QLA : روشی جدید برای ایجاد همکاری در سیستمهای چند عامله

برنا جعفر پور محمدرضا میبدی
دانشکده مهندسی کامپیوتر و فناوری اطلاعات
دانشگاه صنعتی امیر کبیر
تهران ایران
(jafarpour@aut.ac.ir, mmeybodi@aut.ac.ir)

چکیده

اتوماتای یادگیر یک مدل انتزاعی است که تعداد محدودی عمل دارد که می‌تواند آنها را در محیط خود انجام دهد. اتوماتای یادگیر با توجه به پاسخی که از محیط در مقابل عمل خود میگیرد رفتار خود را به گونه ای تغییر می‌دهد که بیشترین پاداش را از محیط در یافت کند. در این مقاله یک مدل جدید بر پایه اتوماتاهای یادگیر برای ایجاد همکاری در سیستمهای چند عاملی پیشنهاد میگردد. در این مدل در هر حالت محیط یک اتوماتای یادگیر قرار داده می‌شود که حالت بعد را برای عاملی که در آن حالت قرار دارد تعیین می‌کند. برای بررسی کارایی این روش، از آن برای حل مسئله جمع آوری اشیاء استفاده شده است. آزمایشها نشان می‌دهد که این روش از کارایی بالاتری در مقایسه با روشهای برپایه ی Q-Learning و فرومون برخوردار می‌باشد.

کلمات کلیدی : سیستمهای چند عاملی، هماهنگی، اتوماتاهای یادگیر، یادگیری Q، جمع آوری اشیاء

QLA: A New Coordination Method for Multi Agent Systems

B. Jafarpour M. R. Meybodi
Computer Engineering and Information Technology Department
Amirkabir University of Technology
Tehran Iran
(jafarpour@.aut.ac.ir, mmeybodi@aut.ac.ir)

Abstract

A Learning automaton is an abstract model which randomly selects one action out of its finite actions and performs it on a random environment. Based on the responses received from the automaton changes its behavior so that it receives maximum reward from the environment. In this paper a new model based on learning automata for multi agent coordination is proposed. In this model, a learning automaton is put in each state of the environment which is responsible for determining the next state for the visiting agent. To show the performance of proposed model, it is used to solve multi agent Foraging Problem and compared the results with Q-Learning and Pheromone based solutions. Comparisons show the superiority of QLA over Q-Learning and Pheromones.

Keywords: Multi Agent Systems, Coordination, Learning Automata, Q-Learning, Foraging Problem.

۱- مقدمه

ایجاد هماهنگی در یک سیستم چند عاملی امری بسیار مهم و در عین حال دشوار می‌باشد. چنانچه عاملها در یک سیستم همکاری لازم با یکدیگر را نداشته باشند، مزیتهایی که سیستمهای چند عاملی ارائه می کنند قابل دسترسی نخواهد بود. ایجاد هماهنگی بین عاملها عبارت است از همکاری عاملها به شکلی که پاداش دریافتی تیم در تعامل با محیط به صورت کلی (و نه صرفا به صورت فردی) افزایش پیدا کند. برای ایجاد هماهنگی بین عاملها در یک سیستم چند عاملی از روشهای مختلفی از جمله لگوریتمهای تکاملی[1] ، اتوماتاهای یادگیر[2] و فرمون[3] استفاده شده است.

از پردازش تکاملی برای یافتن پارامترهای بهینه برای عاملها استفاده شده است [1-4]. در [1] از PSO[4] برای یادگیری وزنهای شبکه ی عصبی ای که رفتار رباتها را کنترل می‌کنند استفاده شده

[1] Evolutionary Computation
[2] Learning Automata
[3] Pheromone
[4] Particle Swarm Optimization

است. در [2] از برنامه ریزی ژنتیک[5] برای تکامل یک تیم فوتبال استفاده شده است. الگوریتم ژنتیک در [3] برای تعیین پارامترهای مربوط به قرار دادن فرومون در تیمی از ربات‌ها استفاده شده است. در [4] از PSO برای تعیین پارامترهای یک تیم نا همگن از عاملها که عمل خوشه بندی را انجام می‌دهند استفاده شده است. اتوماتاهای یادگیر اخیرا بعنوان مدلی برای ایجاد هماهنگی در سیستمهای چند عامله مورد توجه محققان قرار گرفته است[9-5] . در [5] از اتوماتای یادگیر برای ایجاد هماهنگی در بین عاملهایی که عمل خوشه بندی را انجام می‌دهند استفاده شده است. در این مدل هر عامل دارای یک اتوماتای یادگیر است که تعیین می‌کند داده ی حمل شده توسط عامل باید در کجای صفحه ای که عمل خوشه بندی انجام می‌پذیرد قرار داده شود. نشان داده شده است که عامل ها بدون داشتن ارتباط مستقیم با یکدیگر قادر به ایجاد هماهنگی در قرار دادن داده ها بر روی صفحه می‌باشند. در [6] از اتوماتای یادگیر برای یادگیری اعمال بهینه در عاملهای یک تیم فوتبال استفاده شده است. در [7] نشان دهد شده است که از اتوماتای یادگیر سلسله مراتبی[6] می‌توان به عنوان یک یادگیر تقویتی در سیستمهای چند عاملی مورد استفاده قرار گیرد. در [8] از اتوماتای یادگیر برای تعیین پارامتر بهینه کولونی مورچه ها[7] استفاده شده است. در [9] از اتوماتای یادگیر برای هماهنگی در اشتراک از یک خط مشترک ارتباطی در شبکه های بیسیم تک کاره[8] استفاده شده است. در این روش هر عامل دارای یک اتوماتای یادگیر می باشد که با توجه به داده های نویزی تخمین می زند چه عاملی در هر لحظه ارسال پیام خواهد کرد.

فرومونها یکی دیگر از مدلهایی است که که برای در ایجاد هماهنگی در سیستمهای چند عامله مورد استفاده قرار گرفته است.[12-10]. در [10] فرومونها برای تعیین کوتاهترین مسیر بین لانه و غذا در یک سیستم چند عاملی استفاده شده اند. در [11] نشان داده شده است که فرمونها گزینه ی مناسبی برای حل مسئله ی گشت زنی[9] در سیستمهای چند عاملی می‌باشند. در [12] نوع جدیدی فرومون برای سیستمهای چند عاملی معرفی و در حل مسئله ی A^* فیزیکی[10] استفاده شده است.

