حل مساله فروشنده دوره‌گرد احتمالی توسط اتوماتای یادگیر توزیع شده

محمد علیپور **محمدرضا میبدی**

آزمایشگاه سیستمهای نرم افزاری
دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی امیرکبیر
تهران ایران

چکیده

در مساله فروشنده دوره‌گرد احتمالی[1] 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[2] 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[3] الگوریتمی برای حل مساله فروشنده دوره‌گرد احتمالی ارائه می شود. الگوریتم پیشنهادی با دو الگوریتم مکاشفه ای بهترین تصادفی[4] و مرتب سازی محوری[5]، 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[6] (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 یکبا بازدید نماید و سپس به شهر آغازین بر گردد. مساله فروشنده دوره‌گرد احتمالی[7] (PTSP) یکی از انواع مساله TSP می باشدکه در آن هر شهر با احتمال معینی که مستقل از بازدید سایر شهرها می باشد مورد بازدید قرار می 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[8] گفته میشود. مساله فروشنده دوره گرد احتمالی از مسائل NP-hard بوده [۳, ۱] و برای اولین بار در رساله‌ی دکترای آقای جیلت[9] معرفی شده است [4].

---

[1] Probabilistic Traveling Salesman Problem
[2] Expected length
[3] Dirstributed Learning Automata
[4] Random Best
[5] Radial Sort
[6] TSP
[7] PTSP
[8] A priori tour
[9] Jaillet

استراژی های 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 بروز رسانی، شهرهایی که بازدید نخواهند 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 بترییبی 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 استراتژیی، استراتژی پرش[10] نامیده می شود.

تاکنون راهکارهای الگوریتمی و مکاشفه ای مختلفی برای یافتن جواب بهینه PTSP گزارش شده است [5] . رسی گاویولی[11] از قوانین مکاشفه ای به نامهای نزدیکترین همسایه و معیار پس انداز[12] استفاده کردند [6]. برتیماس[13] و همکارانش در [3] و برتیماس و هاول[14] 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 کرده‌اند. اغلب الگوریتمهای ارائه شده برای PTSP از تطبیق الگوریتم TSP برای PTSP ایجاد شده و یا حتی همان الگوریتم اصلیTSP می باشند که در برخی موارد جوابهای PTSP خوبی تولید می کنند. اتوماتای یادگیر توزیع شده قبلا برای حل مساله فروشنده د دوره گرد و فروشنده دوره گرد پویا مورد استفاده قرار گرفته است[20][13] .

مساله فروشنده دوره‌گرد احتمالی دارای کاربردهای فراوانی می باشد که از آن جمله میتوان به کاربردها ی آن در توزیع کالا اشاره کرد. یک فروشنده به منظور توز یع یکئ کالا هر روز مجموعه‌ای از مشتریان را با ترتیب مشخصی مورد بازد ید قرار م میدهد. (ترتیب معین شده در یک مسیر اولیه که روزانه از آن پیروی می شود) ، اما هر مشتری همیشه نیاز به بازدید نداشته و امکان بهینه سازی و تغییر مسیر اولیه بصورت روزانه وجود ندارد. در اینصورت فروشنده روزانه بر طبق یک مسیر از پیش تعیین شده (مسیر اولیه) حرکت نموده و از مشتریانی که نیاز به بازدید ندارند چشم پوشی می کند.

در این مقاله یک الگوریتم مبتنی بر اتوماتای یادگیر توزیع شده برای حل مساله فروشنده دوره گرد احتمالی ارایه 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 محدوده‌ی 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 تئوریکی می باشد معرفی و کارایی مطلق الگوریتم پیشنهادی با استفاده از این روش ارزیابی میگردد.

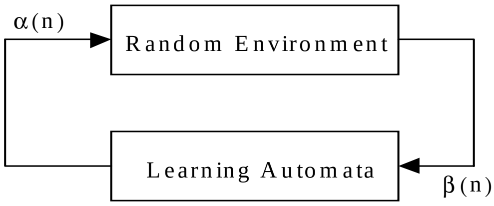
در ادامه مقاله، ابتدا در بخش ۲ اتوماتاهای یادگیر و در بخش ۳ اتوماتاهای یادگیر توزیع معرفی میگردد. در بخش ۴ تابع هدف برای مساله فروشنده دوره‌گرد احتمالی را تعریف میگردد. در بخش ۵ الگوریتم پیشنهادی ارایه میشود و در بخش ۶ نتایج آزمایشها ارایه میشود. بخش ۷ نتیجه گیری میباشد.

