

یک الگوریتم تقریبی مبتنی بر اتوماتای یادگیر توزیع شده برای حل مسئله مسیریابی وسیله نقلیه با درخواستهای احتمالی

اصغر قربانی محمدرضا میبدی

آزمایشگاه سیستم‌های نرم افزاری

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

دانشگاه صنعتی امیرکبیر

تهران ایران

(aghorbani@aut.ac.ir , mmeybodi@aut.ac.ir)

چکیده: مسئله مسیریابی وسیله نقلیه با درخواستهای احتمالی یکی از انواع مسئله مسیریابی وسیله نقلیه میباشد که در آن درخواستهای مشتری ها از قبل مشخص نیست و در زمان ملاقات مشتری مشخص میشود. هدف از حل این مسئله یافتن ترتیبی از مشتری ها می باشد که اگر طبق این ترتیب آنها سرویس دهی شوند متوسط هزینه سفر حداقل گردد. این مسئله یکی از مسائل NP-complete میباشد و بهمین دلیل الگوریتمهای تقریبی متعددی برای آن طراحی شده است. در این مقاله یک الگوریتم تقریبی مبتنی بر اتوماتای یادگیر توزیع شده برای حل مسئله مسیریابی وسیله نقلیه با درخواستهای احتمالی پیشنهاد میگردد. نتایج آزمایشها نشان داده است که الگوریتم پیشنهادی در مقایسه با الگوریتمهای کلونی مورچه ها (ACS) ، جستجوی تابو (TS) و الگوریتم FR نتایج بهتری را تولید کرده است.

کلمات کلیدی: مسیریابی وسیله نقلیه با درخواستهای احتمالی، الگوریتم تقریبی، اتوماتاهای یادگیر، اتوماتای یادگیر توزیع شده

Utilizing Distributed Learning Automata to Solve Vehicle Routing Problem with Stochastic Demands

A. Ghorbani M. R. Meybodi

Computer Engineering and Information Technology
Amirkabir University of Technology
Tehran Iran

Abstract: In the vehicle routing problem with stochastic demands a vehicle has to serve a set of customers whose exact demand is known only upon arrival at the customer's location. A solution to this problem is a permutation of the customers that minimizes the expected distance traveled by the vehicle. This problem is one of the NP-Complete problems and for this reason many approximate algorithms have been designed for solving it. In this paper an approximate algorithm based on distributed learning automata for solving vehicle routing problem with stochastic demands is proposed. To show the performance of the proposed algorithm computer simulations have been conducted and the results obtained are compared with the results of three existing algorithms, Ant Colony Systems, Tabu Search and Randomized Farthest Insertion. The results of comparison have shown the efficiency of the proposed algorithm.

Keywords: Vehicle Routing Problem with Stochastic Demands, Approximate algorithms, Learning Algorithm, Distributed Learning Automata

۱. مقدمه

مسئله مسیریابی وسیله نقلیه با درخواستهای احتمالی^۱ $G = (V, A, D)$ بروی گراف کامل (VRPSD) تعریف می‌شود، که $D = \{d_{ij} : i, j \in V, i \neq j\}$ مجموعه گره‌ها $V = \{(i, j) : i, j \in V, i \neq j\}$ مجموعه بالهای گراف و $A = \{(i, j) : i, j \in V, i \neq j\}$ ماتریس هزینه یک ماتریس متقارن است که قاعده نامساوی مثلثی در آن برقرار می‌باشد. گره ۰ انبار و سایر گره‌ها، مشتری‌ها می‌باشند. ماتریس هزینه یک ماتریس متقارن است که قاعده نامساوی تقاضای مشتریان دارای توزیعهای احتمال گستته $i = 1, \dots, n$ که مستقل از یکدیگر هستند می‌باشد. Q ظرفیت وسیله نقلیه می‌باشد و فرض بر این است که تقاضای هر مشتری از این ظرفیت بیشتر نمی‌باشد. $p_{ik} = \text{Prob}(\xi_i = k), k = 1, \dots, K < Q$ احتمال اینکه مشتری i دارای تقاضای k می‌باشد است.

وسیله نقلیه طبق دنباله $s = s(1), s(2), \dots, s(n)$ که $s(1)$ انبار می‌باشد مشتریان را سرویس دهی می‌کند. سرویس دهی هر مشتری بدین شکل می‌باشد که پس از مشخص شدن مقدار تقاضای مشتری، وسیله نقلیه تصمیم گیری می‌کند که مشتری بعدی را طبق دنباله s سرویس دهد و یا برای بارگیری مجدد به انبار بازگردد. چنانچه تصمیم بر بارگیری مجدد باشد، پس از رفتن به انبار و بارگیری مجدد، سرویس دهی مشتریان طبق دنباله s ادامه پیدا می‌کند. این عمل تا سرویس دهی کامل به تمام مشتری‌ها ادامه می‌پیدد. بارگیری مجدد وسیله نقلیه پیش از تخلیه کامل بدین دلیل است که مقدار دقیق تقاضای مشتری بعدی از قبل مشخص نیست و لذا ممکن است تقاضای مشتری بعدی پیش از مقدار بار جاری وسیله نقلیه باشد که بارگیری مجدد پیشگیرانه^۲ نامیده می‌شود، باystی انجام گیرد.

