

## Solving Urban Bus Terminal Location Problem Using Genetic Algorithm

S.M. Seyed-Hosseini\*, R. Heydari & T. Heydari

S.M. Seyed-Hosseini, Department of Industrial Engineering, Iran University of Science & Technology, Tehran, Iran

R. Heydari, Department of Industrial Engineering, Shahed University

T. Heydari, Department of Industrial Engineering, Iran University of Science & Technology, Tehran, Iran

### Keywords

Bus Networks,  
Location Problem, Genetic  
Algorithm (GA),  
Mixed Integer Programming  
(MIP)

### ABSTRACT

Bus network design is an important problem in public transportation. The main step to this design, is determining the number of required terminals and their locations. This is an especial type of facility location problem, a large scale combinatorial optimization problem that requires a long time to be solved. Branch & bound and simulated annealing methods have already been used for solving Urban Bus Terminal Location Problem (UBTLP) that first method requires much time and second method doesn't reach desired solution. In this paper, a Genetic Algorithm is suggested for solving the problem. The main advantages of proposed algorithm are reaching better solution and taking less time. The demonstrated results have shown that proposed Genetic Algorithm can be an efficient and confident approach for solving UBTLP. For verification of proposed methodology, two illustrative practical examples are solved and obtained outcomes are reported.

© (نشریه بین المللی مهندسی صنایع و مدیریت تولید) شماره ۳، جلد ۲۰، ۱۳۸۸

## حل مساله مکان یابی پایانه های شبکه اتوبوس رانی درون شهری با استفاده از الگوریتم ژنتیک

سید محمد سید حسینی، روح اله حیدری و طاهره حیدری

### چکیده:

طراحی شبکه های اتوبوس رانی، یک مساله مهم در حمل و نقل عمومی است. یکی از گام های مهم در این راستا، محاسبه تعداد و مکان های بهینه پایانه های مورد نیاز است. در واقع این مساله به عنوان حالت خاصی از مساله مکان یابی تسهیلات، که یک مساله بهینه سازی ترکیبیاتی با مقیاس بزرگ است، نیاز به زمان زیادی برای حل دارد.

### کلمات کلیدی

شبکه ای اتوبوس رانی،  
مساله مکان یابی،  
الگوریتم ژنتیک،  
برنامه ریزی عدد صحیح مختلط

تاریخ وصول: ۸۶/۱۰/۲۲

تاریخ تصویب: ۸۷/۸/۱۵

دکتر سید محمد سید حسینی، دانشکده مهندسی صنایع، دانشگاه علم و صنعت ایران [Seyedhoseini@yahoo.com](mailto:Seyedhoseini@yahoo.com)

روح اله حیدری، دانشکده فنی و مهندسی، دانشگاه شاهد، تهران. [Haidari@shahed.ac.ir](mailto:Haidari@shahed.ac.ir)

طاهره حیدری، گروه فناوری اطلاعات، دانشگاه پیام نور دامغان. [T\\_heydari@pnu.ac.ir](mailto:T_heydari@pnu.ac.ir)

تا کنون برای حل این مساله، از روش‌های شاخه و کران، شمارش ضمنی و گرم و سرد کردن شبیه سازی شده استفاده شده است. هر چند روش سوم که یک روش فرا ابتکاری است بسیار کارا تر از دو روش دیگر است؛ اما زمان اجرای الگوریتم، برای شهرهای بزرگ هنوز هم طولانی است. در این مقاله، برای حل مساله یک الگوریتم ژنتیک پیشنهاد شده است. مهمترین مزیت الگوریتم ژنتیک پیشنهادی، رسیدن به جواب دقیق‌تر در زمان کمتر می‌باشد. نتایج مشاهده شده، نشان داد که متدولوژی پیشنهادی یک الگوریتم کارا و مطمئن برای این مساله است. برای تایید این مطلب، الگوریتم برای شبکه های اتوبوسرانی مشهد و تهران اجرا و نتایج آن با نتایج کوشش های پیشین مقایسه شده است.

## ۱. مقدمه

یکی از مسایل مطرح در طراحی سیستم‌های اتوبوس‌رانی، مشخص کردن مکان‌های مناسب جهت ایجاد پایانه‌های اتوبوس‌رانی درون شهری است. در حالت کلی، مساله‌ی مکان‌یابی تسهیلات نوعی مساله‌ی بهینه سازی است که هدف آن انتخاب زیر مجموعه‌ای از یک مجموعه محل‌های کاندید برای قراردادن تسهیلات است که بیشترین خدمت‌دهی یا کمترین هزینه را فراهم سازد. برای مطالعه‌ی بیشتر در زمینه‌ی انواع مسایل مکان‌یابی تسهیلات، به مرجع [۱] مراجعه نمایید.

در زمینه مکان‌یابی و مسیریابی تسهیلات حمل و نقل جاده‌ای، تا کنون تلاش‌های زیادی صورت گرفته است. Lapierre و همکاران چارچوبی برای طراحی سیستم توزیع ارایه کردند که در آن برای تعیین تعداد و مکان پایانه‌های بارگیری از یک الگوریتم جستجوی ممنوع بهره گرفته شده است [۲]. Bander و White مدلی ابتکاری برای طراحی مسیر بهینه پیشنهاد دادند که مجموع هزینه حمل و نقل را کمینه می‌کند [۳]. Bouzaïene-Ayari و همکاران مدلی برای بهینه‌سازی میزان توقف اتوبوس‌ها در دو حالتی که محدودیت ظرفیت شبکه نرم یا سخت باشد؛ ارایه کردند [۴]. Ginkel و Schöbel از یک بهینه سازی دو معیاره برای ارضای دو هدف متناقض کاهش میزان تاخیر اتوبوس‌ها و کاهش تعداد افرادی که به علت حرکت یک اتوبوس، آن اتوبوس را از دست می‌دهند؛ بهره گرفتند [۵].

در مورد سایر تسهیلات حمل و نقل عمومی نیز، Abbink و همکاران، مدلی برای تخصیص بهینه قطارهای درون‌شهری به مسیرهای پرتراфик در ساعات اوج ترافیک ارایه کردند که برای پاسخ گویی به تقاضای ساعات شروع و پایان کار اداره‌ها و شرکت‌ها، قطارهایی که ظرفیت مسافر بیشتری دارند را به این مسیرها اختصاص می‌دهد [۶].

در یک سیستم اتوبوس‌رانی، ایستگاه‌های آن، که در هر یک، تعدادی مسافر سوار و پیاده می‌شوند؛ می‌توانند به عنوان نقاط کاندید جهت احداث پایانه در نظر گرفته شوند. هدف، انتخاب تعدادی از این ایستگاه‌ها، به عنوان پایانه است که میزان خدمت‌دهی بیشینه گردد.

برای حل مساله مکان‌یابی پایانه های شبکه اتوبوس‌رانی درون‌شهری که از این پس با UBTLP<sup>۲</sup> نیز نشان خواهیم داد؛ می‌توان آن را در قالب یک برنامه‌ریزی عدد صحیح مختلط مدل کرد که به یک مساله-ی بهینه‌سازی ترکیبی با مقیاس بزرگ تبدیل خواهد شد. حل این مدل برای شهرهای بزرگ، بسیار پیچیده و وقت‌گیر است. یک راه حل آن کاهش ابعاد مساله است. با توجه به آنکه مهمترین مشکل مساله، تعداد متغیرهای صحیح آن، یعنی تعداد گره‌های کاندید است؛ می‌توان تنها از گره‌هایی از شبکه استفاده کرد که شانس بیشتری در انتخاب نهایی دارند و با استفاده از روش شاخه و کران، به کمک نرم افزار GAMS مساله را حل کرد [۷].

این روش حتی برای شهرهای کوچک نیز وقت گیر است و برای شهرهای بزرگ، قابل اجرا نیست. در مرجع [۸] از روش گرم و سرد کردن شبیه سازی شده که از این پس با SA<sup>۳</sup> هم نشان خواهیم داد؛ استفاده شده و در مرجع [۹] نتایج حاصل از این دو روش و روش شمارش ضمنی برای شبکه اتوبوس‌رانی شهر مشهد با هم مقایسه شده است. با در نظر گرفتن دو عامل زمان اجرا و دقت جواب به عنوان معیار کارایی، روش SA در تمامی موارد کارا تر از روش های شمارش ضمنی و شاخه و کران بوده است. اما تاکنون سایر الگوریتم‌های فراابتکاری برای حل مساله به کار گرفته نشده است تا مشخص شود از روش SA کارا تر است یا خیر.

در این مقاله، ابتدا مساله را به صورت یک برنامه‌ریزی عدد صحیح مختلط مدل‌سازی کرده، سپس برای حل آن در ابعاد بزرگ از طریق الگوریتم ژنتیک، که از این پس با GA<sup>۴</sup> نیز نشان می‌دهیم؛ روش‌های مناسبی برای کدگذاری، ایجاد جمعیت اولیه، عملگرهای انتخاب، تقاطع و جهش ارایه کرده‌ایم.