۲- اتوماتای یادگیر[15]

اتوماتای یادگیر 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 یادگیر داده می شود. اتوماتای یادگیر از این پاسخ استفاده نموده وعمل خود را برای مرحله بعد انتخاب می کند. شکل ۱ ارتباط بین اتوماتای یادگیر و محیط را نشان می دهد.

---

[10] Skipping strategy
[11] Rossi-Gavioli
[12] Savings
[13] Bertsimas
[14] Howell
[15] Learning Automata



شکل ۱: ارتباط بین اتوماتای یادگیر و محیط

محیط[16]: محیط را می توان توسط سه تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ مجموعه ورودیها، $\beta \equiv \{\beta_1, \beta_2, ..., \beta_m\}$ مجموعه خروجیها و $c \equiv \{c_1, c_2, ..., c_r\}$ مجموعه احتمالهای جریمه می باشد. هر گاه β مجموعه دو عضوی باشد ، محیط از نوع P می باشد. در چنین محیطی $\beta_1 = 1$ به عنوان جریمه و $\beta_2 = 0$ به عنوان پاداش در نظر گرفته می شود. در محیط از نوع Q ، $\beta(n)$ می تواند به طور گسسته یک مقدار از مقادیر محدود در فاصله $[0,1]$ و در محیط از نوع S، $\beta(n)$ متغیر تصادفی در فاصله $[0,1]$ است. c_i احتمال اینکه عمل α_i نتیجه نامطلوب داشته باشد می باشد. در محیط ایستا[17] مقادیر c_i بدون تغییر می مانند ، حال آنکه در محیط غیر ایستا[18] این مقادیر در طی زمان تغییر می کنند. اتوماتاهای یادگیربه دو گروه با ساختار ثابت و با ساختار متغیر تقسیم بندی میگردند. در ادامه به شرح مختصری درباره‌ی اتوماتاهای یادگیر با ساختار متغییر که در این مقاله از آنها استفاده شده است، می پردازیم.

اتوماتاهای یادگیر با ساختار متغیر[19]: اتوماتاهای یادگیر با ساختار متغیر توسط ۴ تائی $\{\alpha, \beta, p, T\}$ نشان داده می شوند که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \cdots, \alpha_r\}$ مجموعه عملهای اتوماتا، $\beta \equiv \{\beta_1, \beta_2, \cdots, \beta_m\}$ مجموعه ورودیهای اتوماتا، $p \equiv \{p_1, p_2, \cdots, p_r\}$ بردار احتمال انتخاب هر یک از عملها، و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می باشد. در این نوع از اتوماتاها، اگر عمل α_i در مرحله n ام انتخاب شود و پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ افزایش یافته و سایر احتمالها کاهش می یابند. و برای پاسخ نامطلوب احتمال $p_i(n)$ کاهش یافته و سایر احتمالها افزایش می یابند. در هر حال، تغییرات به گونه ای صورت می گیرد تا حاصل جمع $p_i(n)$ ها همواره ثابت و مساوی یک باقی بماند. الگوریتم زیر یک نمونه از الگوریتمهای یادگیری خطی برا ی اتوماتای یادگ یر با ساختار متغ ی یر م یباشد.

الف- پاسخ مطلوب

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)]$$
$$p_j(n+1) = (1-a)\,p_j(n) \qquad j \neq i \quad \forall j$$

ب- پاسخ نامطلوب

$$p_i(n+1) = (1-b)\,p_i(n)$$
$$p_j(n+1) = \frac{b}{r-1} + (1-b)\,p_j(n) \qquad j \neq i \quad \forall j$$