هدف از حل مسئله VRPSD یافتن دنباله ای از مشتریان است که کمترین متوسط هزینه را داشته باشد. فرض کنید $f_j(q) = 0, 1, \dots, n$ یک تور باشد و همچنین فرض کنید پس از سرویس دهی به مشتری j ام بار جاری وسیله نقلیه q باشد، و $L_j(Q)$ مشخص کننده متوسط هزینه سفر از گره j تا انتهای تور باشد. بدین ترتیب هزینه متوسط تور برابر با $L_j(Q)$ خواهد بود. اگر $L_j(Q)$ مجموعه مقدار بارهایی باشد که وسیله نقلیه پس از اتمام سرویس دهی به مشتری j ام ممکن است داشته باشد، در این صورت $q \in L_j$ برابر است با:

$$f_j(q) = \min \{f_j^p(q), f_j^r(q)\} \quad (1)$$

که

$$\begin{aligned} f_j^p(q) &= d_{j,j+1} + \sum_{k:k \leq q} f_{j+1}(q-k) p_{j+1,k} \\ &\quad + \sum_{k:k > q} (b + 2d_{0,j+1} + f_{j+1}(q+Q-k) p_{j+1,k}) \end{aligned} \quad (2)$$

$$f_j^r(q) = d_{j,0} + d_{0,j+1} + \sum_{k=1}^K f_{j+1}(Q-k) p_{j+1,k} \quad (3)$$

شرایط مرزی معادلات فوق از این قرار است: $f_j^p(q) = d_{n,0}, q \in L_n$. در معادلات فوق $f_j^p(q) = d_{n,0}$ متوسط هزینه سفر در صورتی که مستقیماً مشتری بعدی سرویس دهی شود و $f_j^r(q) = d_{0,j+1}$ متوسط هزینه در صورتی که بارگیری مجدد پیشگیرانه انجام گیرد می‌باشد. چنانچه مقدار Q بی نهایت باشد مسئله VRPSD به مسئله TSP تبدیل می‌شود.

تا کنون الگوریتمهای تقریبی متعددی برای حل مسئله مسیریابی وسیله نقلیه و مسیریابی وسیله نقلیه با درخواستهای احتمالی گزارش شده است [4][9-15]. در [4] الگوریتمهای مبتنی بر کلونی مورچه‌ها^۳ (ACO)، جستجوی تابو^۴ (TS)، روش سرد شدن^۵ و FR^۶ برای این مسئله گزارش شده است. برتسیماس^۷ در [18] یک الگوریتم با نام هیوریستیک دوره‌ای^۸ برای حل VRPSD و

¹ Vehicle Routing Problem with Stochastic Demands (VRPSD)

² Preventive Restocking

³ Ant Colony Optimization

⁴ Tabu Search

⁵ Simulated Annealing

⁶ Randomized Farthest Insertion

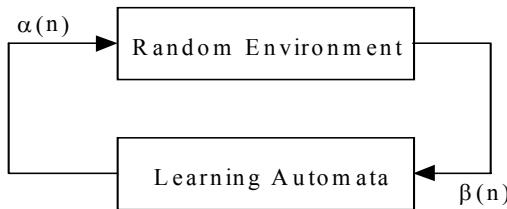
ینگ⁹ در [17]، الگوریتمی برای حل مسأله VRPSD در حالتی که چندین وسیله نقلیه وجود داشته باشد ارائه کرده است. در مقاله مروری [19] الگوریتمهای حل مسأله VRPSD جمع آوری شده است.

در این مقاله یک الگوریتم تقریبی مبتنی بر اتوماتای یادگیر توزیع شده برای حل مسأله مسیریابی وسیله نقلیه با درخواستهای احتمالی پیشنهاد میگردد. الگوریتم پیشنهادی ابتدا یک اتوماتای یادگیر توزیع شده متناظر با گراف مسأله VRPSD را ایجاد میکند. در اتوماتای یادگیر توزیع شده هر گره (مشتری) به یک اتوماتای یادگیر با ساختار متغیر مجهر میباشد. یالهای خروجی این گره اقدامهای این اتوماتای یادگیر متناظر با آن گره می باشند. خروجی اتوماتای یادگیر توزیع شده ترتیبی از اقدامهای انتخاب شده توسط اتوماتاهای یادگیر می باشد که یک تور در گراف را نشان می دهد. با توجه به مطلوب و یا نامطلوب بودن تور تولید شده، اعمال انتخاب شده توسط اتوماتاهای یادگیر در طول این تور پاداش و یا جریمه داده میشود. فرایند ایجاد تور و بروز کردن بردار احتمالات اتوماتاهای یادگیر ادامه پیدا میکند تا توری که یک معیار از پیش تعیین شده را ارضا کند پیدا شود. به منظور ارزیابی، الگوریتم پیشنهادی با الگوریتم های ACS و FR که در [4] به تفصیل شرح داده شده اند مقایسه میگردد. نتایج مقایسه نشان از عملکرد بهتر الگوریتم پیشنهادی دارد.

ادامه مقاله بدین صورت سازماندهی شده است. در بخش ۲ اتوماتاهای یادگیر و در بخش ۳ اتوماتای یادگیر توزیع شده به اختصار شرح داده شده است. در بخش ۴ الگوریتم پیشنهادی و در بخش ۵ نتایج آزمایشها ارایه میشود. بخش ۶ نتیجه گیری میباشد.

۲. اتوماتاهای یادگیر

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 و عمل خود را برای مرحله بعد انتخاب می کند. شکل ۱ ارتباط بین اتوماتای یادگیر و محیط را نشان می دهد.



شکل ۱: ارتباط بین اتوماتای یادگیر و محیط

محیط را می توان توسط سه تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه ورودیها، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه خروجیها و $c \equiv \{c_1, c_2, \dots, c_r\}$ 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 i نتیجه نامطلوب داشته باشد می باشد. در محیط ایستا¹⁰ مقادیر c_i بدون تغییر می مانند، حال آنکه در محیط غیر ایستا¹¹ این مقادیر در طی زمان تغییر می کنند. اتوماتاهای یادگیریه دو گروه با ساختار ثابت و با ساختار متغیر تقسیم بندی میگردد. در ادامه به شرح مختصری درباره اتوماتای یادگیر با ساختار متغیر که در این مقاله از آنها استفاده شده است می پردازیم.

