

## حل مساله فروشنده دوره‌گرد احتمالی توسط اتوماتای یادگیر توزیع شده

محمد علیپور      محمد رضا میبیدی

آزمایشگاه سیستمهای نرم افزاری  
دانشکده مهندسی کامپیوتر و فناوری اطلاعات  
دانشگاه صنعتی امیرکبیر  
تهران ایران  
(alipour, meybodi)@ce.aut.ac.ir

گیرد. هدف مساله فروشنده دوره‌گرد احتمالی یافتن توری است با کمترین متوسط طول بطوریکه اگر هر زیر مجموعه تصادفی از مجموعه شهرها را انتخاب کرده و با همان ترتیبی که در این تور قرار گرفته‌اند بازدید نمائیم دارای کمترین متوسط طول باشند. به این تور که دارای کمترین طول متوسط است، یک تور اولیه<sup>۶</sup> گفته میشود. مساله فروشنده دوره‌گرد احتمالی از مسائل NP-hard بوده [۳، ۱] و برای اولین بار در رساله‌ی دکترای جیل<sup>۷</sup> معرفی شده است [4]. استراتژی‌های مختلفی برای تعیین یک تور اولیه گزارش شده است. استراتژی یک تور اولیه شامل دو جزء می باشد. تعیین یک راه حل اولیه و یک روش بروزرسانی. یک راه حل اولیه، توری است که در آن طول هر زیر مجموعه ممکن از شهرها، کمینه می باشد. روش بروزرسانی، شهرهایی که بازدید نخواستند شد را حذف نموده و بقیه شهرها را با همان ترتیبی که در تور اولیه ظاهر شده‌اند، بازدید می کند. مثالی ساده از روش بروزرسانی بقرار زیر می باشد: هر زیر مجموعه از شهرها را بتریبی که در تور اولیه ظاهر شده‌اند ملاقات کن و از شهرهایی که در تور اولیه بوده و متعلق به این زیر مجموعه نمی باشند، صرفنظر کن (پرش از شهرهایی که نیاز به ملاقات ندارند). این استراتژی، استراتژی پرش<sup>۸</sup> نامیده می شود.

تاکنون راهکارهای الگوریتمی و مکاشفه ای مختلفی برای یافتن جواب بهینه PTSP گزارش شده است [5]. رسی گاویولی<sup>۹</sup> از قوانین مکاشفه ای به نامهای نزدیکترین همسایه و معیار پس انداز<sup>۱۰</sup> استفاده کردند [6].

**چکیده:** در مساله فروشنده دوره‌گرد احتمالی فروشنده دوره‌گرد، هر شهر با احتمال معینی مورد بازدید قرار میگیرد و هدف پیدا کردن یک تور اولیه است که نه تنها دارای حداقل طول متوسط میباشد بلکه اگر از هر زیر مجموعه‌ی تصادفی از شهرها، با همان ترتیبی که در تور اولیه ظاهر شده‌اند، بازدید شود، دارای کمترین طول متوسط باشند. در این مقاله با استفاده از اتوماتای یادگیر توزیع شده<sup>۱</sup> الگوریتمی برای حل مساله فروشنده دوره‌گرد احتمالی ارائه می شود. الگوریتم پیشنهادی با دو الگوریتم مکاشفه ای بهترین تصادفی<sup>۲</sup> و مرتب سازی محوری<sup>۳</sup>، مقایسه شده است. طبق نتایج بدست آمده الگوریتم پیشنهادی نسبت به هر دو الگوریتم فوق الذکر نتایج بهتری را تولید می کند.

**کلمات کلیدی:** فروشنده دوره‌گرد احتمالی، اتوماتاهای یادگیر، اتوماتای یادگیر توزیع شده، مسایل مشکل

### ۱- مقدمه

مساله فروشنده دوره‌گرد<sup>۴</sup> (TSP) مساله‌ای است که در آن بایستی برای تعدادی شهر که فواصل بین آنها داده شده است توری با کمترین هزینه پیدا شود که از یک شهر شروع شده ، کلیه شهرها را فقط یکبار بازدید نماید و سپس به شهر آغازین بر گردد. مساله فروشنده دوره‌گرد احتمالی<sup>۵</sup> (PTSP) یکی از انواع مساله TSP می باشد که در آن هر شهر با احتمال معینی که مستقل از بازدید سایر شهرها می باشد مورد بازدید قرار می