شرط توقف نیز تولید تعداد معلومی نسل در نظر گرفته شده است. برای محاسبه‌ی پارامترهای بهینه‌ی الگوریتم ژنتیک (نرخ تقاطع و نرخ جهش) از روش سعی و خطای عددی روی مقادیر مختلف، پارامترها در یک مساله‌ی تجربی، در محیط نرم‌افزاری MATLAB (نسخه ۷) استفاده کرده‌ایم. در انتها، الگوریتم را برای شبکه های

<sup>2</sup> Urban Bus Terminal Location Problem

<sup>3</sup> Simulated Annealing

<sup>4</sup> Genetic Algorithm

شبکه پوشش داده شود. با استفاده از نتایج حاصل از این تخصیص، پتانسیل سفر برای گره‌های شبکه مشخص می‌شود. به علاوه فرض می‌شود که چنانچه پایانه‌ای در یک گره ایجاد شود نه تنها می‌تواند به تمام مسافری خود آن گره خدمت دهی کند؛ بلکه می‌تواند به مسافری تمام گره‌های همسایه نیز خدمت دهد.

**تعریف:** همسایگی گره  $i$  که با  $J_i^*$  نشان داده می‌شود؛ به مجموعه گره‌هایی در شبکه اطلاق می‌شود که در فاصله‌ای کوچکتر یا مساوی  $r$  از گره  $i$  قرار گرفته‌اند. در این تعریف،  $r$  یک ثابت است و شعاع همسایگی نامیده می‌شود.

### ۳. مدل سازی مساله به عنوان یک برنامه ریزی عدد

#### صحيح مختلط

برای مدل‌سازی ریاضی مساله مکان‌یابی پایانه‌های سیستم اتوبوس‌رانی، پارامترها و متغیرهای زیر را تعریف می‌کنیم [۷]:

- پارامترهای مورد نیاز:

$J$ : مجموعه گره‌های شبکه

$I \subseteq J$ : مجموعه گره‌های کاندید برای ایجاد پایانه

$c_{ij}$ : فاصله هوایی بین گره‌های  $i$  و  $j$

$d_j$ : پتانسیل گره  $j$  برای  $j \in J$

$f(c_{ij})$ : یک تابع نمایی نزولی از  $c_{ij}$  که در این مقاله برابر  $e^{-c_{ij}}$  در نظر گرفته شده است.

$J_i^* \subseteq J$ : مجموعه گره‌هایی از شبکه که می‌توانند از گره  $i$  خدمت بگیرند.

$k$ : تعداد پایانه‌های مورد نیاز

$M$ : یک عدد بزرگ مثبت

- متغیرهای مورد نیاز:

$x_{ij}$ : سهم خدمت‌دهی گره  $i$  ( $i \in I$ ) به گره  $j$  ( $j \in J - I$ )

$y_i$ : متغیری صفر و یک، که اگر گره  $i$  به عنوان پایانه انتخاب شود مقدار یک می‌گیرد و در غیر این صورت مقدار آن برابر صفر خواهد بود.

تابع هدف و محدودیت‌های مدل به صورت زیر تعریف می‌شوند:

$$\text{Max} \sum_{i \in I} \sum_{j \in J_i^*} d_j \cdot f(c_{ij}) \cdot x_{ij} \quad (1)$$

$$\text{s.t.} \sum_{i \in I: j \in J_i^*} x_{ij} \leq 1 \quad \forall j \in J - I \quad (2)$$

$$\sum_{j \in J_i^*} x_{ij} \leq M \cdot y_i \quad \forall i \in I \quad (3)$$

اتوبوس‌رانی مشهد و تهران به کار گرفته و نتایج حاصل را با نتایج حاصل از سایر روش‌ها مقایسه نموده‌ایم.

### ۲. تعریف مساله

رشد روز افزون جمعیت، تمرکز اکثریت آن‌ها در محل‌های مناسب‌تر و در عین حال پراکندگی سایر افراد در نقاط دیگر، افزایش تعداد اتومبیل‌ها در خیابان‌ها و ترافیک سنگین در خیابان‌های شهرهای بزرگ، گسترش سیستم حمل‌ونقل عمومی درون شهری را به عنوان یک نیاز اساسی پیش روی دولت‌ها قرار داده است که در این میان گسترش شبکه‌ی اتوبوس‌رانی نقش بسزایی در کاهش مشکلات فوق خواهد داشت.

مساله‌ی طراحی سیستم حمل و نقل عمومی، از جمله اتوبوس‌رانی، یکی از مسایل پیچیده در برنامه ریزی حمل و نقل است که شامل زیر مساله‌های مختلفی می‌شود. این مساله از یک دیدگاه به دو بخش تعیین ساختار شبکه و تعیین برنامه‌ی عملیاتی سیستم تقسیم می‌شود. تعیین ساختار شبکه، شامل تعیین تعداد و محل پایانه‌های شبکه و تعیین مسیر خطوط می‌شود [۹]. و برنامه‌ی عملیاتی شامل تعداد اتوبوس‌های هر مسیر، زمان بندی حرکت و مدیریت تاخیرها می‌شود [۵]. در تحقیق حاضر، قسمتی از بخش اول مساله، یعنی تعیین محل پایانه‌های شبکه مورد نظر قرار می‌گیرد. برای مطالعه بیشتر در زمینه طراحی شبکه‌های اتوبوس‌رانی به مرجع [۱۰] مراجعه کنید.

از دیدگاه دیگر، توابع هدف در مسایل مکان‌یابی به دو دسته تقسیم می‌شوند. دسته اول، سعی در افزایش کارایی کل سیستم دارند و به تک تک نقاط تقاضا توجه چندانی ندارند. در مرجع [۱۱] برای کاهش کل آلودگی صوتی ایجاد شده از استقرار یک تسهیل آلودگی ساز از تابع *minisum* استفاده شده است. دسته‌ی دیگر که معمولاً برای مکان‌یابی تسهیلات اورژانسی و یا تجهیزات خطر آفرین کاربرد دارد؛ سعی می‌کنند تسهیل یا تسهیلات جدید را در مکانی مستقر کنند که میزان بهره‌مندی یا گله مندی نقاط تقاضا از آن یکسان باشد. مانند تابع هدف *minimax* یا *maximin* این توابع به معیارهای مساوات معروفند. برای آشنایی با انواع مدل‌های مساوات به مرجع [۱۲] مراجعه کنید. تابع هدفی که در این مقاله استفاده می‌شود نیز سعی در افزایش کارایی کل شبکه اتوبوس‌رانی دارد.

برای بیان مساله‌ی مکان‌یابی پایانه‌های سیستم اتوبوس‌رانی، یک شبکه خیابانی مد نظر قرار می‌گیرد. فرض می‌شود تمام گره‌های شبکه می‌توانند به عنوان ایستگاه‌های سیستم اتوبوس‌رانی در نظر گرفته شوند. همچنین فرض می‌شود که میزان سوار و پیاده شدن مسافری در هر گره (پتانسیل گره)، در دست است. همان‌گونه که در مرجع [۷] آمده است؛ برای تعیین این پارامتر، می‌توان یک سیستم فرضی اتوبوس‌رانی تعریف کرد؛ به گونه‌ای که از تمام خیابان‌های اصلی شبکه، حداقل یک خط اتوبوس عبور کند و تا حد امکان تمام

متغیرهای صحیح، در هر شاخه یک برنامه ریزی خطی برای تعیین مقادیر سایر متغیرها حل می‌شود.

روش شمارش ضمنی حالت خاصی از روش شاخه و کران است که در هر شاخه مقادیر تمام متغیرهای صحیح مشخص اند. این روش وقتی کارایی دارد که پس از مشخص شدن مقادیر متغیرهای صحیح، حل باقی‌مانده‌ی مساله برای تعیین متغیرهای پیوسته ساده باشد و نیازی به حل یک برنامه ریزی خطی نباشد. در مساله‌ی مکان‌یابی پایانه‌ها، پس از متغیرهای صحیح  $(y_i)$ ، حل مساله‌ی باقی‌مانده و تعیین متغیرهای پیوسته  $(x_{ij})$  ساده است و نیازی به حل برنامه ریزی خطی نیست. زیرا هر گرهی از نزدیک‌ترین پایانه خدمت می‌گیرد.

#### ۵. روش‌های فرا ابتکاری حل مساله

در بخش قبل، توضیحاتی پیرامون روش‌های دقیق حل مسایل بهینه‌سازی ارائه شد. باید توجه داشت که روش‌های مذکور برای مسایل با ابعاد بزرگ کارایی خود را از دست می‌دهند و عملکردی بهتر از شمارش کامل نخواهند داشت. به دلایل بالا، اخیراً تمرکز بیشتری بر روی روش‌های ابتکاری صورت گرفته است. روش جستجوی ابتکاری، روشی است که با جستجو در بین جواب‌های ممکن، جوابی خوب (نزدیک به بهینه) در زمانی مطلوب برای یک مساله ارائه می‌دهد. معمولاً هیچ تضمینی برای بهینه بودن جواب وجود ندارد و حتی نمی‌توان میزان نزدیکی جواب به دست آمده به جواب بهینه را نیز تعیین کرد. به منظور مطالعه بیشتر در زمینه انواع الگوریتم‌های فراابتکاری به مراجع [۱۵-۱۳] مراجعه کنید.

