

# یک الگوریتم ژنتیکی-عصبی جدید برای حل مسأله فروشنده دوره گرد

مهدی سعادت‌مند طرزجان

دانشکده برق، دانشگاه صنعتی خواجه نصیر الدین طوسی

[811301003@ee.kntu.ac.ir](mailto:811301003@ee.kntu.ac.ir)

محمد اکبرزاده توتونچی

دانشکده مهندسی، دانشگاه فردوسی مشهد

[akbazar@eece.unm.edu](mailto:akbazar@eece.unm.edu)

## ۱- مقدمه

مسائل بهینه‌سازی ترکیبی، از قبیل مسأله فروشنده دوره گرد (TSP)، از خانواده مسائل NP-Complete هستند [۱]. مدت زمان لازم برای حل این نوع مسائل با افزایش تعداد پارامترها بصورت نمایی افزایش می‌یابد و این درحالی است که با پیشرفت تکنولوژی، نیاز به حل سریعتر و مناسبتر این مسائل بسرعت رو به افزایش است. تعیین مسیر بهینه حرکت مته برای سوراخ کردن صفحه‌های PCB، تعیین مسیر بهینه انتقال داده در شبکه‌های کامپیوتری و پردازش تصویر و تشخیص الگو از جمله زمینه‌هایی هستند که حل TSP برایشان بسیار راه گشاست. TSP، همانطور که از نامش بر می‌آید، عبارت است از یافتن کوتاهترین مسیر بسته ممکن بین  $N$  شهر. روشهایی که تاکنون برای حل این مسأله ارائه شده‌اند، معمولاً بر پایه جستجوگرها و آشکارسازهای ایستا یا احتمالاتی هستند که از آنجمله می‌توان به الگوریتمهای کلاسیک جستجوی محلی، بازپخت تطبیقی (SA) [۲]، شبکه‌های عصبی مصنوعی، الگوریتمهای ژنتیکی (GA) [۳]، و سیستم کولونی مورچه‌ها (ACO) [۴]، اشاره کرد. هر یک از این روشها دارای نقاط ضعف و قوت خاص خود است. به عنوان مثال، از روشهای تکاملی مانند GA معمولاً به دلیل حجم زیاد محاسبات بیشتر در کاربردهای خارج خط استفاده می‌شود ولی درعوض توانایی بالایی در بهینه‌سازی مسائل غیرخطی و پیچیده از خود نشان داده‌اند. همچنین با اینکه SA و ACO نسبت به الگوریتمهای

**چکیده:** در این مقاله یک الگوریتم ژنتیکی جدید برای بهینه‌سازی پاسخ‌های شبکه عصبی CNN-TSP ارائه شده است. با اینکه CNN-TSP پاسخی بسیار مناسب و قابل قبول به TSP ارائه می‌دهد، اما در مینیمم‌های محلی تابع انرژی‌اش بدام می‌افتد. الگوریتم ژنتیکی پیشنهادی با استفاده از ساختار سازنده شبکه و جایگزینی یک تصمیم‌گیرنده تکاملی بجای تصمیم‌گیرنده رقابتی آن توانسته است پاسخ‌های شبکه را بنحو چشمگیری بهبود بخشد. تا آنجا که طبق شبیه‌سازی‌های انجام شده، الگوریتم ژنتیکی پیشنهادی توانسته است از میان ۲۲ مسأله‌ای که پاسخ بهینه آنها از کتابخانه TSPLIB استخراج شده بود، در ۷ مورد پاسخی بهتری از پاسخ‌های کتابخانه، در ۹ مورد همان پاسخ‌ها و در ۶ مورد پاسخی بدتر از پاسخ کتابخانه ارائه دهد. همچنین، پاسخ‌های الگوریتم ژنتیکی پیشنهادی برای ۷ مسأله استاندارد از کتابخانه TSPLIB با چند روش متداول دیگر مقایسه گردیده است. شبیه‌سازی‌ها بیانگر آن است که دقت روش پیشنهادی در مقایسه با دیگر روش‌ها حداقل ۱/۵٪ بیشتر است. خصوصیات چون دقت بالا و سرعت همگرایی مناسب الگوریتم ژنتیکی پیشنهادی را به ابزاری مناسب برای کاربردهای خارج خط (off-line) مبدل کرده است.