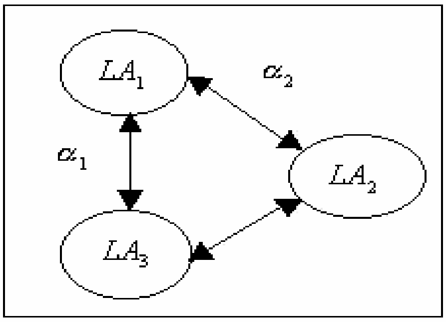
---

16 Environment
17 Stationary
18 Non-Stationary
19 Variable Learning Automata

در روابط فوق، a پارامتر پاداش و b پارامتر جریمه می باشد. با توجه به مقادیر a و b سه حالت را می توان در نظر گرفت. زمانیکه a و b با هم برابر باشند، الگوریتم ر، L_{RP} [20] می نامیم. زمانیکه b از a خیلی کوچکتر باشد، الگوریتم را $L_{R\varepsilon P}$ [21] می نامیم. زمانیکه b مساوی صفر باشد، الگوریتم را L_{RI} [22] می نامیم. برای اطلاعات بیشتر در باره اتوماتاهای یادگیر می توان به [14],[15],[16],[17] مراجعه کرد.

۳- اتوماتای یادگیر توزیع شده[۲۳]

اتوماتای یادگیر توزیع شده (DLA) ، شبکه‌ای از اتوماتاهای یادگیر است که برای حل یک مساله با یکدیگر همکاری می نمایند [2]. تعداد اقدامهای یک اتوماتا در DLA برابر تعداد اتوماتاهای یادگ یر متصل به این اتوماتای یادگیر می باشد. انتخاب یک اقدام توسط یک اتوماتا ی یادگ یر در شبکه، اتوماتای یادگ یر متناظر با این اقدام را فعال می سازد. بعنوان مثال در شکل ۲ هر اتوماتا یادگ یر دارای دو اقدام می باشد. انتخاب اقدام α_2 توسط LA_1، اتوماتا یادگیر LA_3 را فعال خواهد کرد. اتوماتای یادگیر فعال شده (LA_3) سپس یکی از اقدامهای خود را انتخاب می کند که در نتیجه آن یکی از اتوماتاهای یادگ یر متصل به آن اتوماتا ی یادگ یر که متناظر با اقدام انتخاب شده می باشد فعال می شود. در هر زمان فقط یک اتوماتای یادگیر در شبکه فعال میباشد. بطور رسمی DLA را میتوان توسط گراف $DLA=(V,E)$ که $V=\{LA_1,LA_2,\ldots,LA_n\}$ مجموعه اتوماتاهای یادگیر و n تعداد اتوماتاهای یادگیر در DLA و $E\subset V\times V$ مجموعه لبه‌های گراف می باشد، تعریف کرد. لبه (i,j) اقدام j اتوماتا ی یادگیر LA_i را نشان می دهد. LA_j زمانی فعال خواهد شد که اقدام j اتاماتای یادگیر LA_i انتخاب شود. تعداد اقدامهای اتوماتای یادگیر LA_k ($k=1,2,\ldots,n$) برابر درجه‌ی خروجی گره متناظر با اتوماتای یادگیر LA_k می باشد. برای اطلاعات بیشتر در باره اتوماتاهای یادگیر توزیع شده میتوان به مراجع [11] و [12] مراجعه نمود.



شکل ۲: اتوماتای یادگیر توزیع شده (DLA) با ۳ اتوماتا ی یادگیر

۴- تابع هدف مساله فروشنده دوره‌گرد احتمالی

برای تشریح تابع هدف مساله فروشنده دوره‌گرد احتمالی، گراف کاملی با n گره (شهر) را که گره‌های آن از مجموعه $V=\{i=1,2,\ldots,n\}$ 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 مینیمایز کردن تابع $E[L_\lambda]$ که به قرار زیر تعریف شده است میباشد.