اتوماتای یادگیر با ساختار متغیر¹²: اتوماتای یادگیر با ساختار متغیر توسط ۴ تایی $\{\alpha, \beta, p, T\}$ نشان داده می شود که در آن $p \equiv \{p_1, p_2, \dots, p_r\}$ مجموعه عملهای اتوماتا، $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عملهای اتوماتا و $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودیهای اتوماتا، $T[\alpha(n), \beta(n), p(n)] = p(n+1)$ بردار احتمال انتخاب هر یک از عملهای α را می باشد. در این نوع از اتوماتاهای، اگر عمل i در مرحله n ام انتخاب شود و پاسخ مطلوب از محیط دریافت نماید، احتمال p_i افزایش یافته و سایر احتمالها کاهش

⁷ Bertsimas

⁸ Cyclic Heuristic

⁹ Yang

¹⁰ Stationary

¹¹ Non-Stationary

¹² Variable Learning Automata

می یابند. و برای پاسخ نامطلوب احتمال $(n)_i$ کاهش یافته و سایر احتمالها افزایش می یابند. در هر حال، تغییرات به گونه ای صورت می گیرد تا حاصل جمع $(n)_i$ همواره ثابت و مساوی یک باقی بماند. الگوریتم زیر یک نمونه از الگوریتمهای یادگیری خطی برای اتوماتای یادگیر با ساختار متغیر میباشد

الف- پاسخ مطلوب

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)]$$

$$p_j(n+1) = (1-a)p_j(n) \quad j \neq i \quad \forall j$$

ب- پاسخ نامطلوب

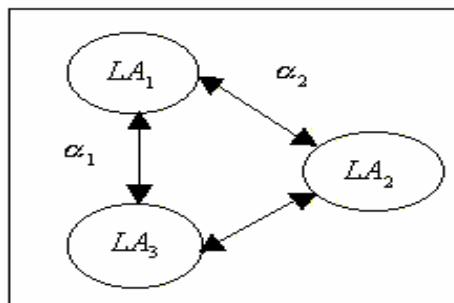
$$p_i(n+1) = (1-b)p_i(n)$$

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

در روابط فوق، پارامتر پاداش و a 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 را L_{RP} ¹³ می نامیم. زمانیکه b از a خیلی کوچکتر باشد، الگوریتم را L_{REP} ¹⁴ می نامیم. زمانیکه b مساوی صفر باشد، الگوریتم را L_{RI} ¹⁵ می نامیم. برای مطالعه بیشتر درباره اتوماتاهای یادگیر می توان به [5][6][7][8] مراجعه کرد.

۳. اتوماتای یادگیر توزیع شده^{۱۶}

اتوماتای یادگیر توزیع شده (DLA)، شبکه‌ای از اتوماتاهای یادگیر است که برای حل یک مسئله با یکدیگر همکاری می نمایند [2]. تعداد اقدامهای یک اتوماتای یادگیر در DLA برابر تعداد اتوماتاهای متعلق به این اتوماتای یادگیر می باشد. انتخاب یک اقدام توسط یک اتوماتا در شبکه، اتوماتای متناظر با این اقدام را فعال می سازد. عنوان مثال در شکل ۲ هر اتوماتا 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 متعلق به آن اتوماتا که متناظر با اقدام انتخاب شده می باشد فعال می شود. در هر زمان فقط یک اتوماتا در شبکه فعال میباشد. بطور رسمی DLA را میتوان توسط گراف (V, E) که $DLA = (V, E)$ که $V = \{LA_1, LA_2, \dots, LA_n\}$ و $E \subset V \times V$ مجموعه اتوماتای یادگیر و n تعداد اتوماتاهای در DLA و E مجموعه لبه‌های گراف می باشد، تعریف کرد. لبه (i, j) اقدام j اتوماتا LA_i را نشان می دهد. LA_j زمانی فعال خواهد شد که اقدام j اتوماتون i LA_i انتخاب شود. تعداد اقدامهای اتوماتا LA_k ($k = 1, 2, \dots, n$) برابر درجه‌ی خروجی آن گره می باشد. برای اطلاعات بیشتر درباره اتوماتاهای یادگیر توزیع شده میتوان به [7][8] و [20] مراجعه نمود.



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

¹³ Linear Reward Penalty

¹⁴ Linear Reward Epsilon Penalty

¹⁵ Linear Reward Inaction

¹⁶ Distributed Learning Automata

۴. الگوریتم پیشنهادی

ابتدا یک اتوماتای یادگیر توزیع شده متناظر با گراف مسئله VRPSD ایجاد می شود. اتوماتای یادگیر در هر گره (مشتری) از اتوماتای یادگیر توزیع شده یک اتوماتای یادگیر با ساختار متغیر میباشد. یالهای خروجی این گره اقدامهای این اتوماتای یادگیر می باشد. یعنی تعداد اقدامهای یک اتوماتای یادگیر معادل تعداد گره هایی می باشد که میتوان بطور مستقیم از این گره به آنها حرکت کرد. خروجی اتوماتای یادگیر توزیع شده ترتیبی از اقدامهای انتخاب شده توسط اتوماتاهای یادگیر می باشد که یک تور در گراف را نشان می دهد. با توجه به هزینه این تور $(f_0(Q))$ ، اعمال انتخاب شده توسط اتوماتاهای یادگیر در طول این تور پاداش و یا جریمه داده میشود. فرایند ایجاد تور و بروز کردن بردار احتمالات اتوماتاهای یادگیر ادامه پیدا میکند تا توری که یک معیار از پیش تعیین شده را ارضاء کند پیدا شود.