<sup>۶</sup> A priori tour

<sup>۷</sup> Jailliet

<sup>۸</sup> Skipping strategy

<sup>۹</sup> Rossi-Gavioli

<sup>۱۰</sup> Savings

<sup>۱</sup> Distributed Learning Automata

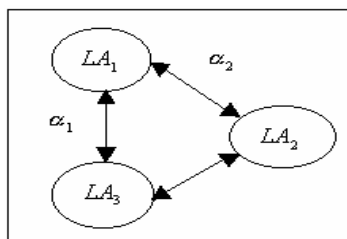
<sup>۲</sup> Random Best

<sup>۳</sup> Radial Sort

<sup>۴</sup> TSP

<sup>۵</sup> PTSP

یادگ یز در شبکه، اتوماتای یادگ یز متناظر با این اقدام را فعال می سازد. بعنوان مثال در شکل ۱ هر اتوماتا یادگ یز دارای دو اقدام می باشد. انتخاب اقدام  $\alpha_2$  توسط  $LA_1$ ، اتوماتا یادگ یز  $LA_3$  را فعال خواهد کرد. اتوماتای یادگ یز فعال شده ( $LA_3$ ) سپس یکی از اقدامهای خود را انتخاب می کند که در نتیجه آن یکی از اتوماتاهای یادگ یز متصل به آن اتوماتا ی یادگ یز که متناظر با اقدام انتخاب شده می باشد فعال می شود. در هر زمان فقط یک اتوماتای یادگ یز در شبکه فعال می باشد. بطور رسمی DLA را میتوان توسط گراف  $DLA = (V, E)$  که  $V = \{LA_1, LA_2, \dots, LA_n\}$  مجموعه اتوماتاهای یادگ یز و  $n$  تعداد اتوماتاهای یادگ یز در DLA و  $E \subset V \times V$  مجموعه لبه های گراف می باشد، تعریف کرد. لبه  $(i, j)$  اقدام  $j$  اتوماتا ی یادگ یز  $LA_i$  را نشان می دهد.  $LA_j$  زمانی فعال خواهد شد که اقدام  $j$  اتوماتای یادگ یز  $LA_i$  انتخاب شود. تعداد اقدامهای اتوماتای یادگ یز  $LA_k$  ( $k = 1, 2, \dots, n$ ) برابر درجه ی خروجی گره متناظر با اتوماتای یادگ یز  $LA_k$  می باشد. برای اطلاعات بیشتر در باره اتوماتاهای یادگ یز توزیع شده میتوان به مراجع [11] و [12] مراجعه نمود.



شکل ۱: اتوماتای یادگ یز توزیع شده (DLA) با ۳ اتوماتای یادگ یز

#### ۴- تابع هدف مساله فروشنده دوره گرد احتمالی

برای تشریح تابع هدف مساله فروشنده دوره گرد احتمالی، گراف کاملی با  $n$  گره (شهر) را که گره های آن از مجموعه  $V = \{i = 1, 2, \dots, n\}$  می باشند، بعنوان نمونه مساله نظر گرفته می شود. فرض میکنیم  $p_i$  احتمال ملاقات شهر  $i$ ، که مستقل از احتمال ملاقات سایر شهرها است باشد. هدف مینیمیز کردن تابع  $E[L_\lambda]$  که به قرار زیر تعریف شده است میباشد.

$$E[L_\lambda] = \sum_{S \subseteq V} p(S) L_\lambda(S) \quad (1)$$

که  $S$  زیر مجموعه ای از گره های  $V$  و  $L_\lambda(S)$  متوسط مسافت طی شده برای بازدید شهرهای مجموعه  $S$  بترتیبی که در یک تور اولیه ظاهر شده باشند میباشد.  $p(S)$  احتمال بازدید شهرهای موجود در آن مجموعه است. یک تور اولیه، (تور  $\lambda$ ) شامل همه ی گره های مجموعه  $V$  بوده و دارای طول متوسط زیر می باشد:

برتیماس<sup>۱۱</sup> و همکارانش در [3] و برتیماس و هاول<sup>۱۲</sup> در [2] با بررسی کامل برخی ویژگیهای PTSP تعدادی قوانین مکاشفه ای که شامل قانون مکاشفه ای ایجاد تور و قانون مکاشفه ای بهبود تور می باشند پیشنهاد کرده اند. اغلب الگوریتمهای ارائه شده برای PTSP از تطبیق الگوریتم TSP برای PTSP ایجاد شده و یا حتی همان الگوریتم اصلی TSP می باشند که در برخی موارد جوابهای PTSP خوبی تولید می کنند.

در این مقاله یک الگوریتم مبتنی بر اتوماتای یادگ یز توزیع شده برای حل مساله فروشنده دوره گرد احتمالی ارائه میگردد. نتایج بدست آمده از الگوریتم پیشنهادی، با دو الگوریتم، مکاشفه ای بهترین تصادفی و مرتب سازی محوری، مقایسه میگردد و نشان داده میشود که برای محدوده ی وسیعی از احتمالات بازدید شهرها، الگوریتم پیشنهادی جوابهای بهتری تولید می کند. در این مقاله روشی برای ارزیابی کارایی مطلق الگوریتم که براساس حد پایین تئوریک می باشد معرفی و کارایی مطلق الگوریتم پیشنهادی با استفاده از این روش ارزیابی میگردد. اتوماتای یادگ یز توزیع شده قبلا برای حل مساله فروشنده دوره گرد و فروشنده دوره گرد پویا مورد استفاده قرار گرفته است [20][13]. در ادامه مقاله، ابتدا در بخش ۲ اتوماتاهای یادگ یز و در بخش ۳ اتوماتاهای یادگ یز توزیع معرفی میگردد. در بخش ۴ تابع هدف برای مساله فروشنده دوره گرد احتمالی را تعریف میگردد. در بخش ۵ الگوریتم پیشنهادی ارائه میشود و در بخش ۶ نتایج آزمایشها ارائه میشود. بخش ۷ نتیجه گیری میباشد.

