الگوريتم جديدي براي مسأله فروشنده دورهگرد با مقياس بزرگ

مهندس حجت‌الهرضازاده   کارشناس پژوهشی مرکز تحقيقات مهندسي صنايع و بهرهوري دانشگاه اميرکبير
مهندس ناهيد شيخان   مربی پژوهشی مرکز تحقيقات مهندسي صنايع و بهرهوري دانشگاه اميرکبير

چکيده
مسآله کلاسيك فروشنده دورهگرد عبارت از تعيين توري است كه كل مسافت يا هزينه بازديد چندين شهر را حداقل نمايد و از هر شهر فقط يكبار عبور نمايد و دوباره به نقطه شروع برگردد. اين مقاله الگوريتم جديدي را براي سرعت بخشيدن به اجراي محاسباتي الگوريتمهاي ابتكاري تعويض شاخه براي مسآله متقارن فروشنده دورهگرد ارائه ميدهد. بهبود اجرا با در نظر گرفتن فقط تعويضاتي كه يك شانس خوب براي ايجاد يك پاسخ بهتر را دارند، بهدست ميآيد. اين الگوريتم از ساير الگوريتمهاي ابتكاري سريعتر ميباشد و زمان محاسباتي آن خيلي كمتر از ساير الگوريتمها ميباشد. الگوريتم جديد امكان پاسخ دادن به مسائل فروشنده دورهگرد را در مقياس بزرگ فراهم ميسازد.

کلمات کليدی: مسألة فروشنده دورهگرد، الگوريتم ابتکاری.

A new Algorithm for Large-Scale Traveling Salesman Problems

H.Rezazadeh   Researcher, Ind. Eng. & Productivity Research Center, Amirkabir University, Tehran, Iran
N.Sheykhan   Member of Scientific board, Ind. Eng. & Productivity Research Center, Amirkabir University, Tehran, Iran

Abstract
The classic traveling – salesman problem is to determine a tour that will minimize the total distance or cost involved in visiting several cities and returning to the starting point. This paper describes a new algorithm for accelerating the computational performances of branch exchange heuristics for symmetric traveling salesman problem. The improvement in performance is obtained by considering only exchanges that have a good chance of producing a better solution. This algorithm is faster than other heuristics' algorithms and computation times increase much less rapidly with problem size. The new algorithm makes it possible to solve large-scale traveling salesman problems.

Key words: Traveling salesman problem, Heuristic algorithm.

۱- مقدمه

مسأله معروف فروشنده دوره گرد عبارت از پيدا كردن توري است كه فروشنده دورهگرد از يك شهر شروع كند و از شهرهاي باقي مانده فقط يكبار بگذرد و دوباره به شهر مبدا برگردد. تور بهينه، توري است كه كل مسافت طي شده يا هزينه سفر آن حداقل باشد. اگر فاصله يا هزينه سفر از شهر i به j برابر با فاصله يا هزينه سفر از شهر j به i باشد ($C_{ij} = C_{ji} \quad \forall$ i,j) مسأله متقارن (SYMMETRIC) و در غير اينصورت غير متقارن (ASYMMETRIC) ناميده ميشود. در يك مسأله فروشنده دورهگرد با بعد n (n تعداد شهرها ميباشد) در حالت متقارن 2/ ! (n-1) و در حالت غير متقارن ! (n-1) تور وجود دارد.

بهدليل كاربردهاي فراوان اين مسأله (بهعنوان مسأله بهينه تركيبي Combinatorial Optimization Problem) توجه محققين از چند دهه قبل به ارائه راه حلهاي اين مسأله معطوف گشته است. حتي در سال ۱۹۸۵ نيز كتابي شامل خلاصهاي از گزارشات مقالات در زمينه فرموله كردن، پيچيدگيهاي مسأله، روشهاي توسعه داده شده براي بهدست آوردن يك پاسخ خوب و پاسخ بهينه چاپ شده است. بهطور كلي روشهاي حل مسأله فروشنده دورهگرد به دو دسته روشهاي پاسخ دقيق (EXACT) و روشهاي حل ابتكاري (HEURISTIC) تقسيم ميشود.