در این مقاله یک مدل جدید بر پایه اتوماتاهای یادگیر برای ایجاد همکاری در سیستمهای چند عاملی پیشنهاد میگردد. در این مدل در هر حالت محیط یک اتوماتای یادگیر قرار داده می‌شود. وظیفه ی اتوماتای یادگیر تعیین حالت بعدی برای عاملی است که در آن حالت قرار دارد. چنانچه اتوماتای یادگیر عامل را به حالت هدف ببرد، پاداش می‌گیرد در غیر این صورت از مفهوم آنتروپی برای تعیین پاداش و جریمه استفاده می‌شود. برای ارزیابی کارایی روش پیشنهادی از آن برای حل مسئله ی جمع آوری اشیاء[11] استفاده شده است. مقایسه ها با روشهای بر پایه ی فرومون و Q-Learning بعنوان یک روش یادگیری تقویتی نشان می‌دهد که روش پیشنهادی ابزار مناسبی برای ایجاد هماهنگی در سیستمهای چند عاملی می باشد. ادامه این گزارش بدین صورت سازماندهی شده است. در بخش ۲ اتوماتاهای یادگیر بطور خلاصه معرفی می‌شود. بخش ۳ روش پیشنهادی را مطرح می‌کند. در بخش ۴ مسئله ی جمع آوری اشیاء و روشهای حل آن برپایه فرومون و Q-Learning و روش پیشنهادی معرفی می‌شوند. نتایج شبیه سازی و نتیجه‌گیری به ترتیب در بخشهای ۵ و ۶ آمده است.

۲- اتوماتاهای یادگیر

اتوماتای یادگیر یک مدل انتزاعی است که تعداد معدودی 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 حالت درونی خود را به روز می کند و دوباره عمل خود را برای مرحله بعد انتخاب می کند[13]. اتوماتا با تعامل با محیط عمل بهینه را فراگیری می کند و به این شکل پاداش دریافتی خود را از محیط حد اکثر 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 یک 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 α ها 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، β متغیر تصادفی در فاصله [0,1] است. c_i احتمال اینکه عمل α_i نتیجه نامطلوب[12] داشته باشد، می باشد. در محیط ایستا[13] مقادیر c_i بدون تغییر می مانند ، حال آن که در محیط غیر ایستا[14] این مقادیر در طی زمان تغییر می کنند.

[5] Genetic Programming
[6] Hierarchical
[7] Ant Colony
[8] Ad hoc Wireless Network
[9] Patrolling Problem
[10] Physical A^*
[11] Foraging
[12] Unfavorable
[13] Stationary
[14] Non-Stationary

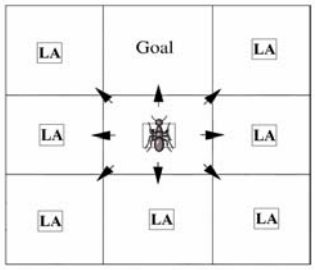
اتوماتای یادگیر به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم می گردد [13]. اتوماتای یادگیر با ساختار متغیر[15] توسط ۴ تایی $\{\alpha,\beta,p,T\}$ نشان داده می شود که در آن $\alpha \equiv \{\alpha_1,\alpha_2,...,\alpha_r\}$ مجموعه عملهای اتوماتا، $\beta \equiv \{\beta_1,\beta_2,...,\beta_m\}$ مجموعه ورودیهای اتوماتا و $p = \{p_1,p_2,...p_r\}$ بردار احتمال انتخاب هر یک از اعمال 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 این عمل، پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ افزایش یافته و سایر احتمالها کاهش می یابند. برای پاسخ نامطلوب احتمال $p_i(n)$ کاهش یافته و سایر احتمالها افزایش می یابند. این تغییرات به گونه ای صورت می پذیرد که جمع احتمالات برابر با یک باقی بماند.. فرمول (۴) یکی از الگوریتمهای یادگیری خطی در اتوماتای با ساختار متغیر مدل S را نشان می دهد. فرض کنید که عمل i توسط اتوماتای یادگیر انتخاب شده باشد و مقدار β توسط محیط به اتوماتا داده شده باشد :

$$\begin{aligned} p_i(n+1) &= p_i(n)+a(1-\beta_i(n))(1-p_i(n))-b\beta_i(n)p_i(n) \\ p_j(n+1) &= p_j(n)-a(1-\beta_i(n))p_j(n)+b\beta_i(n)(\frac{1}{r-1}-p_j(n)) \qquad \forall j,\ j\neq i \end{aligned} \quad (۱)$$

a و b به ترتیب پارامتر پاداش و پارامتر جریمه می باشد. زمانی که a و b با هم برابر باشند، الگوریتم L_{R-P}[16] ، زمانیکه b از a خیلی کوچکتر باشد، الگوریتم $L_{R-\varepsilon P}$[17] و زمانی که b مساوی صفر باشد، الگوریتم L_{R-I}[18] نامیده می شود.

۳- روش پیشنهادی

در این بخش روش پیشنهادی برای یادگیری در سیستمهای چند عاملی به کمک اتوماتای یادگیر را بررسی می‌کنیم. در این مدل در هر حالت محیط یک اتوماتای یادگیر با ساختار متغیر مدل S قرار داده می‌شود. تعداد اعمال اتوماتا در هر حالت برابر است با تعداد حالتهای همسایه ی آن حالت و هر عمل متناظر با یکی از حالات همسایه می‌باشد. هر عامل برای تعیین حالت بعدی خود از اتوماتای یادگیر کمک می‌گیرد؛ به این معنی که عامل به حالت متناظر با عمل انتخاب شده توسط اتوماتا می‌رود. شکل ۲ چگونگی چینش اتوماتای یادگیر در حالتهای یک محیط نشان می‌دهد.