یکی از روش‌های فرا ابتکاری که در مرجع [۹] برای حل مساله مکان‌یابی پایانه‌ها به کار گرفته شده است روش گرم و سرد کردن شبیه‌سازی شده است. این روش برای اولین بار توسط متراپلیس، کرک پاتریکا و سرنی ابداع شد و وجه تسمیه آن شباهت به پدیده‌ی گرم و سرد کردن جامدات مذاب است. جواب حاصل از این روش به جواب اولیه وابسته نیست و به طور معمول می‌توان توسط آن، جوابی نزدیک به بهینه به دست آورد. در مرجع [۹] برای حل مساله‌ی UBTLP یک الگوریتم SA پیشنهاد شده است که قسمتی از نتایج آن در این مقاله نیز آمده است.

یکی دیگر از الگوریتم‌های فرا ابتکاری، الگوریتم ژنتیک است که سرعت همگرایی آن از روش SA بیشتر است [۱۶]. از آن‌جا که در این مقاله از یک الگوریتم ژنتیک برای حل مساله‌ی UBTLP بهره گرفته ایم؛ در قسمت بعد به معرفی کامل این الگوریتم می‌پردازیم.

#### ۶. حل مساله به کمک الگوریتم ژنتیک

##### ۶-۱. تاریخچه‌ی الگوریتم ژنتیک

محققان با مشاهده‌ی سازگاری، بقا، خودترمیمی، هدایت، تولید مثل و دیگر خصوصیات سیستم‌های طبیعی و تامل در این که

$$\sum_{i \in I} y_i = k \quad (۴)$$

$$x_{ij} \geq 0 \quad \forall i \in I, j \in J_i^* \quad (۵)$$

$$y_i = 0, 1 \quad \forall i \in I \quad (۶)$$

تابع هدف (۱)، میزان خدمت‌دهی گره‌های کاندید به تمام گره‌ها را مشخص می‌کند. مسلماً هر چه پتانسیل گره  $j$  بالاتر باشد؛ یعنی تعداد مسافرین سوار و پیاده شده در آن بیشتر باشد و فاصله کمتری تا گره کاندید  $i$  داشته باشد، میزان خدمت‌دهی گره  $i$  باید بیشتر شود. بنابراین تابع  $f$  در (۱) به صورت زیر تعریف می‌شود.

$$f(c_{ij}) = e^{-c_{ij}} \quad (۷)$$

محدودیت (۲) نشان می‌دهد که مجموع سهم خدمتی که هر مسافر گره  $j$  از گره‌های کاندید می‌گیرد، باید حداکثر برابر یک باشد. محدودیت (۳) نشان می‌دهد که تنها زمانی که گره  $i$  به عنوان پایانه انتخاب شود ( $y_i = 1$ ) به گره‌های همسایه خدمت می‌دهد. در غیر این صورت به هیچ گره‌ای خدمت نداده و در نتیجه  $x_{ij}$  برای تمام گره‌های  $j \in J_i^*$  برابر با صفر می‌شود. محدودیت (۴) تعداد پایانه‌های مورد نیاز را کنترل می‌کند. و محدودیت‌های (۵) و (۶) نیز محدودیت‌های علامت و صفر و یک مساله هستند.

#### ۴. روش‌های دقیق حل مساله

در دهه‌های پنجاه و شصت، بهینه‌سازی بیشتر به مدل‌های خطی و غیر خطی با متغیرهای پیوسته، مربوط می‌شد. بهینه‌سازی ترکیبی که مربوط به مسایل بهینه‌سازی با متغیرهای گسسته می‌شود در دهه‌ی ۷۰ مورد توجه قرار گرفت [۹].

یک روش ناشیانه برای حل مسایل بهینه‌سازی ترکیبی، شمارش کامل است؛ که در آن برای تمامی ترکیب‌های ممکن، مقدار تابع هدف محاسبه شده و در نهایت، بهترین جواب موجه انتخاب می‌شود. هرچند این شیوه، در نهایت به جواب دقیق مساله می‌رسد؛ اما به دلیل زیاد بودن ترکیبات ممکن در مسایل واقعی، عملاً استفاده از آن غیر ممکن است. با توجه به مشکلات مربوط به روش شمارش کامل، تلاش بسیاری برای توسعه‌ی روش‌های موثرتر و کاراتر صورت گرفته است. در همین راستا، الگوریتم‌های مختلفی به وجود آمده است که مشهورترین نمونه‌های آن، شاخه و کران و شمارش ضمنی اند.

ایده‌ی اصلی روش شاخه و کران، تقسیم ناحیه‌ی موجه به زیر بخش‌های قابل کنترل‌تر و در صورت نیاز تقسیم آن‌ها به زیر بخش‌های کوچک‌تر است. در روش عمومی شاخه و کران، برای حل مسایل برنامه ریزی مختلط، پس از مشخص شدن مقادیر تعدادی از

۲. محاسبه‌ی تابع برزندگی هر کروموزوم در جمعیت
۳. ایجاد جمعیت جدید (عمل تولید نسل):  $N$  بار قدم‌های زیر را انجام ده:
- ۱-۳. انتخاب دو کروموزوم والد از جمعیت جاری با استفاده از عملگر انتخاب
- ۲-۳. اعمال عملگر تقاطع روی والدین با احتمال وقوع  $pc$  و ایجاد فرزند
- ۳-۳. انجام جهش روی فرزند با احتمال وقوع  $pm$
- ۴-۳. اضافه کردن فرزند جدید به جمعیت
۴. انتخاب جمعیت جدید و جای‌گذاری جمعیت جدید با جمعیت جاری با استفاده از عملگر انتخاب
۵. آزمون شرایط توقف و برگرداندن بهترین جواب در صورت توقف
۶. بازگشت به قدم ۲

### ۳-۶. ساختار الگوریتم ژنتیک برای مساله‌ی UBTLP

#### ۱-۳-۶. کدگذاری (CODING)

الگوریتم ژنتیک به جای کار بر روی متغیرهای مساله، با کد شده جواب‌ها، یعنی کروموزوم‌ها سر و کار دارد. در حقیقت عمل کد کردن جواب‌ها، یکی از اصلی‌ترین کارها در پیاده‌سازی الگوریتم ژنتیک بر روی مسایل مختلف است. روش‌های متنوعی برای کدگذاری وجود دارد که کدگذاری دودویی، حقیقی، ترتیبی و درختی از آن جمله‌اند [۲۲-۲۱]. در این مقاله از کدگذاری دودویی استفاده شده است. به این ترتیب که یک رشته با طول  $|I|$ ، یعنی تعداد گره‌های کاندید برای ایجاد پایانه در نظر گرفته شده است. هر یک از عناصر این رشته نقش یک ژن را بازی می‌کند. در یک کروموزوم (یا جواب شدنی)، ژن حاوی مقدار یک نشان می‌دهد که کاندید  $i$ -ام برای احداث پایانه انتخاب شده است و برعکس. به عنوان مثال، اگر در مساله UBTLP، داشته باشیم:

$$J = \{1, 2, 3, \dots, 60\}$$

$$I = \{1, 5, 15, 20, 25, 30, 35, 40, 45, 50\}$$

$$k = 5$$

|   |   |   |   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 1 | 1 | 0 | 0 | 0 | 1 | 1 |
|---|---|---|---|---|---|---|---|---|---|

در این صورت کروموزوم بالا نشان دهنده‌ی یک جواب شدنی مساله است که در آن گره‌های ۱ و ۲۰ و ۲۵ و ۴۵ و ۵۰ برای احداث پایانه انتخاب شده‌اند (به ترتیب عناصر  $I$  به عنوان مکان ژن توجه شود).

#### ۲-۳-۶. ایجاد جمعیت اولیه

در این مقاله یک روند کاملاً تصادفی برای ایجاد جمعیت اولیه (انتخاب تصادفی مکان ژن‌های با محتوای یک، در کروموزوم‌ها) صورت می‌پذیرد. البته در بعضی مسایل، بسته به نوع مساله و برای

طبیعت چگونگی مشکلات خود را حل می‌کند؛ به فکر تقلید از روش‌های طبیعی در حل مسایل پیچیده و طراحی سیستم‌ها افتادند. ایده اصلی الگوریتم ژنتیک هم براساس این فکر و بر پایه‌ی نظریه تکامل داروین شکل گرفت.

در سال ۱۹۶۰ رچنبرگ با طرح تکنیک‌های محاسباتی تکاملی که با تقلید از طبیعت به حل مسایل سخت می‌پرداختند، اولین گام را در این زمینه برداشت. اما اصول اولیه الگوریتم ژنتیک در سال ۱۹۶۳ توسط هالند و همکارانش ارائه شد. آنان برای مدل‌سازی سیستم‌های مصنوعی به فرایند سازگاری سیستم‌های طبیعی روی آوردند. این تحقیقات باعث به وجود آمدن الگوریتم ژنتیک گردید. در سال ۱۹۷۵ هالند به مبانی ریاضی الگوریتم ژنتیک در کتاب مشهور خود پرداخت [۱۸]. با توجه به موفقیت الگوریتم ژنتیک، جان کوزا در سال ۱۹۹۱ برای پیدا کردن الگوریتمی که بتواند جواب هر مساله‌ای را بیابد، مفهوم برنامه‌ریزی تکاملی و برنامه‌ریزی ژنتیکی را ارائه کرد. گلدبرگ، تعریف زیر را برای الگوریتم ژنتیک ارائه کرده است. برای مطالعه بیشتر در زمینه الگوریتم‌های تکاملی به مرجع [۱۷] مراجعه کنید.

