

۴-۳ الگوریتم‌های پیشنهادی

• الگوریتم‌های فراابتکاری

انگیزه اصلی استفاده از الگوریتم‌های فراابتکاری به جای روش‌های دیگر به دلیل توانایی آن‌ها در ارائه راه حل‌های تقریباً بهینه در زمان محاسباتی معقول است. فراابتکاری‌ها همچنین عملکرد و انعطاف‌پذیری بالایی برای حل مسائل متفاوت بهینه‌سازی دارند. این الگوریتم‌ها به دو دسته عمده مبتنی بر یک جواب و مبتنی بر جمعیت تقسیم می‌شوند. الگوریتم‌های مبتنی بر یک جواب در حین فرایند جستجو یک جواب را تغییر می‌دهند، در حالی که در الگوریتم‌های مبتنی بر جمعیت در حین جستجو، یک جمعیت از جواب‌ها در نظر گرفته می‌شوند. الگوریتم‌های فراابتکاری منفرد ممکن است کمبودهایی مانند همگرایی زود هنگام داشته باشند. برای غلبه بر این نقص، محققان سعی در توسعه الگوریتم‌های فراابتکاری ترکیبی مثل (آکپونار و آکپینار^۲، ۲۰۲۱) کردند، زیرا الگوریتم‌های ترکیبی عموماً یک هم‌افزایی مثبت را نشان می‌دهند.

• GA-VNS

GAها از تکامل بیولوژیکی الهام گرفته شده‌اند و بر اساس ایده‌ی بقای بهترین‌ها طراحی می‌شوند. قدرت GAها در توانایی آنها برای جلوگیری از جستجوی تصادفی با بهره‌برداری هوشمندانه از اطلاعات تاریخی برای هدایت فرآیند جستجو به سمت مناطق با عملکرد بهتر در فضای راه حل نهفته است. GAها در نهایت به سمتی همگرا می‌شود که بهترین سازگاری را با محیط دارد و در نتیجه راه حل بهینه یا رضایت بخش را به دست می‌آورند. الگوریتم فراابتکاری ترکیبی از یک طرف به الگوریتم‌های مبتنی بر جمعیت تعلق دارد و از طرف دیگر به VNS که الگوریتم مبتنی بر یک جواب است تعلق دارد. روش ما با مجموعه‌ای از راه حل‌ها به عنوان GAها کار می‌کند. با این حال راه‌حل‌های اولیه با استفاده از یک الگوریتم جستجوی

² Akpunar and Akpinar

مضاعف تولید می‌شوند که قادر به ارائه مسیرهای امکان‌پذیر بین هر دو نقطه است. جزئیات این الگوریتم در ادامه داده شده است. هنگامی که جمعیتی از راه‌حل‌ها (مسیرهای امکان‌پذیر) با موفقیت ایجاد شد، ما یک عملیات بهبود را روی هر فرد در اولین جمعیت انجام می‌دهیم. یعنی یک VNS را روی جمعیت اول اعمال می‌کنیم. ما با بهبود راه‌حل‌های اولیه به افزایش کیفیت الگوریتم کمک می‌کنیم تا راه حل بهینه را پیدا کند.

• ALNS-VNS

جستجوی همسایگی بزرگ (LNS) توسط (شاو، ۱۹۹۸) برای یک مسئله مسیریابی وسیله نقلیه ظرفیت‌دار معرفی شد. LNS با یک راه حل اولیه شروع می‌شود و مقدار هدف را به تدریج با اعمال یک تخریب و یک عملگر تعمیر در هر تکرار بهبود می‌بخشد. روپکه و پیسینگر (۲۰۰۶) الگوریتم ALNS را برای بهبود LNS پیشنهاد کردند. ALNS شامل چندین عملگر تخریب و تعمیر است که همگی ابتکاری برای دستیابی به هدف صرفه جویی در زمان هستند. در هر تکرار، یک جفت عملگر به‌طور تصادفی انتخاب می‌شوند تا حرکتی انجام دهند. این روش تطبیقی است زیرا اغلب جفت‌ها ممکن است در طول جستجو تغییر کنند. VNS یک الگوریتم فراابتکاری است که معمولاً برای حل مسائل بهینه‌سازی ترکیبی و کلی استفاده می‌شود. VNS اولین بار توسط (ام‌لادنووچ و هانسن، ۱۹۹۷) به عنوان روشی برای تغییر سیستماتیک همسایگی بر اساس تکرار جستجوهای همسایگی پیشنهاد شد. VNS یک الگوریتم ساده است و مزیت آن تعداد کم پارامترهای آن است که باعث افزایش سرعت در حل مسائل می‌شود. تا جایی که ما می‌دانیم، ایده ترکیب ALNS و VNS برای اولین بار توسط (علینقیان و شکوهی، ۲۰۱۸) مطرح شد.

۴-۴ شبه کد الگوریتم‌های پیشنهادی

در این تحقیق از ۲ الگوریتم فراابتکاری ترکیبی جستجوی همسایگی متغیر و ژنتیک (VNS-GA) و الگوریتم ترکیبی جستجوی همسایگی متغیر و جستجوی همسایگی انطباقی (ALNS-VNS) استفاده شده است. در ادامه شبه کد الگوریتم ترکیبی ALNS-VNS به کار رفته جهت حل مسئله پیشنهادی در شکل (۳-۴) ارائه شده است.

در الگوریتم ALNS همسایگی‌ها توسط روش‌های ابتکاری ساده و سریع جستجو می‌شوند به طوری که در هر تکرار آن، یک رویکرد برای تخریب جواب جاری و یک رویکرد برای اصلاح آن با توجه به مکانیزم انتخاب چرخ رولت، انتخاب و اجرا می‌شود. این روش‌های ابتکاری مخرب و سازنده به ترتیب برای ایجاد پراکندگی و همگرایی جستجو به کار برده می‌شوند. نحوه‌ی انتخاب رویکردهای تخریب و تعمیر وابسته به عملکردشان در گذشته‌ی فرآیند جستجو الگوریتم می‌باشد.

در این الگوریتم، جواب‌های بدست آمده از عملگرهای تخریب و تعمیر در الگوریتم ALNS، توسط روش VNS و رویکرد همسایگی بهبود پیدا می‌کنند. این الگوریتم از سه بخش تولیدجمعیت اولیه، تولید فرزندان از عملگر تخریب و تعمیر ALNS، و بهبود جواب‌های بدست آمده در مرحله قبل توسط روش VNS تشکیل شده است.