شکل ۲- چینش اتوماتای یادگیر در حالتهای محیط

در ابتدا اتوماتاهای یادگیر تمام عملهای خود را با احتمالی یکسان انتخاب می‌کنند. چنانچه عمل اتوماتا منجر به ورود عامل به حالت هدف شود اتوماتا پاداش می‌گیرد، در غیر اینصورت از آنتروپی بردار احتمال اتوماتای یادگیر حالت بعد برای تعیین پاداش یا جریمه استفاده می‌شود. آنتروپی بردار احتمال میزان عدم اتوماتای یادگیر حالت بعد را در انتخاب عمل خود نشان می‌دهد. هر چه آنتروپی بیشتر باشد میزان عدم قطعیت بیشتر است. عدم قطعیت بالا در بردار احتمال اتوماتای یادگیر به این معنی است که این اتواماتای یادگیر دارای اطلاعات مفیدی برای رسیدن به هدف نیست و عملهای خود را به صورت تصادفی انتخاب می‌کند. ولی چنانچه عدم قطعیت کم باشد به این معنی است که اتوماتای یادگیر با احتمال بالایی یکی از اعمال خود را انتخاب می‌کند و دارای اطلاعات مفیدی برای رسیدن به هدف می‌باشد. فرض کنید که $\{p_1,p_2,...p_r\}$ بردار احتمال اعمال یک اتوماتای یادگیر باشد . آنتروپی این بردار احتمال به شکل زیر تعیین می‌شود.

$$E(P) = -\sum_{i=1}^{r} p_i \log(p_i) \quad (۲)$$

هرچه میزان عدم قطعیت بیشتر باشد مقدار آنتروپی بیشتر خواهد بود. زمانی E بیشترین مقدار را خواهد داشت که تمام اعمال احتمالی یکسان داشته باشند($P_{equal}=\{p_1=p_2=...p_{r=1/r}\}$) و زمانی کمترین مقدار(برابر با ۰) را خوهد داشت که $\exists i\ p_i = 1 \wedge \forall\ j \neq i\ p_j = 0$. برای اینکه مقدار انتروپی را به مقداری بین ۰ و ۱ تبدیل کنیم تا به عنوان بردار تقویتی در اتوماتای ساختار متغیر مدل S قابل استفاده باشد از فرمول زیر استفاده می‌شود.

$$\beta = E((P)/E(P_{equal}))^K \quad (۳)$$

K پارامتر روش می‌باشد. میزان این پارامتر باید به دقت تعیین شود. مقادیر بالای این پارامتر باعث می‌شود که β ها مقادیر کمی داشته باشند و اتوماتاهای یادگیر بیش از حد پاداش ببینند. این امر باعث می‌شود که انتخاب یک عمل با احتمال بیشتری پاداش بگیرد. به این شکل اتوماتاها با احتمال بیشتری حالتهای نه چندان مناسب را انتخاب

[15] Variable Structure Learning Automata
[16] Linear Reward Penalty
[17] Linear Reward Epsilon Penalty
[18] Linear Reward Inaction

می‌کنند و این به معنی جستجوی[19] بیشتر در محیط است. زمانی که K مقدار بالایی داشته باشد این احتمال وجود دارد که عاملها تشکیل یک حلقه دهند و به دنبال یکدیگر شروع به حرکت کنند. به این شکل عاملها ممکن است در یک حلقه بسته قرار بگیرند. هر چه میزان K کمتر باشد β ها مقدایر بیشتری خواهند داشت. این امر باعث می‌شود که اتوماتاها بیشتر جریمه شوند. این مسئله باعث می‌شود که حتی حالتهای مطلوب نیز پاداش لازم را نگیرند. به این شکل عاملها قادر به یافتن مسیر مناسبی به سمت هدف نخواهند بود و در صفحه بی‌هدف پرسه می‌زنند. برای تعیین میزان نرخ جستجو(Exploration) و بهره برداری(Exploitation) می‌توان از تغییر پارامتر یادگیری(a در فرمول ۱) استفاده کرد. هرچه نرخ یادگیری بیشتر باشد عاملها حالتهای خوب را با احتمال بیشتری در ادامه انتخاب خواهند کرد ولی چنانچه مقدار این پارامتر کم باشد، عاملها دیرتر به سمت حالتهای خوب جذب می‌شوند و بیشتر به جستجو می‌پردازند. فرض کنید که عامل در حالت s باشد و اتوماتای یادگیر آن(LA_s) عامل را به حالت s' هدایت کند. در اینصورت تعیین سیگنال تقویتی به این شکل تعیین می‌شود :

$$\beta_s = \begin{cases} 1 & s' = Out\ of\ bound \\ 0 & s' = Goal \\ (E(P(LA_{s'})) / E(P_{equal}))^K & otherwise \end{cases} \qquad (۴)$$

$P(LA_s)$ نشان دهنده ی بردار احتمال اتوماتای یادگیر قرار گرفته در حالت s می‌باشد. چنانچه چندین عامل در یک حالت قرار داشته باشند عاملها به صورت تصادفی انتخاب می‌شوند و انتخاب حالت بعدی و آموزش اتوماتای یادگیر انجام می‌شود. به علت شباهتی که این روش به Q-Learning دارد آن را QLA نام گذاری کرده ایم. شبه کد این الگوریتم را می‌توان به شکل زیر نوشت :

```
Initialize.
While not done do
   For each agent i do in parallel
      Activate LA_i (LA residing in current state of agent i (S_i))
      Move to S_i' (next state selected by LA_i)
      Compute β_i (reinforcement signal of LA_i)
      Train LA residing in S_i according to β_i
   End parallel for
End while
```

شکل ۳- شبه کد QLA

۴- مسئله ی جمع آوری اشیاء

در این بخش مسئله ی جمع آوری اشیاء را مطرح می‌کنیم و توضیح می‌دهیم که این مسئله چگونه توسط Q-Learning ، فرومون ها و روش پیشنهادی قابل حل می‌باشد.

