۲
۶	۱۰	۳	۳	۳
۷	۱۰	۳	۴	۲
۸	۱۰	۳	۴	۳
۹	۱۱	۲	۳	۲
۱۰	۱۱	۲	۳	۳
۱۱	۱۱	۲	۴	۲
۱۲	۱۱	۲	۴	۳
۱۳	۱۱	۳	۳	۲
۱۴	۱۱	۳	۳	۳
۱۵	۱۱	۳	۴	۲
۱۶	۱۱	۳	۴	۳
۱۷	۱۲	۲	۳	۲
۱۸	۱۲	۲	۳	۳
۱۹	۱۲	۲	۴	۲
۲۰	۱۲	۲	۴	۳
۲۱	۱۲	۳	۳	۲
۲۲	۱۲	۳	۳	۳
۲۳	۱۲	۳	۴	۲
۲۴	۱۲	۳	۴	۳

جدول (۴): ویژگی‌های اصلی مسائل نمونه کوچک طراحی شده

شماره	تعداد	تعداد	تعداد	تعداد مشتریانی
مثال	مشتری	دیو	وسایل	که تقاضای آنها برش می‌یابد
			نقلیه در هر دیو	
۱	۱۰	۲	۳	۲
۲	۱۰	۲	۳	۳
۳	۱۰	۲	۴	۲
۴	۱۰	۲	۴	۳
۵	۱۰	۳	۳	۲
۶	۱۰	۳	۳	۳
۷	۱۰	۳	۴	۲
۸	۱۰	۳	۴	۳
۹	۱۱	۲	۳	۲
۱۰	۱۱	۲	۳	۳
۱۱	۱۱	۲	۴	۲
۱۲	۱۱	۲	۴	۳
۱۳	۱۱	۳	۳	۲
۱۴	۱۱	۳	۳	۳
۱۵	۱۱	۳	۴	۲
۱۶	۱۱	۳	۴	۳
۱۷	۱۲	۲	۳	۲
۱۸	۱۲	۲	۳	۳
۱۹	۱۲	۲	۴	۲
۲۰	۱۲	۲	۴	۳
۲۱	۱۲	۳	۳	۲
۲۲	۱۲	۳	۳	۳
۲۳	۱۲	۳	۴	۲
۲۴	۱۲	۳	۴	۳

برای سنجش اعتبار الگوریتم‌ها و مقایسه آنها با یکدیگر از چند شاخص استفاده شده است که در ادامه معرفی می‌شوند.

شاخص  $RPD$  یا درصد انحراف نسبی برای هر کدام از الگوریتم‌های پیشنهادی با رابطه (۳۳) تعریف شده است:

$$RPD = (Alg_S - Alg_{BS}) / Alg_{BS} \quad (33)$$

در این رابطه  $Alg_S$  و  $Alg_{BS}$  به ترتیب میانگین جواب و بهترین جواب به دست آمده توسط الگوریتم پیشنهادی در ۱۰ بار اجرای هر مسأله نمونه است.

شاخص‌های  $GAP_S$  و  $GAP_T$  به ترتیب درصد اختلاف جواب و زمان اجرای الگوریتم‌ها را نشان می‌دهند که از روابط (۳۴) و (۳۵) محاسبه می‌شوند:

$$GAP_S = (DCGA_S - DGAPSO_S) / DGAPSO_S \quad (35)$$

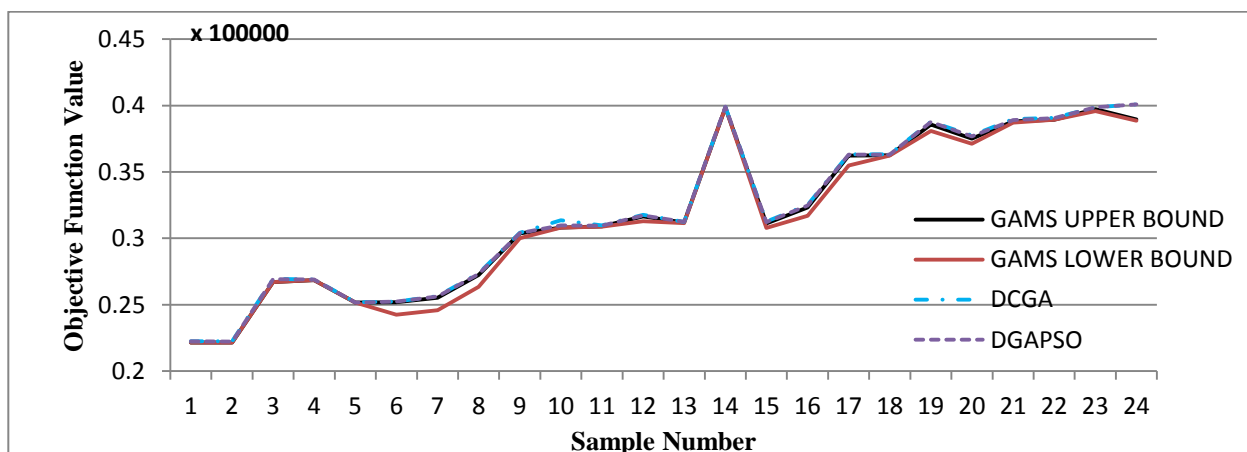
$$GAP_T = (DGAPSO_T - DCGA_T) / DCGA_T \quad (36)$$