همچنین در شکل (۴-۴) شبه کد الگوریتم ژنتیک و VNS ارائه شده است. در الگوریتم ترکیبی ژنتیک و VNS فرزندان به دست آمده از عملگرهای انتخاب، ترکیب و جهش در الگوریتم ژنتیک نیز توسط روش VNS و رویکرد همسایگی آن بهبود پیدا می‌کنند. این الگوریتم از سه بخش تولید جمعیت اولیه، تولید فرزندان از عملگر انتخاب، ترکیب و جهش، و بهبود فرزندان به دست آمده از عملگرهای الگوریتم ژنتیک توسط رویکرد همسایگی روش VNS تشکیل شده است.

```

1. Set algorithm's parameters ( $It\_Max$ ,  $Nmove$ ,  $Kmax$ ,  $Phi$ ,  $T$ ,  $\Omega^+$ ,  $\Omega^-$ )
2. Generate Initial Solution
3. Evaluate  $P$  (Calculate objective function)
4.  $t = 0$  (number of iterations)
5. While ( $t \leq It\_Max$ ):
6.    $k=1$ 
7.   While ( $k \leq Nmove$ ):
8.      $P' \leftarrow P$ 
9.     Select the destroy method by roulette wheel
10.    Perform the selected destroy method on  $P'$ 
11.    Select the repair method by roulette wheel
12.    Perform the selected repair method on  $P'$ 
13.    Check feasibility of  $P'$  solution and repairing it (if necessary)
        calculate objective function  $F(P')$ 
14.    If  $F(P') \leq F(P)$ 
15.       $P \leftarrow P'$ 
16.      If  $F(P') \leq F(P_{Best})$ 
17.         $P_{Best} \leftarrow P'$ 
18.    Else
19.       $Deltaf = F(P') - F(P)$ 
20.      If  $rand < \exp(-\frac{Deltaf}{T})$ 
21.         $P \leftarrow P'$ 
22.    Update weights of selected destroy  $\Omega^-$  and repair  $\Omega^+$  operators
23.    Update Temperature  $T$ 
24.    If  $Rem(t, It\_Vns) == 0$ 
25.       $j=1$ 
26.      While ( $j \leq MAxit\_VNS$ ):
27.         $P' = Shaking(P)$ 
28.         $P'^* = local\ search(P')$ 
29.        If  $F(P'^*) \leq F(P)$ 
30.           $P \leftarrow P'^*$ 
31.         $j=j+1$ 
32.      Else
33.         $j=j+1$ 
34.      If  $Rem(t, Phi) == 0$ 
35.        Reset weight of operators
36.         $t=t+1$ 
37. Report  $P_{Best}$ 

```

شکل ۴-۳ شبه کد الگوریتم ترکیبی ALNS-VNS

یکی از نقاط ضعف الگوریتم ژنتیک، قرار گرفتن در جواب‌های محلی می‌باشد. جهت افزایش کارایی این الگوریتم از رویکردهای مبتنی بر جستجوی محلی استفاده شده است. یکی از بهترین الگوریتم‌ها در این

خصوص VNS می باشد. لذا در این تحقیق از قابلیت جستجوی محلی الگوریتم VNS برای افزایش کارایی الگوریتم ژنتیک استفاده شده است. رویکرد پیشنهادی بر اساس یک فرآیند ترکیبی انجام شده بین الگوریتم ژنتیک (GA) که متعلق به فراابتکاری مبتنی بر جمعیت است و یک جستجوی همسایگی متغیر (VNS) است که به صورت همزمان قصد دارند جواب بهینه مسئله را پیدا کنند. در شکل (۴-۴) شبه کد این الگوریتم را نشان می دهد.

```

1) Set algorithm's parameters (Max it, Te, Kmax)
2) For each  $i = 1..M$  ( $M$  is the size of the population)
    a) Generate  $P[i]$  randomly
    b) Evaluate  $P[i]$  (Calculate objective function)
3)  $t = 0$  (number of iterations)
4) sort solution based on Best objective
5) repeat
    a) select parent
    b) crossover
    c) mutation
    d) check feasibility of  $P_{new}[i]$  solution and do possible repairing
    e) evaluate objective function
6) set  $i=1$  for each offspring until  $I_{max}$ 
    a) apply local search
        I.  $j=1$ 
        II. While ( $j \leq MAXit\_VNS$ ):
        III.  $P_{new}'[i] = Shaking(P_{new}[i])$ 
        IV.  $P_{new}'^* = local\ search(P_{new}'[i])$ 
        V. If  $F(P_{new}'^*) \leq F(P_{new}[i])$ 
        VI.  $P_{new}[i] \leftarrow P_{new}'^*$ 
        VII.  $j=j+1$ 
        VIII. Else
        IX.  $j=j+1$ 
    b) If  $F(P_{new}[i]) \leq F(P[i])$ 
    c)  $P[i] \leftarrow P_{new}[i]$ 
    d) If  $F(P[i]) \leq F(P_{Best})$ 
    e)  $P_{Best} \leftarrow P[i]$ 
7) add new offspring population ( $P_{new}$ ) to parent population ( $P$ )
8) sort solution based on best objective
9) elitism selection technique and select parent population for next generation
10)  $t=t+1$ 

```

شکل ۴-۴ شبه کد الگوریتم ترکیبی ژنتیک - VNS

۴-۵ نحوه نمایش جواب‌ها

در این بخش نحوه‌ی نمایش جواب‌ها مورد بررسی قرار می‌گیرد. در مدل مفروض نحوه‌ی نمایش جواب مسئله از دو قسمت تشکیل شده است. بخش اول تورهای مربوط به پهبادهای را نشان می‌دهد و بخش دوم باتری انتخاب شده برای هر پهباد را نشان می‌دهد. در بخش اول نحوه‌ی نمایش جواب از یک بردار تشکیل شده است. در این مسئله به دنبال یافتن بهترین مسیرها برای هر یک از پهبادهای هستیم که نقطه‌های (نقاط آسیب‌دیده و ایستگاه‌های شارژ) تخصیص داده شده به هر پهباد و توالی بازدید آن‌ها را نشان می‌دهد. نحوه نمایش جواب یک بردار است. اعداد صفر، پایان هر تور پهباد و شروع تور پهباد دیگر را نشان می‌دهد. در هر بردار نحوه نمایش جواب به تعداد پهبادهای منهای یک (تعداد پهباد - ۱) صفر وجود دارد و تور پهبادهای با اعداد صفر جداسازی می‌شوند. عدد منفی یک (۱-) در هر خانه نشان از استفاده‌ی مجدد پهباد در تور مورد نظر دارد و قرارگیری شماره‌ی نقطه ایستگاه‌های شارژ در هر خانه نشان از مراجعه به ایستگاه شارژ دارد. بردار جواب دارای مشخصه‌های زیر است:

✓ شماره نقاط آسیب‌دیده در بردار جواب یک بار تکرار می‌شوند

✓ تعداد اعداد صفر برابر تعداد پهبادهای منهای یک است

✓ عدد ۱- (استفاده مجدد در دیو) و شماره ایستگاه‌های شارژ می‌توانند چند بار تکرار شوند (استفاده

مجدد در ایستگاه شارژ) که این امر به جواب مسئله بستگی دارد.

✓ در محاسبات هر تور، پهباد از دیو به نقطه اول و بعد از نقطه آخر به دیو باز می‌گردد.