در روشهاي حل دقيق تور بهينه تعيين ميشود ولي در روشهاي حل ابتكاري، تورهاي نزديك بهبهينه بهدست ميآيد. هنوز هم با توسعه كامپيوترها بهدست آوردن پاسخ بهينه از روشهاي حل دقيق وقتي كه بعد مسأله افزايش مييابد، مشكل و مستلزم بكارگيري وقت زيادي از كامپيوتر است. بنابراين استفاده از الگوريتمهاي ابتكاري (بهخصوص با افزايش بعد مسأله) كه تورهاي خيلي نزديك بهبهينه ايجاد ميكنند و سرعت آنها نيز نسبت به روشهاي پاسخ دقيق بسيار كمتر ميباشد، هنوز معتبر است.

در اين مقاله روش جديدي براي سرعت دادن به عمليات محاسباتي الگوريتمهاي ابتكاري تعويض شاخه كه از روشهاي خوب براي پاسخ مسأله متقارن فروشنده دورهگرد ميباشد، ارائه ميشود و ميزان تأثير آن در كيفيت اين حل و نيز كاهش زمان محاسباتي با روشي مشابه سنجيده ميشود.

۱- روشهاي حل مسأله

روشهاي حل مسأله فروشنده دورهگرد به دو بخش تقسيم ميشود:

الف- روش حل دقيق (EXACT)

در اين روش محاسبات در صورتي متوقف ميشود كه تور بهينه بهدست آمده باشد [۱]. الگوريتمهاي پاسخ دقيق عبارتند از:

الف- برنامهريزي پويا (Dynamic Programming)

ب- برنامهريزي عدد صحيح (Integer Programming) و انشعاب و تحديد (Branch and Bound)

ب- روش حل ابتكاري (HEURISTIC)

روشهاي ابتكاري زيادي براي پاسخ مسأله فروشنده دورهگرد ارائه شده است كه برحسب مورد هريك قابليت بيشتري نسبت بهديگري دارد، اصولاً معيار ارزيابي در مورد الگوريتمهاي ابتكاري يكي ميزان نزديكي بهپاسخ بهينه و ديگري كوتاه بودن زمان محاسباتي است. الگوريتمهاي ابتكاري معمولا" بهساخت تور [۲] يا بهبود تور [۳] يا تركيبي از اين دو روش ميپردازند.

يكي از بهترين روشهاي ابتكاري براي پاسخ مسأله فروشنده دورهگرد، روشهاي تعويض شاخه بهخصوص روشهاي ابتكاري 2-opt و 3-opt ميباشد. در سال ۱۹۸۲ S.Ellon و N.Christofides [۴] الگوريتم جديدي را براي تعويض شاخه r-opt مسأله فروشنده دورهگرد ارائه كردند كه زمان محاسباتي با زياد شدن اندازه مسأله افزايش كمي دارد. اين الگوريتم امكان پاسخ مسائل با ابعاد بزرگ را ممكن ميسازد، همچنين ادعا ميشود كه هر تعويض r-opt معادل يك توالي محدود از 2-opt ميباشد و نيز با استفاده از نتايج محاسباتي نشان ميدهند كه حل بدست آمده از 5-opt بهتر از حل بدست آمده از 3-opt و نزديكتر به پاسخ بهينه ميباشد.

در سال ۱۹۸۳ يك روش ابتكاري بسيار موثر براي بدست آوردن پاسخهاي بهينه يا نزديك بهينه توسط B.W.Kernighan , S.Lin [۵] ارائه شده است. اساس اين الگوريتم بر پايه تعويض K شاخه از تور موجه فعلي است تا تور موجه بهبود يافته بهدست آيد و اين عمل تا آنجا ادامه مييابد كه با تعويض بهبود حاصل نشود و

در این صورت پاسخ بهينه محلي بهدست آمده است. در آزمايش ديگري، با چند بار اجرا از يك مسأله شاخههاي مشترك را ثابت نگهداشته و الگوريتم فوق را براي بقيه شاخههاي تور بكار بردند كه نتايج اين روشها بسيار خوب بوده و زمان اجراي آن براي يك مسأله n گرهاي فروشنده دورهگرد، حدود n^2 تخمين زده ميشود.

در سال ۱۹۹۲ مطالعه جامعي توسط T.Doyle, L.Bodin, B.Golden و W.stewart.JR [۶] انجام گرفت كه در آن الگوريتمهاي حل مسأله فروشنده دورهگرد با سه روش، مورد بررسي قرار دادند.