۴-۱ تعریف مسئله

در این مسئله تعدادی عامل بر روی صفحه ای دو بعدی مانند اتوماتای سلولی قرار دارند. عامل ها از یک لانه شروع به جستجو برای یافتن غذا (اشیاء) می‌کنند. هنگامی که این عاملها به منبع غذا برسند یک واحد از آن را برمی‌دارند و سعی می‌کنند آن را به لانه تحویل دهند. عاملها تنها در سلولهای صفحه می‌توانند حرکت کنند و بر خلاف اتوماتای سلولی صفحه مدور[20] نمی‌باشد. یعنی عامل نمی‌تواند از یک طرف صفحه خارج شود و از طرف دیگر وارد شود. کارایی یک تیم از عامل ها در این مسئله برابر است با تعداد واحدهایی از غذا که در مدت زمان مشخصی به لانه تحویل داده می‌شود. بدیهی است که هرچه مسیری که عاملها از لانه به غذا و برعکس طی می‌کنند کوتاهتر باشد و عاملهای بیشتری در جمع آوری اشیاء شرکت کنند کارایی تیمی بیشتر خواهد شد. این مسئله را می‌توان به دو زیر مسئله ی یافتن کوتاهترین مسیر از لانه به غذا و از غذا به لانه تقسیم کرد. در این مسئله عاملها حافظه ای ندارند و قادر تشخیص مکان خود در صفحه نمی‌باشند و تنها به اطلاعاتی که در سلولهای همسایه توسط دیگر عاملها نوشته شده دسترسی دارند. هر عامل می‌تواند این داده ها را بخواند و با توجه به آنها برای مسیر حرکت خود تصمیم گیری کند. علاوه بر این، عامل می‌تواند این داده ها را تغییر دهد. عاملها باید به کمک هم و با نوشتن داده های مناسب با یکدیگر همکاری کرده تا کوتاهترین مسیر از لانه به غذا و برعکس را بیابند. برای اطلاعات بیشتر در مورد این مسئله می‌توانید به [14] مراجعه کنید. در این مقاله سه روش را برای حل این مسئله بررسی و آنها را با هم مقایسه می‌کنیم. این ۳ روش عبارتند از Q-Learning ، فرومونها و روش پیشنهادی.

۴-۲ حل مسئله با Q-Learning

Q-Learning [15] یکی از روشهای یادگیری تقویتی است که به علت کارایی بالا و سادگی استفاده در کابردهای فراوانی مورد استفاده قرار گرفته است.. در این روش در هر حالت مسئله یک آرایه به نام Q وجود دارد. طول این آرایه برابر است با تعداد حالتهای همسایه ی آن حالت. هر خانه ی آرایه متناظر است با یکی از حالتهای همسایه و مقدار موجود در آن خانه میزان مطلوبیت همسایه ی متناظر را نشان می‌دهد. فرض کنید که یک عامل در حالت s قرار داشته باشد. عامل برای انتخاب حالت بعد با توجه به مقادیر Q تصمیم گیری می‌کند. یکی از روشهای متداول برای انتخاب عمل برابر است با انتخاب حالت بعد با احتمال متناسب با مقدار Q متناظر با آن. به عنوان مثال به شکل زیر :

$$P(s,a) = Q(s,a)^K \ / \ \sum_{a \in A_s} Q(s,a)^K \qquad (۵)$$

[19] Exploration

[20] Wraparound

در این الگوریتم هرچه میزان K بیشتر باشد الگوریتم بیشتر به بهره برداری(Exploitation) و هرچه کمتر باشد بیشتر به جستجو(Exploration) می‌پردازد. فرض کنید که عامل در حالت s باشد و بردار Q عامل را به حالت s' هدایت کند. در اینصورت بردار Q حالت s به این شکل به روز می‌شود :

$$Q(s,a) = r + \gamma \max_{a' \in A_{s'}} (Q(s', a')) \quad (۶)$$

A_S برابر است با مجموعه اعمال ممکن در حالت s و γ پارامتر الگوریتم می‌باشد که معمولا مقداری نزدیک به یک در نظر گرفته می‌شود. برای حل مسئله ی جمع آوری اشیاء توسط این روش در هر سلول صفحه دو آرایه ی Q با طول ۸ قرار داده می‌شود. Q_1 برای یافتن کوتاهترین مسیر از لانه به سمت غذا و Q_2 برای مسیر برعکس استفاده می‌شود. چنانچه عامل حامل غذا نباشد به سلول غذا وارد شود و یا چنانچه عامل حامل غذا باشد و به سلول لانه وارد شود پاداش می‌گیرد. توجه کنید که زمانی که عاملها حامل غذا باشند تنها به Q_2 دسترسی دارد و آن را به روز می‌کند در غیر اینصورت تنها به Q_1 دسترسی دارد و آن را به روز می‌کند.

۳-۴ حل مسئله با فرومون

برای حل این مسئله از فرومونها نیز استفاده شده است[10]. در این روش عاملها زمانی که حامل غذا نباشند فرمون "به سمت لانه" بر روی سلولها قرار می‌دهند و فرومون "به سمت فرومونهای غذا" را دنبال می‌کنند. عاملها هر چه از لانه دور شوند میزان فرومونی که برروی سلولها قرار می‌دهند کاهش می‌یابد. زمانی که عامل به منبع غذا برسد یک واحد از آن را بر می‌دارد و پس از آن در هنگام حرکت، فرمون "به سمت غذا" بر روی سلولها قرار می‌دهد و فرمون "به سمت لانه" را دنبال می‌کند. لانه و غذا همیشه به ترتیب دارای بیشترین مقدار فرومون به سمت لانه و فرمون به سمت غذا می‌باشند. عاملها در این روش برای قرار دادن فرمون از فرمول زیر استفاده می‌کنند :

$$Phr(s) = \max_{s' \in N(s)} (Phr(s')) - 2 \quad (۷)$$

$Phr(s)$ برابرست با با مقدار فرومون در سلول s و $N(s)$ برابر است با ۸ همسایه مجاور سلول s. عاملها سلولهای همسایه را با احتمالی متناسب با مقدار فرومون در آن سلولها از طریق فرمول زیر انتخاب می‌کنند.

$$p(s') = (phr(s') + c)^K \quad (۸)$$

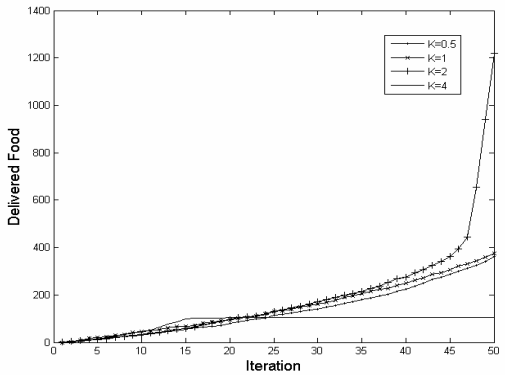
همانند Q-Learning در این الگوریتم هرچه میزان K بیشتر باشد الگوریتم بیشتر به بهره برداری (Exploitation) و هرچه کمتر باشد بیشتر به جستجو (Exploration) می‌پردازد. برای جلوگیری از گیر افتادن در مینیمم محلی در هر تکرار فرومونها در ضریبی به نام ρ که کوچکتر از ۱ می‌باشد ضرب می‌شوند.