**تعریف:** الگوریتم‌های ژنتیکی، تکنیک‌های جستجوی تصادفی هستند که براساس انتخاب طبیعی و نسل‌شناسی طبیعی کار می‌کنند [۱۹].

#### ۲-۶. ساختار کلی الگوریتم ژنتیک

اجازه دهید ابتدا برخی از واژگان ژنتیک را به عنوان اجزای مهم الگوریتم ژنتیک معنی کنیم [۲۰]. سایر مفاهیم نیز در طول متن، معرفی خواهند شد.

- ژن: عناصر تشکیل دهنده‌ی یک کروموزوم که می‌تواند عدد یا کلمه یا حتی دستورات برنامه نویسی باشد.
- کروموزوم: به رمز درآمده یا کد شده‌ی یک جواب یا قسمتی از جواب را کروموزوم می‌گویند. به عنوان مثال، یک کروموزوم با  $n$  ژن، می‌تواند به شکل زیر باشد (اعداد در مبنای  $m$  هستند):

|       |       |       |     |       |
|-------|-------|-------|-----|-------|
| $b_1$ | $b_2$ | $b_3$ | ... | $b_n$ |
|-------|-------|-------|-----|-------|

$$\{0, 1, \dots, m-1\}, i: 1, \dots, n. \in b_i$$

- مکان: هر ژن در کروموزوم موقعیت خاص خودش را داراست که به این موقعیت ژن، مکان می‌گویند.
- جمعیت: به مجموعه‌ای از کروموزوم‌ها جمعیت گفته می‌شود.

در الگوریتم ژنتیک،  $N$  اندازه‌ی جمعیت،  $pc$  احتمال وقوع تقاطع (نرخ تقاطع) و  $pm$  احتمال وقوع جهش (نرخ جهش) در هر تکرار باید معلوم باشند. ساختار کلی الگوریتم به صورت زیر است:

۰. تعیین روش کدگذاری

۱. تولید جمعیت اولیه به اندازه  $N$

که  $f_k$  در این فرمول معرف میزان برآزندگی کروموزوم  $k$ -ام است. برای پیاده‌سازی مدل چرخ رولت در کامپیوتر، از یک شبیه‌سازی به صورت ذیل بهره می‌گیریم:

ابتدا تمام جواب‌های جمعیت به صورت یک دنباله در آورده می‌شوند و احتمال انتخاب هر یک (یعنی مقدار  $p_k$  برای کروموزوم  $k$ -ام) محاسبه می‌شود. سپس مقدار احتمال تجمعی ( $q_k$ ) برای هر جواب، از رابطه زیر محاسبه می‌شود:

$$q_k = \sum_{i=1}^k p_i \quad (9)$$

با ایجاد عدد تصادفی  $r$  در فاصله  $[0,1]$ ، اندیس  $k$  به صورت زیر تعیین می‌شود:

$$k = \min\{j : q_j \geq r\} \quad (10)$$

به عبارت دیگر، کوچکترین اندیسی که احتمال تجمعی بیشتر از  $r$  داشته باشد، انتخاب می‌شود.

#### ۶-۳-۴. عملگرهای تقاطع و جهش

عملگر انتخاب به تنهایی قادر به تولید کروموزوم جدید در جمعیت نیست. تولید کروموزوم‌های جدید توسط عملگرهای ژنتیکی به نام تقاطع و جهش صورت می‌پذیرد که ماهیتی احتمالی دارند.

تقریباً تمام عملگرهای تقاطعی، دو کروموزوم را به طور تصادفی به عنوان والدین برگزیده و به طور تصادفی قسمت یا قسمت‌هایی از دو کروموزوم والد را با هم عوض می‌کند. به هر یک از دو کروموزوم حاصل یک نوزاد گفته می‌شود. عملگر جهش، جهش طبیعی را شبیه‌سازی می‌کند و باعث ورود اطلاعات جدید به جامعه می‌شود؛ به این ترتیب که به صورت تصادفی مقدار یک یا چند ژن از یک کروموزوم را عوض می‌کند. کروموزوم مورد نظر اغلب حاصل فرایند تقاطع است.

در این مقاله با توجه به نوع مساله عملگرهای تقاطع و جهش خاصی پیشنهاد می‌کنیم که به نظر می‌رسد در کوتاه شدن زمان رسیدن به جواب راضی‌کننده موثر باشند. عملگر تقاطع بر دو کروموزوم اثر کرده و حاصل یک کروموزوم نوزاد است؛ به این ترتیب که کلیه ژن‌های با محتوای برابر در والدین عیناً به نوزاد منتقل می‌شود. سپس برای اینکه تعداد ژن‌های با محتوای یک، در کروموزوم نوزاد برابر با  $k$  (تعداد پایانه‌های مورد نیاز) شود، تعداد مورد نیاز از ژنهای نوزاد را به طور تصادفی، یک قرار داده و در بقیه مکانها نیز صفر قرار می‌دهیم. به طور مثال اگر الگوریتم تقاطع مورد نظر بر دو کروموزوم  $y1$  و  $y2$  اعمال شود، کروموزوم نوزاد می‌تواند به صورت  $y$  باشد.

افزایش سرعت همگرایی الگوریتم، از روش‌های ابتکاری مانند الگوریتم JMOGIS یاسکوویچ استفاده می‌شود [۲۰].

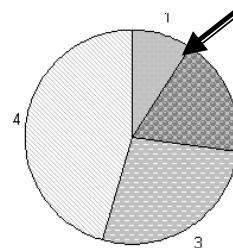
#### ۶-۳-۳. عملگر انتخاب

مطابق نظریه تکاملی داروین، جمعیت، بیشتر به داشتن موجوداتی تمایل دارد که با تغییر کروموزوم‌های جمعیت، ساختار و رفتار آن تکامل یابد.

در الگوریتم ژنتیک نیز وظیفه‌ی اصلی عملگر انتخاب، هدایت الگوریتم به نواحی امیدبخش فضای جواب است. چون در الگوریتم ژنتیک اندازه جمعیت ثابت است، عملگر انتخاب با شناسایی جواب‌های خوب، ساختن کپی از آن‌ها و حذف جواب‌های بد، علاوه بر حفظ اندازه‌ی جمعیت، باعث زیاد شدن جواب‌های خوب در جمعیت نیز می‌شود.

عملگر انتخاب برای ایجاد نسل جدید، یا از همه‌ی نوزادان و والدین استفاده می‌کند (اصطلاحاً گفته می‌شود فضای نمونه گیری توسعه یافته است) یا هر نوزاد بلافاصله جایگزین والد یا والدین خود می‌شود (اصطلاحاً گفته می‌شود فضای نمونه گیری عادی است). همچنین روش‌های مختلفی برای انتخاب بهترین کروموزوم‌ها از یک جمعیت وجود دارد مانند چرخ رولت، انتخاب مسابقه‌ای و انتخاب بر اساس رتبه بندی [۲۰]. در این مقاله از روش چرخ رولت در فضای نمونه گیری پیوسته برای انتخاب نسل جدید بهره گرفته‌ایم.

در این روش که توسط هالند پیشنهاد شد؛ سطح چرخ به بخش‌هایی تقسیم می‌شود که تعداد آن‌ها برابر تعداد اعضای جامعه‌ی جاری (شامل والدین و نوزادان) است. سطح هر بخش متناسب با مقدار برآزندگی جواب متناظر است. چرخ به گردش درمی‌آید و پس از توقف، جواب مقابل پیکان انتخاب می‌شود. این فرایند  $N$  بار انجام می‌شود تا جمعیت جدید انتخاب شود.



- 1 ■ برآزندگی کروموزوم ۱
- 2 ■ برآزندگی کروموزوم ۲
- 3 ■ برآزندگی کروموزوم ۳
- 4 ■ برآزندگی کروموزوم ۴

شکل ۱. چرخ رولت

احتمال انتخاب کروموزوم  $k$ -ام برابر است با:

$$p_k = \frac{f_k}{\sum_{i=1}^N f_i} \quad (8)$$

سعی و خطا برای محاسبه ی مقادیر مناسب (و نه الزاما بهینه)  $pc$  و  $pm$  بهره می گیریم و اندازه ی جمعیت را به پیشنهاد مرجع [۲۰] برابر ۱۰ می گیریم.

قسمتی از شبکه ی اتوبوس رانی شهر مشهد را که در بخش ۱-۷ مورد بررسی قرار خواهد گرفت؛ در نظر گرفته و الگوریتم ژنتیک را به ازای  $pc$  و  $pm$  های مختلف در بازه های  $pc \in [0.8, 0.99]$  و  $pm \in [0.1, 0.2]$  اجرا نموده و زوج پارامتری که منجر به جواب بهتر و زمان اجرای کمتر شود را به عنوان پارامترهای مناسب الگوریتم می پذیریم و در مطالعه های موردی به کار می گیریم.