**واژه‌های کلیدی:** مسأله فروشنده دوره گرد، شبکه عصبی

CNN-TSP، الگوریتم‌های ژنتیکی

تکاملی سریعتر هستند اما معمولاً از دقت کمتری برخوردارند [۴]. ما قبلاً در [۱] و [۵] شبکه جدیدی تحت عنوان CNN-TSP، معرفی و ساختار و خواص آن را بررسی نموده‌ایم. علیرغم اینکه شبکه پیشنهادی در مقایسه با سایر روش‌ها، از دقت و سرعت مناسبی برخوردار است، اما هنوز هم در مینیمم‌های محلی تابع انرژی خود بدام می‌افتد [۱]. در این مقاله الگوریتم ژنتیکی جدیدی جهت بهینه‌سازی پاسخ‌های شبکه عصبی مذکور ارائه گردیده است. در شبیه‌سازی‌ها، به منظور ارزیابی عملکرد الگوریتم تکاملی پیشنهادی، از تعدادی از مسائل کتابخانه استاندارد TSPLIB استفاده شده است. از مقایسه پاسخ‌های الگوریتم پیشنهادی با پاسخ‌های بهینه گزارش شده در کتابخانه مذکور و پاسخ‌های چند الگوریتم متداول دیگر شامل SA [۶]، CNN-TSP، SOM [۶]، Budinich's (BSOM) [۶]، ESOM [۷] و eISOM [۸] می‌توان نتیجه گرفت که روش پیشنهادی از دقت بسیار مناسبی برخوردار است. در ادامه مقاله، بخش بعدی به معرفی شبکه عصبی CNN-TSP و الگوریتم آموزش آن اختصاص یافته است. در بخش سوم الگوریتم‌های ژنتیکی به صورت اجمالی شرح داده شده‌اند. بخش چهارم به شرح تصمیم‌گیرنده تکاملی الگوریتم پیشنهادی می‌پردازد. در بخش پنجم الگوریتم ژنتیکی پیشنهادی معرفی شده است. و در نهایت بخش‌های ششم و هفتم به ترتیب شبیه‌سازی‌ها و نتیجه‌گیری اختصاص یافته‌اند.

## ۲- شبکه عصبی CNN-TSP

شبکه عصبی پیشنهادی دارای دو بخش و هر بخش شامل چهار لایه می‌باشد. خروجی نرون‌های هر لایه به شکل بردار است. شکل (۱) این شبکه را برای یک مسیر سه شهری از بین شش شهر نشان می‌دهد. در شکل، ورودی‌ها، خروجی‌ها و مقادیر آستانه نرون‌ها نیز نشان داده شده‌اند.

در شبکه CNN-TSP، نرون‌های لایه اول (Path layer) بیانگر مسیر ارائه شده توسط شبکه در هر مرحله هستند. شبکه به شهرها اجازه می‌دهد که در بین نرون‌های این لایه جابجا شوند. در لایه دوم (Link layer) هر نرون یک کمان از مسیر را مشخص می‌کند. لایه‌های سوم (Link competitive layer) و چهارم شبکه (Path competitive layer) به دو قسمت تقسیم می‌شوند. نرونهایی که بر روی مسیر قرار گرفته‌اند در بخش

اول (بخش بهینه‌ساز) جای می‌گیرند و بقیه نرون‌ها در بخش دوم (بخش سازنده) قرار داده می‌شوند. وظیفه بخش اول بهینه‌سازی مسیر موجود و وظیفه بخش دوم توسعه مسیر است. در این دو لایه متناظر با هر شهر یک نرون قرار دارد. بعلاوه در لایه سوم یک نرون آستانه نیز وجود دارد (که در شکل ۱ با T نشان داده شده است).

همانطور که گفتیم، CNN-TSP یک شبکه دو قسمتی است. بر همین اساس الگوریتم آموزشی آن نیز الگوریتمی دو فازه است. در فاز اول با اضافه شدن شهرهای جدید به مسیر، شبکه توسعه می‌یابد. این مرحله برعهده بخش سازنده شبکه است. در فاز دوم با جابجایی شهرهای موجود در مسیر، مسیر فعلی بهبود می‌یابد که این وظیفه برعهده فاز بهینه‌ساز شبکه است. فرآیند آموزش چنان طراحی گردیده است که شبکه در هر مرحله، حتی قبل از پایدارشدن، مسیری معتبر ارائه دهد (مسیری که از تمام شهرهای مسیر تنها یکبار عبور کرده باشد). برای مطالعه بیشتر درباره شبکه CNN-TSP و نحوه آموزش آن به [۱] و [۵] مراجعه شود.