۳-۴ حل مسئله با QLA

برای حل این مسئله در هر سلول صفحه دو اتوماتای یادگیر با ۸ عمل قرار می‌دهیم. هر عمل متناظر است با یکی از خانه های همسایه. یک اتوماتا وظیفه هدایت عاملها به سمت غذا و دیگری وظیفه هدایت عاملها به سمت لانه را دارد. بسته به اینکه عامل حامل غذا باشد یا خیر، یکی از دو اتوماتای یادگیر غذا یا لانه وظیفه ی هدایت عامل را بر عهده دارد. همانند شکل ۲ اتوماتای یادگیر در سلولهای صفحه قرار می‌گیرند و زمانی که یک عامل در آنها قرار بگیرد اتوماتای یادگیر محل بعدی آن را در صفحه تعیین می‌کند. عامل نیز با توجه به به فرمول (۴) مقدار سیگنال تقویتی را تعیین می‌کند و با توجه به فرمول (۱) بردار احتمال اتوماتای یادگیر را به روز می‌کند.

۵- نتایج شبیه سازیها

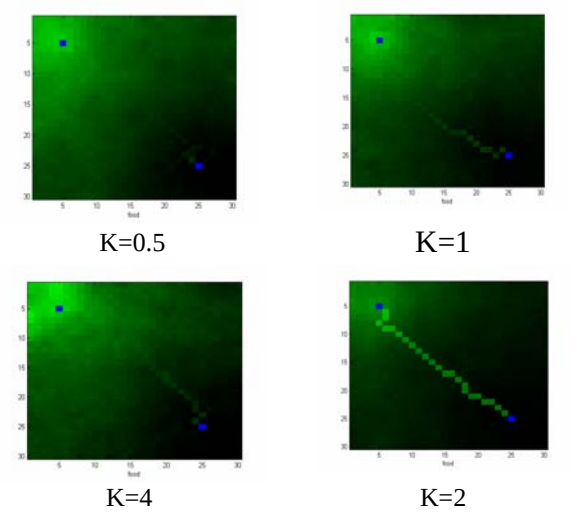
در آزمایش اول تاثیر پارامتر K را بر روی الگوریتم پیشنهادی بررسی کرده ایم. در این آزمایش از یک صفحه ی ۳۰×۳۰ استفاده کرده‌ایم. لانه در خانه‌ی (۵،۵) و غذا در خانه‌ی (۲۵،۲۵) قرار دارد. در QLA از روش یادگیری L_{RI} با a و b به ترتیب برابر با 0.3 و 0 استفاده شده است. K در تکرارهای مختلف برابر با 0.5، 1، 2 و 4 قرار داده شده است. تعداد عاملها در تمام آزمایشها برابر با ۱۰ قرار داده شده است. در شکل ۴ میزان غذای تحویل داده شده به لانه را در ۵۰۰۰۰ تکرار می‌بینید. شکل ۴ میانگین ۲۰ تکرار را گزارش می‌کند.



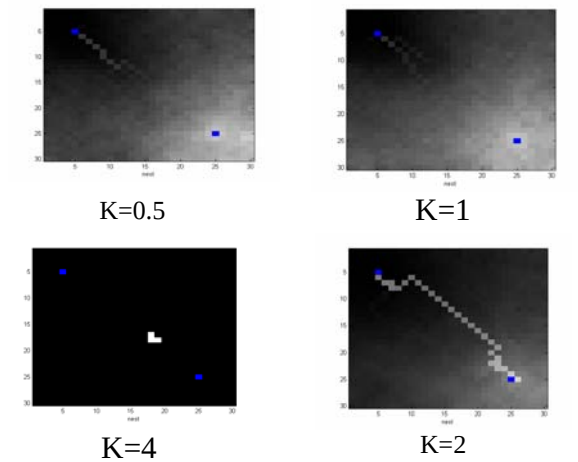
شکل ۴ - میانگین غذای تحویل داده شده به لانه توسط QLA با K های مختلف

همانطور که می‌بینید K=2 در این آزمایش بهترین جواب را تولید کرده است. بقیه ی مقادیر K برای الگوریتم مناسب نمی‌باشند. برای نمایش مسیر حرکت عامل ها بر روی آنها را به فرومون مجهز کرده‌ایم.

دقت کنید که این فرومونها در QLA صرفا جنبه ی نمایش دارند و هیچ تاثیری در الگوریتم ندارند. . عاملها زمانی که حرکت می‌کنند بر روی سلولها فرومون قرار می‌دهند. به این شکل روی مسیری که عاملها طی می‌کنند فرومون بیشتری جمع می‌شود. با تبدیل این فرومونها به تصویر، مسیری که عاملها در هر روش طی می‌کنند مشخص می‌شود. شکل ۵ مسیر طی شده از غذا به سمت لانه و شکل ۶ مسیر طی شده از لانه به سمت غذا به ازای K های مختلف را در انتهای ۵۰۰۰۰ اجرا نشان می‌دهد. در شکلهای ۵ و ۶ هرچه میزان فرومون در یک سلول بیشتر باشد آن سلول روشنتر نشان داده شده است.