### الف) تعریف زیر مساله نمونه

یک شبکه اتوبوس رانی با ۶۰ گره و ۲۰ گره کاندید احداث پایانه به صورت زیر در نظر بگیرید. با توجه به محدودیت های مالی و سایر ملاحظات مهم، شهرداری در نظر دارد تعداد ۵ پایانه ی اتوبوس رانی در این شبکه ایجاد نماید. اگر هر پایانه، توان خدمت دهی به گره های اطراف خود تا شعاع ۵ واحد را داشته باشد، مقادیر بهینه ی پارامترهای الگوریتم ژنتیک را بیابید. مقادیر  $x$  و  $y$  که نشان دهنده طول و عرض گره های شبکه در دستگاه مختصات می باشند، بر حسب ۱۰۰ متر و مقادیر  $d$  بر حسب نفر در ساعت داده شده است.

$$J = \{1, 2, \dots, 60\}$$

$$I = \{2, 3, 5, 11, 15, 17, 18, 20, 21, 23, 27, 28, 35, 43, 44, 49, 50, 52, 56, 59\}$$

$$k = 5$$

$$r = 5$$

مختصات گره ها در بردارهای  $x$  و  $y$  به صورت زیر داده شده است:

$$X = \{58/92, 20/86, 26/62, 31/77, 9/38, 57/68, 44/09, 29/80, 18/97, 47/42, 23/99, 5/51, 49/95, 27/93, 5/14, 42/93, 58/77, 42/04, 31/00, 45/28, 56/58, 32/58, 51/68, 15/74, 20/24, 56/83, 42/12, 59/17, 24/29, 23/13, 31/61, 23/61, 1/81, 42/86, 26/80, 50/11, 8/10, 54/60, 37/39, 0/21, 58/71, 0/86, 24/98, 29/29, 46/92, 7/58, 39/77, 20/77, 4/07, 20/09, 57/13, 46/76, 38/95, 8/87, 53/92, 49/14, 57/34, 0/31, 18/52, 4/86\}$$

$$Y = \{24/74, 40/09, 50/39, 15/43, 33/47, 57/08, 13/35, 0/42, 44/30, 24/02, 15/10, 36/13, 17/61, 2/73, 30/79, 2/13, 29/85, 48/19, 30/16, 14/29, 5/32, 54/21, 37/89, 22/74, 18/88, 32/11, 52/50, 53/12, 37/63, 50/87, 48/44, 57/70, 57/22, 38/79, 10/48, 58/21, 15/06, 40/59, 30/73, 13/61, 51/68, 29/15, 46/37, 31/35, 35/47, 6/58, 59/83, 10/56, 18/56, 22/57, 43/16, 37/06, 45/38, 35/97, 10/32, 4/16, 19/04, 45/60, 42/92, 50/75\}$$

مقادیر پتانسیل گره ها نیز عبارتند از:

$$D = \{24, 44, 41, 32, 40, 33, 17, 14, 17, 26, 36, 15, 41, 28, 18, 35, 27, 22, 34, 31, 20, 14, 40, 23, 13, 30, 15, 14, 11, 23, 40, 13, 41, 21, 12, 12, 33, 30, 51, 36, 28, 40, 45, 23, 58, 51, 17, 38, 41, 21, 33, 45, 40, 29, 8, 55, 23, 37, 15, 19\}$$

توجه شود که همه ی مجموعه های بالا مرتب هستند.

شکل ۲ را می توان برای مساله متصور شد. در این شکل محور  $x$  و  $y$  مختصات محل نقاط گره را نشان می دهند. اعداد داخل نمودار نیز بیانگر شماره ی گره می باشند. سرانجام، همسایگی نقاط کاندید با دواپری به مرکز این نقاط و شعاع ۵ رسم شده است.

$$y1 \begin{bmatrix} 1 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

$$y2 \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix}$$

$$y \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 & 1 \end{bmatrix}$$

عملگر جهش تعریف شده نیز به این صورت عمل می کند که ابتدا یک ژن با محتوای یک، به طور تصادفی انتخاب شده و مقدار صفر می گیرد. سپس ژنی با محتوای صفر، که بیشترین مقدار  $\sum_{j \in J_i^*} d_j$  را داراست، انتخاب شده و محتوای آن را به یک تبدیل می کنیم.

$$\sum_{j \in J_k^*} d_j = \max_{\{i: y_i=0\}} \left\{ \sum_{j \in J_i^*} d_j \right\} \Rightarrow y_k = 1$$

با اعمال این جهش انتظار می رود میزان برازندگی نوزاد از والد خود بهتر باشد.

### ۵-۳-۶. تابع برازندگی

تابعی است که خوب یا بد بودن جواب را تعیین می کند و معمولا بین صفر و یک نرمال می شود. تابع برازندگی را معمولا تابع هدف یا تابعی یکنوا از آن می گیرند. تابع برازندگی در الگوریتم ژنتیک پیشنهادی همان تابع هدف است.

### ۶-۳-۶. شرط توقف الگوریتم

عمده ترین شرایط توقف الگوریتم عبارتند از:

۱. رسیدن به جوابی با برازندگی یک در حالت نرمال. این روش معمولا خیلی زمان بر است.
۲. رسیدن به جوابی که دارای برازندگی بیشتر از مقدار داده شده باشد.
۳. رسیدن به جمعیتی که برازندگی اعضای آن تقریبا یکسان باشد یا واریانس برازندگی جوابها از مقدار مشخصی کمتر شود.
۴. تولید تعداد معلوم نسل

در این مقاله، با تولید تعداد ۲۰ نسل، الگوریتم پایان می یابد.

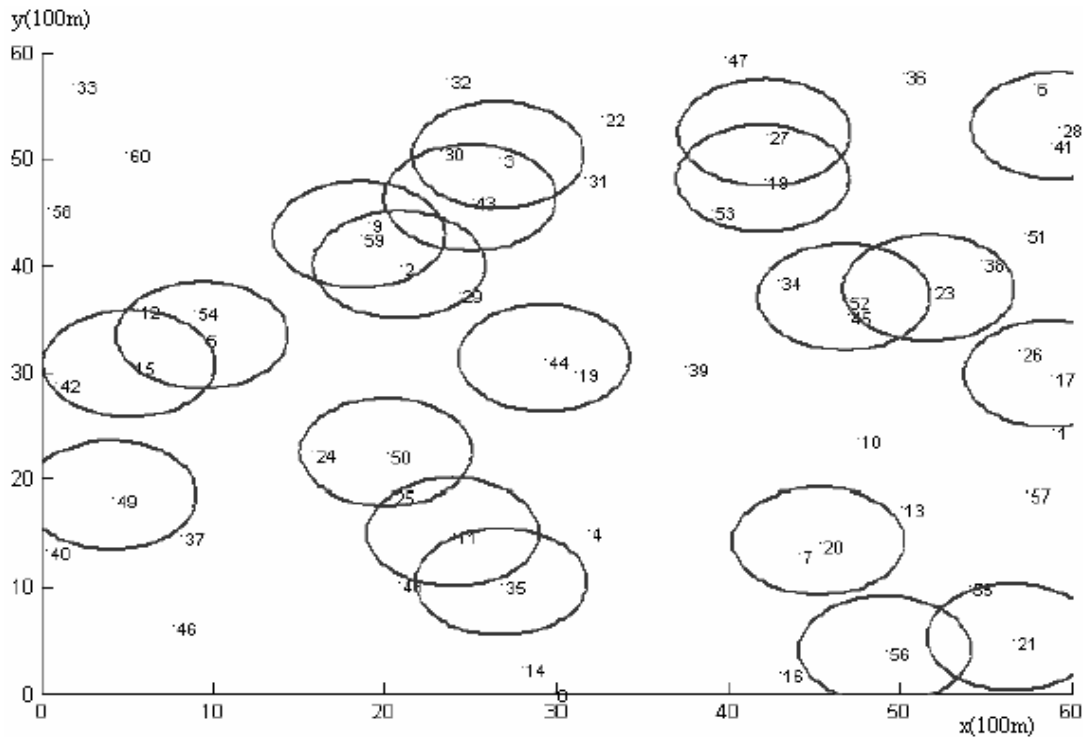
### ۷-۳-۶. محاسبه ی پارامترهای بهینه ی الگوریتم

تعیین پارامترهای  $pm$  و  $pc$  و اندازه ی جمعیت ( $N$ ) تاثیر مستقیم بر عملکرد الگوریتم ژنتیک می گذارد. مثلا اگر  $pm$  زیاد باشد؛ به خاطر جهش های فراوان، احتمال ورود اطلاعات جدید افزایش می یابد ولی در عوض باعث منقرض شدن مواد ژنتیکی می شود. افزایش جمعیت نیز باعث افزایش گوناگونی می شود اما سرعت الگوریتم را پایین می آورد. از آن جا که یافتن پارامترهای بهینه برای هر مساله، نیاز به محاسبات فراوان دارد، در این جا از یک روش

## ب) محاسبه پارامترهای مناسب

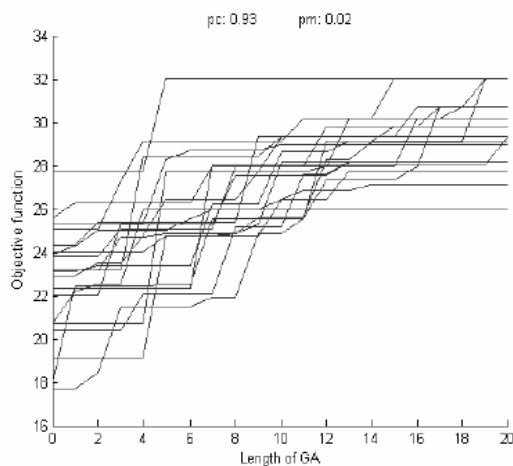
با توجه به مشخص بودن مختصات نقاط گره و کاندید، محاسبه  $C_{ij}$  بسیار آسان است. اندازه جمعیت برابر ۱۰ و شرط توقف الگوریتم،