در رابطه (۳۴)  $DCGA_S$  و  $DGAPSO_S$  جواب‌های الگوریتم‌های  $DCGA$  و  $DGAPSO$  را نشان می‌دهند. همچنین در رابطه (۳۵)  $DGAPSO_T$  و  $DCGA_T$  بیانگر زمان‌های محاسباتی الگوریتم‌های

نتایج جدول (۵) نشان می‌دهد که حل‌کننده سیپلکس در تعدادی از مسائل نمونه کوچک مدت زمان حداکثر ۱۰۰۰ ثانیه‌ای که برای اجرای نرم‌افزار در نظر گرفته شده، به جواب دقیق می‌رسد و در سایر نمونه‌ها جوابی نزدیک به بهینه را ارائه می‌دهد. همین موضوع نشان می‌دهد که پیچیدگی محاسباتی مسأله پیشنهادی بالا است به طوری که حل‌کننده سیپلکس برای تعداد زیادی از مسائل کوچک طراحی شده قادر نیست در مدت زمان کوتاهی به جواب بهینه دست یابد. عملکرد الگوریتم‌های پیشنهادی در مسائل کوچک خوب است به طوری که در نمونه مسائلی که سیپلکس به جواب بهینه دست می‌یابد الگوریتم‌های پیشنهادی نیز در زمان به مراتب کمتری به جواب بهینه می‌رسند.

جدول (۵): نتایج و زمان محاسباتی حل دقیق و الگوریتم‌های پیشنهادی بر روی مسائل نمونه کوچک در MDCLRSPDS