#### ۲- اتوماتاهای یادگ یز

اتوماتای یادگ یز یک مدل انتزاعی است که تعداد محدودی عمل را می تواند انجام دهد. هر عمل انتخاب شده توسط محیطی احتمالی ارزیابی شده و پاسخی به اتوماتای یادگ یز داده می شود. اتوماتای یادگ یز از این پاسخ استفاده نموده و عمل خود را برای مرحله بعد انتخاب می کند. اتوماتاهای یادگ یز به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم بندی میگردند برای اطلاعات بیشتر در باره اتوماتاهای یادگ یز با ساختار متغیر که در این مقاله از آنها استفاده شده است می توان به [14],[15],[16],[17] مراجعه کرد

#### ۳- اتوماتای یادگ یز توزیع شده<sup>۱۳</sup>

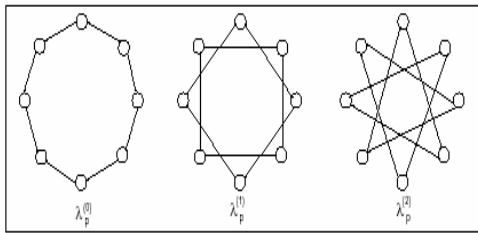
اتوماتای یادگ یز توزیع شده (DLA)، شبکه ای از اتوماتاهای یادگ یز است که برای حل یک مساله با یکدیگر همکاری می نمایند [2]. تعداد اقدامهای یک اتوماتا در DLA برابر تعداد اتوماتاهای یادگ یز متصل به این اتوماتای یادگ یز می باشد. انتخاب یک اقدام توسط یک اتوماتای

<sup>11</sup> Bertsimas

<sup>12</sup> Howell

<sup>13</sup> Distributed Learning Automata

و بهمین صورت ادامه می دهیم. برای روشن شدن بیشتر مطلب شکل ۲ را در نظر می گیریم، طبق آنچه از شکل قابل مشاهده است این نمونه ۸ شهری و متجانس (احتمال بازدید هر ۸ شهر یکسان است) بوده و  $\lambda_p^{(0)}$  را میتوان بعنوان یک تور اولیه برای این نمونه در نظر گرفت. در این شکل زیر تور  $\lambda_p^{(1)}$  از بازدید یک شهر و پرش از یک شهر بعدی و  $\lambda_p^{(2)}$  از بازدید یک شهر و پرش از دو شهر بعدی در یک تور اولیه نمونه داده شده  $(\lambda_p^{(0)})$  حاصل شده اند.



شکل ۲: زیر تورهای  $\lambda_p^{(0)}$ ،  $\lambda_p^{(1)}$  و  $\lambda_p^{(2)}$  یک نمونه ۸ شهری متجانس که طول متوسط آنها بترتیب برابر عبارتهای  $L_\lambda^{(0)}$ ،  $L_\lambda^{(1)}$  و  $L_\lambda^{(2)}$  معادله ۴ می باشد.

#### ۵- الگوریتم پیشنهادی

ابتدا شبکه‌ای از اتوماتاهای یادگیر که متناظر<sup>۱۶</sup> با گراف مساله PTSP میباشد ایجاد می شود. در این شبکه هر گره (معادل یک شهر در مساله)، یک اتوماتای یادگیر با ساختار متغیر بوده و هر لبه‌ی خروجی این گره (جاده‌ای که توسط آن از این شهر میتوان به شهری دیگری رفت) یکی از اقدامهای آن اتوماتای یادگیر می باشد. تعداد اقدامهای یک اتوماتای یادگیر متناظر با یک شهر معادل تعداد شهرهایی می باشد که میتوان بطور مستقیم از این شهر به آنها مسافرت کرد. خروجی DLA ترتیبی از اقدامهای انتخاب شده توسط اتوماتاهای یادگیر می باشد که یک تور را در گراف نشان می دهد. که از آن بعنوان یک تور اولیه استفاده خواهد شد. محیط از طول این تور برای تولید خروجی استفاده می کند. این خروجی با توجه به مطلوب یا نامطلوب بودن آن، باعث پاداش و یا جریمه دادن به اقدامهای اتوماتاهای یادگیر واقع در این تور (تور هامیلتونی ایجاد شده البته اگر تور هامیلتونی در گراف وجود داشته باشد) می شود.