اجرای ۲۰ تکرار الگوریتم فرض می‌شود. تنها پارامترهای مجهول مساله مقادیر  $pm$  و  $pc$  می‌باشند. در الگوریتم ژنتیک فرض بر آن است که  $pc \in [0, 0.8, 0.99]$  و  $pm \in [0, 0.1, 0.2, 0.4]$



شکل ۲. مختصات گره‌های شبکه، نقاط کاندید و گره‌های همسایه

IVOF (Initial Value Of Objective Function) : بهترین مقدار تابع هدف در جمعیت اولیه (Final Value Of Objective Function) : مقدار بهینه تابع هدف  
Times (s) : زمان اجرای الگوریتم بر حسب ثانیه  
Avg(Average) : میانگین مقادیر هر ستون



شکل ۳. مقادیر تابع هدف بر حسب طول الگوریتم در ۲۰ بار اجرای الگوریتم ژنتیک با پارامترهای بهینه

در این قسمت برای محاسبه‌ی مقادیر بهینه  $pm$  و  $pc$  از روش آزمایش و خطا بهره گرفته‌ایم. برای این منظور مقادیر مختلف این دو پارامتر را در بازه‌های گفته شده، به عنوان پارامترهای الگوریتم ژنتیک در نظر گرفته و الگوریتم به ازای این مقادیر، بر مساله پیاده شده است. برای آن‌که بتوان با تقریب خوبی به مقادیر  $pm$  و  $pc$  دست آمده اطمینان کرد؛ به ازای هر جفت  $pm$  و  $pc$  مورد آزمون، ۲۰ بار برنامه را اجرا کرده و میانگین جواب‌ها را به عنوان معیار ارزیابی پارامترها در نظر گرفته‌ایم.

با توجه به نتایج حاصل، مقادیر بهینه‌ی پارامترهای الگوریتم ژنتیک عبارتند از:  $pc=0.93$  و  $pm=0.02$ . در خروجی هر اجرا، یک جدول و یک نمودار مشاهده می‌شود. اما از آنجا که نمایش تمام جداول و نمودارها در این مختصر نمی‌گنجد، به نمودار و جدول حاصل از پارامترهای بهینه بسنده می‌کنیم.

شکل ۳، نحوه‌ی تغییرات تابع هدف را بر حسب تعداد تکرارهای الگوریتم به ازای پارامترهای بهینه در ۲۰ اجرای مستقل نشان می‌دهد. همان‌گونه که مشاهده می‌شود مقادیر تابع هدف بعد از ۲۰ تکرار به ثبات تقریبی می‌رسند. بنابر این انتخاب عدد ۲۰ برای تعداد تکرارهای الگوریتم در شرط توقف نیز دور از واقعیت نیست. پارامترهای موجود در جدول ۱ نیز عبارتند از:



جدول ۱. مقادیر تابع هدف به ازای پارامترهای بهینه در ۲۰ بار

#### اجرای الگوریتم ژنتیک

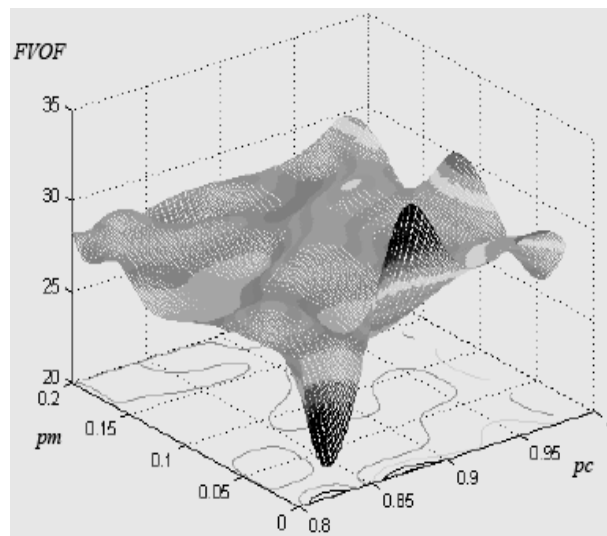
| IVOF  | FVOF  | Time(s) |
|-------|-------|---------|
| ۲۳،۱۶ | ۲۹،۱۰ | ۰،۰۷    |
| ۲۷،۷۹ | ۳۲،۰۲ | ۰،۰۷    |
| ۲۴،۰۰ | ۲۸،۱۸ | ۰،۰۶    |
| ۲۲،۰۴ | ۲۹،۰۰ | ۰،۰۸    |
| ۲۰،۷۸ | ۲۷،۱۱ | ۰،۱۰    |
| ۲۰،۳۸ | ۲۸،۱۸ | ۰،۰۷    |
| ۲۲،۹۳ | ۲۹،۳۲ | ۰،۰۷    |
| ۲۵،۲۷ | ۲۸،۱۸ | ۰،۰۷    |
| ۲۵،۰۸ | ۳۲،۰۲ | ۰،۰۶    |
| ۲۳،۱۹ | ۲۹،۳۲ | ۰،۰۶    |
| ۲۳،۹۰ | ۲۵،۹۸ | ۰،۰۷    |
| ۲۴،۳۵ | ۳۲،۰۲ | ۰،۰۷    |
| ۲۰،۶۹ | ۳۰،۱۸ | ۰،۰۷    |
| ۱۹،۱۴ | ۳۷،۷۶ | ۰،۰۷    |
| ۲۳،۱۲ | ۳۲،۰۲ | ۰،۰۷    |
| ۲۲،۳۶ | ۲۹،۳۲ | ۰،۰۷    |
| ۱۸،۰۲ | ۳۰،۶۹ | ۰،۰۷    |
| ۲۵،۵۸ | ۲۹،۸۱ | ۰،۰۷    |
| ۱۷،۷۱ | ۳۰،۶۹ | ۰،۰۷    |
| ۲۳،۸۲ | ۳۰،۶۹ | ۰،۰۷    |
| Avg:  | ۲۹،۵۸ | ۰،۰۷    |

#### ۷. حل چند مثال واقعی

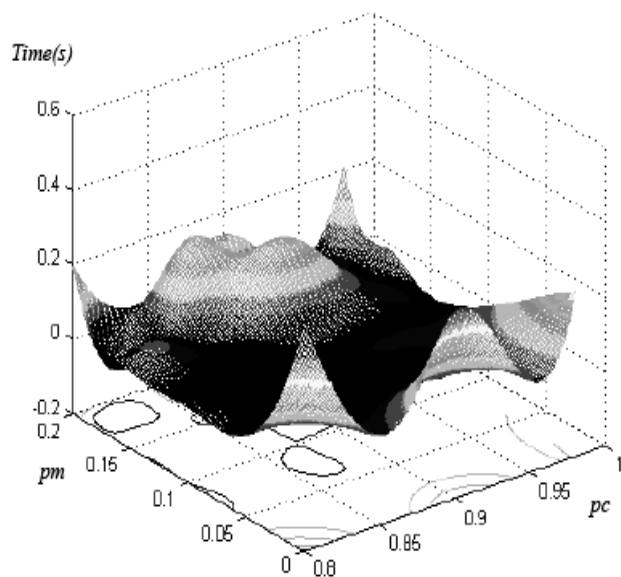
##### ۷-۱. شبکه‌ی اتوبوس‌رانی شهر مشهد

شبکه‌ی اتوبوس‌رانی شهر مشهد که در مرجع [۷] بررسی شده است؛ دارای ۲۷۴ گره (ایستگاه) است. مدل مکان‌یابی پایانه‌ها برای این شبکه در حالتی که تعداد پایانه‌ها برابر ۲۰ باشد؛ دارای ۲۰ متغیر صحیح، حدود ۳۰۰ متغیر پیوسته و در حدود ۶۱۰ محدودیت است. این مدل توسط نرم افزار GAMS حل شده و نتایج آن، در مرجع [۷] ارائه شده است. در مرجع [۹] نیز این مساله به کمک روش‌های شمارش ضمنی و گرم و سرد کردن شبیه‌سازی شده با ۳۵ تکرار در هر دما حل شده است. برای استفاده از الگوریتم ژنتیک، برنامه‌ای در محیط نرم‌افزاری MATLAB 7 نوشته‌ایم. (هر چهار برنامه‌ی SA، GA، IE و GAMS توسط 66 DX2 486 MHz اجرا شده است). نتایج حاصل از حل مدل مکان‌یابی پایانه‌ها برای شهر مشهد با ۲۰ گره کاندید توسط روش‌های SA، GA، IE و GAMS در جدول ۲ خلاصه شده است. به دلیل کوچک بودن ابعاد مساله، در هر ۴ روش، جواب بهینه حاصل شده است. از این رو، زمان اجرای برنامه‌ها را می‌توان به عنوان مبنای مقایسه‌ی روش‌ها

شکل ۴ نحوه‌ی تغییرات میانگین مقادیر تابع هدف و شکل ۵ نحوه‌ی تغییرات متوسط زمان اجرای الگوریتم را بر حسب مقادیر مختلف  $pc$  و  $pm$  نشان می‌دهند. از روی این دو شکل می‌توان مقادیر بهینه‌ی پارامترهای الگوریتم را به دست آورد.