## ۳- الگوریتم‌های ژنتیکی

CNN-TSP در مقایسه با دیگر شبکه‌های عصبی چون کوهونن پاسخ‌های قابل قبولی ارائه می‌دهد؛ به طوری که طول پاسخ‌های ارائه شده توسط آن، به طور متوسط (برای مسیرهای ۵۰ شهری)، تقریباً ۲/۵٪ کوتاهتر از پاسخ‌های کوهونن می‌باشد. این در حالی است که زمان همگرایی CNN-TSP بسیار کمتر از آن و در حدود ۱/۲۰ است [۱]. در [۹] نیز پاسخ‌های CNN-TSP با استفاده از یک تصمیم‌گیرنده فازی-ژنتیکی بجای تصمیم‌گیرنده رقابتی شبکه بهبود یافته است. اما با وجود این تمهیدات هنوز هم شبکه در مینیمم‌های محلی خود به دام می‌افتد. این شبکه عصبی برای کاربردهای بلادرنگ بسیار مناسب است، اما چنانچه هدف ما یافتن بهترین پاسخ برای یک مسأله خاص باشد، آنگاه باید برای بهینه‌سازی پاسخ‌های شبکه تمهیداتی اندیشید. آنچه که بر پاسخ‌های شبکه CNN-TSP به طور مستقیم اثر می‌گذارد، این است که در هر مرحله، کدام شهر در مسیر قرار گیرد. بدیهی است که چنانچه این ترتیب به بهترین حالت ممکن باشد آنگاه شبکه قادر به یافتن بهترین پاسخ نیز خواهد بود. ما برای یافتن این آرایش از

الگوریتم‌های ژنتیکی استفاده کرده‌ایم. الگوریتم‌های ژنتیکی خانواده‌ای از مدل‌های محاسباتی هستند که از تکامل موجودات (مسئله انتخاب طبیعی در طبیعت) الهام گرفته شده‌اند [۱۰]. برای مطالعه درباره الگوریتم‌های ژنتیکی به [۱۱] مراجعه شود.

#### ۴- تصمیم‌گیرنده تکاملی

برای بیان چگونگی استفاده از الگوریتم‌های ژنتیکی در بهینه‌سازی پاسخ‌های شبکه CNN-TSP، قبل از هر چیز باید از دیدگاهی نو به شبکه نگریست. همانطور که قبلاً شرح دادیم، نرون‌هایی که بر روی مسیر قرار ندارند (در لایه‌های سوم و چهارم)، بخش سازنده شبکه را تشکیل می‌دهند. در این دو لایه، طی یک فرآیند رقابتی -در هر مرحله- درباره اینکه کدام نرون باید در کجای مسیر قرار بگیرد، تصمیم‌گیری می‌شود. اینکه کدام نرون باید در مسیر قرار گیرد، در لایه چهارم تعیین می‌شود. عاملی که در این انتخاب، نقش اصلی دارد همان افزایش طولی است که نرون با قرار گرفتن در مسیر ایجاد می‌کند. از این دیدگاه می‌توان به لایه چهارم شبکه (در بخش سازنده) به عنوان یک تصمیم‌گیرنده رقابتی نگریست. این موضوع در شکل (۲) نشان داده شده است. بدیهی است که تصمیم‌گیری تنها بر اساس افزایش طول نمی‌تواند به نتایج بهینه‌ای منجر شود و می‌توان به جای تصمیم‌گیرنده رقابتی از دیگر تصمیم‌گیرنده‌ها همچون تصمیم‌گیرنده‌های فازی سود جست [۹]. ایده ما در بهینه‌سازی پاسخ‌های شبکه استفاده از یک تصمیم‌گیرنده تکاملی است. به این ترتیب که هر کروموزوم بیانگر آرایش وارد شدن شهرها به مسیر باشد و در طول فرآیند تکاملی این آرایش بهینه گردد. بلوک دیاگرام الگوریتم ژنتیکی پیشنهادی و نحوه ارتباط بخش‌های مختلف آن با یکدیگر در شکل (۳) نشان داده شده است. فرآیند تصمیم‌گیری به این ترتیب است که در هر مرحله یک شهر از کروموزوم استخراج می‌شود و در مسیر قرار می‌گیرد. محلی که باید شهر در آنجا قرار گیرد، توسط بخش سازنده شبکه (خروجی نرون متناظر با شهر) در لایه سوم، تعیین می‌شود. بنابراین هر کروموزوم عبارت است از رشته‌ای از شهرها،  $\{a_1, a_2, \dots, a_N\}$ . روال تولید مسیر چنین است که ابتدا با استفاده از سه شهر  $a_1, a_2$  و  $a_3$  یک مسیر بسته ایجاد کرده، آنرا به عنوان ورودی به شبکه می‌دهیم. در مرحله  $k$ ام شهر  $a_{k+3}$  را

از کروموزوم استخراج کرده، در محلی که بخش سازنده شبکه معرفی می‌کند قرار می‌دهیم. سپس اجازه می‌دهیم شبکه در فاز بهینه‌سازی‌اش همگرا گردد. به همین ترتیب در مراحل بعد، شهرهای بعدی به ترتیب در مسیر قرار داده می‌شوند، تا آنجا که همه شهرها وارد مسیر گردند. پس از آخرین مرحله، طول مسیر معرفی شده توسط شبکه میزان کیفیت آن کروموزوم را تعیین می‌کند. در حقیقت، CNN-TSP نقش تابع هزینه الگوریتم ژنتیکی را ایفا می‌کند.