قبل از اینکه به شرح الگوریتم پیشنهادی بپردازیم بایستی به این نکته اشاره نمود که آزمایشهای انجام شده نشان داده است که اگر یک اتوماتای یادگیر فعال فقط از بردار احتمال اقدامها برای انتخاب اقدام خود استفاده کند، جوابهای بدست آمده تقریباً غیر قابل قبول بوده و همچنین نرخ همگرایی الگوریتم بسیار پایین میباشد. جهت رفع این مشکل، به هر اتوماتای یادگیر اجازه داده شد در انتخاب اقدام خود، علاوه بر استفاده از

$$p(S) = \prod_{i \in S} p_i \prod_{i \in V-S} (1 - p_i) \quad (2)$$

تابع هدف مساله فروشنده دوره‌گرد احتمالی (فرمول ۱) در زمان  $O(n^2)$  قابل ارزیابی است (Jaillet [4]). اگر یک تور اولیه،  $\lambda = (1, 2, \dots, n)$  را در نظر بگیریم، در اینصورت طول متوسط آن از رابطه زیر قابل محاسبه است:

$$E[L_\lambda] = \sum_{i=1}^n \sum_{j=i+1}^n d_{ij} p_i p_j \prod_{k=i+1}^{j-1} (1 - p_k) + \sum_{i=1}^n \sum_{j=1}^{i-1} d_{ij} p_i p_j \prod_{k=i+1}^n (1 - p_k) \prod_{l=1}^{i-1} (1 - p_l) \quad (3)$$

رابطه فوق با در نظر گرفتن احتمال بازدید هر یک از کمانهای موجود در گراف نمونه مساله، و پرش از شهرهایی که نیاز به بازدید ندارند، بدست زمانی مورد استفاده قرار خواهد  $(i, j)$  می آید. بعنوان مثال کمان  $p_i \cdot p_j$  بایستی بازدید شده و باید با احتمال  $j$  و  $i$  گرفت که هر دو شهر حرکت کرده و از شهرهایی که در یک تور  $j$  به شهر  $i$  مستقیماً از شهر  $i+1, i+2, \dots, j-1$  اولیه بین این دو شهر قرار دارند، (پرش) نمائیم و یا بعبارتی شهرهای فوق بایستی بازدید شوند. احتمال

$$p_i p_j \prod_{k=i+1}^{j-1} (1 - p_k) \quad (\text{when } j \leq n)$$

می باشد. نوع خاصی از مساله فروشنده دوره‌گرد احتمالی که در آن احتمال بازدید همه شهرها برابر است، مساله فروشنده دوره‌گرد احتمالی متجانس<sup>۱۴</sup> نامیده می شود که در این مقاله به حل آن پرداخته شده است. برای مساله فروشنده دوره‌گرد احتمالی متجانس معادله ۳ بصورت زیر خواهد بود:

$$E[L_\lambda] = p^2 \sum_{r=0}^{n-2} (1-p) L_\lambda^{(r)} \quad (4)$$

که  $L_\lambda^{(r)} = \sum_{j=1}^n d(j, (j+1+r) \bmod n)$  می باشد. در این

رابطه  $L_\lambda^{(r)}$  به طول زیر تورهای  $\lambda_p^{(r)}$  اطلاق می شود که تعداد آنها برابر بزرگترین مقسوم علیه مشترک  $n$  و  $r+1$  می باشد<sup>۱۵</sup>. این زیر تورها از تور  $\lambda$  حاصل شده‌اند بطوریکه در آنها یک شهر بازدید شده و از  $r$  شهر که در یک تور اولیه بعد از آن قرار دارند، صرفنظر می کنیم (پرش از  $r$  شهر بعدی) و مجدداً  $r+1$  امین شهر (در صورت وجود) را بازدید کرده

<sup>14</sup> Homogeneous Probabilistic Traveling Salesman Problem

<sup>15</sup>  $\gcd(n, r+1)$

<sup>16</sup> Isomorphic

بردار احتمال اقدامها، از مقدار عکس فاصله بین دو گره (عکس فاصله بین گره فعال (اتوماتای یدگ پر فعال) و گره‌های متصل به این لیه نیز استفاده کند. آزمایشها نشان داد که استفاده از این مقدار برای انتخاب شهر بعدی، بهبود قابل ملاحظه‌ای در کارایی و نرخ همگرایی الگوریتم ایجاد میکند و باعث تولید جوابهای بهینه و یا خیلی نزدیک به جواب بهینه<sup>۱۷</sup> میشود. مقدار عکس فاصله بین دو گره  $j$  و  $i$  در گراف PTSP توسط تابع  $W^{-1}(j, i)$  نشان داده میشود. برای استفاده از این تابع در انتخاب اقدام اتوماتای یادگیر فعال، بردار احتمال اقدامهای اتوماتای یادگیر  $j$ ،  $P^j$  را بطور موقت طبق روابط زیر به بردار  $P'^j$  تغییر می دهیم. پس از انتخاب اقدام، بردار احتمال اقدامها مجدداً به مقدار قبلی خود  $P^j$  برگردانده می شود. این کار در ابتدای هر تکرار انجام میگردد. روابط زیر چگونگی محاسبه بردار  $P'^j$  را از بردار احتمال اقدام  $P^j$  نشان می دهد