[20] Linear Reward Pealty
[21] Linear Reward Epsilon Penalty
[22] Linear Reward Inaction
[23] Distributed Learning Automata

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

که S زیر مجموعه‌ای از گره‌های 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 λ) شامل همه‌ی گره‌های مجموعه V بوده و دارای طول متوسط زیر می باشد:

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

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

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

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 می آید. بعنوان مثال کمان (i, j) زمانی مورد استفاده قرار خواهد گرفت که هر دو شهر i و j بایستی بازدید شده و باید با احتمال $p_i.p_j$ مستقیما از شهر i به شهر j حرکت کرده و از شهرهایی که در یک تور اولیه بین این دو شهر قرار دارند، $i+1, i+2,..., j-1$ ، صرفنظر (پرش) نمائیم و یا بعبارتی شهرهای فوق نبایستی بازدید شوند. احتمال عدم بازدید این شهرها برابر $p_i p_j \prod_{k=i+1}^{j-1}(1-p_k) (when \quad j \le n)$ می باشد. نوع خاصی از مساله فروشنده دوره‌گرد احتمالی که در آن احتمال بازدید همه شهرها برابر است، مساله فروشنده دوره‌گرد احتمالی متجانس[۲۴] 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:

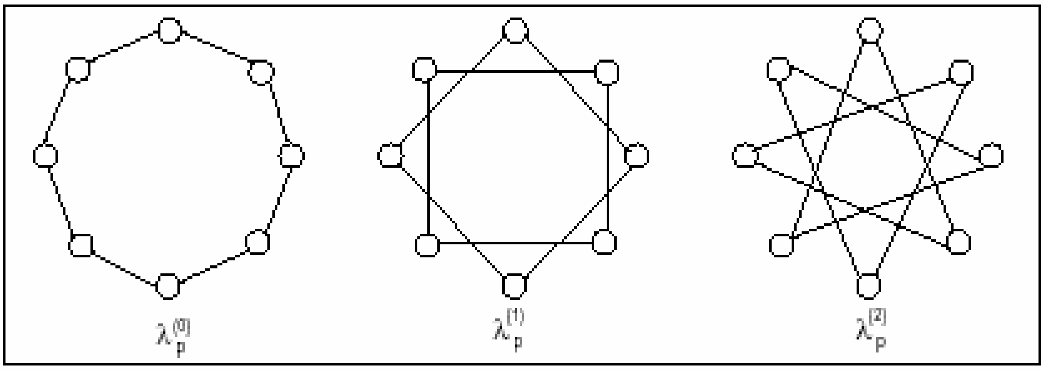
$$E[L_\lambda] = p^2 \sum_{r=0}^{n-2} (1-p) L_\lambda^{(r)} \qquad (4)$$

که $L_\lambda^{(r)} = \sum_{j=1}^{n} d(j, (j+1+r) \quad \text{mod} \quad n)$ 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[۲۵]. این زیر تورها از تور λ 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 مجددا $r+1$ امین شهر (در صورت وجود) را بازدید کرده و بهمین صورت ادامه می دهیم. برای روشن شدن بیشتر مطلب شکل ۳ 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 شده‌اند.

---

[24] Homogeneous Probabilistic Traveling Salesman Problem

[25] $\gcd(n, r+1)$



شکل ۳: 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[۲۶] با گراف مساله PTSP میباشد ایجاد می شود. در این شبکه هر گره (معادل یک شهر در مساله)، یک اتوماتا ی یادگیر با ساختار متغیر بوده و هر لبه‌ی خروجی این گره(جاده ای که توسط آن از این شهر میتوان به شهری دیگری رفت) یکی از اقدامهای آن اتوماتا ی یادگیر می باشد. تعداد اقدامهای یک اتوماتا ی 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 که اگر یک اتوماتای یادگیر فعال فقط از بردار احتمال اقدامها برای انتخاب اقدام خود استفاده کند، جوابهای بدست آمده تقریبا 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 بردار احتمال اقدامها، از مقدار عکس فاصله بین دو گره (عکس فاصله بین گره فعال (اتوماتای یدگ یر فعال) و گره‌های متصل به این لبه 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[۲۷] 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 اتوماتا ی یادگیر j، P^j را بطور موقت طبق روابط زیر به بردار P'^j تغییر می دهیم. پس از انتخاب اقدام، بردار احتمال اقدامها مجددا به مقدار قبلی خود P^j برگردانده می شود. این کار در ابتدای هر تکرار انجام میگیرد. روابط زیر چگونگی محاسبه بردار P'^j را از بردار احتمال اقدام P^j نشان می دهد

بردار احتمال اقدام اتوماتا ی یادگ یر j : $P^j = [p_1{}^j, p_2{}^j, \ldots, p_r{}^j]^T$