#### ۵- الگوریتم ژنتیکی پیشنهادی

همانطور که می‌دانید هر الگوریتم ژنتیکی از بخش‌های رمزنگاری (coding)، انتخاب والدین (selection)، برش و جهش تشکیل شده است. رمزنگاری کروموزومها در بخش قبل شرح داده شد. برای برش، از دو عملگر مشابه با عملگر OX (order crossover) [۱۲] استفاده شده است. عملکرد عملگر اول چنین است که همانند روش برش دو نقطه‌ای [۱۰]، دو نقطه برش بر روی کروموزومهای پدر انتخاب می‌شوند. ناحیه بین این دو نقطه را ناحیه برش می‌نامیم. بدیهی است که چنانچه کروموزومهای فرزند مطابق روش دو نقطه‌ای، تنها با جابجایی نواحی برش بدست آید، آنگاه ممکن است بعضی شهرها از مسیرهای فرزند حذف شوند یا بیش از یکبار تکرار گردند. برای پرهیز از این مشکل راه حل پیشنهادی، در شکل (۴) نشان شده است. در مرحله اول جای دو ناحیه برش را بایکدیگر عوض می‌کنیم، در مرحله دوم از مجموعه ژنهای هر کروموزوم پدر، ژنهای ناحیه برش مربوط به کروموزوم پدر دیگر را حذف می‌کنیم. و در نهایت در مرحله سوم ژنهای باقیمانده از هر پدر را در کروموزوم فرزند متناظرش (فرزند اول متناظر پدر اول و فرزند دوم متناظر پدر دوم است) قرار می‌دهیم بطوری که اولین ژن در اولین جای خالی قرار گیرد. عملگر دوم همانند عملگر اول است با این تفاوت که بجای نواحی برش، از معکوس آنها (همان ژنها ولی با ترتیب معکوس) در تشکیل کروموزومهای فرزند استفاده می‌شود. در حین تکامل، در هر مرحله یک عدد تصادفی بین صفر و یک تولید می‌شود و اگر این عدد کمتر از ۰/۵ بود از عملگر اول و در غیر این صورت از عملگر دوم در تولید نسل استفاده می‌شود. از عملگر جهش به منظور فرار از مینیمم‌های محلی