شکل ۴. میانگین مقادیر تابع هدف بر حسب  $pc$  و  $pm$



شکل ۵. میانگین زمان اجرای الگوریتم بر حسب  $pc$  و  $pm$

در این بخش از مقاله، عملکرد روش‌های ژنتیک (GA)، گرم و سرد کردن شبیه‌سازی شده (SA)، شمارش ضمنی (IE) و شاخه و کران (به کمک نرم افزار GAMS) در حل مساله‌ی مکان‌یابی پایانه‌ها مقایسه می‌شود. برای این منظور، دو شبکه‌ی اتوبوس‌رانی واقعی و در ابعاد بزرگ که مربوط به شهرهای مشهد و تهران می‌شوند؛ بررسی می‌شوند. مبنای مقایسه‌ی کارایی روش‌ها، مقادیر تابع هدف و زمان اجرای برنامه‌هاست.

برای حل مساله حتی با ۳۸ گره کاندید بیش از چهار شبانه روز شد [۹]. همچنین حل این مساله با ۷۵ گره کاندید (پس از حذف تعدادی از گره‌های کاندید که احتمال انتخاب بسیار کمی داشتند) توسط روش SA با ضریب کاهش دمای ۰.۹۸، و تعداد ۲۰۰ تکرار در هر دما در مرجع [۹] آمده است.

با توجه به بدیهی بودن برتری روش GA بر GAMS در مکان‌یابی پایانه‌های تهران، تنها به مقایسه‌ی روش‌های GA و SA می‌پردازیم. جدول ۳ مقدار تابع هدف و زمان اجرای این دو الگوریتم را برای ۷۵ گره کاندید و حداکثر ۳۵ پایانه‌ی قابل احداث، مقایسه می‌کند.

جدول ۳. مقایسه‌ی زمان اجرای SA و GA برای شهر تهران با ۱۶۵۰ گره، ۷۵ گره کاندید و حداکثر ۳۵ پایانه‌ی مورد نیاز

| روش GA            |          | روش SA            |          | تعداد پایانه |
|-------------------|----------|-------------------|----------|--------------|
| زمان اجرا (ثانیه) | تابع هدف | زمان اجرا (ثانیه) | تابع هدف |              |
| ۱۵                | ۷۹۰۱۵    | ۶۰                | ۷۸۹۵۴    | ۱            |
| ۲۰                | ۱۴۰۰۰۱   | ۱۸۰               | ۱۳۷۲۱۹   | ۲            |
| ۲۲                | ۱۷۸۲۶۱   | ۱۸۰               | ۱۷۷۱۵۲   | ۳            |
| ۲۲                | ۲۴۳۶۰۲   | ۱۸۰               | ۲۱۵۶۹۱   | ۴            |
| ۲۵                | ۲۵۱۰۱۰   | ۳۰۰               | ۲۴۴۷۸۲   | ۵            |
| ۴۰                | ۲۶۸۲۹۱   | ۳۰۰               | ۲۶۸۲۸۱   | ۶            |
| ۳۹                | ۲۹۰۵۲۲   | ۳۰۰               | ۲۸۹۵۷۴   | ۷            |
| ۳۹                | ۳۰۷۵۵۰   | ۴۲۰               | ۳۰۷۵۵۷   | ۸            |
| ۴۱                | ۳۲۴۸۰۲   | ۴۲۰               | ۳۲۴۹۵۴   | ۹            |
| ۵۰                | ۳۴۲۴۲۸   | ۴۸۰               | ۳۴۲۰۴۲   | ۱۰           |
| ۵۰                | ۳۶۲۰۱۱   | ۵۴۰               | ۳۵۸۹۱۰   | ۱۱           |
| ۵۱                | ۳۷۵۹۰۲   | ۵۴۰               | ۳۷۵۶۱۶   | ۱۲           |
| ۵۰                | ۳۸۸۹۵۵   | ۶۰۰               | ۳۹۱۷۲۶   | ۱۳           |
| ۵۳                | ۴۰۸۶۵۲   | ۶۰۰               | ۴۰۷۰۲۸   | ۱۴           |
| ۵۶                | ۴۳۰۸۱۲   | ۶۶۰               | ۴۲۱۷۹۸   | ۱۵           |
| ۵۷                | ۴۳۸۰۰۰   | ۷۲۰               | ۴۳۳۹۱۱   | ۱۶           |
| ۵۵                | ۴۴۵۱۵۳   | ۷۲۰               | ۴۴۵۱۲۷   | ۱۷           |
| ۵۵                | ۴۵۶۹۰۰   | ۷۸۰               | ۴۵۵۹۴۸   | ۱۸           |
| ۵۶                | ۴۷۱۳۱۷   | ۷۸۰               | ۴۶۶۳۶۹   | ۱۹           |
| ۶۰                | ۴۷۴۹۱۱   | ۲۵۸۰              | ۴۷۵۳۳۰   | ۲۰           |
| ۶۱                | ۴۹۱۰۸۵   | ۲۶۴۰              | ۴۸۴۱۵۷   | ۲۱           |
| ۶۱                | ۴۹۳۲۷۶   | ۲۷۶۰              | ۴۹۳۳۵۴   | ۲۲           |
| ۵۹                | ۵۰۵۲۱۳   | ۲۸۸۰              | ۵۰۰۳۰۹   | ۲۳           |
| ۵۹                | ۵۱۰۵۱۱   | ۲۹۴۰              | ۵۰۸۱۵۹   | ۲۴           |
| ۶۵                | ۵۱۵۷۴۱   | ۳۰۶۰              | ۵۱۵۵۶۳   | ۲۵           |
| ۶۰                | ۵۲۴۷۰۲   | ۳۱۸۰              | ۵۲۲۸۱۲   | ۲۶           |
| ۶۰                | ۵۳۱۴۲۳   | ۳۳۰۰              | ۵۲۹۳۷۶   | ۲۷           |
| ۶۳                | ۵۳۹۵۱۱   | ۳۴۲۰              | ۵۳۷۴۰۶   | ۲۸           |
| ۶۵                | ۵۴۲۵۷۲   | ۳۴۸۰              | ۵۴۲۵۸۶   | ۲۹           |
| ۷۸                | ۵۵۱۱۲۰   | ۳۶۰۰              | ۵۵۰۰۵۷   | ۳۰           |
| ۸۰                | ۵۵۸۵۹۰   | ۳۷۲۰              | ۵۵۵۵۹۹   | ۳۱           |
| ۹۰                | ۵۶۵۱۸۹   | ۳۷۸۰              | ۵۶۰۹۹۱   | ۳۲           |
| ۹۵                | ۵۷۰۷۶۰   | ۳۹۰۰              | ۵۶۸۶۶۱   | ۳۳           |
| ۹۵                | ۵۷۲۹۶۲   | ۴۰۲۰              | ۵۷۲۹۴۱   | ۳۴           |
| ۹۶                | ۵۸۵۲۳۳   | ۴۰۸۰              | ۵۸۰۴۱۹   | ۳۵           |
| ۱۹۸۳              | ۱۴۷۲۵۹۸۳ | ۶۲۱۰۰             | ۱۴۶۳۹۳۵۹ | جمع          |

در نظر گرفت. اگر جمع زمان اجرای برنامه‌ها برای ۲۰ حالت مبنای مقایسه در نظر گرفته شود؛ مشاهده می‌شود که روش GA حدود ۸ برابر سریعتر از روش SA، حدود ۴۸۵ برابر سریعتر از روش IE و حدود ۱۲۰۰ برابر سریعتر از GAMS است. چنانچه شهرداری قصد احداث ۸ پایانه را داشته باشد این نسبت‌ها به ترتیب برابر ۸.۳۷ و ۹۲۵ و ۵۴۰۵ خواهد بود.

همانطور که در جدول ۲ دیده می‌شود زمان الگوریتم ژنتیک پیشنهادی، برخلاف سایر روش‌ها، حساسیت کمتری به تعداد پایانه‌ها دارد. در حقیقت این زمان، بیش از آن که به تعداد پایانه‌ها حساس باشد؛ به تعداد نقاط کاندید حساس است و این، به علت نوع کروموزم و عملگرهای تقاطع و جهش خاص تعریف شده و نیز شرط پایان الگوریتمی است که تعریف کرده ایم.