بردار احتمال اقدام اتوماتای یادگیر  $j$ :  $P^j = [p_1^j, p_2^j, \dots, p_r^j]^T$   
 بردار احتمال اقدام تغییر یافته اتوماتای یادگیر  $j$ :

$$P'^j = \{p_i'^j | p_i^j = \frac{[p_i^j \times W^{-1}(j, i)]^\beta}{\sum_{i=1}^r [p_i^j \times W^{-1}(j, i)]^\beta} : i = 1, 2, \dots, r\}$$

در این رابطه  $P_i^j$ ، احتمال انتخاب اقدام  $i$  توسط اتوماتای یادگیر  $j$  می باشد و  $W^{-1}(j, i)$ ، عکس فاصله بین دو گره  $j$  و  $i$  در گراف PTSP بوده و  $\beta \geq 1$  اهمیت نسبی فاصله بین دو گره در انتخاب یک اقدام را مشخص میکند. بعبارتی دیگر احتمال انتخاب یالهایی (اقدامهایی) که دارای احتمال انتخاب بیشتر و طول یال کمتر باشند، بالاتر رفته و بالعکس. آزمایشها نشان داده اند که با اعمال تغییرات فوق در بردار احتمال اقدام و با توجه به در نظر گرفته شدن فاصله بین دو گره، نرخ همگرایی الگوریتم پیشنهادی بمیزان قابل ملاحظه‌ای افزایش می یابد در ضمن اینکه جوابهای نزدیک به بهینه‌ای تولید خواهد شد.

اکنون به توصیف الگوریتم پیشنهادی می پردازیم. در گام نخست اتوماتای یادگیر مبدا بصورت تصادفی از گراف DTSP بعنوان شهر آغازین تور انتخاب می شود. این اتوماتای یادگیر یکی از اقدامهای خود را طبق بردار احتمال اقدام تغییر یافته،  $P'^j$ ، انتخاب می کند. انجام این اقدام، اتوماتای یادگیر طرف دیگر لیه که متناظر با اقدام انتخاب شده میباشد را فعال می سازد. با توجه به اینکه در تور هامیلتونی هر شهر نبایسد بیش از یکبار ملاقات شود بایستی ترتیبی اتخاذ نمود تا هیچ شهری (گره‌ای) بیش از یکبار انتخاب نشود. برای این منظور اگر اتوماتای یادگیر اقدام  $k$  را از

لیست اقدامهای خود انتخاب کند، همزمان با آن اتوماتاهای یادگیر غیرفعال اقدام  $k$  را در لیست اقدامهای خود غیر فعال<sup>۱۸</sup> می سازند (اما حذف نمی کنند). در ابتدای هر تکرار تمام اقدامهای غیر فعال شده مجدداً فعال خواهند شد. اتوماتای یادگیر فعال شده با استفاده از بردار احتمال اقدام تغییر یافته‌ی خود، اتوماتای یادگیر طرف دیگر لیه انتخاب شده را فعال می سازد. فرایند انتخاب اقدام و فعال سازی اتوماتای یادگیر متناظر با اقدام انتخاب شده تا ملاقات همه گره‌های (شهرهای) موجود در گراف PTSP و برگشت به شهر آغازین و یا بنابر برخی دلایل که امکان انتخاب اقدام بعدی برای اتوماتای یادگیر فعال وجود نداشته باشد (امکان ایجاد تور هامیلتونی با توجه به شهرهایی که قبلاً انتخاب شده‌اند و یا ساختار گراف نمونه مساله داده شده، وجود نداشته باشد)، تکرار می شود.

پس از پیدا کردن یک تور، طول متوسط آن طبق رابطه‌ی ۴ که در بخش قبل به آن اشاره شد محاسبه می شود و با طول متوسط بهترین توری که تا بحال توسط الگوریتم ایجاد شده است مقایسه میگردد. بر طبق نتیجه مقایسه، بردار احتمال اقدام اتوماتاهای یادگیر DLA بروزرسانی خواهد شد. نحوه‌ی بروزرسانی بردار احتمال اقدام بدینصورت است که اگر طول متوسط تور ایجاد شده کوچکتر و یا مساوی طول متوسط بهترین توری که تا بحال ایجاد شده است باشد، همه اتوماتاهای یادگیر DLA، اقدام انتخابی خود را طبق الگوریتم یادگیری  $L_{R-I}$ ، پاداش می دهند. برای روشن شدن بیشتر این مطلب الگوریتم  $L_{R-I}$  ذکر شده در بخش اتوماتای یادگیر را مجدداً یادآوری می نماییم. بعنوان مثال اگر طول متوسط تور ایجاد شده در یک تکرار کوچکتر و یا مساوی طول متوسط بهترین تور ایجاد شده تاکنون باشد، و در این تکرار (t) اتوماتای یادگیر  $j$  از مجموعه اقدامهای مجاز خود اقدام  $i$  را انتخاب کرده باشد، احتمال انتخاب اقدام  $i$  طبق رابطه‌ی زیر افزایش خواهد یافت:

$$p_i(t+1) = p_i(t) + a[1 - p_i(t)]$$
  
 و احتمال انتخاب سایر اقدامهای اتوماتای یادگیر  $j$  بصورت زیر کاهش خواهد یافت:

$$p_k(t+1) = (1-a)p_k(t) \quad k \neq i \quad k = 1, 2, \dots, r$$
  
 در رابطه‌ی بالا پارامتر  $a$  نرخ یادگیری<sup>۱۹</sup>،  $r$  تعداد اقدامهای اتوماتای یادگیر  $j$  می باشد. فرایند ایجاد تور تا رسیدن به شرط پایانی ادامه مییابد. آخرین تور هامیلتونی ایجاد شده توسط الگوریتم جواب تولید شده توسط الگوریتم می باشد که "یک تور اولیه" میباشد که در بین همه تورهای ایجاد شده توسط الگوریتم دارای کمترین طول متوسط است.

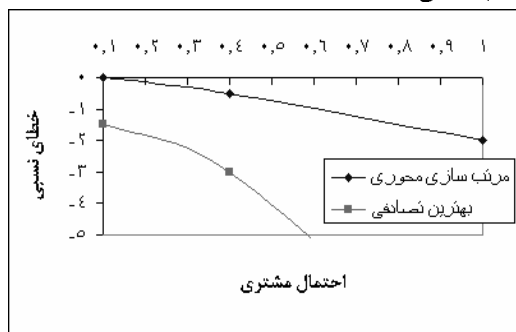
## ۶- ارزیابی الگوریتم پیشنهادی

<sup>18</sup> Disable

<sup>19</sup> Learning Rate

<sup>17</sup> Optimal

شهرها و محور عمودی خطای نسبی که از رابطه  $\frac{E[PTSPDLA] - E[heuristic]}{E[PTSPDLA]}$  محاسبه می شود را نشان می دهد که در این رابطه  $E[PTSPDLA]$  طول متوسط تور ایجاد شده توسط الگوریتم پیشنهادی و  $E[heuristic]$  طول تور ایجاد شده توسط هر یک از دو الگوریتم بهترین تصادفی و مرتب سازی محوری می باشد. هر نقطه از شکل میانگین نتایج بدست آمده برای ۱۶ نمونه مساله فروشنده دوره گرد احتمالی می باشد. اولین نکته قابل مشاهده در این شکل، بالاتر بودن کارایی الگوریتم پیشنهادی از هر دو الگوریتم مورد مقایسه برای همه احتمالات می باشد. مشاهده میشود که الگوریتم بهترین تصادفی در مقایسه با الگوریتم پیشنهادی و الگوریتم مرتب سازی محوری دارای نتایج ضعیفتری می باشد. دومین نکته قابل توجه، نتایج تقریباً یکسان الگوریتم پیشنهادی و مرتب سازی محوری در احتمالات کوچک (تقریباً برابر ۰,۱) می باشد. نتیجه فوق مؤید این نکته است که تورهای ایجاد شده توسط الگوریتم مرتب سازی محوری برای مقادیر کوچک احتمالات بازدید نزدیک به بهینه می باشند.



شک ۳۴: کارایی نسبی الگوریتم پیشنهادی<sup>۲۱</sup> در مقایسه با مرتب سازی محوری و بهترین تصادفی

کارایی مطلق الگوریتم پیشنهادی: با توجه به اینکه جواب بهینه هیچیک از نمونه مساله های فروشنده دوره گرد احتمالی در نظر گرفته شده در این مقاله، مشخص نمی باشد بنابراین کارایی مطلق الگوریتم پیشنهادی با استفاده از حد پایین جواب بهینه نمونه مساله مورد ارزیابی قرار میگردد. میتوان با استفاده از حد پایین جواب بهینه، حد بالای خطای انجام گرفته توسط الگوریتم را بدست آورد. عبارتی اگر LB حد پایین جواب بهینه نمونه مساله و  $E[L_{\lambda}^*]$  جواب بهینه آن باشد، طبق تعریف داریم:

$$E[L_{\lambda}^*] \geq LB \quad (5)$$

و اگر جواب بدست آمده از الگوریتم برابر  $E[L_{\lambda}]$  باشد، در اینصورت برای خطای نسبی الگوریتم، نامعادله زیر برقرار خواهد بود:

برای تولید نمونه های فروشنده دوره گرد احتمالی متجانس (احتمال بازدید همه شهرها برابر است) میتوان از نمونه های مساله فروشنده دوره گرد استفاده کرد. برای این منظور به هر یک از شهرها یک احتمال بازدید نسبت داده میشود. با توجه به اینکه مساله مورد مطالعه در این مقاله از نوع متجانس میباشد احتمال بازدید برای تمام شهرها یکسان در نظر گرفته شده است. آزمایشها برای مقادیر احتمال بازدید  $p = 0.1, 0.2, \dots, 0.9$  انجام گرفته شده است. الگوریتم پیشنهادی بر روی ۱۶ نمونه مساله فروشنده دوره گرد متقارن، اجرا و نتایج آن با نتایج دو الگوریتم مکاشفه ای به نامهای مرتب سازی محوری<sup>۲۰</sup> و "بهترین تصادفی" که برای حل این مساله گزارش شده اند، مقایسه گردیده است. ۵ نمونه از ۱۶ نمونه مذکور از کتابخانه TSPLIB [18] انتخاب شده و شامل نمونه های eil101، kroa100، eil76، berlin52، eil51 و eil101 می باشند که تعداد شهرهای آنها بین ۵۱ تا ۱۰۱ شهر است. ۱۱ نمونه باقیمانده توسط کد تولید کننده نمونه تصادفی DIMACS Challenge [19] تولید شده اند که متقارن و دارای ۲۵ تا ۱۵۰ شهر می باشند. برای انجام آزمایشها از یک کامپیوتر خانگی AMD 1300 MHz با ۱۲۸ مگابایت حافظه اصلی استفاده شده است.

الگوریتم بهترین تصادفی ابتدا تعدادی تور بصورت تصادفی که در آنها شهرهای مورد بازدید در تور بصورت تصادفی انتخاب می شوند، ایجاد کرده و سپس از بین تورهای ایجاد شده، توری که دارای کمترین طول متوسط می باشد را انتخاب می کند. الگوریتم مرتب سازی محوری ابتدا شهرهای موجود در نمونه مساله را براساس زاویه ایجاد شده بین آنها و مرکز جرم توزیع فضایی شهرها<sup>۲۰</sup>، مرتب می نماید و سپس طبق نتیجه مرتب سازی شهرهای واقع در تور انتخاب می شوند. مختصات مرکز جرم تعدادی شهر را میتوان با میانگین گیری از مختصات شهرها محاسبه کرد.

یک تور اولیه ایجاد شده توسط الگوریتم مرتب سازی محوری وابسته به احتمال بازدید شهرها نبوده و بهمین دلیل احتمال بازدید شهرها در نتیجه الگوریتم دخیل نمی باشد. . برغم سادگی مرتب سازی محوری، تورهایی که با این الگوریتم ایجاد می شوند زمانیکه احتمال بازدید شهرها پایین باشد، جویهای نزدیک به بهینه تولید میکند [2]. ولی این الگوریتم برای نمونه های نامتقارن PTSP و یا نمونه هایی که مختصات شهرها مشخص نیست، قابل استفاده نمی باشد که یکی از نقاط ضعف این الگوریتم میباشد.

در شکل ۳ کارایی نسبی الگوریتم پیشنهادی در مقایسه با الگوریتمهای مرتب سازی محوری و بهترین تصادفی با توجه به مقدار مختلف احتمال بازدید نشان داده شده است. در این شکل محور افقی احتمال بازدید

<sup>21</sup> PTSPDLA

<sup>20</sup> Center of mass' of the City spatial distribution

صورتیکه احتمال بازدید شهرها بالا باشد، جوابهای با خطای نسبی پایین تولید می کنند.

## مراجع

- [1] Bertsimas, D. J., Probabilistic Combinatorial Optimization Problems. PhD thesis, MIT, Cambridge, MA, 1988.
- [2] Bertsimas, D. J. and Howell, L. "Further results on the probabilistic traveling salesman problem", European Journal of Operational Research, 65:68-95, 1993.
- [3] Bertsimas, D. J., Jaillet, P. and Odoni, A. "A priori optimization. Operations Research", 38:1019-1033, 1990.
- [4] Jaillet, P. "Probabilistic Traveling Salesman Problems", PhD thesis, MIT, Cambridge, MA, 1985.
- [5] J\_ez\_equel, A. "Probabilistic Vehicle Routing Problems", Master's thesis, MIT, Cambridge, MA, 1985.
- [6] Rossi, F. A. and Gavioli, I. "Aspects of Heuristic Methods in the Probabilistic Traveling Salesman Problem", pages 214-227. World Scientific, Singapore, 1987.
- [7] Lakshmivarahan, S. Learning Algorithms: Theory and Applications, New York: Springer-verlag, 1981.
- [8] Meybodi, M. R. and S. Lakshmivarahan, S. "On a Class of Learning Algorithms which have Symetric Behavior under Success and Failer", pp. 145-155. Lecture Notes in Statistics, Berlin: Springer-Verlag, 1984.
- [9] Mars, P., Chen, J. R. and Nambir, R. "Learning Algorithms: Theory and Applications in Signal Processing, Control, and Communication", CRC Press Inc., 1996.
- [10] Narendra, K. S. and Thathachar, K. S. "Learning Automata: An Introduction", New York: Prentice-Hall, 1989.
- [11] Meybodi, M. R. and Beigy, H. "Solving Stochastic Shortest Path Problem Using Distributed Learning Automata", Proceedings 6<sup>th</sup> Annual CSI Computer Conference, University of Isfahan's Computer Engineering Department, 2001.
- [12] Beigy, H. and Meybodi, M. R. "A New Distributed Learning Automata for Solving Stochastic Shortest Path Problem", Proceedings of the Sixth International Joint Conference on Information Science, Durham, USA, pp. 339-343, 2002.
- [13] Alipour, M. and Meybodi, M. R. "Solving Traveling salesman Problem Using Distributed Learning Automata", Proceedings of 10th Annual CSI Computer Conference, Computer Engineering Department, Iran Telecommunication Research Center, Tehran, Iran, pp. 271-280, Feb. 2005.
- [14] Lakshmivarahan, S. "Learning Algorithms: Theory and Applications", New York: Springer-verlag, 1981.
- [15] Meybodi, M. R. and Lakshmivarahan, S. "On a Class of Learning Algorithms which have Symetric Behavior under Success and Failer", pp. 145-155. Lecture Notes in Statistics, Berlin: Springer-Verlag, 1984.
- [16] Mars, P., Chen, J. R. and Nambir, R. "Learning Algorithms: Theory and Applications in Signal Processing, Control, and Communication", CRC Press Inc., 1996.
- [17] Narendra, K. S. and Thathachar, K. S. "Learning Automata: An Introduction", New York: Prentice-Hall, 1989.
- [18] <http://www.iwr.uniheidelberg.de/groups/comopt/software/TSPLIB>
- [19] <http://www.research.att.com/~dsj/chtsp/download.html>
- [20] Alipour, A. and Meybodi, M. R. "Solving Dynamic Traveling Salesman Problem Using Responsive Distributed Learning Automata", Proceedings of IKT2005 Tehran, Iran, May 24-26, 2005