شماره مثال	نتایج CPLEX			نتایج DCGA			نتایج DGAPSO			GAP <sub>S</sub>	GAP <sub>T</sub>
	حد بالا	حد پایین	زمان	میانگین	زمان	RPD	میانگین	زمان	RPD		
۱	۲۲۱۵۴.۴۲	۲۲۱۵۴.۴۲	۷۳۱	۲۲۱۵۴.۴۲	۹۰.۸۲	۰	۲۲۱۵۴.۴۲	۱۰۳.۶۴	۰	۰	-۰.۱۲
۲	۲۲۱۶۳.۴۲	۲۲۱۶۳.۴۲	۵۸۹.۸	۲۲۱۶۳.۴۲	۱۰۷.۱	۰	۲۲۱۶۳.۴۲	۷۲.۸۵	۰	۰	۰.۴۷
۳	۲۶۷۰۱.۱۳	۲۶۷۰۱.۱۳	۵۹۶.۳	۲۶۷۰۱.۱۳	۱۶۹.۰۱	۰	۲۶۷۰۱.۱۳	۶۲.۲۸	۰	۰	۱.۷۱
۴	۲۶۸۶۰.۱۳	۲۶۸۴۹.۹۲	۱۰۰۰	۲۶۹۰۹.۱۴	۱۷۵.۱۷	۰.۰۰۰۴۷	۲۶۹۰۷.۶۲	۷۱.۱۳	۰.۰۰۰۰۵	۰.۰۰۰۰۵	۱.۴۶
۵	۲۵۱۵۷.۶۴	۲۵۱۵۷.۶۴	۵۵۹.۴	۲۵۱۵۷.۶۴	۸۸.۳۴	۰	۲۵۱۵۷.۶۴	۵۶.۰۶	۰.۰۰۰۰۵۱	۰	۰.۵۷
۶	۲۵۱۹۰.۹۸	۲۴۴۵۵.۹۱	۱۰۰۰	۲۵۲۳۷.۶۴	۱۲۱.۶۱	۰.۰۰۰۰۵۸	۲۵۲۲۵.۷۴	۱۱۲.۸۶	۰.۰۰۱۱۸	۰.۰۰۰۰۴۷	۰.۰۷
۷	۲۵۵۳۰.۰۵	۲۴۵۸۱.۴۷	۱۰۰۰	۲۵۶۵۰.۶۶	۱۰۸.۳۳	۰.۰۰۰۰۲۳	۲۵۶۳۰.۴۲	۶۴.۸۳	۰.۰۰۰۰۷۲	۰.۰۰۰۰۷۸	۰.۶۷
۸	۲۷۲۲۸.۲۱	۲۶۳۳۴.۶۷	۱۰۰۰	۲۷۳۶۰.۹۳	۱۴۸.۴۴	۰.۰۰۰۰۳۵	۲۷۳۰۳۰.۰۴	۸۴.۷۵	۰.۰۰۰۰۶۸	۰.۰۰۰۲۱۲	۰.۷۵
۹	۳۰۳۶۲.۹۳	۳۰۰۱۶.۶۳	۱۰۰۰	۳۰۴۲۲.۲۴	۱۶۸.۴۸	۰.۰۰۰۰۱۴	۳۰۴۰۰۰.۱	۶۱.۰۵	۰.۰۰۰۰۵۱	۰.۰۰۰۰۷۳	۱.۷۵
۱۰	۳۰۷۹۷.۲۹	۳۰۷۹۷.۲۹	۷۷۹.۶	۳۱۶۸۲.۴۵	۱۴۸.۳۱	۰.۰۰۰۰۳۵	۳۰۹۶۸.۲۵	۶۸.۷۶	۰.۰۰۰۰۴۱	۰.۰۰۲۳۰۶	۱.۱۵
۱۱	۳۰۸۹۱.۷	۳۰۸۷۸.۳۷	۱۰۰۰	۳۰۹۷۸.۹۴	۱۰۴.۳۴	۰.۰۰۰۰۹۴	۳۰۹۵۲.۷۶	۶۶.۷۰	۰.۰۰۰۰۲۸	۰.۰۰۰۰۸۴	۰.۵۶
۱۲	۳۱۶۵۰.۹۷	۳۱۲۸۳.۹۱	۱۰۰۰	۳۱۷۷۰.۵۵	۱۴۶.۰۴	۰.۰۰۰۰۸۶	۳۱۷۲۸.۰۲	۸۷.۳۳	۰.۰۰۰۰۴۱	۰.۰۰۱۳۴	۰.۶۷