جدول ۲. مقایسه‌ی زمان اجرای GAMS، IE، SA و GA برای شبکه‌ی مشهد با ۲۷۴ گره و ۲۰ گره کاندید (ثانیه)

| تعداد پایانه | زمان اجرای GAMS | زمان اجرای IE | زمان اجرای SA | زمان اجرای GA |
|--------------|-----------------|---------------|---------------|---------------|
| ۱            | ۱۲              | ۱             | ۷             | ۲             |
| ۲            | ۱۰۳             | ۱             | ۱۱            | ۲.۱           |
| ۳            | ۵۱۰             | ۳             | ۱۶            | ۲.۱           |
| ۴            | ۱۷۸۷            | ۱۷            | ۱۹            | ۳.۵           |
| ۵            | ۴۶۳۹            | ۹۳            | ۲۲            | ۳.۸           |
| ۶            | ۹۶۶۵            | ۴۵۵           | ۲۵            | ۳.۵           |
| ۷            | ۱۶۳۶۸           | ۱۵۰۶          | ۲۸            | ۳.۶           |
| ۸            | ۲۰۰۰۱           | ۳۴۲۰          | ۳۱            | ۳.۷           |
| ۹            | ۱۹۰۱۲           | ۵۶۴۰          | ۳۴            | ۴             |
| ۱۰           | ۱۲۰۴۵           | ۷۱۴۰          | ۳۶            | ۴             |
| ۱۱           | ۵۶۳۲            | ۷۲۰۰          | ۳۸            | ۴.۱           |
| ۱۲           | ۲۲۶۳            | ۵۷۰۰          | ۴۱            | ۴.۲           |
| ۱۳           | ۱۱۱۰            | ۳۷۸۰          | ۴۳            | ۴.۲           |
| ۱۴           | ۴۷۵             | ۱۹۸۰          | ۴۵            | ۵             |
| ۱۵           | ۱۴۹             | ۸۴۰           | ۴۷            | ۵             |
| ۱۶           | ۶۶              | ۲۸۲           | ۴۹            | ۴.۸           |
| ۱۷           | ۲۹              | ۷۰            | ۵۱            | ۴.۹           |
| ۱۸           | ۱۴              | ۱۳            | ۵۳            | ۵.۲           |
| ۱۹           | ۷               | ۲             | ۵۵            | ۴             |
| ۲۰           | ۲               | ۱             | ۱             | ۳             |
| جمع          | ۹۳۸۸۹           | ۳۸۱۴۴         | ۶۵۲           | ۷۸.۷          |

## ۲-۷. شبکه‌ی اتوبوس‌رانی شهر تهران

دومین شبکه‌ی واقعی بررسی شده مربوط به شهر تهران می‌شود. این شبکه دارای ۱۶۵۰ گره با ۹۹ گره کاندید است. در این حالت مساله دارای ۹۹ متغیر صحیح، حدود ۲۵۰۰ متغیر پیوسته و در حدود ۴۳۰۰ محدودیت است.

حل این مساله توسط GAMS به عنوان یک مساله‌ی واحد (بدون تقسیم بندی شبکه) عملاً ناممکن است. به عنوان مثال، زمان لازم

## ۸. نتیجه گیری

مساله‌ی مکان‌یابی پایانه‌های شبکه اتوبوس‌رانی، حالت خاصی از مساله‌ی مکان‌یابی تسهیلات است که علاوه بر طراحی شبکه‌های اتوبوس‌رانی، در مسایل دیگری مانند احداث تعداد معینی مدرسه‌ی شبانه روزی در بین تعداد زیادی روستای کوچک و پراکنده، مکان‌یابی تعداد مشخص حوزه‌های امتحانی آزمون کارشناسی ارشد در سطح کشور و زمینه‌های مشابه آن نیز کاربرد دارد.

در کوشش‌های پیشین برای حل این مساله، روش‌های شاخه و کران، شمارش ضمنی و گرم و سرد کردن شبیه‌سازی شده به کار گرفته شده است. در این مقاله، برای حل مساله، یک الگوریتم ژنتیک پیشنهاد کردیم. به این منظور، عملگرهای تقاطع، جهش و انتخاب مناسب معرفی شد. برای به دست آوردن پارامترهای بهینه‌ی الگوریتم یعنی نرخ تقاطع و نرخ جهش از یک مساله‌ی تجربی بهره گرفتیم.

نتایج حاصل از حل مساله برای شهر مشهد، کارایی فوق‌العاده‌ی این روش را در مقایسه با سه روش دیگر نشان داد. هرچند هر چهار روش در مورد شبکه مشهد به جواب بهینه رسیدند؛ اما زمان حل مساله توسط الگوریتم ژنتیک، به ترتیب حدود ۸، ۴۸۵ و ۱۲۰۰ برابر سریعتر از روش‌های گرم و سرد کردن شبیه‌سازی شده، شمارش ضمنی و شاخه و کران شد. پس از آن که کارایی الگوریتم پیشنهادی در حل مساله، مشخص شد؛ مساله‌ی مکان‌یابی شهر تهران نیز توسط این روش حل شد. ابعاد شبکه‌ی شهر تهران حدود ۵ برابر شهر مشهد است و حل مساله‌ی مذکور با روش شاخه و کران (توسط نرم افزار GAMS)، مستلزم تقسیم کردن شهر به چند زیر ناحیه و حل مجزای آن‌ها است. در غیر این صورت به دلیل بالا رفتن زمان اجرا، حل مساله توسط GAMS تقریباً غیر عملی است. این مساله بدون نیاز به تقسیم بندی، با استفاده از الگوریتم ژنتیک پیشنهادی حل و زمان اجرای آن ۳۰ برابر کمتر از رقیب اصلی اش، SA شد. میانگین دقت جواب نیز بیشتر از روش SA است.

## مراجع

- [۱] ابوالقاسم، آرزو، « مکان‌یابی پایانه‌های درون‌شهری برای شبکه‌ی اتوبوس‌رانی»، پایان نامه کارشناسی ارشد، دانشکده مهندسی عمران، دانشگاه صنعتی شریف، شهریور ۱۳۷۸.
- [۲] حجازی، بهرنگ، « حل مساله‌ی مکان‌یابی پایانه‌های شبکه‌ی اتوبوس‌رانی با استفاده از روش گرم و سرد کردن شبیه‌سازی شده»، پایان نامه کارشناسی ارشد، دانشکده مهندسی عمران، دانشگاه صنعتی شریف، شهریور ۱۳۷۸.
- [۳] ذکایی آشتیانی، هدایت، بهرنگ، « کاربرد روش گرم و سرد کردن شبیه‌سازی شده در حل مساله‌ی مکان‌یابی پایانه‌های شبکه‌ی اتوبوس‌رانی»، استقلال، جلد ۲۰، شماره ۲، صفحه ۱۴۰-۱۲۵، ۱۳۸۰.
- [۴] قنبری، رضا، « کاربرد روش جستجوی محلی یاسکوویچ در حل مسایل بهینه‌سازی ترکیبیاتی چند هدفه»، پایان نامه کارشناسی ارشد، دانشکده علوم ریاضی، دانشگاه صنعتی شریف، آبان ۱۳۸۳.
- [5] Francis, R.L., White, J.A., *Facility Layout and Location: an Analytical Approach*, Prentice-Hall, 1974.
- [6] Lapierre, S.D., Ruiz, A.B., Soriano, P., "Designing Distribution Networks: Formulations and Solution Heuristic", *TRANSPORTATION SCIENCE*, Vol. 38, No. 2, May 2004, pp. 174-187.
- [7] Bander, J.L., White, C.C., "A Heuristic Search Approach for a Nonstationary Stochastic Shortest Path Problem with Terminal Cost", *TRANSPORTATION SCIENCE*, Vol. 36, No. 2, May 2002, pp. 218-230.
- [8] Bouzaïene-Ayari, B., Gendreau, M., Nguyen, S., "Modeling Bus Stops in Transit Networks: A Survey and New Formulations", *TRANSPORTATION SCIENCE*, Vol. 35, No. 3, August 2001, pp. 304-321.
- [9] Ginkel, A., Schöbel, A., "To Wait or Not to Wait? The Bicriteria Delay Management Problem in Public Transportation", *TRANSPORTATION SCIENCE*, Vol. 41, No. 4, November 2007, pp. 527-538.
- [10] Abbink, E., Van den Berg, B., Kroon, L., Salomon, M., "Allocation of Railway Rolling Stock for Passenger Trains", *TRANSPORTATION SCIENCE*, Vol. 38, No. 1, February 2004, pp. 33-41.
- [11] Van Oudheusden, D.L., Ranjithan, S., Singh, K.N., "The design of bus route systems; An interactive location-allocation approach", *TRANSPORTATION SCIENCE*, Vol. 14, No. 3, 1987, pp.253-270.
- [12] Drezner, T., Drezner, Z., Scott, C.H., "Location of a Facility Minimizing Nuisance to or From a Planar Network", *Computers & Operations Research*, Vol. 36, 2009, pp.135 – 148
- [13] Drezner, T., Drezner, Z., "Equity Models in Planar Location", *CMS*, Vol. 4, 2007, pp.1-16.
- [14] Gonzales, F.T., *HandBook of Approximation Algorithms and Metaheuristics*, Champan and Hall, 1 Ed, 2007.
- [15] Pham, D.P., Karaboga, D., *Intelligent Optimization Techniques: Genetic Algorithm, Tabu Search, Simulated Annealing, Neural Network*, Springer, 1Ed, 2000.
- [16] Rego, C., Alidaee, B., *Metaheuristic Optimization via Memory and Evolution: Tabu Search and Scatter Search*, Springer, 1 Ed, 2005.
- [17] Haupt, R.L., Haupt, S.E., *Practical Genetic Algorithm*, John Wiley, 2 Ed, 2004.
- [18] Eiben, A.E., Smith, J.E., *Introduction to Evolutionary Computing*, Springer, 2007.
- [19] Holland J.H., *Adaptation in Natural and Artificial Systems*, The University of Michigan Press, 1975.

- [20] Goldberg, D.E., "Real-Coded Genetic Algorithms, Virtual Alphabets and Blocking" , Complex Systems, Vol.5, No.2, 1991, pp. 139-168.
- [21] Goldberg, D.E., *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison-Wesley, 1989.
- [22] Goldberg, D.E., Lingle, R., "Alleles, Loci, and the Travelling Salesman Problem", Proceedings of 1st International Conference on Genetic Algorithms and their Applications, 1985, pp. 154-159.