بردار احتمال اقدام تغییر یافته اتوماتا ی یادگ یر j : $P'^j = \{p_i'^j \mid p_i'^j = \dfrac{[p_i{}^j \times W^{-1}(j,i)]^\beta}{\sum_{i=1}^{r}[p_i{}^j \times W^{-1}(j,i)]^\beta} \quad : \quad i = 1,2,\ldots,r\}$

---

[26] Isomorphic
[27] Optimal

در این رابطه P_i^j، احتمال انتخاب اقدام i توسط اتوماتا ی 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 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 لبه 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 اتوماتا ی 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[۲۸] می سازند (اما حذف نمی کنند). در ابتدای هر تکرار تمام اقدامهای غیر فعال شده مجددا فعال خواهند شد. اتوماتای یادگیر فعال شده با استفاده از بردار احتمال اقدام تغییر یافته‌ی خود، اتوماتای یادگیر طرف دیگر لبه انتخاب شده را فعال می سازد. فرآیند 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 اتوماتا ی یادگ یر فعال وجود نداشته باشد (امکان ایجاد تور هامیلتونی با توجه به شهرهایی که قبلا 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 تابحال ایجاد شده است باشد، همه اتوماتاهای یادگیرDLA، 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 مجددا یادآوری می نمائیم. 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:

$$p_k(t+1) = (1-a)\,p_k(t) \qquad k \neq i \quad k = 1,2,\ldots,r$$

در رابطه‌ی بالا پارامتر a نرخ یادگیری[۲۹]، r تعداد اقدامهای اتوماتای یادگیر j می باشد. فرآیند ایجاد تور تا رسیدن به شرط پایانی ادامه میبابد. 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. الگوریتم پیشنهادی با ذکر جزییات بیشتر در شکل ۴ آمده است. این الگوریتم را میتوان بشرح زیر توصیف کرد.

الگوریتم شکل ۴ ابتدا بردار احتمال اقدامهای اتوماتاهای یادگیر در DLA مقدار دهی اولیه و سپس مراحل ۱ تا ۴ که در زیر آمده است را اجرا میکنند. احتمال انتخاب اقدامها مساوی و برابر با 1/k در نظر گرفته میشود. k تعداد اعمال اتوماتای یادگیر میباشد.

---
[28] Disable
[29] Learning Rate

```
Procedure PTSP
Begin
  initialize the probability vector of each automaton
  Repeat
    //Phase 1
    InitialCity := Generate a random number between 1 and n (choosing the intial City of
    the tour randomly)
    CurrentCity := InitialCity
    disable action 'CurrentCity' of all Unactivated LAs
    //Phase 2
    for i := 1 to n
      if i < n then
        //choose next node as a sample realization of active LA modified action probability
      vector, P'^j
        NextCity := GetNextCity()
        //if there's no edge between the current City and any unselected City, the  current
        path could not be hamilton cycle
        disable action 'NextCity' of all Unactivated LAs
        if i = n-1 then
          Enable action 'InitialCity' of LA NextCity
        end if
      else
          NextCity := InitialCity
          //if there's no edge between current City and intial City, so current path could not
        be hamilton cycle
      end if
      CurrentCity := NextCity
    next i
    //Phase 3
    compute the current tour expected length
    if CurrentTourExpectedLength < BestTourExpectedLength then
      //Reward selected actions of all LAs along the tour according to L_{R-I} learning
      algorithm
      for j := 1 to n
        // <i> is the selected action of automata <j>
          p_i(t + 1) = p_i(t) + a[1 - p_i(t)]
          p_k(t+1) = (1-a) p_k(t)      k ≠ i    k = 1,2,..., r
      next j
    end if
    //Phase4
    enable all the disabled actions of LAs
  until (stop condition)
end procedure
```

شکل ۴: الگوریتم حل PTSP با استفاده DLA

مرحله۱: یکی از گره‌های گراف بصورت تصادفی بعنوان شهر آغازین تور (InitialCity) انتخاب شده و سپس اتوماتای یادگیر متناظر با این گره(CurrentCity) فعال میگردد این اقدام (CurrentCity) در اتوماتاهای یادگیر دیگر غیر فعال میگردد.