۱۳	۳۱۱۸۲.۴۵	۳۱۱۵۴.۶۶	۱۰۰۰	۳۱۲۴۶.۱۷	۹۹.۷۶	۰.۰۰۰۰۴۴	۳۱۲۴۵.۰۷	۶۴.۳۳	۰.۰۰۰۰۳۴	۰.۰۰۰۰۰۳	۰.۵۵
۱۴	۳۹۸۵۸.۰۲	۳۹۸۵۶.۹۶	۱۰۰۰	۳۹۸۹۵.۴۱	۱۶۱.۲۶	۰.۰۰۰۰۹۳	۳۹۹۱۲.۲۱	۸۳.۹۸	۰.۰۰۰۰۳۲	-۰.۰۰۰۰۴۲	۰.۹۲
۱۵	۳۱۱۱۲.۱۹	۳۰۷۷۲.۱۶	۱۰۰۰	۳۱۲۶۰.۳۵	۱۱۲.۷۷	۰.۰۰۰۰۲۴	۳۱۲۱۹.۰۷	۶۸.۸۴	۰.۰۰۰۱۴۸	۰.۰۰۱۳۲	۰.۶۳
۱۶	۳۲۳۲۹.۵۹	۳۱۶۹۶.۲۵	۱۰۰۰	۳۲۵۲۰.۹۷	۱۵۲.۶۱	۰.۰۰۰۰۹۸	۳۲۴۶۵.۹۵	۹۴.۴۹	۰.۰۰۰۰۹۴	۰.۰۰۱۶۹	۰.۶۱
۱۷	۳۶۲۴۰.۳۷	۳۵۴۷۶.۷	۱۰۰۰	۳۶۳۱۸.۲۳	۹۵.۲۲	۰.۰۰۰۰۷۹	۳۶۲۹۲.۱۸	۶۰.۱۹	۰.۰۰۰۰۳۸	۰.۰۰۰۰۷۱	۰.۵۸
۱۸	۳۶۲۵۱.۰۴	۳۶۲۴۰	۱۰۰۰	۳۶۳۲۹.۸۹	۱۰۷.۰۸	۰.۰۰۰۰۵۶	۳۶۳۰۸.۱۲	۶۷.۲۸	۰.۰۰۱۱۹	۰.۰۰۰۰۵۹	۰.۵۹
۱۹	۳۸۵۶۴.۹	۳۸۰۹۸.۰۷	۱۰۰۰	۳۸۷۹۱.۴۶	۱۸۷.۶۲	۰.۰۰۶۸۳۶	۳۸۷۵۸.۵۴	۹۶.۵۶	۰.۰۰۰۰۷۹	۰.۰۰۰۰۸۴	۰.۹۴
۲۰	۳۷۵۱۶.۵۳	۳۷۱۲۸.۷۵	۱۰۰۰	۳۷۷۴۳.۵۵	۱۳۹.۱۵	۰.۰۰۰۰۳۲	۳۷۶۸۳.۸۵	۸۴.۱۳	۰.۰۰۰۰۹۷	۰.۰۰۱۵۸	۰.۶۵
۲۱	۳۸۸۰۳.۹۶	۳۸۷۱۷.۱۶	۱۰۰۰	۳۸۹۸۵.۳۸	۱۰۹.۳۲	۰.۰۰۰۰۲۸۱	۳۸۹۰۶.۹	۷۰.۱۴	۰.۰۰۰۱۶۹	۰.۰۰۲۰۱	۰.۵۶
۲۲	۳۸۹۳۲.۰۸	۳۸۹۳۲.۰۸	۷۰۴.۰۵	۳۹۰۶۸.۸۳	۱۳۲.۱۶	۰.۰۰۰۰۸۵	۳۹۰۵۰.۴۶	۷۹.۳۱	۰.۰۰۰۰۶۹	۰.۰۰۰۰۴۷	۰.۶۶
۲۳	۳۹۷۱۵.۰۷	۳۹۵۹۰.۳۲	۱۰۰۰	۳۹۸۶۸.۳۲	۱۴۳.۳۲	۰.۰۰۰۰۸۶	۳۹۸۷۰	۸۱.۶۲	۰.۰۰۰۰۸۴	-۰.۰۰۰۰۰۴	۰.۷۵
۲۴	۳۸۹۴۴.۲۳	۳۸۸۴۵.۹۳	۱۰۰۰	۴۰۰۸۹.۶۸	۱۶۰.۴۷	۰.۰۰۰۰۶۲	۴۰۰۸۲.۰۱	۹۸.۹۷	۰.۰۰۰۰۵۱۲	۰.۰۰۰۰۱۹	۰.۶۲