۱- ساخت تور

۲- بهبود تور

۳- تركيبي از ساخت و بهبود تور

با انجام آزمايشات متفاوت به اين نتيجه رسيدند كه بسياري از روشهاي ساخت تور براي پيدا كردن تور، حدود % ۹۵-۹۳ بهينگي ميباشد. روشهاي بهبود تور، بويژه روش 2-opt (بهترين آن از ۲۵ اجرا) و 3-opt (فقط يكبار) با تور تصادفي ابتدايي، مانند بهترين روشهاي ساخت تور عمل ميكنند. تور روش تركيبي سه مرحلهاي داراي % ۹۸-۹۷ بهينگي است. براي بهدست آوردن يك تور با %۹۹-۹۸ بهينگي، يك كاربرد تكراري از روش تركيبي سه مرحلهاي ميتواند مناسبترين روش باشد. در اين روش تور اوليه، تور به دست آمده از روش جايگذاري دلخواه (Arbitrary Insertion) است. اگر اين روش تركيبي را با ۱۰ نقطه شروع اوليه مختلف تكرار كنيم، نتيجه واقعاً عالي و در خور توجه ميباشد.

در سال ۱۹۹۵، W.R.Stewart, Jr [۷] نيز روشي را براي سريعتر كردن عمليات محاسباتي الگوريتمهاي ابتكاري تعويض شاخه براي مسأله فروشند ه دورهگرد ارائه ميدهد. در اين روش فقط شاخههايي در نظر گرفته ميشود كه شانس خوبي براي بهوجود آوردن يك پاسخ بهتر را دارند، ولي اين شاخهها را از طريق حداقل درخت گسترش يافته (Minimal Spanning trees) انتخاب ميكند. اين روش در الگوريتم ابتكاري 3-opt، تعداد عمليات موردنياز را براي بهدست آوردن يك مسأله n گرهاي فروشنده دورهگرد از $O(n^3)$ به $O(n^2)$ كاهش ميدهد، بدون آنكه از كيفيت پاسخ كاسته شود. محقق گزارش ميدهد كه براي يك مسأله ۱۰۰ گرهاي، تعداد مراحل مورد نياز حداقل درخت گسترش يافتــه حـدود ۱۰ الـي ۱۲ مرحله (يعنـي حدود $\frac{n}{10}$) ميباشد.

۲- روش برخورد با مسأله

در اين مقاله روش جديدي جهت انتخاب شاخه براي الگوريتمهاي ابتكاري تعويض شاخه كه هنوز هم از روشهاي حل خوب براي مسأله متقارن فروشنده دورهگرد است ارائه ميشود. ميتوان ادعا نمود اين روش از سرعت محاسباتي بسيار بالا برخوردار است و علاوه بر اينكه دركيفيت پاسخ مسأله تأثير منفي نميگذارد، در خيلي از موارد نيز پاسخ را بهبود ميدهد.

تاكنون روشهايي جهت انتخاب شاخه (در نتيجه محدود كردن تعـداد تست تعـويضات ممكن) ابـداع شده كه مـيتوان از [۴]، [۶] و [۷] نام برد. هدف از روشها اين است كه انتخاب شاخه در كيفيت حل مسأله تأثيري نگذارد و زمان محاسباتي را نيز بهطور قابل توجهي كاهش دهد. بهطوري كه با افزايش بعد مسأله (n) بتوانيم از الگوريتمهاي ابتكاري تعويض شاخه براي بهدست آوردن يك پاسخ خوب استفاده كنيم.

با نظر اجمالي بهروشهاي انتخاب شاخه كه تاكنون ارائه شده است، پي ميبريم كه زمان محاسباتي اين روشها نيز نسبت به زمان مورد نياز جهت تعويض شاخه قابل توجه ميباشد، بهعنوان مثال در [۷] از روش حداقل درخت گسترش يافته (MST) جهت انتخاب شاخه استفاده شده است در حاليكه اين روش نيز وقتي كه بعد مسأله بهطور قابل توجهي افزايش مييابد، زمان نسبتا" زيادي نياز دارد.