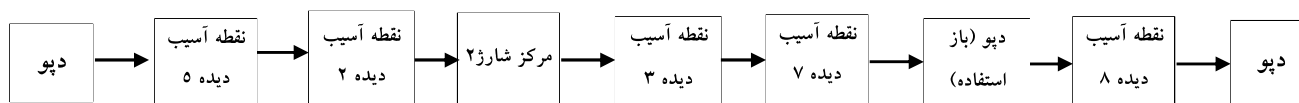
بطور مثال برنامه حرکتی یک مسئله با ۴ پهباد، ۱۰ نقطه آسیب‌دیده و ۲ ایستگاه شارژ و یک بار استفاده

مجدد در دیو ارائه شده است:

پهپاد ۴		پهپاد ۳	پهپاد ۲	پهپاد ۱
۹	۰	۶ -۱ ۴ ۱ ۱۰	۰ ۰	۸ -۱ ۷ ۳ ۱۲ ۲ ۵

شکل ۴-۵ بردار نمایش جواب

در بالا ماتریس بخش اول نشان داده شده است. حال برای روشن تر شدن این نوع کدگذاری مثالی شرح داده شده است. بردار جواب تور پهپاد شماره یک (قسمت آبی رنگ) به صورت ذیل است:



طبق ماتریس نمایش جواب، پهپاد یک از دپو خارج می شود و نقاط ۵ و ۲ را بازدید می کند سپس به مرکز شارژ ۲ می رود و باتری پهپاد شارژ می شود. سپس به نقاط ۳ و ۷ می رود و در انتها به دپو رفته، پس از شارژ باتری و بارگیری مجدد کالا، نقطه ی ۸ را بازدید می کند و سپس به دپو باز می گردد. مطابق شکل ۴، پهپاد دوم انتخاب نشده است و بعد از عدد صفر عدد غیر صفری نیامده است که این نشان از عدم انتخاب پهپاد ۲ دارد. پهپاد ۳ نیز مانند پهپاد ۱ نقاط آسیب دیده ی ۱، ۱۰ و ۴ را بازدید کرده سپس به دپو رفته، باتری خود را شارژ می کند، بارگیری مجدد انجام می دهد و نقطه ۶ را بازدید کرده و با بازگشت به دپو تور خود را پایان می دهد. پهپاد ۴ نیز از دپو خارج شده نقطه ۹ را بازدید می کند و به دپو باز می گردد.

بخش دوم شامل یک بردار است که به تعداد پهپادها دارای خانه است. این بردار نوع باتری انتخاب شده برای هر پهپاد را نشان می دهد.

	پهپادها			
باتری انتخاب شده	۲	۰	۱	۲

شکل ۴-۶ نمایش جواب باتری انتخاب شده برای ۴ پهپاد

در شکل ۴-۶ باتری انتخاب شده برای پهباد ها نمایش داده شده است. همان‌طور که مشخص است پهباد ۲ انتخاب نشده است. برای پهباد ۱ باتری نوع ۲، پهباد ۳ باتری نوع ۱ و پهباد ۴ باتری نوع ۲ انتخاب شده است.

۴-۶ نحوه تولید جمعیت اولیه

جهت تولید جمعیت اولیه از رویکرد تصادفی استفاده شده است. جهت تولید جمعیت اولیه از رویکرد زیر استفاده شده است.

۱- یک توالی تصادفی از نقاط و پهبادها ایجاد می‌کنیم.

۲- یک نقطه از نقاط تخصیص داده نشده انتخاب می‌شود.

۳- براساس توالی ایجاد شده برای پهبادها، یک پهباد از پهبادهای انتخاب نشده تعیین می‌شود و به مرحله

۴ می‌رویم در صورت نبود وجود پهباد در مجموعه انتخاب نشده به مرحله ۸ می‌رویم.

۴- اگر نقطه دیگری قبلاً به پهباد انتخاب شده تخصیص داده شده است به مرحله ۵ در غیر این صورت به مرحله ۷ می‌رویم.

۵- براساس محدودیت ظرفیت باتری اگر امکان تخصیص نقطه به پهباد وجود دارد، نقطه تخصیص داده می‌شود و به مرحله ۲ می‌رویم در غیر این صورت به مرحله ۶ می‌رویم.

۶- در صورتی که قبل از بازدید نقطه مربوطه، امکان رفتن به ایستگاه شارژ جهت شارژ باتری و سپس بازدید نقطه مربوطه وجود دارد، نقطه تخصیص داده می شود و به مرحله ۲ می رویم در غیر این صورت به مرحله ۳ می رویم.

۷- یک نوع باتری برای پهباد انتخاب می شود و به مرحله ۵ می رویم.

۸- جهت استفاده مجدد از پهباد (سفر جدید، بارگیری و شارژ مجدد) امکان تخصیص نقطه آسیب دیده به یک پهباد در تور جدید، مرحله ۴، ۵، ۶ و ۷ بررسی شده و در صورت امکان نقطه در تور جدید به پهباد مربوطه تخصیص می یابد.

۹- پس از تخصیص همه نقاط به پهبادها تابع هدف مسئله محاسبه می شود.

۴-۷ عملگر جهش و ترکیب

اولین عملگر الگوریتم ژنتیک، عملگر ترکیب می باشد که حاصل از ترکیب دو والد است که نتیجه ی آن ایجاد دو فرزند و جواب جدید است. در الگوریتم ژنتیک برای کروموزم مسئله از عملگر تقاطع تک نقطه ای باهوش برای ترکیب استفاده شده است:

$$\begin{aligned} ch_1 &= [Paret_1(1 : Alpha), Paret_2(1 + Alpha : N)] \\ ch_2 &= [Paret_2(1 : Alpha), Paret_1(1 + Alpha : N)] \end{aligned} \quad (4-1)$$

که $paret1$ و $paret2$ به ترتیب پدر و مادر انتخابی بوده و α یک نقطه ی برش بین ۱ تا $N-1$ می باشد و $ch1$ و $ch2$ فرزندان حاصل شده هستند. این عملیات به صورت شماتیک در شکل ۴-۷ ارائه شده است: (مقدار α برابر ۳ است)

$Paret_{11}$						$Paret_{22}$				فرزند اول							
۴	-۱	۲	۰	۵		۲	-۱	۴	→	۴	-۱	۲	۰	۵	۲	-۱	۴
$Paret_{21}$						$Paret_{12}$				فرزند دوم							
۵	۰	۳	۱	-۱		۳	۶	۱	→	۴	۰	۳	۱	-۱	۳	۶	۱

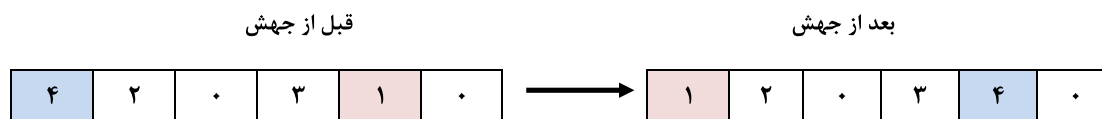
شکل ۴-۷ عملگر ترکیب

در این مسئله از روش تک نقطه‌ای باهوش استفاده شده است. در کروموزم مسئله، نقاط آسیب‌دیده تنها یک بار می‌توانند تکرار شوند و شماره استفاده مجدد مراکز شارژ و استفاده مجدد دیو (عدد ۱-) می‌توانند چند بار تکرار شوند. همچنین تعداد اعداد صفر نیز به تعداد پهباد منهای یک در کروموزم وجود دارد. در مثال بالا تعداد ۲ پهباد، ۵ نقطه آسیب‌دیده و یک مرکز شارژ مجدد (شماره ۶) وجود دارد. پس از ترکیب جواب‌ها نقاضا ۲ و ۴ در فرزند اول و نقاط ۳ و ۱ در فرزند دوم تکرار شده‌اند. در رویکرد تک نقطه باهوش پس از ترکیب جواب‌ها، در صورت تکرار نقاط آسیب‌دیده در جواب فرزندان، از بخش اول والدها نقاط استفاده نشده جا به جا می‌شود. به طور مثال در فرزند اول نقاط ۲ و ۴ تکرار شده‌اند. جهت جایگزینی آن‌ها از بخش $paret1$ استفاده می‌شود. اولین نقطه استفاده نشده در بخش $paret21$ نقطه شماره ۳ است که جایگزین شماره ۲ می‌شود، دومین نقطه استفاده نشده نقطه ۱ است که جایگزین نقطه ۴ می‌شود. در نهایت جواب‌های بدست آمده به شرح زیر است.