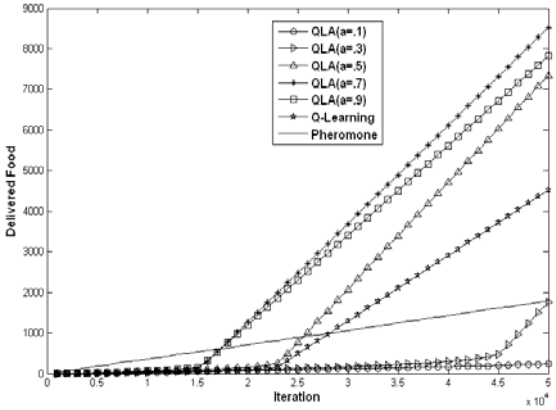
شکل ۵ - مقایسه ی مسیر طی شده از لانه به سمت غذا توسط QLA(a=.3,b=0) با K های مختلف



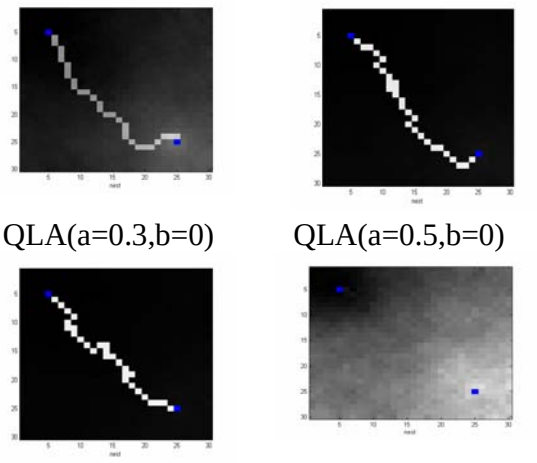
شکل۶- مقایسه ی مسیر طی شده ازغذا به سمت لانه توسط QLA(a=.3,b=0) با K های مختلف

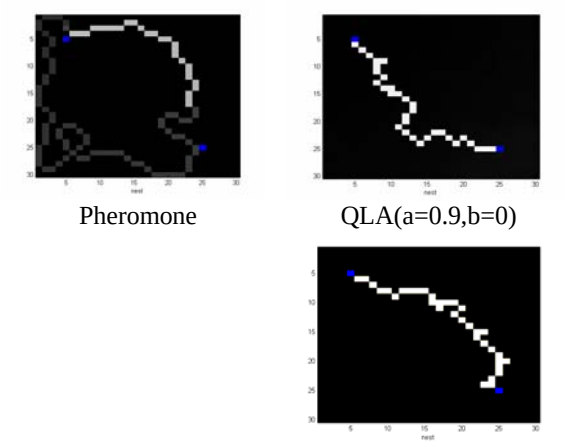
همانطور که در شکلهای ۵ و۶ می‌بینید K=2 در این آزمایش بهترین مسیر را یافته است. این شکلها نتیجه ی شکل ۴ را تأیید می‌کنند. همانطور که در بخش ۳ مطرح کردیم مقادیر بالای K باعث می‌شود که در عاملها در یک حلقه به دنبال یکدیگر شروع به حرکت کنند(شکل ۶ پایین سمت راست).

برای بررسی بیشتر ۲ آزمایش دیگر ترتیب داده ایم. در آزمایش اول عاملها در یک صفحه ی ۳۰×۳۰ می‌تونند حرکت کنند. لانه در خانه (۵،۵) و غذا در خانه ی (۲۵،۲۵) قرار دارد. مقدار K در روش Q-Learning، فرومون و QLA به ترتیب برابر با ۱۰ ، ۱۰ و ۲ قرار داده شده است. در روش بر پایه ی فرومون مقدار c برابر با 0.001 قرار داده شده است.در روش QLA از روش یادگیری L_{RI} با a های برابر با 0.1، 0.3، 0.5 ، 0.7 و 0.9 و b برابر با 0 استفاده شده است تا تاثیر نرخ یادگری در عملکرد الگوریتم را بیابیم. در روش Q-Learning میزان پاداش(r) برای ورود به لانه و غذا برابر ۵ و γ برابر با 0.95 قرار داده شده است. در شکل ۷ میزان غذای تحویل داده شده به لانه را در ۵۰۰۰۰ تکرار می‌بینید. شکل ۷ میانگین ۲۰ تکرار را گزارش می‌کند.

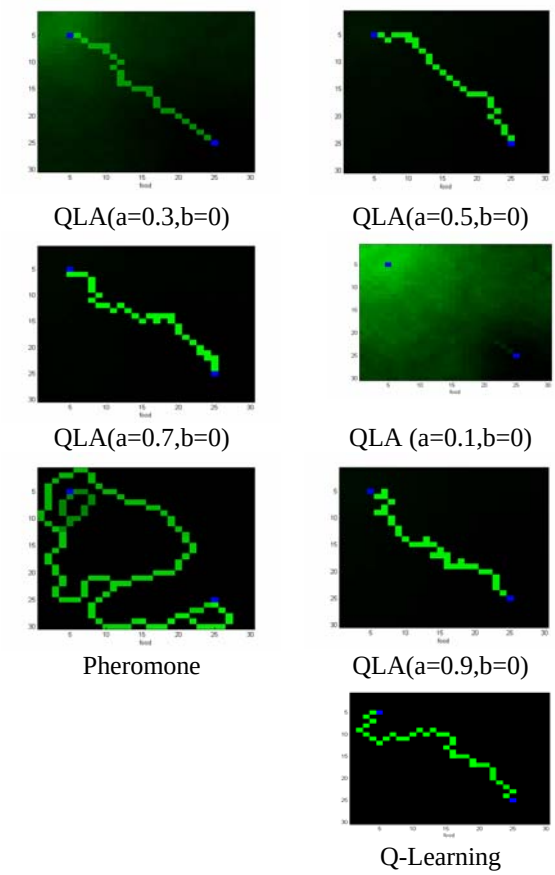


شکل ۷- میانگین غذای تحویل داده شده به لانه توسط سه روش مورد بررسی در صحفه ی ۳۰×۳۰





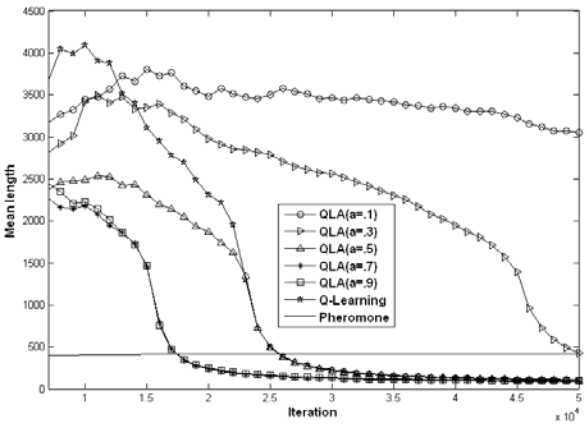
شکل ۸.- مقایسه ی مسیر طی شده از غذا به سمت لانه در ۳ روش مورد بررسی در صفحه ی ۳۰×۳۰