شکل (۸): مقایسه نتایج به‌دست آمده از DCGA, DGAPSO و GAMS برای مسائل نمونه کوچک در MDCLRSPDSL

جدول (۶): نتایج و زمان محاسباتی حل دقیق و الگوریتم‌های پیشنهادی بر روی مسائل نمونه متوسط در MDCLRSPDSL

شماره مثال	نتایج CPLEX		نتایج DCGA		نتایج DGAPSO			GAP <sub>S</sub>	GAP <sub>T</sub>
	حد بالا	حد پایین	زمان	میانگین	زمان	میانگین	RPD		
۱	۳۷۸۸۱.۱۸	۲۷۴۹۸.۵۹	۳۶۰۰	۳۰۸۳۱.۶۵	۳۱۹.۹۵	۳۰۷۸۰.۱۲	۱۹۳.۴۷	۰.۰۰۱۶۷	۰.۶۵
۲	۳۷۴۷۷.۳۳	۲۸۵۳۶.۸۸	۳۶۰۰	۳۲۸۷۵.۹۲	۳۱۴.۴۴	۳۲۸۴۳.۶۹	۲۱۳.۱۴	۰.۰۰۰۹۸	۰.۴۷
۳	-	۱۸۶۰۱.۱۴	۳۶۰۰	۲۳۷۹۲.۶۴	۳۲۱.۶۵	۲۳۷۲۹.۴۴	۲۰۶.۳۰	۰.۰۰۲۶۳	۰.۵۶
۴	-	۲۴۴۶۹.۱۲	۳۶۰۰	۳۰۵۸۱.۹۵	۳۷۵.۹۷	۳۰۵۳۱.۶۶	۲۱۸.۴۰	۰.۰۰۱۶۴	۰.۷۲
۵	-	۲۹۶۴۵.۸	۳۶۰۰	۲۹۹۴۱.۷۳	۵۶۳.۱۴	۳۷۱۳۰.۶۷	۲۴۸.۹۵	-۰.۱۹۳۶۱	۱.۲۶
۶	-	۳۳۸۹۴.۹	۳۶۰۰	۴۶۲۰۵.۱۴	۶۲۶.۱۶	۴۶۱۹۹.۶۳	۲۷۰.۹۶	۰.۰۰۰۳۳	۱.۳۱
۷	-	۲۹۲۴۱.۳	۶۰۰۰	۳۴۹۲۱.۸۷	۵۲۷.۵۷	۳۴۸۷۷.۸۴	۲۶۷.۷۸	۰.۰۰۰۵۸	۰.۹۷
۸	-	۳۳۶۲۲.۲	۶۰۰۰	۳۸۰۷۸.۸۸	۵۱۸.۹۵	۳۸۰۳۲.۰۶	۳۱۰.۰۳	۰.۰۰۱۳۲	۰.۶۷
۹	-	۳۱۹۲۳.۸	۶۰۰۰	۳۳۴۷۴.۷۲	۶۸۰.۳۶	۳۳۴۱۵.۲۴	۳۸۰.۷۶	۰.۰۰۱۷۸	۰.۷۸
۱۰	-	۳۱۳۸۲.۹	۶۰۰۰	۳۴۴۹۲.۰۴	۸۱۴.۱۸	۳۴۴۷۴.۴۱	۳۱۳.۰۶	۰.۰۰۰۵۱	۱.۶۰
۱۱	-	۳۲۱۶۱.۶	۶۰۰۰	۳۶۹۴۲.۰۴	۵۷۸.۱۵	۳۶۹۴۳.۱۳	۳۶۰.۵۳	-۰.۰۰۰۰۲	۰.۶۰
۱۲	-	۳۳۵۲۹.۱	۶۰۰۰	۴۰۵۰۲.۴۱	۶۰۱.۶۷	۴۰۴۸۹.۸	۳۹۹.۰۶	۰.۰۰۰۳۱	۰.۵۱

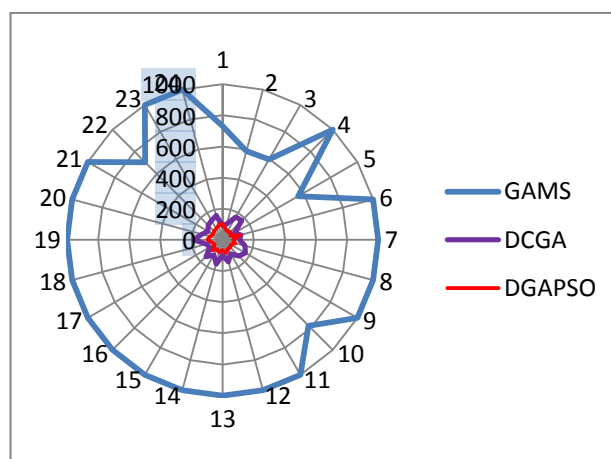
۲۳، کمی بهتر از DCGA عمل می‌کند و از لحاظ زمان اجرا بسیار بهتر از DCGA است. شاخص‌های  $GAP_S$  و  $GAP_T$  نیز بیانگر این واقعیت‌ها هستند. شکل (۸) مقایسه نتایج به‌دست آمده برای مقدار تابع هدف مسأله از روش حل دقیق و الگوریتم‌های پیشنهادی در مسائل نمونه کوچک را نشان می‌دهد.

همچنین شکل (۹) نمایش‌دهنده زمان محاسباتی برای حل دقیق و روش فراابتکاری در مسائل نمونه با ابعاد کوچک می‌باشد. در این شکل میانگین زمان حل برای مسائل مختلف به‌دست آمده با هم مقایسه می‌شوند. این شکل مؤثر بودن الگوریتم‌های پیشنهادی از لحاظ زمان محاسباتی را به وضوح نشان می‌دهد.