الگوريتم پيشنهادي زير اين نياز زماني را بهطرز قابل توجهي كاهش ميدهد. شرح الگوريتم به قرار زير است:

۲-۱- الگوريتم پيشنهادي

الگـوريتـم پيشنهـادي كه آن را متـوسط شاخـهها (Average of Arcs) مينامیم، همانطور كه از نام آن نيز پيداست بر مبناي معدلگيري از شاخهها استوار است. اين الگوريتم فقط نسبتي از شاخههايي را كه از يك گره خارج ميشوند، را انتخاب ميكند. فرض كنيد كه يك شبكه n گرهاي داريم، از هر گره n-1 شاخه خارج ميشود و آنرا به n-1 گره مرتبط ميسازد. حال اگر يك گره فرضي مثلاً I را در نظر بگيريم و مجموع طول n-1 شاخههايي كه از اين گره خارج ميشوند را بر n-1 تقسيم كرده و آنرا حد بناميم، سپس شاخههايي را از اين گره انتخاب كنيم كه طول آنها از اين حد كمتر باشد، با اين روش توانستهايم حدود ۵۰% از كوتاهترين شاخههايي كه از اين گره خارج ميشوند را انتخاب نماييم و اگر

حد را بر ۲ تقسیم کنیم حدود ۲۵% و اگر بر ۳ تقسیم کرده حدود ۱۷% و ... شاخه انتخاب میشوند. حال اگر این عمل را براي بقيه گرهها نيز انجام دهيم بهطرز بسيار ساده و بـا سرعت چشمگيري ميتوانيم حدود ۵۰%، ۲۵%، ۱۷% و ... كوتاهترين شاخههاي يك مسأله n تايي فروشنده دورهگرد را انتخاب كنيم. الگوريتم زير مراحل مختلف محاسبات را نشان ميدهد.

مرحله ۱: فرض كنيد $C : n \times n$ ماتريس فواصل بين گرهها و $C' : n \times n$ ماتريس شاخههاي انتخابي باشد. P (درصد انتخاب شاخهها) را مساوي ۵۰% ، ۲۵% ، ۱۷% و ... قرار دهيد.

مرحله ۲: اولين سطر (i=1) از ماتريس C را در نظر بگيريد.

مرحله ۳ : حد سطر يعني b_i را از رابطه ذيل بهدست آوريد.

$$b_i = \frac{\sum_{j=1,\ j \neq i}^{n} C_{ij}}{(n-1)} \times 2p$$

مرحله ۴: اگر $C_{ij} \langle= b_i\ (\forall j)$ باشد، $C'_{ij} = C_{ij}$ و $C'_{ji} = C_{ij}$ قرار دهيد.

مرحله ۵: اگر i=n باشد متوقف شويد، درغير اينصورت i را يك واحد افزايش داده و به مرحله ۳ برويد.

بهعنوان مثال اگر ماتريس زير نشان دهنده فواصل بين گرهها در يك مسأله (n=10) متقارن فروشنده دورهگرد باشد.

$$C = \begin{bmatrix} - & 75 & 40 & 38 & 20 & 77 & 91 & 84 & 7 & 5 \\ 75 & - & 62 & 30 & 52 & 16 & 9 & 80 & 66 & 49 \\ 40 & 62 & - & 99 & 37 & 76 & 85 & 6 & 52 & 71 \\ 38 & 30 & 99 & - & 82 & 40 & 25 & 85 & 80 & 55 \\ 20 & 52 & 37 & 82 & - & 80 & 75 & 91 & 93 & 8 \\ 77 & 16 & 76 & 40 & 80 & - & 78 & 41 & 22 & 7 \\ 91 & 9 & 85 & 25 & 75 & 78 & - & 81 & 27 & 36 \\ 84 & 80 & 6 & 85 & 91 & 41 & 81 & - & 52 & 56 \\ 7 & 66 & 52 & 80 & 93 & 22 & 27 & 52 & - & 51 \\ 5 & 49 & 71 & 55 & 8 & 7 & 36 & 56 & 51 & \end{bmatrix}$$

به ازاي P=0.5 خواهيم داشت:

$$C' = \begin{bmatrix} - & - & 40 & 38 & 20 & - & - & - & 7 & 5 \\ - & - & - & 30 & 52 & 16 & 9 & - & - & - \\ 40 & - & - & - & 37 & - & - & 6 & 52 & - \\ 38 & 30 & - & - & - & 40 & 25 & - & - & 55 \\ 20 & 52 & 37 & - & - & - & - & - & - & 8 \\ - & 16 & - & 40 & - & - & - & 41 & 22 & 7 \\ - & 9 & - & 25 & - & - & - & - & 27 & 36 \\ - & - & 6 & - & - & 41 & - & - & 52 & 56 \\ 7 & - & 52 & - & - & 22 & 27 & 52 & - & - \\ 5 & - & - & 55 & 8 & 7 & 36 & 56 & - & - \end{bmatrix}$$

و براي بقيه موارد نيز مانند حالات فوق ميتوان عمل كرد.