شکل ۹- مقایسه ی مسیر طی شده از لانه به سمت غذا در ۳ روش مورد بررسی در صفحه ی ۳۰×۳۰

همانطور که مشاهده میشود روش ارائه شده با نرخ یادگیری بالا دارای کارایی بسیار بالایی در مقایسه با Q-Learning و فرومونها می‌باشد. از میان نرخهای یادگیری، مقدایر0.7 و 0.9 همگرایی سریعی دارند و مسیرهای خوبی می‌یابند. البته نرخ بالای یادگیری باعث گرفتار شدن در مینیمم محلی نیز می‌شود. همانطور که در شکل بالا می‌بینید نرخ یادگیری 0.9 با وجود همگرایی سریع، نسبت به مقدار 0.7 دارای کارایی کمتری می‌باشد. دلیل این امر را می‌توان گرفتار شدن در مینیمم محلی دانست. راز کارایی بالای QLA در مسیری است که عاملها توسط این روش بین لانه و غذا می‌پیمایند. همانند آزمایش قبل از فرومونها برای نمایش مسیر طی شده توسط عاملها استفاده کرده ایم. شکل ۸ مسیر طی شده از غذا به سمت لانه و شکل ۹ مسیر طی شده از لانه به سمت غذا در ۳ روش مورد بررسی را در انتهای ۵۰۰۰۰ اجرا نشان می‌دهد.

همانطور که مشاهده میشود نرخهای یادگیری بزرگتر از 0.3 قادر به یافتن مسیرهای خوبی بین لانه و غذا می‌باشند. البته همانطور که در قبل مطرح کردیم مسیرهای یافته شده نشان می‌دهد که نرخ یادگیری 0.9 با وجود همگرایی بالا گاهی در مینیمم محلی گرفتار می‌شود. مسیر طی شده توسط QLA به خط راست بین لانه و غذا نزدیک است. عاملها تقریبا هیچگاه از مسیر پیدا شده خارج نمی‌شوند. شکلهای ۸ و ۹ نشان می‌دهد روش پر پایه ی فرومون ها و Q-Learning جواب مناسبی را نتوانسته اند بیابند. برای مقایسه ی بیشتر، در تمام روش ها میانگین مسیر طی شده برای آوردن غذا ها به لانه در هر تکرار محاسبه شده است. شکل ۱۰ میانگین این پارامتر را در ۲۰ تکرار نشان می‌دهد.

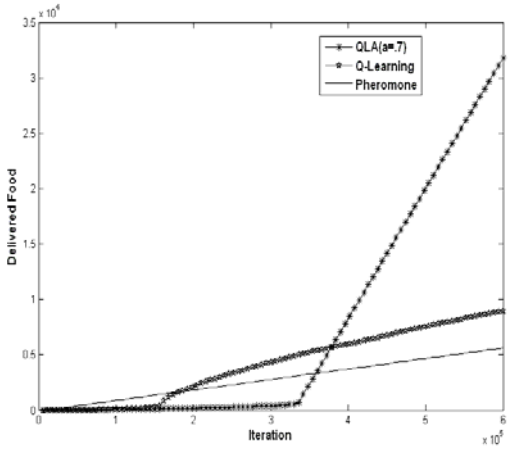


شکل ۱۰ - میانگین مسیر طی شده برای تحول غذا به لانه توسط سه روش مورد بررسی در طول اجرا در صفحه‌ی ۳۰×۳۰

شکل ۱۰ نشان می‌دهد که روش بر پایه ی فرومونها به سرعت مسیر نسبتا خوبی بین لانه و غذا پیدا می‌کند ولی پس از آن قادر به بهبود مسیر یافته شده نیست. پس از آن سریعترین همگرایی مربوط به QLA با نرخ های یادگیری 0.7 و 0.9 می‌باشد.

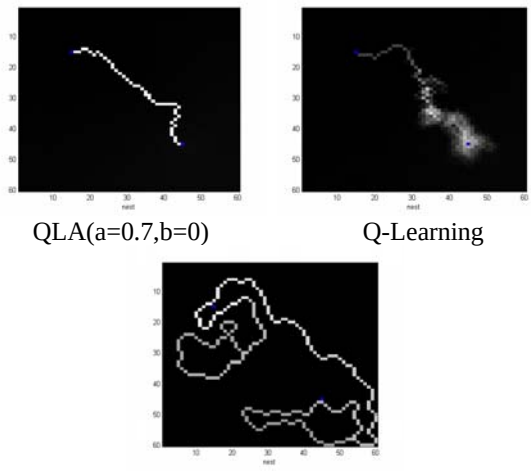
آزمایش قبل را برای صفحه ای ۶۰×۶۰ نیز تکرار کرده ایم. لانه در خانه (۱۵،۱۵) و غذا در خانه ی (۴۵،۴۵) قرار دارد. از QLA(a=0.7,b=0) استفاده کرده ایم. سایر پارامترها مشابه آزمایش

قبل استفاده شده اند. در شکل ۱۱ میزان غذای تحویل داده شده به لانه را در ۶۰۰۰۰۰ تکرار می‌بینید. شکل ۱۱ میانگین ۲۰ تکرار را گزارش می‌کند.



شکل ۱۱- میانگین غذای تحویل داده شده به لانه توسط سه روش مورد بررسی در صفحه ی ۶۰×۶۰

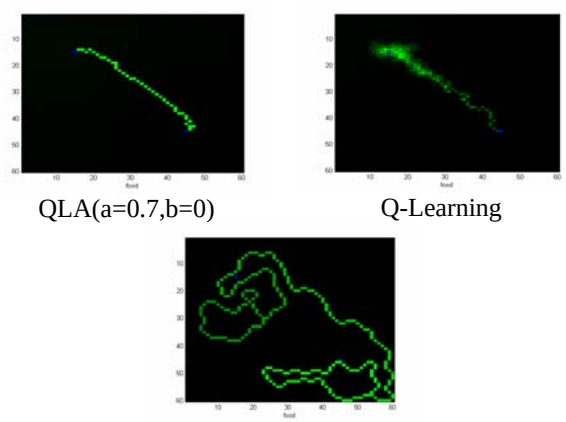
همانطور که مشاهده میشود a=0.7، K=2 و b=0.0 پارامترهای مناسبی برای روش ارائه شده می‌باشند شکل ۱۲ و ۱۳ به ترتیب نمونه مسیر یافته شده از غذا به لانه و برعکس را در ۳ روش مورد بررسی نشان می‌دهد.