در مسائل با ابعاد متوسط حل‌کننده سیپلکس به جز در مسائل نمونه شماره ۱ و ۲، در مدت زمان ۳۶۰۰ ثانیه قادر به ارائه یک جواب‌شدنی هم نمی‌باشد (جدول (۶)). در مسائل شماره ۷ تا ۱۰ فرصت محاسباتی بیشتری (۶۰۰۰ ثانیه) به حل‌کننده سیپلکس داده شده است اما با این وجود سیپلکس قادر به ارائه حد بالا نیست، همین موضوع نشان می‌دهد که با افزایش ابعاد مسأله روش‌های فراابتکاری پیشنهادی کارایی بیشتری از خود به نمایش می‌گذارند. برای حل دقیق زمان لازم برای دستیابی به جواب‌ها در مسائل نمونه، با افزایش تعداد مشتریان و تعداد تقاضاهای برش افزایش می‌یابد. با افزایش ابعاد مسأله کیفیت جواب‌ها کاهش یافته و زمان حل مسأله افزایش می‌یابد و این بدان معناست که عملکرد حل‌کننده سیپلکس در نرم‌افزار گمز در مسائل نمونه MDCLRSPDSL با افزایش ابعاد مسأله، جالب توجه نیست.

عملکرد الگوریتم‌های پیشنهادی با افزایش ابعاد مسأله کاملاً قابل دفاع است، زیرا در مسائل با ابعاد متوسط در زمان نسبتاً کم جواب‌های شدنی تولید می‌کند در حالی سیپلکس در زمان‌های به

نتایج جدول (۵) نشان می‌دهد که حل‌کننده سیپلکس در تعدادی از مسائل نمونه کوچک مدت زمان حداکثر ۱۰۰۰ ثانیه‌ای که برای اجرای نرم‌افزار در نظر گرفته شده، به جواب دقیق می‌رسد و در سایر نمونه‌ها جوابی نزدیک به بهینه را ارائه می‌دهد. همین موضوع نشان می‌دهد که پیچیدگی محاسباتی مسأله پیشنهادی بالا است به‌طوری‌که حل‌کننده سیپلکس برای تعداد زیادی از مسائل کوچک طراحی شده قادر نیست در مدت زمان کوتاهی به جواب بهینه دست یابد. عملکرد الگوریتم‌های پیشنهادی در مسائل کوچک خوب است به‌طوری‌که در نمونه مسائلی که سیپلکس به جواب بهینه دست می‌یابد الگوریتم‌های پیشنهادی نیز در زمان به مراتب کمتری به جواب بهینه می‌رسند.



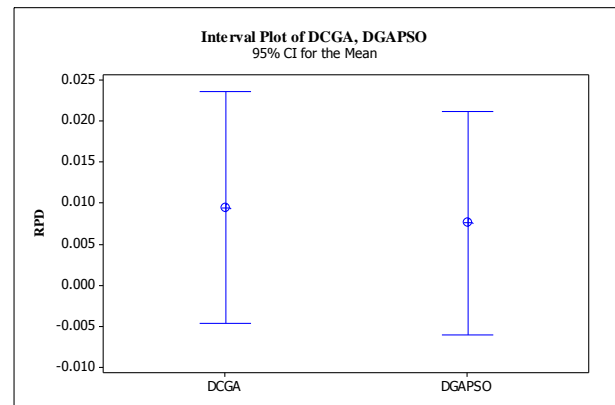
شکل (۹): زمان محاسباتی الگوریتم‌های پیشنهادی و حل‌کننده CPLEX برای حل مسائل نمونه کوچک در MDCLRSPDSL

با نگاهی به جواب‌های به‌دست آمده می‌توان فهمید که DGAPSO از لحاظ مقدار تابع هدف به جز در مسائل نمونه ۱۴ و