از اين شاخههاي انتخاب شده ميتوان در روشهاي تعويض شاخه 2-opt و 3-opt استفاده نمود.

۳- نتايج محاسباتي

براي ارزيابي الگوريتم پيشنهادي و ميزان تأثير آن در الگوريتمهاي تعويض شاخه، از روشهاي تركيبي استفاده ميكنيم. اين روشها از روشهاي ابتكاري سريع و تور حاصل از اجراي آن خيلي نزديك به توربينه (درحد % ۹۸-۹۷ بهينگي) ميباشد. اساس كار روشهاي تركيبي به قرار زير است:

مرحله ۱: يك تور ابتدايي پيدا كنيد.

مرحله ۲: الگوريتم 2-OPT را براي تور به دست آمده در مرحله ۱ بهكار ببريد.

مرحله ۳: الگوريتم 3-OPT را براي تور بهدست آمده در مرحله ۲ بهكار ببريد.

با توجه به اينكه از الگوريتم خوب تورسازي جايگذاري دلخواه و نيز از دو روش انتخاب شاخه (الگوريتم MST و الگوريتم پيشنهادي) بهره ميبريم، پس روش كار براي مسأله متقارن n گرهاي فروشنده دورهگرد به قرار زير ميباشد:

قدم ۱- تور ابتدايي را بهوسيله الگوريتم جايگذاري دلخواه ايجاد كنيد.

قدم ۲- الگوريتم 2-opt را براي تور بهدست آمده از قدم ۱ با داشتن ماتريس انتخاب شاخه (اين ماتريس انتخاب شاخه از الگوريتم MST يا الگوريتم پيشنهادي تشكيل ميشود و در غيراينصورت (درحالت كلي) همان ماتريس فواصل بين گرهها در نظر گرفته ميشود) اجرا كنيد.

قدم ۳- الگوريتم 3-opt را براي تور بهدست آمده از قدم ۲ و با داشتن ماتريس انتخاب شاخهاي كه درقدم ۲ استفاده شده اجرا كنيد.

براي تشكيل ماتريس انتخاب شاخه در يك مساله n بعدي، MST را در k=n/10 [۷] مرتبه انجام ميدهيم (بعبارت بهتر اگر بعد مسأله n=۴۰ باشد الگوريتم MST را براي $k = \frac{۴۰}{۱۰} = ۴$ مرتبه انجام ميدهيم و از شاخههاي انتخاب شده آن در الگوريتم 2-opt و 3-opt استفاده ميكنيم) و الگوريتم پيشنهادي را نيز در ۵ حالت مختلف ($P =$ %۵ ، %۲۵ ، %۱۷ ، %۱۲۵ ، %۱۰) بررسي ميكنيم. پس بنابر حالات مختلف استفاده از الگوريتمها در ماتريس انتخاب شاخه و نيز روش تورسازي جايگذاري دلخواه، الگوريتمهاي تعويض شاخه را بهكار ميبريم.

براي آزمايش اين روش بهطور تصادفي ۲۰ مسئله به اين طريق ايجاد نموديم كه براي هر مسئله n بعدي فروشنده دورهگرد ۵ مسأله ايجاد نموديم (كه n برابر با ۴۰ , ۶۰ , ۸۰ و ۱۰۰ ميباشد) با حل آن توسط روش فوق نتايجي حاصل شد كه نتايج اين آزمايشات در جدول (۱-۳) آورده شده است.

۴- ارزيابي نتايج

همانطـور كه مـيدانيم روشهـاي تـركيبــي تــور را در حد % ۹۸-۹۷ بهينگـي پيدا مـيكنند. نتـايج جـدول (۱-۳) نيز نشان مـيدهد كه تـور روش (n/10)MST در حـد % ۹۸/۱، روش AV(p=0.5) در حد % ۹۹/۹، روش AV(p=0.25) در حد %۹۷/۸، روش AV(p=0.17) در حد % ۹۷/۷، روش AV(p=0.125) در حد % ۹۰/۹ و سرانجام روش AV(p=0.1) در حد % ۸۷/۲ تور بهدست آمده از روشهاي تركيبي ميباشند. البته در بيشتر موارد تور روشهاي AV(p=0.25),n/10MSTو AV(p=0.17) بهتر از روشهاي تركيبي و تور AV(p=0.5) مساوي با تور بهدست آمده از روشهاي تركيبي ميباشد، الگوريتم پيشنهادي بخصوص در موارد AV(p=0.5)، AV(p=0.25) و AV(p=0.17) نه تنها مانند روش KMST از كيفيت پاسخ نميكاهد بلكه در موارد زيادي نيز بر آن ميافزايد.