فرزند اول							
۴	-۱	۲	۰	۵	۳	-۱	۱
فرزند دوم							
۵	۰	۳	۱	-۱	۴	۶	۲

شکل ۴-۸ فرزندان تولید شده از ترکیب پس از اصلاح

جهش نیز عملگر دیگری است که جواب‌های ممکن دیگری را متولد می‌کند. در الگوریتم ژنتیک بعد از این که یک عضو جدید در جمعیت جدید به وجود آمد، هر ژن آن با احتمال جهش، جهش می‌یابد. برای عملگر جهش از عملگر SWAP استفاده شده است یعنی اعداد مربوط به دو ستون مربوط به دو نقطه انتخاب شده هستند و وضعیت آن‌ها عوض می‌شود. به‌طور مثال در شکل (۴-۹) نقطه شماره ۱ و ۴ جابه‌جا می‌شوند.



شکل ۴-۹ عملگر جهش

شرط توقف در الگوریتم‌های ارائه شده به دو صورت است. شرط اول توقف برابر حداکثر تعداد تکرار الگوریتم می‌باشد. شرط دوم توقف برای الگوریتم‌ها برابر است با عدم تغییر جواب بهینه بعد از تعداد مشخص تکرار الگوریتم است که این عدد در بخش تنظیم پارامتر مشخص شده است.

۴-۸ عملگرهای تخریب و تعمیر الگوریتم ALNS

الگوریتم ALNS شامل دو عملگر اصلی می‌باشد. این دو عملگر به عنوان عملگرهای تخریب و تعمیر شناخته می‌شوند. در عملگر تخریب، بخشی از نحوه نمایش جواب حذف و تخریب می‌شود و در بخش دیگر یعنی تعمیر، سعی می‌شود جواب مسئله بازسازی شود. برای عملگر تخریب از پنج رویکرد و برای عملگر تعمیر نیز از چهار رویکرد استفاده شده است.

۴-۸-۱ عملگر تخریب

مطابق مشخصات مسئله، جهت تخریب جواب ها در الگوریتم ALNS از رویکردهای ذیل استفاده می شود.

✓ رویکرد حذف یک نقطه به صورت تصادفی:

یک نقطه به صورت تصادفی انتخاب می شود و پهباد تخصیصی به آن حذف می گردد

✓ رویکرد حذف یک نقطه از یک تور با بیش ترین تقاضا:

یک تور پهباد انتخاب می شود و نقطه با بیش ترین تقاضا از آن حذف می گردد

✓ رویکرد حذف یک نقطه با بیش ترین زمان سفر:

یک تور پهباد بصورت تصادفی انتخاب می شود و نقطه با بیش ترین زمان سفر از آن حذف می گردد

✓ رویکرد حذف یک تور با بیش ترین ظرفیت بدون استفاده:

ظرفیت بدون استفاده پهبادهای محاسبه شده و پهباد با بیش ترین ظرفیت خالی حذف می گردد

✓ حذف باتری پهباد:

یک پهباد انتخاب شده و باتری انتخابی آن به صورت تصادفی حذف می گردد.

✓ حذف باتری پهباد با کم ترین ظرفیت:

میزان ظرفیت باتری پهبادها بررسی شده و باتری با کم ترین ظرفیت از یک پهباد حذف می شود.

۴-۸-۲ عملگر تعمیر

پس از تخریب جواب در بخش بالا، جهت تعمیر آن مطابق مشخصات مسئله از رویکردهای ذیل استفاده

می شود.

✓ اضافه کردن نقطه بصورت تصادفی:

نقاط آسیب‌دیده بدون پهیاد انتخابی، به صورت تصادفی به پهیادها تخصیص می‌یابند.

✓ اضافه کردن نقطه به نزدیک‌ترین نقطه ممکن در یک تور پهیاد

برای هر یک از نقاط آسیب‌دیده بدون پهیاد انتخابی، یک تور انتخاب شده و در توالی بازدید مشتریان بعد از نزدیک‌ترین نقطه در آن تور قرار می‌گیرد.

✓ اضافه کردن نقطه به یک تور پهیاد با بیش‌ترین ظرفیت باتری استفاده نشده

نقاط تقاضای بدون پهیاد انتخابی، به تور پهیادی تخصیص می‌یابد که ظرفیت باتری استفاده نشده آن‌ها بیشتر باشد.

✓ ایجاد یک تور پهیادی جدید

نقاط تقاضای بدون پهیاد انتخابی، به پهیاد انتخاب نشده تخصیص می‌یابند و یک تور پهیادی جدید ایجاد می‌کنند.

✓ ایجاد یک تور استفاده مجدد جدید در پهیاد با کم‌ترین ظرفیت باتری استفاده

نشده:

نقاط آسیب‌دیده بدون پهیاد انتخابی، به یک پهیاد انتخاب شده تخصیص می‌یابند و یک تور پهیادی استفاده مجدد ایجاد می‌کنند.

✓ تخصیص باتری به پهیاد به صورت تصادفی

یک نوع باتری به صورت تصادفی به پهیاد تخصیص می‌یابد.

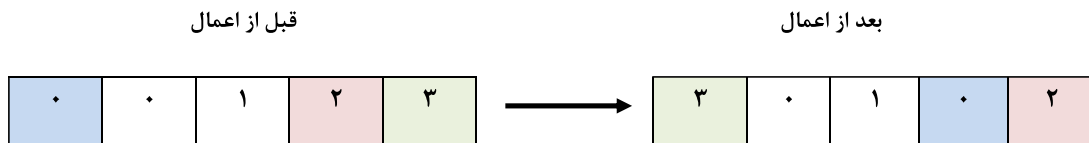
✓ تخصیص باتری با بیش‌ترین ظرفیت ممکن نسبت به نقاط قابل بازدید

برای پهیاد بدون باتری، باتری با ظرفیت مورد نیاز براساس نودهای بازدید شده انتخاب و تخصیص می‌یابد.

۹-۴ عملگرهای همسایگی روش VNS

در این تحقیق برای بخش همسایگی الگوریتم‌های ترکیبی با VNS، سه روش همسایگی در نظر گرفته شده است.

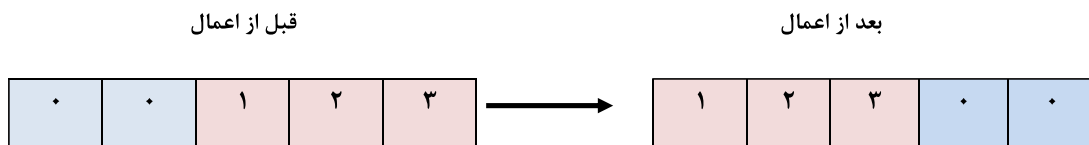
۱- روش اول همسایگی: جابجایی سه نقطه به صورت زیر :



شکل ۴-۱۰ عملگر همسایگی اول

۲- روش دوم همسایگی: انتخاب یک نقطه برش، تقسیم جواب به دو بخش و جابجایی دو بخش

مربوطه مطابق شکل زیر:



شکل ۴-۱۱ عملگر همسایگی دوم

۳- روش سوم همسایگی: این روش بروی بخش دوم کروموزم اعمال می‌شود. بدین صورت که یک

پهپاد انتخاب شده و باتری تخصیص داده شده به آن تغییر می‌کند.