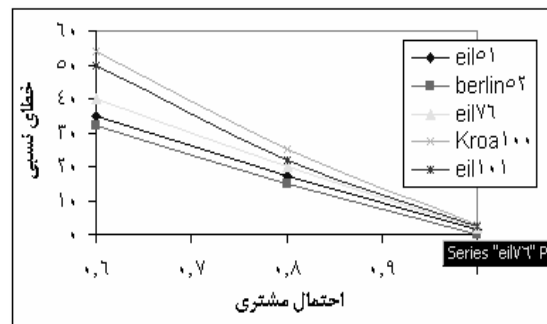
$$\frac{E[L_{\lambda}] - E[L_{\lambda^*}]}{E[L_{\lambda^*}]} \leq \frac{E[L_{\lambda}] - LB}{LB} \quad (6)$$

در ادامه تکنیکی را برای ارزیابی حد پایین جواب بهینه مساله فروشنده دوره گرد احتمالی معرفی نموده و از آن برای ارزیابی کارایی مطلق الگوریتم پیشنهادی استفاده خواهیم کرد.

برای محاسبه حد پایین جواب بهینه مساله فروشنده دوره گرد احتمالی متجانس اگر جواب بهینه نمونه مساله فروشنده دوره گرد ( $L_{TSP}$ ) بکار برده شده مشخص باشد، میتوان از رابطه زیر استفاده کرد [۲].

$$LB = pL_{TSP}(1 - (1 - p)^{n-1}) \quad (7)$$

با قرار دادن مقدار LB از رابطه فوق در سمت راست نا معادله ۶ حد بالای خطای نسبی الگوریتم پیشنهادی بدست می آید. کارایی مطلق الگوریتم پیشنهادی که برای تعدادی از نمونه های موجود در کتابخانه TSPLIB مورد ارزیابی گرفته است در شکل ۴ نشان داده شده است. در این شکل محور افقی احتمال بازدید شهر و محور عمودی مقدار  $[E[L_{\lambda}(PTSPDLA)] - LB] / E[L_{\lambda}(PTSPDLA)] * 100$  را نشان می دهد. نکته قابل توجه در این شکل بالا رفتن حد بالای خطای نسبی با کاهش احتمال بازدید شهرها می باشد که دلیل این امر کاهش دقت حد پایین جواب بهینه در نظر گرفته شده می باشد. برای مثال مشاهده میشود که الگوریتم پیشنهادی جوابی در حدود ۱۵٪ جواب بهینه را در احتمال بازدید برابر ۰٫۹ پیدا می کند.



شکل ۴: حد بالای درصد خطای نسبی الگوریتم پیشنهادی برای ۵ نمونه از کتابخانه TSPLIB

## ۷- نتیجه گیری

در این مقاله الگوریتمی براساس اتوماتای یادگیر توزیع شده برای حل مساله فروشنده دوره گرد احتمالی متجانس ارائه گردید و با دو الگوریتم مرتب سازی محوری و الگوریتم بهترین تصادفی مقایسه شد. بر طبق نتایج بدست آمده، الگوریتم پیشنهادی در مقایسه با این دو الگوریتم از کارایی بالاتری برخوردار است. نشان داده شد که الگوریتم پیشنهادی در