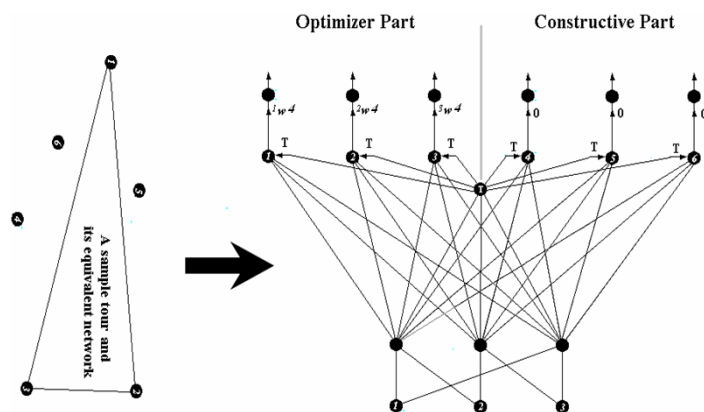
## ۶- شبیه‌سازی

شبیه‌سازی‌های این بخش توسط یک کامپیوتر شخصی AMD Athlon 1600MHz و تحت نرم‌افزار Matlab انجام شده است. برای بررسی عملکرد ENN-TSP، مسأله‌های استاندارد کتابخانه TSPLIB [۱۶] مورد استفاده قرار گرفت. در این کتابخانه ۱۱۱، TSP متقارن آورده شده است که شامل مسیرهایی از ۱۴ شهر تا ۸۵۹۰۰ شهر می‌باشد. همچنین برای بعضی از این مسائل، پاسخ‌های بهینه‌ای نیز معرفی گردیده است. هم‌اکنون محققین از این کتابخانه برای مقایسه الگوریتم‌های مختلف با یکدیگر استفاده می‌کنند [۱۷]. در جدول (۱)، پاسخ ENN-TSP به تعدادی از مسائل TSPLIB بدست آورده و دقت آنها را محاسبه کرده‌ایم. برای هر مسأله، ENN-TSP حداقل دو مرتبه اجرا شده است. در ستون‌های این جدول به ترتیب، نام مسأله، تعداد شهرهای آن، طول پاسخ بهینه‌ای که در کتابخانه TSPLIB گزارش شده، طول بهترین پاسخ ENN-TSP، طول بدترین پاسخ، کمترین تعداد تکرار قبل از همگرایی، بیشترین تعداد تکرار قبل از همگرایی، اندازه جمعیت، زمان محاسباتی متوسط برای یک تولید نسل و در نهایت خطای نسبی بهترین پاسخ نسبت به پاسخ بهینه کتابخانه TSPLIB ذکر گردیده است. از مقایسه ستون‌های چهارم و پنجم می‌توان دریافت که شبکه حداقل وابستگی را به جمعیت اولیه دارد به طوری که در اکثر قریب به اتفاق موارد بهترین و بدترین پاسخ بدست آمده یکسان است. البته با مراجعه به ستون‌های ششم و هفتم می‌بینید که تعداد تولید نسل‌ها برای جمعیت‌های اولیه متفاوت، یکسان نیست، اما الگوریتم ژنتیکی حتی با جمعیت اولیه نامناسب نیز توانسته است همان پاسخی را ارائه دهد که برای یک جمعیت اولیه مناسب در تعداد تکرار کمتر ارائه داده است. با مراجعه به ستون‌های پنجم و ششم در می‌یابیم که تعداد تولید نسل‌های ENN-TSP چندان زیاد نبوده است. با توجه به اینکه در هر تولید نسل تنها دو فرزند جدید تولید می‌شود، بنابراین می‌توان نتیجه گرفت که الگوریتم ژنتیکی پیشنهادی از سرعت قابل قبولی برخوردار است. با اینکه زمان همگرایی CNN-TSP با افزایش تعداد شهرها افزایش می‌یابد، اما می‌توان گفت که در مجموع ENN-TSP برای مسیرهایی با تعداد شهرهای متوسط (در حدود ۲۰۰ شهر) از سرعت همگرایی مناسبی برخوردار است (به ستون پانزدهم مراجعه کنید). در ستون هشتم اندازه

استفاده می‌شود. قبل از اینکه الگوریتم ژنتیکی شروع به کار کند، عددی به عنوان احتمال جهش در نظر گرفته می‌شود. در طول فرآیند تکامل، پس از تولید هر فرزند، برای آن، عددی تصادفی تولید می‌شود. چنانچه این عدد از احتمال جهش کوچکتر باشد، عملگر جهش بر روی فرزند مذکور اعمال می‌گردد. در الگوریتم پیشنهادی از سه عملگر جهش استفاده شده است. عملگر اول، پس از انتخاب دو نقطه برش، ترتیب ژنهای بین این دو نقطه را معکوس می‌نماید. عملگر دوم، پس از انتخاب دو نقطه برش، دو ناحیه خارج این دو نقطه را پس از معکوس کردن با یکدیگر عوض می‌کند و در نهایت عملگر سوم، با انتخاب تعدادی شهر به صورت تصادفی، جای آنها را با یکدیگر عوض می‌کند. برای تعیین اینکه کدامیک از عملگرهای سه گانه را باید برای ایجاد جهش مورد استفاده قرار داد، عددی تصادفی بین صفر و یک تولید می‌شود و چنانچه این عدد کوچکتر از  $0.33$  بود عملگر اول، اگر بین  $0.33$  و  $0.66$  بود عملگر دوم و در غیر این صورت عملگر سوم مورد استفاده قرار می‌گیرد. تکامل بیش از آنکه به حفظ گونه‌هایی پردازد که تطابق بیشتری با محیط اطرافشان داشته‌اند، به حذف گونه‌هایی پرداخته که کمتر با محیط اطرافشان تطبیق داشته‌اند [۱۳]. به عبارت دیگر در تکامل حذف عناصر ضعیفتر به مراتب اثر بیشتری از حفظ عناصر قویتر دارد. در الگوریتم ژنتیکی پیشنهادی مشابه با Genitor [۱۴]، در هر تولید نسل (generation)، از میان جمعیت موجود تنها دو کروموزوم به عنوان ولی (parent) انتخاب می‌شوند و بنابراین تنها دو کروموزوم فرزند تولید می‌گردد. جمعیت جدید با جایگذاری این دو فرزند به جای بدترین کروموزومهای جمعیت قبلی (آنهايي که بدترین درجه شایستگی را دارند) بوجود می‌آید. برای انتخاب والدین در هر تولید نسل نیز از روش "مسابقه (tournament)" [۱۵] استفاده گردید. بطوری که در هر نسل، از بین جمعیت موجود تعداد معینی کروموزوم بصورت تصادفی انتخاب می‌شود. سپس از میان این کروموزوم‌ها، دو کروموزومی که کمترین مقدار هزینه (کوته‌ترین مسیر) را دارند به عنوان والدین در تولید نسل بعدی انتخاب می‌شوند. اکنون الگوریتم ژنتیکی پیشنهادی که ENN-TSP (Evolutionary Neural Network for TSP) نامیده می‌شود، کامل گردیده است.

salesman problem (TSP)," *Iranian Conference on Electrical Engineering (ICEE'01)*, 2001.