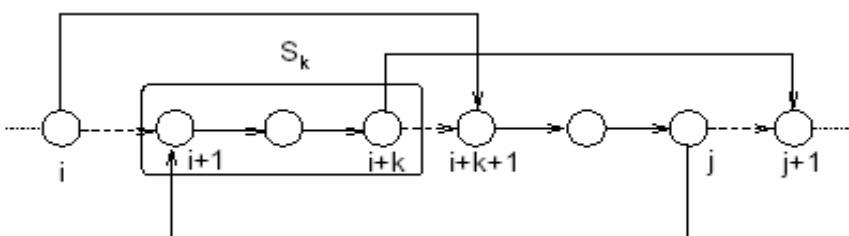
آزمایشهای انجام گرفته نشان داده است که برای افزایش نرخ همگرایی الگوریتم به خصوص برای گرافهای بزرگ بهتر است تور بدست آمده در انتهای هر مرحله از الگوریتم به طریقی بهبود داده شود. یک راهکار برای انجام این کار استفاده از الگوریتم orOptTSP [16] و یا orOpt [4] می باشد. این الگوریتمها از طریق جابجایی گره ها (مشتری ها) در طول تور بدست سعی در پیدا کردن یک تور بهتر مینماید. عمل اصلی در الگوریتم های orOpt و یا orOptTSP جابجایی رشته های s_k که $k \in \{1,2,3\}$ است تور رشته از یک نقطه از تور به نقطه دیگری از تور می باشد. فرض کنید s_k گره i تا گره $i+k+1$ در تور داده شده باشد و پس از حذف s_k از تور، s_k پس از گره j درج میگردد. ابتدا مقدار هزینه ای که بابت حذف s_k کم میشود، محاسبه می گردد و سپس مقدار هزینه ای که بابت درج این رشته پس از گره j بدست آمده محاسبه شود. سپس بررسی شود که تور بهتری حاصل گردیده یا خیر. در orOpt هزینه هر جابجایی از رابطه زیر بدست می آید:

$$\frac{\sum_{q=0}^Q [(f_j''(q) - f_j(q)) - (f_i'(q) - f_i(q))]}{Q + 1} \quad (4)$$

که $f_i(q)$ هزینه سفر از گره i پیش از حذف s_k و $f_i'(q)$ هزینه سفر از گره i تا انتهای تور پس از حذف s_k میباشد. $f_j''(q)$ هزینه سفر از گره j پیش از درج s_k و $f_j'(q)$ مقدار هزینه سفر از گره j پس از درج s_k میباشد. در orOptTSP هر جابجایی از رابطه زیر بدست می آید:

$$d_{i,i+k+1} + d_{j,i+1} + d_{i+k,j+1} - d_{i,i+1} - d_{i+k,i+k+1} - d_{j,j+1} \quad (5)$$

که $d_{i,j}$ فاصله گره های i و j میباشد. در شکل 2 s_k و محل درج آن مشخص شده است.



شکل 2: نحوه عمل تابع orOpt. در این شکل s_k با $k=3$ و محل بعدی آن مشخص شده است.

اکنون به توصیف الگوریتم پیشنهادی می پردازیم. در گام نخست اتوماتای مبدأ، گره (0) از گراف VRPSD که انبار می باشد عنوان نقطه آغازین تور انتخاب می شود. این اتوماتای یادگیر یکی از اقدامهای خود را بر طبق بردار احتمال اقدام P^j انتخاب می کند. انجام این اقدام، اتوماتای طرف دیگر یال که متناظر با اقدام انتخاب شده میباشد را فعال می سازد. با توجه به اینکه در این مسئله هر گره نباید بیش از یکبار ملاقات شود ترتیبی بایستی اتخاذ گردد تا هیچ گره ای بیش از یکبار انتخاب نشود. برای این منظور اگر اتوماتای یادگیری اقدام k را از لیست اقدامهای خود انتخاب کند، همزمان با آن اتوماتاهای غیرفعال اقدام k را در لیست اقدامهای خود غیرفعال می سازند. در ابتدای هر تکرار تمام اقدامهای غیرفعال شده مجدداً فعال میشوند. اتوماتای یادگیر فعال شده با استفاده از بردار احتمال اقدام تعییر یافته‌ی خود، اتوماتا طرف دیگر یال انتخاب شده را فعال می سازد. فرآیند انتخاب اقدام و فعال سازی اتوماتای یادگیر متناظر با اقدام انتخاب شده تا ملاقات همه گرههای موجود در گراف VRPSD و برگشت به انبار (مبدأ) تکرار می شود.

پس از پیدا کردن یک تور، هزینه آن محاسبه شده و با یک آستانه مقایسه می‌شود. مقدار آستانه میانگین هزینه تورهایی که از ابتدای الگوریتم تا بحال پاداش گرفته اند می‌باشد. با توجه به نتیجه مقایسه هزینه تور با مقدار استانه، بردار احتمال اقدام اتوماتاهای DLA بروزرسانی می‌شود. نحوه بروزرسانی بردار احتمال اقدام بدینصورت است که اگر هزینه تور ایجاد شده کوچکتر و یا مساوی آستانه باشد اعمال انتخابی توسط اتوماتاهای یادگیری در طول تور بر طبق الگوریتم یادگیری L_{R-I} ، پاداش می‌گیرند برای روشن شدن بیشتر این مطلب الگوریتم L_{R-I} ذکر شده در بخش اتوماتای یادگیر را مجدداً یادآوری می‌نماییم: عنوان مثال اگر هزینه تور ایجاد شده در یک تکرار (t) کوچکتر و یا مساوی هزینه بهترین توری که تابحال ایجاد شده است، باشد و در این تکرار اتوماتای J از مجموعه اقدامهای مجاز خود اقدام i را انتخاب کرده باشد، احتمال انتخاب اقدام i طبق رابطه زیر افزایش خواهد یافت:

$$p_i(t+1) = p_i(t) + a[1 - p_i(t)]$$

و احتمال انتخاب سایر اقدامهای اتوماتا 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 و r تعداد اقدامهای اتوماتای یادگیر J می‌باشد. فرآیند ایجاد تور تا رسیدن به شرط پایانی ادامه می‌باشد. آخرین تور ایجاد شده تور تقریبی تولید شده توسط الگوریتم می‌باشد. الگوریتم پیشنهادی در شکل ۳ آورده شده است.

```

Procedure VRPSD
Begin
    Initialize the probability vector of each automaton
    repeat
        //Phase 1
        initialPlace := depot (vertex 0)
        CurrentCustomer := initialPlace
        Disable action ‘CurrentCostumer’ of all Unactivated LAs
        //Phase 2
        for i := 1 to n
            if i < n then
                //choose the next node as a sample realization of active LA action probability vector,  $P^j$ 
                NextCustomer := GetNextCustomer()
                Disable action ‘NextCustomer’ of all Unactivated LAs
                if i = n-1 then
                    Enable action ‘InitialPlace’ of LA NextCustomer
                end if
            else
                NextCustomer := InitialPlace
            end if
            CurrentCustomer := NextCustomer
            CurrentTour.add(CurrentCustomer)
        next i

        //Phase 3
        orOpt(CurrentTour)

        //Phase 4
        Compute the tour cost
        if CurrentTourCost <= threshold then
            //Reward the selected actions of all LAs along the tour according to  $L_{R-I}$  learning algorithm
            Reward(CurrentTour)
            BestTour := CurrentTour
        end if
        compute threshold
        Enable all the disabled actions of LAs
        until (stop condition)
    end procedure

```

شکل ۳: الگوریتم پیشنهادی

شرح الگوریتم فوق در ادامه آمده است.

مرحله ۰: گره ۰ که انبار می باشد بعنوان نقطه آغازین تور انتخاب شده و سپس اتماتای متناظر با این گره فعال میگردد و اقدامهایی که در اتماتاهای دیگر منجر به انتخاب این گره شود غیر فعال می گردد.

مرحله ۲: در هر تکرار از این مرحله اتماتای یادگیرفعال شده یکی از اقدامهای مجاز^{۱۷} خود را انتخاب می شود. تابع GetNextCustomer() با استفاده از بردار احتمال P^j ، یکی از اقدامهای مجاز اتماتای یادگیرفعال شده را انتخاب می کند. چنانچه این تابع برای گره فعال اقدام مجازی را بر نگرداند، بدین معنی است که تور کامل شده است و هیچ گره (مشتری) دیگری برای انتخاب باقی نمانده است که در این صورت به مرحله ۳ میرویم.

مرحله ۳: الگوریتم orOpt، با جستجو در همسایگی تور بدست آمده سعی میکند که تور بهتری را پیدا کند. در صورت پیدا شدن تور بهتر، آن تور جایگزین تور فعلی می گردد.

مرحله ۴: هزینه تور ایجاد شده (Q_f) با استفاده از رابطه (1) محاسبه و با مقدار آستانه (threshold) مقایسه میگردد. مقدار آستانه، میانگین هزینه تورهایی است که از ابتدای اجرای الگوریتم تا کنون پاداش گرفته اند میباشد. در صورتیکه هزینه تور ایجاد شده کوچکتر و یا مساوی مقدار آستانه باشد، به اقدامهای انتخاب شده اتماتاهای DLA طبق الگوریتم یادگیری L_{R-I} پاداش داده می شود. اقدامهای غیر فعال شده در مرحله ۲، مجدداً فعال شده و سپس شرط خاتمه الگوریتم بررسی میگردد. در صورتیکه شرط پایان الگوریتم برقرار نباشد الگوریتم به مرحله ۱ بر میگردد و توری جدید آغاز میشود. اگر تعداد تورهای ایجاد شده توسط الگوریتم از مقدار از پیش تعیین شده ای بیشتر باشد و یا احتمال مسیر^{۱۸} که عبارتست از حاصلضرب احتمال یالهای موجود در تور بهینه، از یک آستانه (برای مثال، ۹۰٪) بیشتر شود الگوریتم خاتمه می یابد.

الگوریتم فوق الذکر را الگوریتم ۱ (alg1) مینامیم. با توجه به تعاریف مختلف آستانه، اینکه تابع orOpt مورد استفاده قرار گیرد یا orOptTSP و یا اینکه از الگوریتمهای بهبود دهنده استفاده بشود یا نشود، نسخه های مختلفی از الگوریتم پیشنهادی به دست می آید که در ادامه ۵ نمونه آن مختصرًا توضیح داده میشود.

الگوریتم ۲ (alg2): الگوریتم ۲ همان الگوریتم ۱ است با این تفاوت که به جای استفاده از تابع orOpt از تابع orOptTSP استفاده شده است.

الگوریتم ۳ (alg3): الگوریتم ۳ همان الگوریتم ۱ است که در آن تابع orOpt فراخوانی نمی شود.

الگوریتم ۴ (alg4): الگوریتم ۴ همان الگوریتم ۳ است با این تفاوت که مقدار آستانه برابر است با مقدار هزینه بهترین توری که تابحال بدست آمده باشد.

الگوریتم ۵ (alg5): الگوریتم ۵ همان الگوریتم ۲ است با این تفاوت که مقدار آستانه برابر است با مقدار هزینه بهترین توری که تابحال بدست آمده باشد.

الگوریتم ۶ (alg6): الگوریتم ۶ همان الگوریتم ۱ است با این تفاوت که مقدار آستانه برابر است با مقدار هزینه بهترین توری که تابحال بدست آمده باشد.