براي بررسي بهتر اثرات روشهاي انتخاب شاخه بر الگوريتمهاي تعويض شاخه (2-opt , 3-opt) اجازه دهيد كه زمانهاي محاسباتي كامپيوتر جهت ساخت تور و نيز انتخاب شاخه را از زمان كل ارائه شده حذف نماييم. جدول (۱-۴) اين نتايج را در بردارد.

با بررسي اين جدول پي ميبريم كه الگوريتم انتخاب شاخه AV(p=0.5) سبب كاهش زمان لازم محاسباتي به ميزان كمتر از نصف و بقيه الگوريتمها نيز زمان محاسباتي را به حدود $\frac{۱}{۳}$ زمان لازم تقليل ميدهند. يعني شاخههاي انتخابي از روش متوسط شاخهها، همان كاري را انجام ميدهند كه شاخههاي انتخابي از الگوريتم KMST انجام ميدهند. جــدول (۲-۴) و شكل (۱-۴) نشانگر اين نتيجه ميباشد.

جدول ۱-۳- نتايج محاسباتي آزمايشات بعمل آمده با تورسازي AI

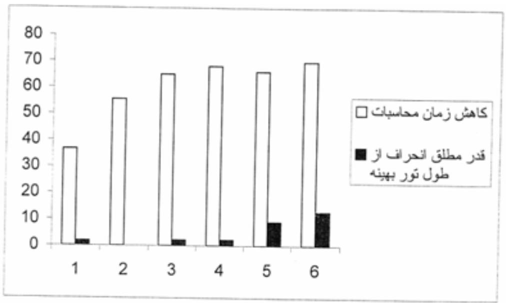
روش كار متوسط۵ مسأله n بعدي	AI 2-opt 3-opt	(n/10)MST AI 2-opt 3-opt	AV(p=0.5) AI 2-opt 3-opt	AV(p=0.25) AI 2-opt 3-opt	AV(P=0.17) AI 2-opt 3-opt	AV(p=0.125) AI 2-opt 3-opt	AV(p=0.1) AI 2-opt 3-opt	(n/10)MST	AV(p=0.1)	*AI*
n=۴۰ طول تور زمان (s)	۲۳۴/۶ ۱۷/۵۹	۲۳۸/۲ ۱۳/۲۴	۲۳۶/۸ ۷/۱۱	۲۴۰/۶ ۵/۴۲	۲۴۹/۲ ۵/۲۷	۲۸۱/۸ ۳/۷۸	۳۳۵/۴ ۲/۶۶	۷/۹۱	۰/۰۸	۴۴۳ ۰/۰۲
n =۶۰ طول تور زمان (s)	۲۴۹/۲ ۶۷/۳۸	۲۴۴/۶ ۷۵/۰۲	۲۴۹/۲ ۳۰/۸۲	۲۵۷/۶ ۲۳/۲۶	۲۵۳ ۲۷/۱۴	۲۵۷/۴ ۲۵/۲۳	۲۷۸/۶ ۲۴/۱۸	۵۰/۱	۰/۱۶	۵۸۴/۲ ۰/۰۶
n=۸۰ طول تور زمان (s)	۲۷۴/۶ ۱۹۲/۷	۲۶۵/۲ ۲۶۷/۳۸	۲۷۴/۶ ۸۶/۸۶	۲۶۸/۶ ۷۳/۹۷	۲۷۵/۸ ۴۹/۳	۲۶۹ ۸۱/۳۵	۲۷۶/۴ ۶۳/۴۹	۱۸۹/۱۱	۰/۲۹	۶۶۶/۲ ۰/۱۱
n=۱۰۰ طول تور زمان (s)	۲۹۷ ۴۳۲/۸	۲۹۳ ۷۱۰/۲۷	۲۹۷ ۱۹۵/۴	۳۰۱ ۱۴۵/۰۴	۲۹۳ ۱۵۰/۸۹	۲۹۰/۲ ۱۳۸/۶۷	۲۹۳/۲ ۱۵۷/۸۱	۵۴۹/۱۵	۰/۴۵	۷۵۰۱/۸ ۰/۷