۴-۱۰ مسئله عددی

در این تحقیق ۱۵ مسئله نمونه تصادفی مطابق جدول (۴-۴) تولید شده است. همچنین از تابع توزیع

یکنواخت برای تولید پارامترهای مربوط به مسئله به منظور ارزیابی عملکرد الگوریتم‌های پیشنهادی استفاده

شده است. همچنین دامنه تولید پارامترها با توجه به مقالات موجود در این زمینه برای الگوریتم‌های پیشنهادی در نظر گرفته شده به شرح جدول (۲-۴) می‌باشد که در بخش ۲-۴ معرفی شد.

جدول ۴-۴ اندازه مسائل نمونه مورد بررسی برای مقایسه الگوریتم‌ها

شماره مساله	نوع مساله	کل نقاط شبکه	نقاط آسیب-دیده	ایستگاه های شارژ	پهپاد	باتری	سطح سرعت
۱	اندازه کوچک	۶	۴	۱	۲	۲	۲
۲		۱۰	۷	۲	۳	۲	۲
۳		۱۵	۱۱	۳	۵	۳	۳
۴		۲۰	۱۶	۳	۶	۳	۳
۵		۲۵	۲۰	۴	۷	۳	۳
۶	اندازه متوسط	۳۰	۲۵	۴	۸	۴	۳
۷		۳۵	۲۹	۵	۱۰	۴	۳
۸		۴۰	۳۴	۵	۱۵	۴	۴
۹		۴۵	۳۸	۶	۱۸	۴	۴
۱۰		۵۰	۴۲	۷	۲۰	۵	۴
۱۱	اندازه بزرگ	۶۰	۵۲	۷	۲۵	۵	۴
۱۲		۷۰	۶۱	۸	۲۵	۶	۵
۱۳		۸۰	۶۹	۱۰	۳۰	۶	۵
۱۴		۹۰	۷۷	۱۲	۳۵	۷	۵
۱۵		۱۰۰	۸۷	۱۲	۴۰	۷	۵

۴-۱۱ تنظیم پارامترهای الگوریتم‌های فراابتکاری پیشنهادی

اگر پارامترهای یک الگوریتم به طور مناسب تعیین نگردد، این امر باعث کاهش کارآمدی الگوریتم خواهد شد. لذا عملکرد الگوریتم‌های فراابتکاری تا حد زیادی به تنظیم پارامترهای الگوریتم وابسته است. تکنیک‌های مختلفی مانند تکنیک طرح‌های چند عاملی طراحی آزمایش‌ها، رویه سطح پاسخ و طراحی

آزمایش‌ها به روش تاگوچی^۳ برای تنظیم پارامتر الگوریتم‌ها مورد استفاده قرار می‌گیرد. در این تحقیق، از روش تاگوچی به منظور تنظیم پارامتر الگوریتم‌های فراابتکاری پیشنهادی استفاده شده است.

در این بخش، پارامترهای به کار رفته در الگوریتم ترکیبی ALNS-VNS و الگوریتم ترکیبی ژنتیک و VNS تنظیم می‌شوند.

برای تنظیم پارامترهای الگوریتم‌های پیشنهادی از روش طراحی آزمایش‌های تاگوچی استفاده شده است. برای روش ترکیبی ALNS-VNS شش فاکتور تعداد همسایگی روش ALNS (Nmove)، حداکثر تعداد تکرار الگوریتم (Maxit)، حداکثر تعداد تکرار الگوریتم VNS (Maxit_VNS)، ضریب فی (Phi)، ضریب اتا (Eta) و معیار توقف (Stop_Criteria) در نظر گرفته شده است. همچنین معیار ضریب نرمالایز شده مقدار تابع هدف و زمان حل که عددی بین صفر و یک است به عنوان معیار پاسخ در نظر گرفته شده‌اند. همچنین برای هر فاکتور سه سطح به صورت جدول زیر در نظر گرفته شده است.

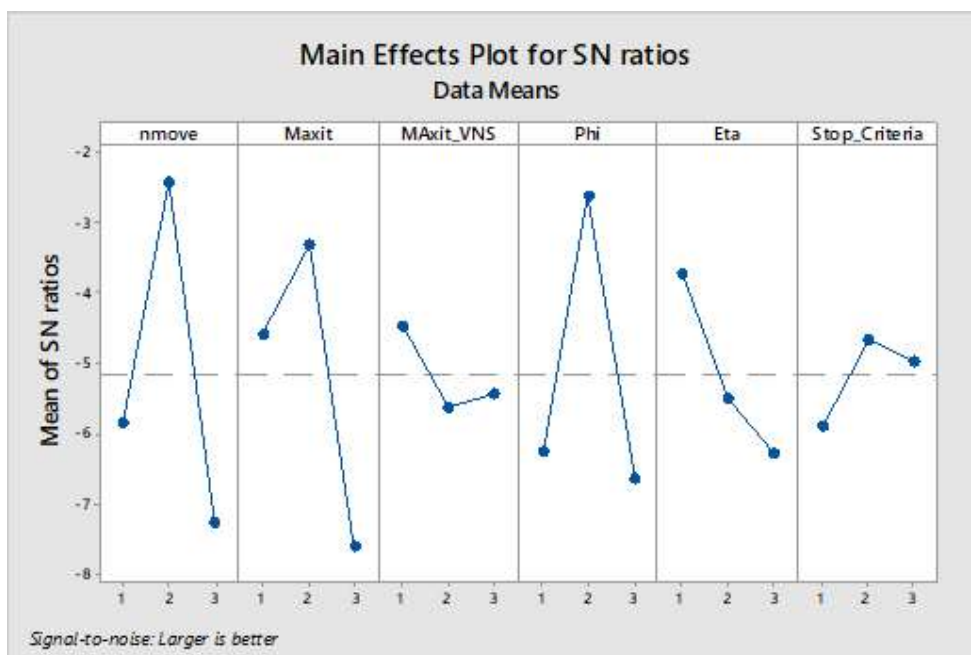
جدول ۴-۵ سطوح معیارهای الگوریتم اول

سطح ۳	سطح ۲	سطح ۱	معیار
۱۰	۸	۵	Nmove
۴۵۰	۳۰۰	۱۵۰	Maxit
۲۰	۱۵	۱۰	MAxit_VNS
۴۰	۳۰	۲۰	Phi
۰/۷	۰/۵۵	۰/۴	Eta
۱۲۵	۱۰۰	۷۵	Stop_Criteria

در روش تاگوچی از معیار (S/N) استفاده می‌شود. این معیار مقدار تغییرات رخ داده شده در متغیر پاسخ را نشان می‌دهد. برای هر فاکتور، مقدار سطحی بهینه و مناسب است که مقدار معیار (S/N) آن بیشتر باشد پس با توجه به شکل (۴-۱۲) برای شش فاکتور تعداد همسایگی روش ALNS (Nmove) سطح دوم، برای پارامتر حداکثر تعداد تکرار الگوریتم (Maxit) سطح دوم، برای فاکتور حداکثر تعداد تکرار الگوریتم VNS (Maxit_VNS) مقدار سطح اول، ضریب فی (Phi) مقدار سطح دوم، ضریب اتا (Eta) مقدار سطح اول و برای معیار توقف (Stop_Criteria) مقدار سطح دوم برای اجرای پارامتر انتخاب شده اند. نتایج نهایی تنظیم پارامترها به شرح جدول (۴-۶) می‌باشد:

جدول ۴-۶ نتایج تنظیم پارامترها الگوریتم ترکیبی ALNS-VNS

سطوح انتخابی	تعداد همسایگی	تعداد تکرار الگوریتم	تعداد تکرار VNS	نرخ Phi	نرخ Eta	معیار توقف
سطوح انتخابی	۸	۳۰۰	۵	۳۰	۰/۴	۱۰۰



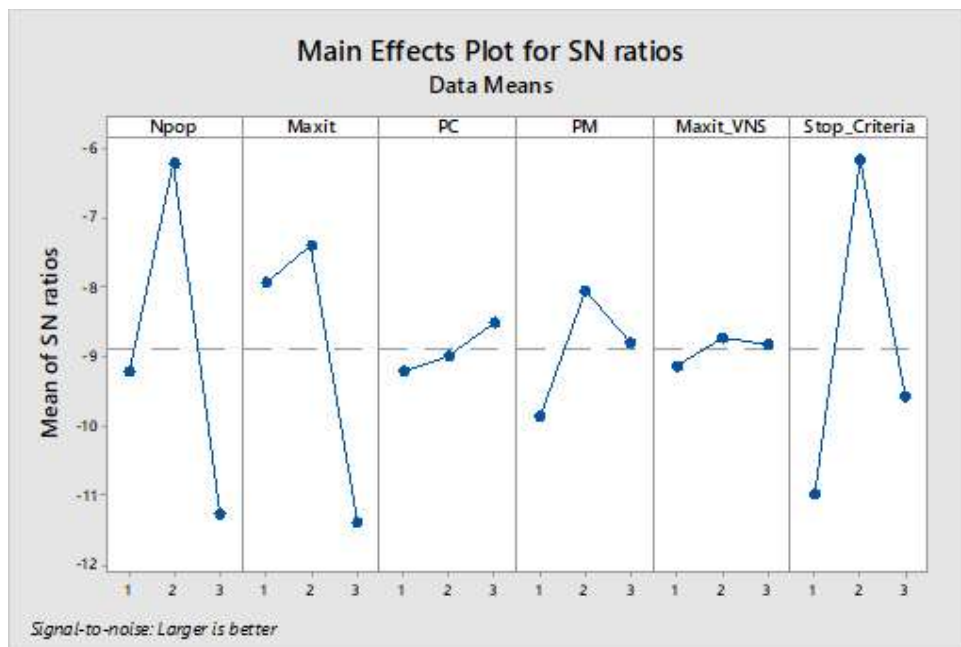
شکل ۴-۱۲ نمودار نرخ S/N برای ضرایب الگوریتم ترکیبی ALNS-VNS

برای روش ترکیبی ژنتیک و VNS نیز شش فاکتور تعداد جمعیت اولیه (Npop)، حداکثر تعداد (Max_it)، نرخ ترکیب (PC)، نرخ جهش (PM)، حداکثر تعداد تکرار الگوریتم VNS (Maxit_VNS) و معیار توقف (Stop_Criteria) در نظر گرفته شده است. همچنین معیار ضریب نرمالایز شده مقدار تابع هدف و زمان حل که عددی بین صفر و یک است به عنوان معیار پاسخ در نظر گرفته شده اند. همچنین برای هر فاکتور سه سطح به صورت جدول زیر در نظر گرفته شده است.:

جدول ۴-۷ سطوح معیارهای الگوریتم دوم

سطح ۳	سطح ۲	سطح ۱	معیار
۷۵	۵۰	۳۰	Npop
۳۰۰	۲۰۰	۱۰۰	Maxit
۰/۹۵	۰/۸۵	۰/۷	PC
۰/۰۵	۰/۱۵	۰/۳	PM
۲۰	۱۵	۱۰	Maxit_VNS
۵۰	۴۰	۳۰	Stop_Criteria

با توجه به شکل (۴-۱۳) برای هر شش فاکتور Npop، Maxit، PC، PM، Maxit_VNS، Stop_Criteria به ترتیب سطح دوم (۵۰)، سطح دوم (۲۰۰)، سطح سوم (۰/۹۵)، سطح دوم (۰/۱۵)، سطح دوم (۱۵) و سطح دوم (۴۰) بهترین مقادیر می باشند.



شکل ۴-۱۳ نمودار نرخ S/N برای ضرایب الگوریتم ترکیبی ژنتیک و VNS

جدول ۴-۸ نتایج تنظیم پارامترهای الگوریتم ترکیبی ژنتیک و VNS

Stop_Criteria	Maxit_VNS	PM	PC	Maxit	Npop	
۴۰	۱۵	۰/۱۵	۰/۹۵	۲۰۰	۵۰	سطوح انتخابی

۴-۱۲ نتایج محاسباتی الگوریتم‌های پیشنهادی

در این بخش، برای اعتبارسنجی مدل از روش دقیق استفاده شده است. کلیه نتایج حاصل از اجرای برنامه‌ها در سیستم یک نوت بوک با پردازنده Intel® Core™ 2 Duo CPU و 2.0 GHz و حافظه 4.00 Gb RAM و سیستم عامل Microsoft Windows 10 می‌باشد. برای حل مسئله از نرم افزار گمز استفاده شده است. با

توجه به NP-Hard مدل با افزایش سائز مسئله، زمان حل محاسباتی به شدت بالا می‌رود. از این رو در این تحقیق بهترین جواب یافت شده توسط نرم افزار گمز در دو ساعت یا ۷۲۰۰ ثانیه گزارش شده است.

در این قسمت نتایج محاسبات مثال‌های حل شده در جداول (۹-۴) تا (۱۰-۴) با الگوریتم پیشنهادی ارائه شده است. این الگوریتم در نرم افزار MATLAB ورژن 2013b برنامه نویسی شده است. ۱۵ مسئله طراحی شده توسط الگوریتم ترکیبی ژنتیک و VNS و الگوریتم ترکیبی ALNS-VNS و نرم افزار گمز حل شده و نتایج آن در جدول (۹-۴) و جدول (۱۰-۴) گزارش شده است. در جدول (۹-۴) مقدار تابع هدف حاصل از الگوریتم‌ها محاسبه شده است و در جدول (۱۰-۴) زمان محاسباتی حاصل از الگوریتم‌ها ارائه شده است. در جدول (۹-۴) ستون اول شماره مسئله براساس جدول (۴-۴) می‌باشد. در ستون دوم جواب بهینه بدست آمده توسط نرم افزار گمز را نشان می‌دهد. حداکثر مدت زمان هر اجرای گمز ۷۲۰۰ ثانیه می باشد و در صورت اینکه زمان حل بیشتر از این زمان باشد بهترین جواب یافت شده در مدت زمان ۷۲۰۰ ثانیه گزارش شده است. همچنین در صورتیکه در مدت زمان ۷۲۰۰ ثانیه جواب یافت نشد، مقدار تابع هدف گزارش نشده است. در این تحقیق از موتور حل Cplex استفاده شده است. با توجه به حجم بالای مدل و NP-Hard بودن مسئله، نرم افزار گمز بعد از دو ساعت نتوانست جواب بهینه برای مسائل بالای نقطه شبکه را به دست بیاورد. ستون‌های بعدی مقدار تابع هدف بدست توسط الگوریتم‌های پیشنهادی ارائه شده و در صورت وجود گپ با جواب بدست آمده از نرم افزار گمز درصد گپ بدست آمده گزارش شده است. در جدول (۱۰-۴) زمان حل محاسباتی ارائه شده است. ستون دوم به بعد نیز حل محاسباتی برای الگوریتم‌های مربوطه نمایش داده شده است. زمان محاسباتی برحسب ثانیه می باشد. همانطور که از جدول مشخص می باشد نرم افزار گمز قادر به یافتن جواب بهینه نهایی در مدت زمان دو ساعت نمی‌باشد. الگوریتم به ازای

۵ اجرا ارائه شده است. ستون GAP حاصل از مسئله براساس فرمول
$$\frac{\text{مقدار تابع هدف گمز} - \text{مقدار تابع هدف فرآیندکاری}}{\text{گمز هدف تابع مقدار}}$$

محاسبه شده و در ستون آخر زمان حل الگوریتم ارائه شده است.