۵. نتایج آزمایشها

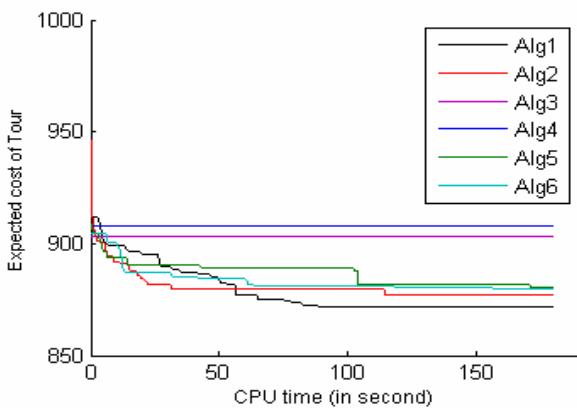
برای ارزیابی الگوریتمهای مسیریابی وسیله نقلیه با درخواست های احتمالی، مسائل استاندارد وجود ندارد. برای انجام آزمایشها در این مقاله از دو نمونه مسئله ۲۰ مشتری و ۲۵ مشتری که از [4] گرفته شده است استفاده میشود. در پیوست این مقاله مشخصات این نمونه ها آورده شده است. برای ارزیابی عملکرد الگوریتمها از دو معیار هزینه متوسط تور و پارامتر کیفیت که طبق رابطه (6) محاسبه میشود استفاده شده است. در این رابطه $ALG_{final\ value}$ هزینه تور حاصل از اجرای الگوریتم، $FR_{starting\ value}$ هزینه تور ابتدایی $FR_{final\ value}$ هزینه تور نهایی حاصل از اجرای الگوریتم FR [4] میباشند. مقادیر منفی پارامتر کیفیت برای یک الگوریتم نشانده‌اند که عملکرد بهتر الگوریتم نسبت به الگوریتم FR میباشد. با توجه به رابطه ۶ پارامتر کیفیت برای الگوریتم FR صفر میباشد.

¹⁸ Path Probability

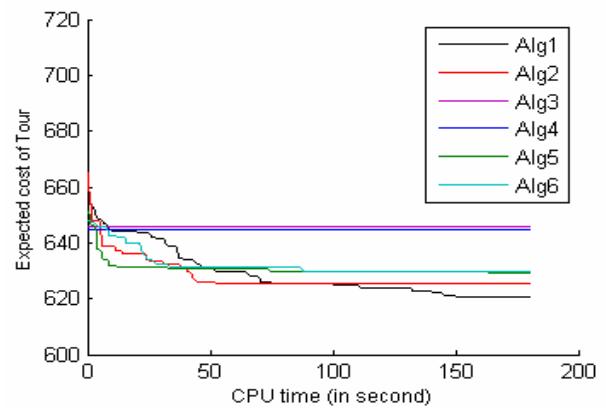
$$\frac{ALG_{final\ value} - FR_{final\ value}}{FR_{starting\ value} - FR_{final\ value}} \quad (6)$$

در شکل ۴ و شکل ۵ هزینه بهترین تورهایی که تا لحظه t توسط الگوریتم‌های پیشنهادی برتری برای مسئله ۱ و مسئله ۲ بدست آمده نشان داده شده است. همانطور مشاهده می‌شود، الگوریتم‌های ۳ و ۴ که از بهبود دهنده‌های محلی مانند orOptTSP و orOpt یا استفاده نمی‌کنند دارای نرخ همگرایی پایین تری می‌باشند. همچنین مشاهده می‌شود الگوریتم‌هایی که از میانگین هزینه‌های تورهای که پاداش گرفته‌اند به عنوان آستانه استفاده می‌کنند در مقایسه با الگوریتم‌هایی که از هزینه بهترین تور برای آستانه استفاده می‌کنند نتایج بهتری تولید کرده‌اند. از مقایسه الگوریتم ۱ با الگوریتم ۲ و مقایسه الگوریتم ۵ میتوان نتیجه گرفت که الگوریتم‌هایی که از تابع orOpt استفاده کرده‌اند (الگوریتم ۲ و ۵) دارای نرخ همگرایی بالاتری و الگوریتم‌هایی که از orOpt استفاده کرده‌اند (الگوریتم ۱ و ۶) دارای دقت بالاتری هستند.

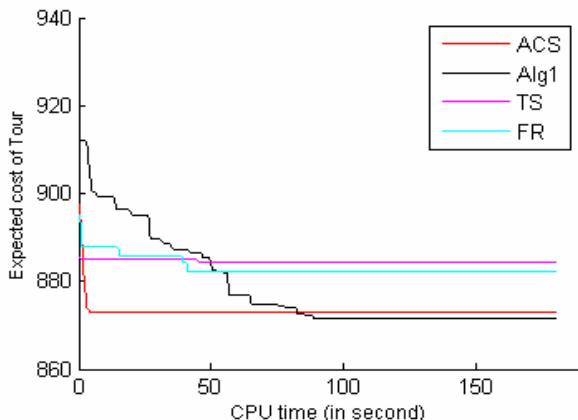
در نمودارهای شکل ۶ و شکل ۷ نتایج همگرایی الگوریتم پیشنهادی ۱ در مقایسه با الگوریتم‌های TS، ACS و FR در نمودار شکل ۸ مقایسه تورهای تولید شده توسط الگوریتم‌هایی که از هزینه بهترین تور برای آستانه استفاده می‌کنند (الگوریتم ۱ تا ۶) در مقایسه با الگوریتم‌های FR، TS و ACS نشان داده شده است. در شکل ۸ مقادیر منفی برای پارامتر کیفیت نشانده‌نده عملکرد بهتر الگوریتم FR نسبت به الگوریتم ۱ مقدار پارامتر کیفیت برای الگوریتم FR صفر می‌باشد. همانطور که مشاهده می‌شود برای هر دو نمونه از مسئله، الگوریتم پیشنهادی ۱ نتایج بهتری را در مقایسه با هر سه الگوریتم کلونی مورچه‌ها (ACS)، جستجوی تابو (TS) و الگوریتم FR تولید کرده است. همچنین مشاهده می‌شود که الگوریتم‌های پیشنهادی ۲، ۵ و ۶ بهتر از الگوریتم‌های جستجوی تابو (TS) و الگوریتم FR عمل کرده‌اند.