جدول ۱-۴- زمانهاي محاسباتي و درصد بهينگي طول تور الگوريتمهاي تعويض شاخه با استفاده از تور سازي AI

رديف	زماذ هاي محاسباتي براساس روش كار (s) متوسط ۵ مسآله n بعدي	AI 2-opt 3-opt	(n/10)MST AI 2-opt 3-opt	AV(p=0.5) AI 2-opt 3-opt	AV(p=0.25) AI 2-opt 3-opt	AV(p=0.17) AI 2-opt 3-opt	AV(p=0.125) AI 2-opt 3-opt	AV(p=0.1) AI 2-opt 3-opt
۱	۴۰ n =	۱۷/۵۷	۵/۳۱	۷/۰۱	۵/۳۲	۵/۱۷	۳/۶۸	۲/۵۶
	۶۰n =	۶۷/۳۲	۲۴/۸۶	۳۰/۶	۲۳/۰۴	۲۶/۹۲	۲۵/۰۱	۲۳/۹۶
	۸۰n =	۱۹۲/۵۹	۷۸/۱۶	۸۶/۴۶	۷۳/۵۷	۴۸/۹	۸۰/۹۵	۶۳/۰۹
	۱۰۰n =	۴۳۲/۶۳	۱۶۰/۹۵	۱۹۴/۷۸	۱۴۴/۴۲	۱۵۰/۲۷	۱۳۸/۰۵	۱۵۷/۱۹
۲	درصد بهينگي طول تور نسبت به عدم استفاده از الگوريتمهاي تعويض شاخه	۱۰۰	۹۸/۱	۱۰۰	۹۷/۸	۹۷/۷	۹۰/۹	۸۷/۲

جدول ۲-۴ – مقايسه طول تور و زمان محاسباتي الگوريتمهاي تعويض شاخه با استفاده از روشهاي انتخاب شاخه نسبت به عدم استفاده از آن در الگوريتمهاي تعويض شاخه روشهاي تركيبي

روش	*کاهش زمان محاسبات*	قدر مطلق انحراف از طول توربهينه
(n/10)MST	% ۳۶/۶	%۱/۹
AV(p=0.5)	%۵۵/۵	%۰
AV(p=0.25)	%۶۵	%۲/۲
AV(p=0.17)	%۶۸	%۲/۳
AV(p=0.125)	%۶۶	%۹/۱
AV(p=0.1)	%۶۹/۸	%۱۲/۸



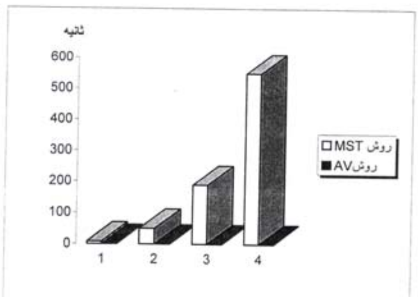
(N/10)MST	1
AV(P=0.5)	2
AV(P=0.25)	3
AV(P=0.17)	4
AV(P=0.125)	5
AV(P=P=0.1)	6

شكل ۱-۴ – مقايسه طول تور و زمان محاسباتي الگوريتمهاي تعويض شاخه با استفاده از روشهاي انتخاب شاخه نسبت به عدم استفاده از آن در الگوريتمهاي تعويض شاخه روشهاي تركيبي

حال به جدول (۳-۴) و شکل (۲-۴) که زمانهاي محاسبه انتخاب شاخه الگوريتمهاي MST و AV را مقايسه ميکند توجه کنيد.