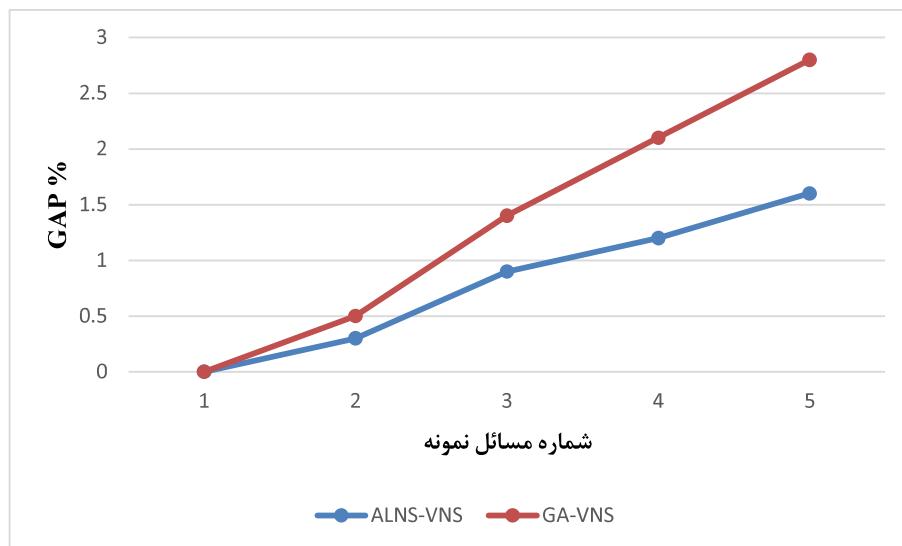
جدول ۹-۴ نتایج الگوریتم های پیشنهادی

نمونه مسئله	تابع هدف				
	مقدار تابع هدف گمز	ALNS-VNS		GA-VNS	
		مقدار تابع هدف	درصد شکاف	مقدار تابع هدف	درصد شکاف
۱	۶۸۴۸	۶۸۴۸	۰/۰	۶۸۴۸	۰/۰
۲	۴۶۵۵۸	۴۶۶۹۸	۰/۳	۴۶۷۹۱	۰/۵
۳	۹۹۱۵۵	۱۰۰۰۴۷	۰/۹	۱۰۰۵۴۳	۱/۴
۴	۱۸۲۹۷۷	۱۸۵۱۷۳	۱/۲	۱۸۶۸۲۰	۲/۱
۵	۳۳۱۱۰۸	۳۳۶۴۰۶	۱/۶	۳۴۰۳۷۹	۲/۸
۶	-	۵۷۹۳۷۹	-	۵۹۲۱۳۵	-
۷	-	۹۵۵۰۶۳	-	۹۸۲۳۹۷	-
۸	-	۱۴۸۸۱۶۰	-	۱۵۴۳۸۲۶	-
۹	-	۲۲۴۵۳۶۳	-	۲۳۵۰۳۹۹	-
۱۰	-	۳۰۷۰۹۲۱	-	۳۲۵۸۳۸۲	-
۱۱	-	۴۰۵۰۲۶۹	-	۴۳۴۷۵۸۴	-
۱۲	-	۴۷۷۰۹۳۹	-	۵۱۶۵۹۴۸	-
۱۳	-	۵۵۳۱۱۰۱	-	۶۰۲۸۴۸۹	-
۱۴	-	۶۲۷۹۶۷۴	-	۶۹۱۴۲۷۲	-
۱۵	-	۷۱۹۴۷۵۳	-	۷۹۶۳۷۰۵	-
میانگین	-	۲۴۵۶۰۵۳	-	۲۶۵۵۲۳۵	-

جدول ۴-۱۰ نتایج زمان حل محاسباتی الگوریتم‌های پیشنهادی

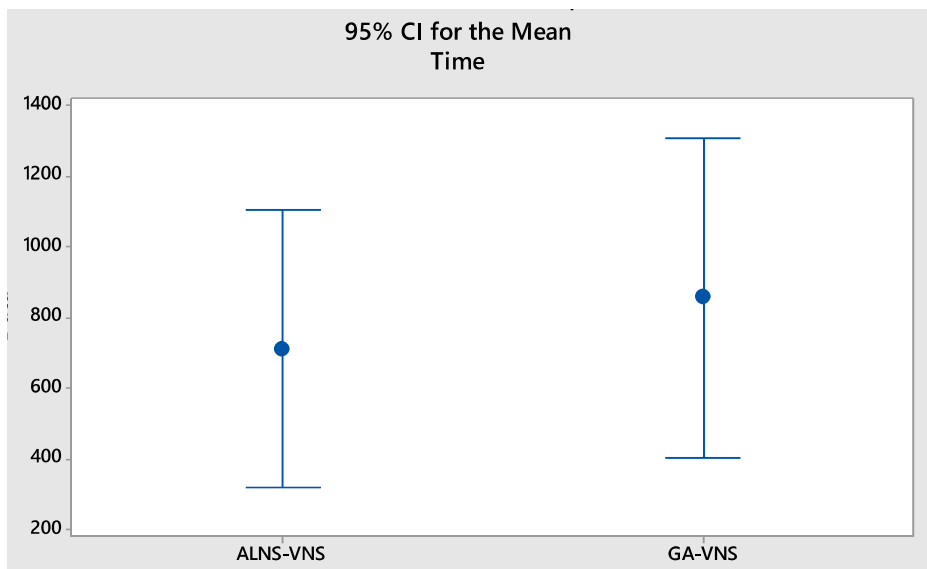
مسئله نمونه	زمان		
	گمز	ALNS-VNS	GA-VNS
۱	۱۶	۹	۱۵
۲	۱۴۵۶	۲۶	۳۷
۳	۷۲۰۰	۴۲	۵۶
۴	۷۲۰۰	۸۹	۱۱۷
۵	۷۲۰۰	۱۸۵	۲۵۳
۶	-	۲۲۹	۲۹۲
۷	-	۳۷۶	۵۲۷
۸	-	۴۵۵	۶۶۷
۹	-	۵۴۰	۸۴۱
۱۰	-	۸۴۱	۱۰۱۶
۱۱	-	۱۰۰۸	۱۲۷۲
۱۲	-	۱۱۶۵	۱۴۲۶
۱۳	-	۱۴۷۰	۱۸۱۳
۱۴	-	۱۹۰۶	۲۱۱۵
۱۵	-	۲۲۱۸	۲۵۶۰

شکل (۴-۱۴) زمان گپ محاسباتی دو الگوریتم را با پارامترهای تنظیم شده توسط روش تاگوچی نشان می دهد. محور عمودی مقدار گپ را نشان می دهد. همانطور که از شکل مشخص است، گپ حاصل از الگوریتم ترکیبی ALNS-VNS نسبت به الگوریتم دیگر کمتر بوده و این الگوریتم کارایی بیشتری از خود نشان داده است. (برای مقایسه گپ الگوریتم‌ها با خروجی نرم‌افزار گمز و رسم نمودار، برای ۲ مسئله نمونه اول در گمز جواب global به دست آوردیم. برای مسئله نمونه ۳، ۴ و ۵ در گمز به ۷۲۰۰ ثانیه اجرای نرم‌افزار اکتفا کردیم).