## مراجع

- [1] Webb, M.H.J. (1968). Cost functions in the location of depots for multiple-delivery journeys. *OR*, 311-320.
- [2] Belenguer, J.M., Benavent, E., Prins, C., Prodhon, C., Wolfler Calvo, R. (2011). A branch-and-cut method for the capacitated location-routing problem. *Computers & Operations Research*, 38(6): 931-941.
- [3] Melechovský, J., Prins, C., Calvo, R.W. (2005). A metaheuristic to solve a location-routing problem with non-linear costs. *Journal of Heuristics*, 11(5-6): 375-391.
- [4] Nagy, G., Salhi, S. (2007). Location-routing: Issues, models and methods. *European Journal of Operational Research*, 177(2): 649-672.
- [۵] ستاک، مصطفی، توکل‌نیا، امین (۱۳۹۲). مسیریابی وسایل نقلیه، انتشارات نشر بازرگانی، تهران، ویرایش اول.
- [6] Minis, I., Tatarakis, A. (2011). Stochastic single vehicle routing problem with delivery and pick up and a predefined customer sequence. *European Journal of Operational Research*, 213(1): 37-51.
- [7] Hoff, A., Gribkovskaia, I., Laporte, G., Løkketangen, A. (2009). Lasso solution strategies for the vehicle routing problem with pickups and deliveries. *European Journal of Operational Research*, 192(3): 755-766.
- [8] Karaoglan, I., Altıparmak, F., Kara, I., Dengiz, B. (2012). The location-routing problem with simultaneous pickup and delivery: Formulations and a heuristic approach. *Omega*, 40(4): 465-477.
- [9] Subramanian, A., Drummond, L.M.D. A., Bentes, C., Ochi, L.S., Farias, R. (2010). A parallel heuristic for the vehicle routing problem with simultaneous pickup and delivery. *Computers & Operations Research*, 37(11): 1899-1911.
- [10] Karaoglan, I., Altıparmak, F., Kara, I., Dengiz, B. (2011). A branch and cut algorithm for the location-routing problem with simultaneous pickup and delivery. *European Journal of Operational Research*, 211(2): 318-332.
- [11] Karaoglan, I., Altıparmak, F. (2010, July). A hybrid genetic algorithm for the location-routing problem with simultaneous pickup and delivery. In *Computers and Industrial Engineering (CIE), 2010 40th International Conference on* (pp. 1-6). IEEE.
- [12] Berbeglia, G., Cordeau, J.F., Gribkovskaia, I., Laporte, G. (2007). Static pickup and delivery problems: a classification scheme and survey. *Top*, 15(1): 1-31.
- [13] Dror, M., Trudeau, P. (1989). Savings by split delivery routing. *Transportation Science*, 141-145.
- [14] Ho, S.C., Haugland, D. (2004). A tabu search heuristic for the vehicle routing problem with time windows and split deliveries. *Computers & Operations Research*, 31(12): 1947-1964.
- [15] Tavakkoli-Moghaddam, R., Safaei, N., Kah, M. M. O., Rabbani, M. (2007). A new capacitated vehicle routing problem with split service for minimizing fleet cost by simulated annealing. *Journal of the Franklin Institute*, 344(5): 406-425.
- [16] Mitra, S. (2007). A parallel clustering technique for the vehicle routing problem with split deliveries and pickups. *Journal of the Operational Research Society*, 59(11): 1532-1546.
- [17] Şahin, M., Çavuşlar, G., Öncan, T., Şahin, G., Tüzün Aksu, D. (2012). An efficient heuristic for the Multi-vehicle One-to-one Pickup and Delivery Problem with Split Loads. *Transportation Research Part C: Emerging Technologies*
- [18] Chen, S., Golden, B., Wasil, E. (2007). The split delivery vehicle routing problem: Applications, algorithms, test

مراتب بیشتر از زمان الگوریتم‌های پیشنهادی قادر به ارائه جواب‌شدنی هم نیست. در مسائل نمونه متوسط نیز عملکرد خوب الگوریتم *DGAPSO* نسبت به الگوریتم *DCGA* ادامه دارد. در ادامه تحلیل نتایج با بررسی مقادیر *RPD* الگوریتم‌ها در جداول ارائه شده و شکل (۱۰) می‌توان فهمید که الگوریتم *DGAPSO* ثبات بیشتری در ارائه جواب‌ها داشته و دارای پراکندگی کمتری است.