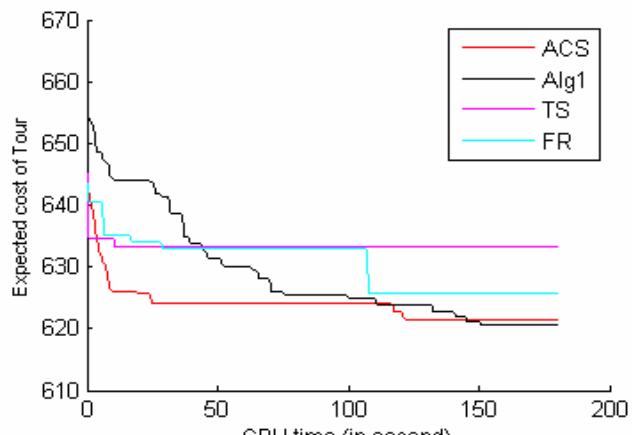
شکل ۵: متوسط هزینه تور حاصل از اجرای الگوریتم‌های ارائه شده در نمونه ۲



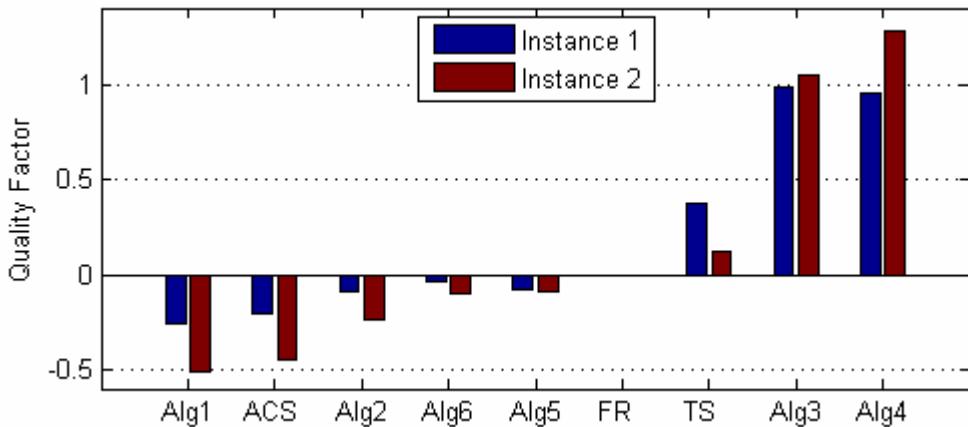
شکل ۴: متوسط هزینه تور حاصل از اجرای الگوریتم‌های ارائه شده در نمونه ۱



شکل ۷: نمودار متوسط هزینه تور حاصل از اجرای الگوریتم‌ها در نمونه ۲



شکل ۶: نمودار متوسط هزینه تور حاصل از اجرای الگوریتم‌ها در نمونه ۱



شکل ۸: نمودار ستونی پارامتر کیفیت تورهای حاصل از اجرای الگوریتم‌ها، مقدار منفی برای یک الگوریتم نشانده‌نده عملکرد بهتر ان الگوریتم نسبت به الگوریتم **FR** می‌باشد.

۶. نتیجه گیری

در این مقاله یک الگوریتم تقریبی مبتنی بر اتمماتی یادگیر توزیع شده برای حل مسئله مسیریابی وسیله نقلیه با درخواستهای احتمالی پیشنهاد گردید. نتایج آزمایشها نشان داد که الگوریتمهای پیشنهادی در مقایسه با الگوریتمهای کلونی مورچه ها (ACS)، جستجوی تابو (TS) و الگوریتم FR نتایج بهتری را تولید کرده است.