- [2] F. Tian, L. Wang, "Chaotic simulated annealing with augmented Lagrange for solving combinatorial optimization problems," *Annual Conference of the IEEE Industrial Electronics Society (IECON'00)*, 4: 2722-2725, 2000.
- [3] L. Jiao and L. Wang, "A novel genetic algorithm based on immunity," *IEEE Trans. Systems, Man, and Cybernetics-Part A: Systems and Humans*, 30(5), 2000.
- [4] S. Boettcher, A. Percus, "Research Note: Nature's way of optimizing," *Artificial Intelligence*, 119:275-286, 2000.
- [5] M. Saadatmand-T., M.-R. Akbarzadeh-T., "A novel constructive hybrid neural network for the symmetric shortest path problem with specified city number," *Iranian Conference on Intelligent Systems (CIS'03)*, 2003.
- [6] M. Budinich, "A self-organizing neural network for the traveling salesman problem that is competitive with simulated annealing," *Neural Comput.*, 8:416-424, 1996.
- [7] K. S. Leung, H. D. Jin, and Z. B. Xu, "An expanding self-organizing neural network for the traveling salesman problem," *Neural Netw.*, 2002, submitted for publication.
- [8] H.-D. Jin, K.-S. Leung, M.-L. Wong, and Z.-B. Xu, "An efficient self-organizing map designed by genetic algorithms for the traveling salesman problem," To Appear in *IEEE Trans. Systems, Man, and Cybernetics-PART B: CYBERNETICS*, 2003.
- [9] M. Saadatmand-T., "A novel genetic fuzzy neural network for the traveling salesman problem," *Iranian Conference on Electrical Engineering (ICEE'03)*, 2003.
- [10] D. Whitley, "A genetic algorithm tutorial," *Statistics and Computer*, 4: 65-85, 1994.
- [11] D. E. Goldberg, "Genetic and evolutionary algorithms in the real world," *IlligAL Report No. 99013*, 1999.
- [12] Z. Michalewicz, *Genetic Algorithms + Data Structures – Evolution Programs*, Springer-Verlage, 2<sup>nd</sup> edition, 1994.
- [13] C. Darwin, *The origin of species by means of natural selection*, Murray, 1859.
- [14] D. Whitley, "The GENITOR algorithm and selective pressure," *Proc. 3<sup>th</sup> Int. Conf. Genetic Algorithms*, ed. Morgan-Kaufmann: 116-121, 1989.
- [15] D. E. Goldberg, "A note on boltzman tournament selection for genetic algorithms and population oriented simulated annealing," *TCGA 90003*, Engineering Mechanics, Alabama University.
- [16] B. Bixby, G. Reinelt, "TSPLIB-A library of traveling salesman and related problem instances," Feb. 1995. <http://sofitlib.rice.edu/pub/tsplib/tsplib/>
- [17] K. Helsgaun, "An effective implementation of the Lin-Kernighan traveling salesman heuristic: Theory and Methodology," *European Journal of Operational Research*, 126:106-130, 2000.



شکل ۱: یک مسیر سه شهری نمونه و شبکه متناظر با آن.