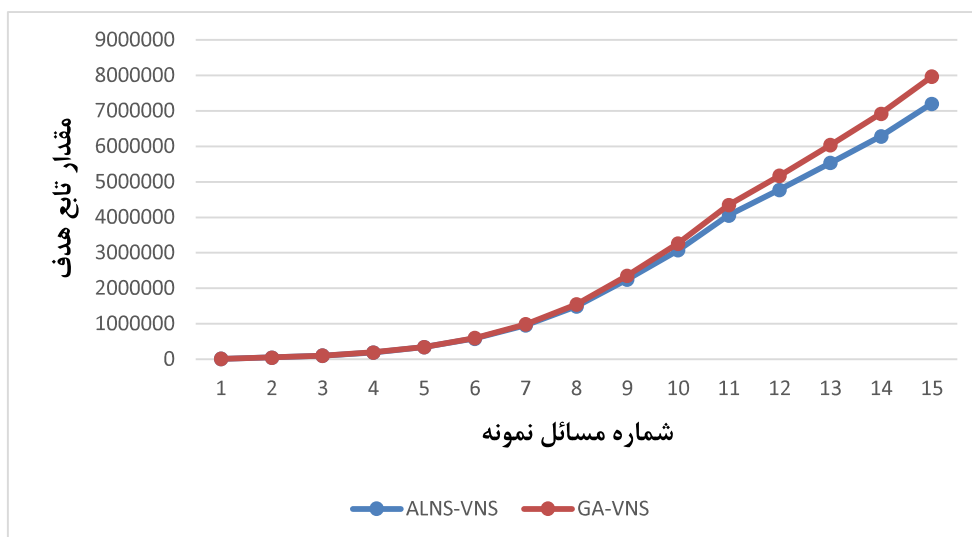
شکل ۴-۱۴ مقایسه گپ محاسباتی روش های حل پیشنهادی

شکل (۴-۱۵) زمان حل دو الگوریتم را با پارامترهای تنظیم شده توسط روش تاگوچی نشان می‌دهد. محور عمودی زمان محاسبه را نشان می‌دهد که بر حسب ثانیه می‌باشد. هر یک از این نمودارها حداقل و حداکثر زمان حل و میانه زمان حل بدست آمده برای ۱۵ مسئله حل شده توسط هر الگوریتم در هر حالت را نشان می‌دهد. همانطور که از شکل مشخص می‌باشد کمترین زمان حل و میانگین زمان حل برای الگوریتم ALNS-VNS کمتر می‌باشد. الگوریتم های ALNS و VNS هر دو براساس جستجوی محلی و تولید همسایگی عمل می‌کنند و معمولاً الگوریتم های مبتنی بر رویکردهای همسایگی نسبت به الگوریتم های مبتنی بر جمعیت مانند الگوریتم ژنتیک زمان محاسباتی کمتر دارند لذا علت این سرعت محاسباتی مناسبتر ALNS-VNS می‌تواند این امر باشد.

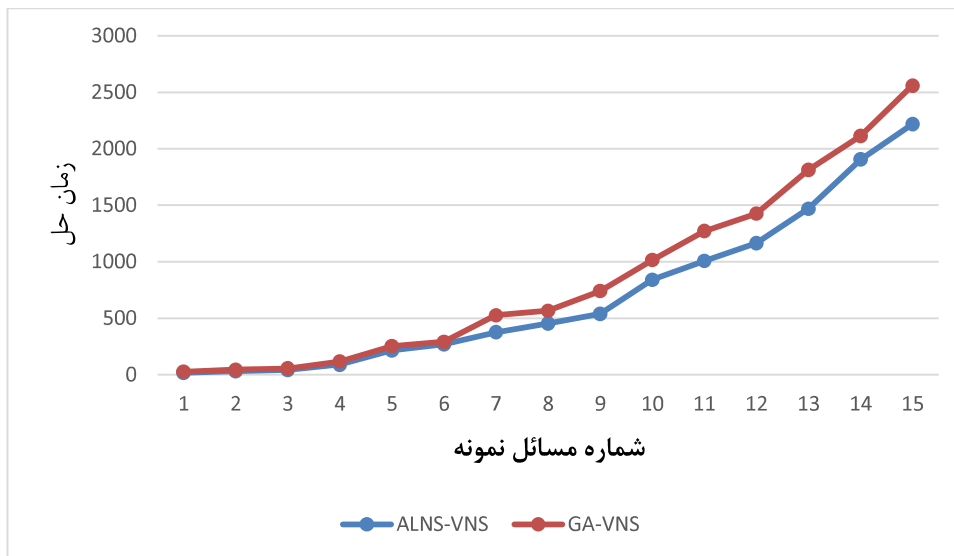


شکل ۴-۱۵ مقایسه نمودار بازه ای زمان محاسباتی روش های حل پیشنهادی

شکل (۴-۱۶) و (۴-۱۷) مقدار تابع هدف و زمان حل دو الگوریتم پیشنهادی برای ۱۵ مسئله نشان داده است. محور افقی شماره مسائل حل شده و محور عمودی مقدار تابع هدف و زمان حل می باشد.

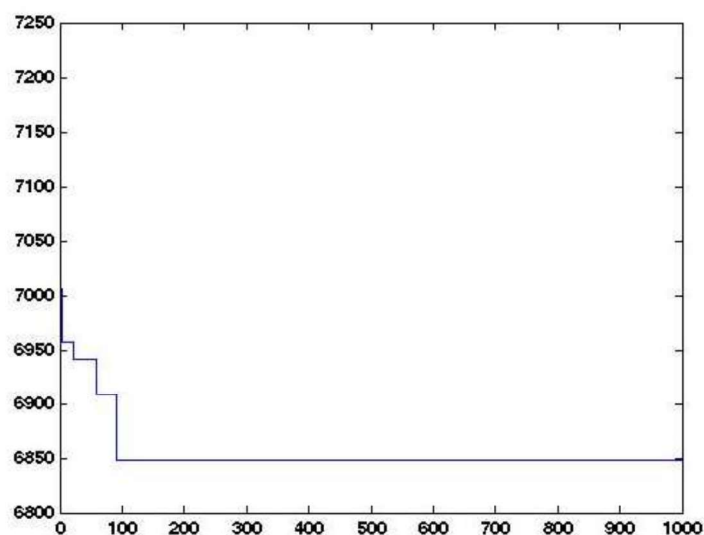


شکل ۴-۱۶ مقایسه مقدار تابع هدف الگوریتم های پیشنهادی



شکل ۴-۱۷ زمان حل الگوریتم‌های پیشنهادی

شکل (۴-۱۸) نیز بطور نمونه نمودار همگرایی مربوط به الگوریتم ALNS-VNS برای مسئله نمونه ۱ نشان داده شده است که نشان از همگرایی مناسب الگوریتم مربوطه دارد.



شکل ۴-۱۸ نمودار همگرایی مسئله ۱ برای الگوریتم ترکیبی ALNS-VNS