مرحله ۲: در هر تکرار از این مرحله یکی از اقدامهای مجاز [30] اتوماتای یادگیر فعال انتخاب می شود. برای این منظور از تابع GetNextCity() استفاد شده است (شکل۵). این تابع با استفاده از بردار احتمال P'^{j}، یکی از اقدامهای مجاز اتوماتای یادگیر فعال (شهر بعدی) را انتخاب می کند. اگر این تابع قادر به تعیین شهر بعدی برای حرکت نباشد در اینصورت مسیر پیمایش شده تا این مرحله، قابل تبدیل به یک تور هامیلتونی نبوده و از آن صرفنظر میگردد و اقدام به ایجاد یک تور جدید میشود. اگر الگوریتم قادر به برگشت به شهر آغازین بشود در این صورت یک تور هامیلتونی توسط الگوریتم پیدا شده که در این صورت به مرحله ۳ میرویم و در غیر اینصورت اگر اتوماتای یادگیر فعال(شهر جاری) امکان انتخاب اقدام مجاز نباشد و یا از آخرین اتوماتای یادگیر فعال امکان بازگشت به گره شروع وجود نداشته باشد، در اینصورت الگوریتم موفق به ایجاد تور هامیلتونی(تور اولیه) نشده که در این صورت به مرحله ۴ می رویم.

مرحله ۳: در این مرحله طول تور ایجاد شده محاسبه و با طول بهترین توری که تابحال بدست آمده است مقایسه میگردد و در صورتیکه طول تور ایجاد شده کوچکتر و یا مساوی طول بهترین توری که تا بحال بدست آمده است باشد، به اقدامهای انتخاب شده‌ی اتوماتاهای یادگیر DLA طبق الگوریتم یادگیری L_{R-I} پاداش داده می شود.

مرحله۴: اقدام‌های غیر فعال شده در حین اجرای مرحله ۲، مجددا فعال شده و در صورتیکه شرط پایان الگوریتم برقرار نباشد مبادرت به ایجاد توری جدید مینماییم. اگر تعداد تورهای ایجاد شده از یک تعداد معینی فراتر رفته باشد و یا احتمال مسیر[31] که عبارتست از حاصلضرب احتمال لبه‌های موجود در مدار بهینه، از یک حد مشخصی (برای مثال ۰,۹) بیشتر شده باشد، در اینصورت الگوریتم خاتمه میابد.

```
Function GetNextCity (j)
  // j is the current automata
  //modify  the action probability vector of  <j> ( P^j ) as:
```

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

```
  Choose an action as a sample realization of j's modified action
  probability vector,  P'^j
  Restore the previous value of  P^j
End function
```

شکل ۵: تابع GetNextCity برای انتخاب یکی از اقدامهای مجاز اتوماتای یادگ یر جاری

۶- ارزیابی الگوریتم پیشنهادی

برای تولید نمونه‌های فروشنده دوره‌گرد احتمالی متجانس (احتمال بازدید همه شهرها برابر است) میتوان از نمونه‌های مساله فروشنده دوره‌گرد 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 دوره‌گرد متقارن، اجرا و نتایج آن با نتایج دو الگوریتم مکاشفه ای به نامهای مرتب سازی محوری" و " بهترین تصادفی" که برای حل این مساله کزارش شده اند، مقایسه گردیده است.. ۵ نمونه از ۱۶ نمونه مذکور از کتابخانه‌یTSPLIB [18] انتخاب شده و شامل نمونه‌های eil51، berlin52، eil76، kroa100 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 شده‌اند که متقارن و دارای ۲۵ تا۱۵۰ شهر می باشند. برای انجام آزمایشها از یک کامپیوتر خانگی AMD 1300 MHZ با ۱۲۸ مگابایت حافظه اصلی) استفاده شده است.