QLA(a=0.7,b=0) Q-Learning

Pheromone

شکل ۱۲ - مقایسه ی مسیر طی شده از غذا به سمت لانه در ۳ روش مورد بررسی در صفحه ی ۶۰×۶۰

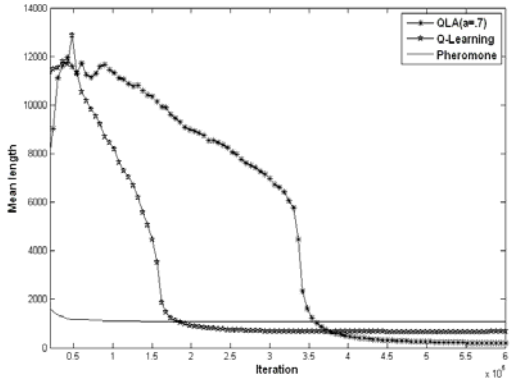


QLA(a=0.7,b=0) Q-Learning

Pheromone

شکل - ۱۳ مقایسه ی مسیر طی شده از لانه به سمت غذا در ۳ روش مورد بررسی در صفحه ی ۶۰×۶۰

همانطور که مشاهده میشود زمانی که طول مسیر بین لانه و غذا زیاد شده است افت مقادیر Q باعث سرگردانی عاملها پس از خروج از لانه و پرسه زدن آنها می‌شود. برای مقایسه ی بیشتر، در تمام روش ها میانگین مسیر طی شده برای آوردن غذا ها به لانه در هر تکرار محاسبه شده است. شکل ۱۴ میانگین این پارامتر را در ۲۰ تکرار نشان می‌دهد.



شکل ۱۴- میانگین مسیر طی شده برای تحول غذا به لانه توسط سه روش مورد بررسی در طول اجرا در صفحه‌ی ۶۰×۶۰

همانند آزمایش قبل مشاهده میشود که QLA مسیر بهتری نسبت به Q-Learning و فرومون ها می‌یابد و غذای بیشتری جمع آوری می‌کند.

۶- نتیجه گیری

در این مقاله روشی برای ایجاد هماهنگی در بین عاملهای یک سیستم چند عاملی بر پایه ی اتوماتای یادگیر پیشنهاد و سپس از آن برای حل مسئله ی جمع آوری اشیاء استفاده گردید. مقایسه روش پیشنهادی با روشهای Q-Learning و فرومونها نشان داد که روش

پیشنهادی ابزار مناسبی برای ایجاد هماهنگی در سیستمهای چند عاملی می‌باشد. دلیل کارایی بالای روش پیشنهادی را می‌توان در استفاده از مزیت هر دو روش Q-Learning و فرومونها دانست.

مراجع

[1] Pugh, J. and Martinoli, A., "Multi-Robot Learning with Particle Swarm Optimization", Proceedings of International Conference on Autonomous Agents and Multiagent Systems, pp. 441 - 448, 2006.

[2] Haynes, T and Sen, S., "Crossover operators for evolving a team". In J. R. Koza, K. Deb, M. Dorigo, D. B. Fogel, M. Garzon, H. Iba and R. L. Riolo, editors, Genetic Programming 1997: Proceedings of the Second Annual Conference, pages 162–167, Stanford University, CA, USA, 13-16 July 1997.

[3] Sauter, J., Matthews, R., Van Dyke Parunak, H. and Brueckner, S., "Evolving adaptive pheromone path planning mechanism"s. In Proceedings of First International Joint Conference on Autonomous Agents and Multi-Agent Systems (AAMAS-02), pp. 434–440, 2002.

[4] Jafarpour**,** B. and Meybodi, M. R., "Adaptation of Ant Clustering Parameters using CLA-PSO ", The 1st National Data Mining Conference (IDMC'2007), Tehran, Iran, 2007.

[5] Khojasteh, M. R. and Meybodi, M. R., "Evaluating Learning Automata as a Model for Cooperation in Complex Multi-Agent Domains", Lecture Notes in Artificial Intelligence, Springer Verlag, LNAI 4434, pp. 409-416, 2007.

[6] Jafarpour**,** B. and Meybodi, M. R., "Improving Ant based Clustering Using Learning Automata", The 13th International Computer Society of Iran Conference (CSICC'2008), Kish Island, Iran, 2008.

[7] Ann Nowé, Katja Verbeeck and Maarten Peeters: "Learning Automata as a Basis for Multi Agent Reinforcement Learning", LAMAS 2005.

[8] Abdali, F. and Meybodi, M. R., "Adaptation of Ants Colony Parameters Using Learning Automata", Proceedings of 10th Annual CSI Computer Conference Iran, Telecommunication Research Center, Tehran, Iran, pp. 972-980, Feb. 2005.

[9] Aggarwal, P. S. and Chenhui Liu, "Coordination within Multiple Learning Automata Agents: a Novel Distributed Permission Switching Protocol", International Conference on Integration of Knowledge Intensive Multi-Agent Systems, 2005.

[10] Panait, L. and Luke, S., "Ant Foraging Revisited", Proceedings of the Ninth International Conference on the Simulation and Synthesis of Living Systems (ALIFE9). 2003.

[11] Chu, Hoang-Nam, Glad, A., Simonin, O., Sempé,F., Drogoul, A. and Charpillet, F., "Swarm Approaches for the Patrolling Problem, Information Propagation vs. Pheromone Evaporation", Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'2007), Patras, Greece, October, 2007.

[12] Felner, A., Shoshani, Y., Wagner, I. A. and Bruckstein, A. M., "Multi-agent Physical A* using Large Pheromones", Proceedings of the Third International Joint Conference on Autonomous Agents and Multiagent Systems, 2004. AAMAS 2004.

[13] Narendra, K. S. and Thathachar, M. A. L., Learning Automata: An introduction, Prentice Hall, 1989.

[14] Sen, S and Weiss, G., Multi-Agent Systems: A Modern Approach to Distributed Artificial Intelligence, MIT Press, 1999.

[15] Watkins,C., "Learning from Delayed Rewards", Thesis, University of Cambridge, England, 1989.