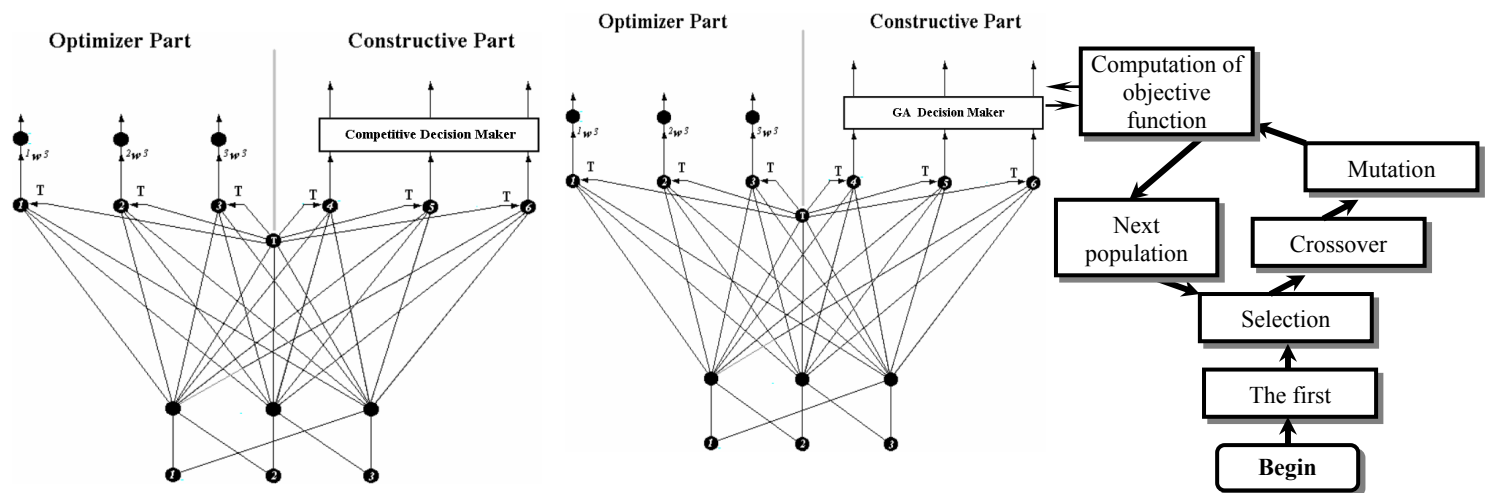
جمعیت مورد استفاده برای هر مسأله ذکر گردیده است. در انتخاب اندازه جمعیت تعداد شهرهای مسأله و پاسخ الگوریتم ژنتیکی به مسأله‌های قبلی مؤثر بوده است. در ستون دوازدهم خطای بهترین پاسخ ارائه شده توسط ENN-TSP نسبت به پاسخ بهینه معرفی شده در کتابخانه TSPLIB ارائه شده است. مقادیر منفی مربوط به مواردی است که پاسخ بدست آمده از الگوریتم ژنتیکی پیشنهادی کوتاهتر از پاسخ ارائه شده در این کتابخانه بوده است. همانطور که مشاهده می‌کنید، الگوریتم ژنتیکی پیشنهادی در ۷ مورد پاسخ‌هایی بهتر از پاسخ‌های TSPLIB و در ۹ مورد همان پاسخ‌ها را ارائه داده است. این در حالی که است که اختلاف نسبی متوسط پاسخ‌های الگوریتم ژنتیکی پیشنهادی در ۶ مورد دیگر نیز تنها ۰/۶۶۵٪ می‌باشد. در جدول (۲)، پاسخ‌های الگوریتم ژنتیکی پیشنهادی با ۵ الگوریتم دیگر شامل CNN-TSP، SA [۶]، Budinich's SOM، (BSOM) [۶]، ESOM [۷] و eISOM [۸] مقایسه شده است. از میان الگوریتم‌های فوق، CNN-TSP، BSOM، ESOM سه شبکه عصبی با کارایی بالا و eISOM یک الگوریتم ژنتیکی-عصبی است. با مراجعه به جدول (۲) بروشنی می‌توان مشاهده نمود که دقت پاسخ‌های الگوریتم ژنتیکی پیشنهادی از تمام روش‌های فوق بیش از ۱/۵٪ بیشتر است.

## ۷- نتیجه‌گیری

در این مقاله یک الگوریتم ژنتیکی جدید برای بهینه‌سازی پاسخ‌های شبکه عصبی CNN-TSP ارائه شده است. این الگوریتم از ساختار سازنده شبکه، برای بهینه‌سازی پاسخ‌های آن استفاده می‌کند. در حقیقت اساس این الگوریتم بر پایه جایگزینی یک تصمیم‌گیرنده تکاملی به جای تصمیم‌گیرنده رقابتی شبکه است. شبیه‌سازی‌های انجام شده نشان می‌دهد که الگوریتم ژنتیکی پیشنهادی قادر است طی زمان قابل قبولی پاسخ‌هایی با دقت بسیار بالا ارائه نماید. این دو ویژگی ENN-TSP را به عنوان ابزاری مناسب برای کاربردهای خارج خط مطرح می‌کند.

## مراجع

- [1] M. Saadatmand-T., M.-R. Akbarzadeh-T., and M. Khademi, "A novel hybrid neural network for the traveling



شکل ۲: شبکه عصبی پیشنهادی از دیدگاهی دیگر (توضیحات لازم در متن مقاله ذکر شده است).