---

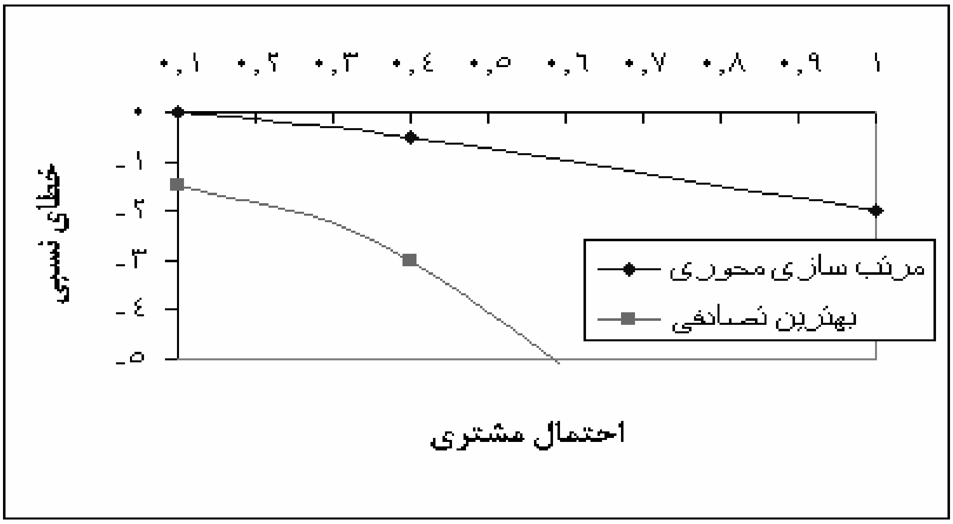
[30] اقدامهای غیر فعالیکه مجاز به انتخاب آنها نیستیم، اقدامهایی می باشند که توسط اتوماتای فعال با توجه به شرایط لازم برای ایجاد تور هامیلتونی، امکان انتخاب آنها وجود ندارد.

[31] Path Probability

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[32]، 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، جوابهای 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 ۶ 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 شهرها و محور عمودی خطای نسبی که از رابطه $\frac{E[PTSPDLA]-E[heurestic]}{E[PTSPDLA]}$ محاسبه می شود را نشان می دهد که در این رابطه $E[PTSPDLA]$ طول متوسط تور ایجاد شده توسط الگوریتم پیشنهادی و $E[heurestic]$ 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 دوره‌گرد احتمالی می باشد. اولین نکته‌ی قابل مشاهده در این شکل، بالاتر بودن کارایی الگوریتم پیشنهادی از هر دو الگوریتم مورد مقایسه برای همه‌ی 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 تقریبا یکسان الگوریتم پیشنهادی و مرتب سازی محوری در احتمالات کوچک (تقریبا برابر ۰٫۱) می باشد. نتیجه‌ی فوق مؤید این نکته است که تورهای ایجاد شده توسط الگوریتم مرتب سازی محوری برای مقادیر کوچک احتمالات بازدید نزدیک به بهینه می باشند.



شکل۶: کارایی نسبی الگوریتم پیشنهادی[33] در مقایسه با مرتب سازی محوری و بهترین تصادفی

---

[32] Center of mass' of the City spatial distribution
[33] PTSPDLA

کارایی مطلق الگوریتم پیشنهادی: با توجه به اینکه جواب بهینه هیچیک از نمونه مساله‌های فروشنده دوره‌گرد 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 میگیرد. میتوان با استفاده از حد پایین جواب بهینه، حد بالای خطای انجام گرفته توسط الگوریتم را بدست آورد. بعبارتی اگر LB حد پایین جواب بهینه نمونه مساله و $E[L_{\lambda*}]$ جواب بهینه آن باشد، طبق تعریف داریم:

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

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

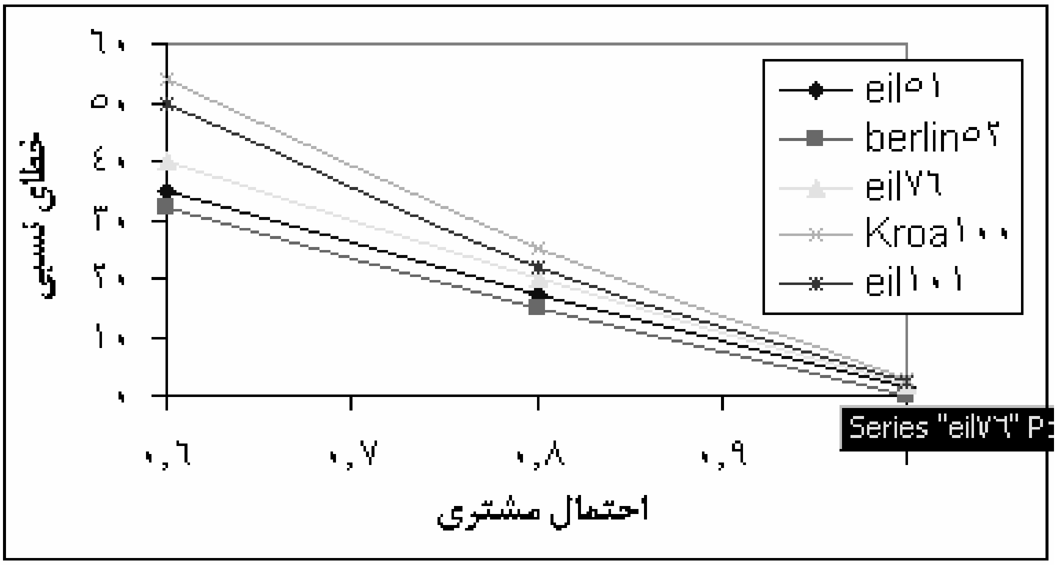
$$\frac{E[L_{\lambda}] - E[L_{\lambda*}]}{E[L_{\lambda*}]} \leq \frac{E[L_{\lambda}] - LB}{LB} \qquad (6)$$

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

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

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

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 نمونه‌های موجود در کتابخانه TSPLIB مورد ارزیابی گرفته است در شکل ۷ 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 ۷: 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 دوره‌گرد 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 صورتیکه احتمال بازدید شهرها بالا باشد، جوابهای با خطای نسبی پایین تولید می کند.

مراجع

[1] D. J. Bertsimas. Probabilistic Combinatorial Optimization Problems. PhD thesis, MIT, Cambridge, MA, 1988.
[2] D. J. Bertsimas and L. Howell, "Further results on the probabilistic traveling salesman problem", European Journal of Operational Research, 65:68-95, 1993.
[3] D. J. Bertsimas, P. Jaillet, and A. Odoni, "A priori optimization. Operations Research", 38:1019-1033, 1990.
[4] P. Jaillet, "Probabilistic Traveling Salesman Problems", PhD thesis, MIT, Cambridge, MA, 1985.
[5] A. J_ez_equel, "Probabilistic Vehicle Routing Problems", Master's thesis, MIT, Cambridge, MA, 1985.
[6] F. A. Rossi and I. Gavioli, "Aspects of Heuristic Methods in the Probabilistic Traveling Salesman Problem", pages 214-227. World Scienti_c, Singapore, 1987.
[7] S. Lakshmivarahan, "Learning Algorithms: Theory and Applications", New York: Springer-verlag, 1981.
[8] M. R. Meybodi and S. Lakshmivarahan, "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] P. Mars, J. R. Chen, and R. Nambir, "Learning Algorithms: Theory and Applications in Signal Processing, Control, and Communication", CRC Press Inc., 1996.
[10] K. S. Narendra and K. S. Thathachar, "Learning Automata: An Introduction", New York: Prentice-Hall, 1989.
[11] M. R. Meybodi & H. Beigy, "Solving Stochastic Shortest Path Problem Using Distributed Learning Automata", Proceedings 6th Annual CSI Computer Conference, University of Isfahan's Computer Engineering Department, 2001.
[12] H. Beigy and M. R. Meybodi "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] M. Alipour and M. R. Meybodi, "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] S. Lakshmivarahan, "Learning Algorithms: Theory and Applications", New York: Springer-verlag, 1981.
[15] M. R. Meybodi and S. Lakshmivarahan, "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] P. Mars, J. R. Chen, and R. Nambir, "Learning Algorithms: Theory and Applications in Signal Processing, Control, and Communication", CRC Press Inc., 1996.
[17] K. S. Narendra and K. S. Thathachar, "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] M. Alipour and M. R. Meybodi, " Solving Dynamic Traveling Salesman Problem Using Responsive Distributed Learning Automata", Proceedings of the Second International Conference on Information and Knowledge Technology (IKT2005), Tehran, Iran, May 24-26, 2005.