جدول ۳-۴- مقايسه زمانهاي محاسبه انتخاب كمان (ثانيه) از دو روش AV,MST

رديف	n	(n/10) MST	AV(p=0.5 ,0.25,0.17,0. 125,0.1)
۱	۴۰	۷/۹۱	۰/۰۸
۲	۶۰	۵۰/۱	۰/۱۶
۳	۸۰	۱۸۹/۱۱	۰/۲۹
۴	۱۰۰	۵۴۹/۱۵	۰/۴۵



شكل ۲-۴- مقايسه زمانهاي محاسبه انتخاب كمان (ثانيه) از دو روش AV,MST

زمانهـاي محاسـباتي الگـوريتم پيـشنهادي صـدها بـار كمتـر از KMST ميباشد براي يك مسئله مثلاً n=۱۰۰ زمـان لازم بـراي پيدا كـردن ۱۰ مرتبـه MST برابـر بـا ۵۴۹/۱۵ ثانيـه اسـت در حاليكه روش متوسط شاخهها اين شاخهها را حدود ۰/۴۵ ثانيـه انتخاب ميكند، كه اين نكته بسيار قابل توجـه و برجـسته بـراي اين الگوريتم ميباشد، زيرا با افزايش بعد مسأله، بكـارگيري ايـن الگوريتم جهت انتخاب شاخه (كه زمان لازم محاسباتي آن براي مسائل خيلي بزرگ نيز در حد چند ثانيه ميباشد) براي سـرعت دادن به محاسبات الگوريتمهاي تعـويض شـاخه بـسيار مـشهود ميباشد.

۵- نتيجه گيری

الگوريتمهاي ابتكاري تعويض شاخه از جمله الگوريتمهاي ابتكاري سريعي است كه هنوز هم جهت بهدست آوردن پاسخ نزديك به بهينه مسأله متقارن فروشنده دورهگرد از آن استفاده ميشود. روشهايي جهت انتخاب شاخه براي سريعتر كردن عمليات محاسباتي اين الگوريتمها وجود دارد كه هدف از اين روشها اين ميباشد كه انتخاب شاخه جدا از اينكه در كيفيت پاسخ مسأله تأثير نگذارد، زمان محاسباتي را نيز بهطور قابل توجهي كاهش دهد تا بتوان از الگوريتمهاي تعويض شاخه وقتي كه بعد مسأله افزايش مييابد بهراحتي استفاده نمود.

در اين مقاله روش جديدي (Average of Arcs) براي انتخاب شاخه ارائه شد و در پاسخ مسائل آزمايشي نشان داد كه زمان محاسباتي الگوريتمهاي تعويض شاخه را به $\frac{۱}{۳}$ زمان مورد نياز تقليل ميدهد بدون آنكه از كيفيت پاسخ بكاهد، در موارد متعددي كيفيت جواب را نيز بهبود داده است. نكته برجسته و قابل توجه در مقايسه اين روش با ساير روشهايي كه تاكنون براي سريعتر كردن عمليات محاسباتي الگوريتمهاي ابتكاري تعويض شاخه ارائه شدهاند، زمان محاسباتي بسياركم اين روش است، كه در مقايسه با روش حداقل درخت گسترش يافته (MST) صدها بار سريعتر عمل ميكند و ميتواند شاخههاي انتخابي براي مسائل خيلي بزرگ را در حد چند ثانيه بهدست آورد. لذا ميتوان براي بهدست آوردن پاسخ نزديك به بهينه مسائل خيلي بزرگ از اين روش در الگوريتمهاي ابتكاري تعويض شاخه استفاده نمود. همچنين پيشنهاد ميشود در الگوريتمهاي ديگري كه بطريقي از تعويض شاخه جهت بهدست آوردن پاسخ نزدیک بهبهینه مسأله فردوشنده دوره گرد استفاده میکنند از این روش استفاده شود.

مراجع

[1] K. R. Baker, "Introduction to sequencing and scheduling", John Wiley & Sons Inc, (1974)94-106.

[2] R. L. Karg and G.L.Thompson, "A HEURISTIC APPROACH TO SOLVING TRAVELLING SALESMAN PROBLEMS", Management Science 10(1994)225-248.

[3] T. C. Raymond, "HEURISTIC Algorithm for the Traveling-Salesman Problem", IBM J.RES. DEVELOP 13 (1999) 400-407.

[4] N. Christrofides and S.Eilon, "Algorithms for Large-Scale Traveling Salesman Problem", Operational Research Quartely. 23 (2000) 511-518.

[5] S. Lin and B. W. Kernighan, "An Effective Heuristic for Symmetric Traveling Salesman Problem", Operation Res. 21 (1997) 498-516.

[6] B. Golden, L. Bodin, T. Doyle and W.Stewart, JR, "Approximate Traveling Salesman Algorithms", Operation Res. 28 (1998) 694-711.

[7] W. Stewart, JR, "Accelerated Branch Exchange Heuristic for Symmetric Traveling Salesman Problems", Networks.17 (1998).