شکل ۳: بلوک دیاگرام الگوریتم ژنتیکی پیشنهادی و نحوه ارتباط شبکه عصبی CNN-TSP با آن. در این شکل از جزئیات مربوط به اولین تولید نسل و شرط خاتمه الگوریتم صرف نظر

$$\begin{aligned}
 & \begin{cases} v_1 = 135 \mid 762 \mid 48 \\ v_2 = 563 \mid 821 \mid 47 \end{cases} \Rightarrow \begin{cases} v_1' = \text{---} \mid 821 \mid \text{---} \\ v_1 = 13576248 \end{cases} \Rightarrow \begin{cases} v_1' = \text{---} \mid 821 \mid \text{---} \\ v_1 = 35764 \end{cases} \Rightarrow v_1' = 357 \mid 821 \mid 64 \\
 & \begin{cases} v_1 = 135 \mid 762 \mid 48 \\ v_2 = 563 \mid 821 \mid 47 \end{cases} \Rightarrow \begin{cases} v_2' = \text{---} \mid 762 \mid \text{---} \\ v_1 = 56382147 \end{cases} \Rightarrow \begin{cases} v_2' = \text{---} \mid 762 \mid \text{---} \\ v_1 = 53814 \end{cases} \Rightarrow v_2' = 538 \mid 762 \mid 14
 \end{aligned}$$

شکل ۴: مراحل اعمال عملگر برش.

جدول ۱: پاسخ الگوریتم ENN-TSP به تعدادی از مسائل TSPLIB.

Tour name	City No.	Optimum answer of TSPLIB	The best answer	The worst answer	Min. Iterations	Max. Iterations	Pop. Size	Average time for a gen. (s)	Prop. dif. of the best answer
ulysses16	16	74.11	73.99	73.99	1	1	50	0.016	-0.16%
ulysses22	22	75.67	75.31	75.31	1	1	50	0.023	-0.36%
bays29	29	9074.15	9074.15	9074.15	1	1	50	0.037	0%
att48	48	33523.71	33523.71	33523.71	11	1	50	0.085	0%
cil51	51	426	428.87	428.87	129	260	50	0.101	0.674%
Berlin52	52	7544.37	7544.37	7544.37	1	1	50	0.104	0%
st70	70	678.60	677.11	677.11	223	746	50	0.226	-0.22%
cil76	76	545.39	544.37	544.37	186	1895	50	0.290	-0.19%
pr76	76	108159.44	108159.44	108159.44	75	964	50	0.268	0%
gr96	96	512.31	510.89	510.89	1613	6215	50	0.598	-0.28%
kroA100	100	21282	21285.44	21285.44	1	55	50	0.672	0.016%
kroC100	100	20750.76	20750.76	20750.76	109	603	50	0.682	0%
kroD100	100	21294.29	21294.29	21294.29	237	267	50	0.658	0%
rd100	100	7910.40	7910.40	7910.40	54	2451	50	0.662	0%
cil101	101	629	640.21	642.45	2108	3072	50	0.710	1.78%
lin105	105	14383.00	14383.00	14383.00	22	29	50	0.769	0%
ch130	130	6110.86	6110.72	6110.72	1143	1742	100	1.708	-0.0023%
gr137	137	698.53	706.29	706.29	1344	1914	100	2.080	1.11%
ch150	150	6532.28	6530.90	6530.90	3629	6882	100	2.795	-0.021%
kroA150	150	26524.86	26524.86	26524.86	1987	2619	100	2.751	0%
kroA200	200	29368	29426.49	29426.49	2385	2585	300	9.014	0.199%
tsp225	225	3859	3867.26	3867.26	10068	10115	400	11.52	0.21%

Pop. = Population, Gen. = Generation, Prop. = Proportional, Dif. = Difference

جدول ۲: اختلاف پاسخ‌های CNN-TSP، الگوریتم ژنتیکی پیشنهادی (ENN-TSP)، بازپخت تطبیقی (SA)،

نگاشت خودسازمانده Budinich، ESOM و eISOM به ۷ مسأله از کتابخانه TSPLIB. برای هر مسأله، بهترین پاسخ با فونت پررنگ (bold) مشخص شده است.

Tour name	No. of Cities	Optimum Answer	Difference of CNN-TSP (%)	Difference of ENN-TSP (%)	Difference of SA (%)	Difference of Budinich's SOM (%)	Difference of ESOM (%)	Difference of eISOM (%)
cil51	51	426	4.09%	0.67%	2.33%	3.10%	2.10%	2.56%
gr96	96	512.31	4.95%	-0.28%	4.12%	2.09%	1.03%	0.81%
kroA100	100	21282	2.58%	0.02%	5.94%	3.68%	1.01%	0.57%
cil101	101	629	6.47%	1.78%	5.74%	5.24%	3.43%	3.59%
gr137	137	698.53	3.12%	1.11%	8.45%	8.61%	4.27%	3.16%
kroA150	150	26524	7.28%	0.00%	4.31%	4.36%	2.04%	1.83%
kroA200	200	29368	5.43%	0.20%	5.61%	6.13%	2.91%	1.64%
Average			4.85%	0.45%	5.21%	4.74%	2.40%	2.02%