مراجع

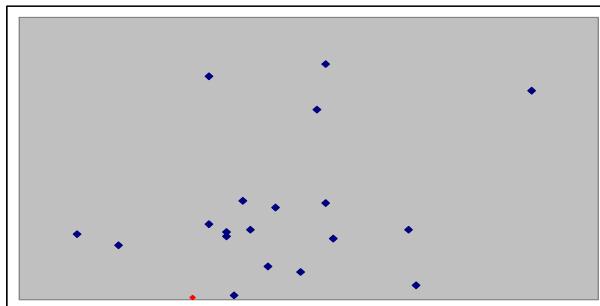
- [1] M. R. Meybodi and H. Beigy, "Solving Stochastic Shortest Path Problem Using Distributed Learning Automata", Proceedings 6th Annual CSI Computer Conference, University of Isfahan, Computer Engineering Department, 2001.
- [2] H. Beigy and M. R. Meybodi, "A New Distributed Learning Automata for Solving Stochastic Shortest Path Problem", Proceedings of the Sixth International Joint Conference on Information Science, Durham, USA, pp. 339-343, 2002,
- [3] M. Alipour and M. R. Meybodi, "A Distributed Learning Automata Based Algorithm for Solving Shortest Path Problem", Technical Report, Computer Engineering Department, Amirkabir University, Tehran, Iran, 2003.
- [4] L. Bianchi, M. Birattari, M. Chiarandini, M. Manfrin, M. Mastrolilli, L. Paquete, O. Rossi-Doria, and T. Schiavinotto, "Metaheuristics for the Vehicle Routing Problem with Stochastic Demands", Lecture Notes in Computer Science, volume 3242, pp. 450-460, Springer-Verlag, Heidelberg, Germany, 2004.
- [5] K. S. Narendra and K. S. Thathachar, Learning Automata: An Introduction, New York: Prentice-Hall, 1989.
- [6] S. Lakshminarayanan, Learning Algorithms: Theory and Applications, New York: Springer-Verlag, 1981.
- [7] M. R. Meybodi and S. Lakshminarayanan, "On a Class of Learning Algorithms which have Symmetric Behavior under Success and Failure", Lecture Notes in Statistics, Berlin: Springer-Verlag, pp. 145-155. 1984.
- [8] P. Mars, J. R. Chen and R. Nambir, Learning Algorithms: Theory and Applications in Signal Processing, Control, and Communication. CRC Press Inc., 1996.
- [9] P. M. Thompson and H. N. Psaraftis. "Cyclic Transfer Algorithms for the Multivehicle Routing and Scheduling Problems", Operations Research 41, pp. 935-946, 1993.
- [10] A. Van Breedam, "An Analysis of the Behavior of Heuristics for the Vehicle Routing Problem for a Selection of Problems with Vehicle-Related, Customer-Related, and Time-Related Constraints", Ph.D. Dissertation, University of Antwerp. 1994.
- [11] M. L. Fisher and R. Jaikumar, "A Generalized Assignment Heuristic for Vehicle Routing", Networks, No. 11, pp. 109-124, 1981.
- [12] D. M. Ryan, C. Hjorring and F. Glover, "Extensions of the Petal Method for Vehicle Routing", Journal of the Operational Research Society, 44, 289-296, 1993.
- [13] J. Renaud and F. F. Boctor, "A Sweep-Based Algorithm for the Fleet Size and Mix Vehicle Routing Problem", European Journal of Operational Research 140, pp. 618-628, 2002.
- [14] É. D. Taillard. "Parallel Iterative Search Methods for Vehicle Routing Problems", Networks, No.23, pp. 661-673, 1993.

- [15] J. Kelly and J. P. Xu., "A Network Flow-Based Tabu Search Heuristic for the Vehicle Routing Problem", *Transportation Science*, No.30, pp. 379-393, 1996.
- [16] I. Or, Traveling Salesman-type Combinatorial Problems and Their Relation to the Logistics of Blood Banking, PhD Thesis, Northwestern University, Evanston, IL, 1976.
- [17] W. Yang, K. Mathur, and R. H. Ballou, "Stochastic Vehicle Routing Problem with Restocking", *Transportation Science*, 34(1), pp. 99–112, 2000.
- [18] D. J. Bertimas, P. Cervi, and M. Peterson, "Computational Approaches to Stochastic Vehicle Routing Problems", *Transportation Science*, Vol. 29, No. 4, pp. 342-352, 1995.
- [19] A. Kenyon and D. P. Morton, "A Survey on Stochastic Location and Routing Problems", *Central European Journal of Operations Research*, No. 9, pp. 277-328, 2002.
- [20] H. Beigy and M. R. Meybodi, "Utilizing Distributed Learning Automata to Solve Stochastic Shortest Path Problem", *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, Vol. 14 No. 5 pp. 591-615, 2006.

پیوست

جدول ۱: مشخصات نمونه ۱

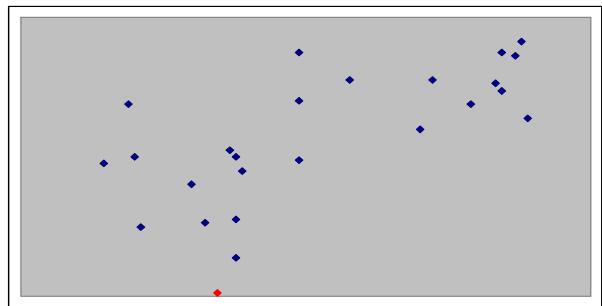
nodes	x	y	avg_demand	spread	nodes	x	y	avg_demand	spread
0	1	1	0	0	10	7	42	5	5
1	-13	28	22	5	11	5	29	24	1
2	18	26	73	5	12	42	89	73	5
3	10	14	37	1	13	14	12	18	1
4	27	30	29	1	14	6	2	91	1
5	5	27	48	5	15	16	81	69	5
6	3	32	44	5	16	8	30	73	5
7	17	100	9	1	17	28	6	20	1
8	17	41	79	5	18	3	95	68	5
9	-8	23	51	5	19	11	39	81	5
Q=203									



شکل ۹: توزیع مشتریان در نمونه ۱. رنگ قرمز نشانده‌هندۀ انبار میباشد.

جدول ۲: مشخصات نمونه ۲

nodes	x	y	avg_demand	spread	nodes	x	y	avg_demand	spread
0	1	1	0	0	13	45	61	42	1
1	-3	32	9	1	14	35	62	65	1
2	33	48	69	1	15	-17	38	44	5
3	-13	55	19	5	16	14	56	11	1
4	22	62	41	1	17	46	70	76	5
5	4	11	96	5	18	46	59	45	1
6	-1	21	27	5	19	3	42	14	5
7	50	51	3	1	20	14	39	50	1
8	4	40	21	5	21	-11	20	30	5
9	48	69	49	1	22	49	73	53	1
10	5	36	60	1	23	-12	40	33	1
11	4	22	48	5	24	41	55	21	1
12	14	70	49	1	13	45	61	42	1
Q = 150									



شکل 10: توزیع مشتریان در نمونه ۲. رنگ قرمز نشانده‌هنده انبار میباشد.