شکل (۱۰): نمودار مقایسه *RPD* برای الگوریتم‌های پیشنهادی

## ۵- نتیجه‌گیری

در این مقاله از ویژگی برش در تقاضا در کنار شرایط دیگری مانند مکانیابی-مسیریابی و برداشت و تحویل همزمان استفاده شده و مسأله‌ای جدید مطرح شد که از جمله کاربردهای این مسأله، استفاده از آن در صنعت نوشیدنی‌ها و صنعت خوارو بار است. این مسأله جدید، مسأله مکان‌یابی مسیریابی چنددپویی ظرفیت‌دار با برداشت و تحویل همزمان و برش یافته نامگذاری شد. در ادامه یک فرمول‌بندی برنامه‌ریزی عدد صحیح مختلط متناظر با این مسأله ارائه شد. سپس دو روش فراابتکاری ترکیبی *DCGA* و *DGAPSO* مبتنی بر روش‌های ژنتیک گسسته و ژنتیک پیوسته و بهینه‌سازی تجمعی ذرات، ارائه شد. در نهایت برای ارزیابی مسأله مطرح شده، تعدادی مسأله نمونه در ابعاد کوچک و متوسط با استفاده از نرم افزار سیپلکس و الگوریتم‌های پیشنهادی حل شد که نتایج حاکی از موثر بودن الگوریتم *DGAPSO* از لحاظ زمان محاسباتی و کیفیت جواب است. از نظر نویسندگان پژوهش حاضر از چند جهت قابل بهبود و گسترش است. تقویت مدل ریاضی با افزودن نامساوی‌های معتبر، در نظر گرفتن محدودیت‌های پنجره زمانی، در نظر گرفتن شرایط عدم قطعیت برای برخی از پارامترهای مسأله و گسترش روش‌های حل بهتر برای رقابت با الگوریتم‌های ارائه شده در این مقاله از جهت کیفیت جواب و زمان اجرا به عنوان برخی جهت‌های تحقیقاتی پیشنهاد می‌شود.

- problems, and computational results. *Networks*, 49(4): 18-329.
- [19] Archetti, C., Speranza, M.G., Savelsbergh, M.W. (2008). An optimization-based heuristic for the split delivery vehicle routing problem. *Transportation Science*, 42(1): 22-31.
- [20] Jin, M., Liu, K., & Eksioglu, B. (2008). A column generation approach for the split delivery vehicle routing problem. *Operations Research Letters*, 36(2), 265-270.
- [21] Sierksma, G., Tijssen, G.A. (1998). Routing helicopters for crew exchanges on off-shore locations. *Annals of Operations Research*, 76: 261-286.
- [22] Archetti, C., Speranza, M.G. (2012). Vehicle routing problems with split deliveries. *International transactions in operational research*, 19(1-2): 3-22.
- [23] Min, H., Jayaraman, V., Srivastava, R. (1998). Combined location-routing problems: A synthesis and future research directions. *European Journal of Operational Research*, 108(1), 1-15.
- [24] Goldberg, D.E. (1989). *Genetic algorithms in search, optimization, and machine learning* (Vol. 412). Reading Menlo Park: Addison-wesley.
- [25] Eberhart, R.C., Kennedy, J. (1995). A new optimizer using particle swarm theory. In *Proceedings of the sixth international symposium on micro machine and human science*, Vol. 1: 39-43.







## Multi depots Capacitated Location-Routing Problem with Simultaneous Pickup and Delivery and Split Loads: Formulation and heuristic methods

M. Setak\*, V. Azizi, H. Karimi

Department of Industrial Engineering, Collage of Engineering, K.N.T. University of Technology, Tehran, Iran

### ARTICLE INFO

#### Article history:

Received 19 May 2014

Accepted 15 February 2015

#### Keywords:

Location-Routing  
Simultaneous Pickup and  
Delivery  
Split  
Mixed Integer Programming  
Genetic Algorithm

### ABSTRACT

Problems in real world include various constraints. So the reliable models should be designed that satisfy the reasonable number of these constraints. These models should be applicable.

In this paper, regarding the mentioned circumstances, we design a model for multi depots capacitated location-routing problem with simultaneous pickup and delivery and split loads. The paper emphasis is on the split property that happens when demand of a customer exceeds vehicle capacity. We propose a mixed integer-programming model. Also we develop two metaheuristics for solving large scale instance problems. The first approach is based on discrete genetic algorithm and continuous genetic algorithm and the second one is based on discrete genetic algorithm and particle swarm optimization. Results of solving instance problems by CPLEX solver and proposed DCGA and DGAPSO algorithms show that both algorithms are effective and DGAPSO outperforms DCGA in solution quality and computation times.

\* Corresponding author. Mostafa Setak

Tel.: +98 21 84063373; E-mail addresses: [setak@kntu.ac.ir](mailto:setak@kntu.ac.ir)