

## یک الگوریتم خود تطبیق جدید مبتنی بر اتوماتای یادگیر برای مسئله تطبیق توپولوژی در شبکه‌های نظیر به نظیر

علی محمد صغیری<sup>۱</sup>، محمد رضا میبدی<sup>۲</sup>

<sup>۱</sup>دانشگاه صنعتی امیرکبیر، a\_m\_saghiri@aut.ac.ir

<sup>۲</sup>دانشگاه صنعتی امیرکبیر، mmeybodi@aut.ac.ir

چکیده - شبکه‌های نظیر به نظیر یک شبکه روگستر بر روی شبکه فیزیکی ایجاد می‌کنند. مدیریت توپولوژی شبکه روگستر در شبکه‌های نظیر به نظیر یک مسئله چالش برانگیز است و عدم تطبیق توپولوژی شبکه روگستر با شبکه فیزیکی یکی از مسائل مطرح در این نوع از شبکه‌ها است. با توجه به عدم وجود یک مدیریت مرکزی و پویایی شبکه، طراحی الگوریتم‌های مدیریتی به شکل خود سازمانده برای حل مشکل ذکر شده امری بدبیهی است، ولی آنچه روش‌های موجود را به جالش می‌کشد عدم تطبیق الگوریتم‌های مدیریتی با شرایط شبکه به شکل برخاست. با پهنه گیری از اتوماتای یادگیر در این مقاله یک الگوریتم خود تطبیق جدید برای این مسئله ارائه می‌شود. الگوریتم ارائه شده با الگوریتم جابجایی همسایگان که یکی از الگوریتم‌های استاندارد در این حوزه است از نظر مقیاس پذیری و بهینه سازی تابع تاخیر انتهای به انتها مقایسه شده است که نتایج نشان دهنده کارایی الگوریتم ارائه شده است.

کلمات کلیدی: شبکه‌های نظیر به نظیر، مسئله تطبیق توپولوژی، اتوماتای یادگیر

قدرت کامپیوترهای کوچک باعث به وجود آمدن مدل نظیر به نظیر شده است [۱].

### - ۱- مقدمه

شبکه‌های نظیر به نظیر<sup>۲</sup> یک نمونه ارسیستم‌های توزیع شده هستند که براساس تبادل اطلاعات بین نظیرها و بدون وابستگی به نقطه‌ای خاص از سیستم طراحی می‌شوند. در این مدل هر نظیر همزمان هم خدمتگزار است و هم مشتری و تعداد بسیار

با افزایش قدرت پردازش کامپیوترهای شخصی و کاهش ارزش قطعات کامپیوترها، شاهد ایجاد مدل‌های توزیع شده متفاوت هستیم. مدل مشتری خدمتگزار<sup>۱</sup> تغییر کرده و انگیزه استفاده از

<sup>2</sup> Peer to peer

<sup>1</sup> Client-Server

## ۲- کارهای مرتبط

مسئله تطبیق توپولوژی دارای متغیرهای تصمیم متعددی است که تنظیم خود به خودی هر یک با توجه به شرایط شبکه میتواند چالش برانگیز باشد. این مسئله در هر دو نوع از شبکه‌های ساختارمند و بی ساختار مطرح شده است و الگوریتم‌های نیز برای آن ارائه شده است [12]-[10].

الگوریتم‌های متعددی در [10][13][14][12][15] برای حل این مسئله ارائه شده اند و از جمله مشکلات الگوریتم‌های موجود می‌توان به موارد زیر اشاره نمود.

- عدم در نظر گرفتن پویایی شبکه [10][13]
- عدم وجود مکانیزم‌هایی به منظور تنظیم خود به خودی الگوریتم‌ها [10][13]
- عدم وجود تحلیل در مورد همگرایی الگوریتم‌ها [12][14][15]

یکی از مشکلات پروتکل‌های موجود در تنظیم پارامترهایی است که برای اجرای درست الگوریتم بایستی به شیوه‌ای تنظیم شده باشند. به عنوان مثال در [10] یک رویه برای تطبیق توپولوژی ارائه شده است ولی پارامتر شاعع همسایگی تاثیر وسیعی بر ترافیک و نرخ بهینه سازی توپولوژی دارد کاملاً رها شده است. تنظیم این پارامتر نیز بایستی به شکل خود مختار و با توجه به شرایط محیطی و همسایگی تعیین شود. در [12] نیز روشی که برای تطبیق توپولوژی ارائه شده است، که یک روش غیر هوشمند را پیشنهاد میکند که به راحتی باعث نوعی جابجایی در شبکه میگردد که ممکن است توقف ناپذیر باشد.

الگوریتم ارائه شده در این مقاله اولین الگوریتمی است که تاکید بر تنظیم خود به خودی الگوریتم تطبیق توپولوژی با توجه به شرایط شبکه دارد.

## ۳- اتوماتای یادگیر

اتوماتای یادگیر یک واحد تصمیم گیر تطبیق پذیر است که میتواند یک عمل بهینه را از طریق تعامل با محیط بیابد [5],[6]. اتوماتای یادگیر برای عملکرد خود نیاز به اطلاعات اضافی ندارد به عبارت دیگر عدم استفاده از اطلاعات اضافی که از یک سو یک

زیبادی ناظر در یک محیط کاملاً پویا و توزیع شده با هم همکاری داشته و منابع و فایل‌های خود را در اختیار هم قرار می‌دهند. این شبکه‌ها در سال‌های اخیر بسیار پرکاربرد شده اند [2], [3]. شبکه‌های ناظر به ناظر را می‌توان به سه دسته تقسیم کرد که عبارتند از: شبکه‌های ساختارمند، بی ساختار و دورگه. در شبکه‌های ساختارمند توپولوژی شبکه، ویژگی‌های مشخصی دارد ولی در شبکه‌های بی ساختار توپولوژی ویژگی مشخصی نخواهد داشت. در شبکه‌های دورگه مبتنی بر شرایط شبکه میتوان ترکیبی از دو نوع ساختارمند و بی ساختار را ایجاد کرد. یکی از مهمترین مشکلات این شبکه‌ها، مدیریت توپولوژی است چرا که در یک شبکه ناظر به ناظر، ناظرها بدون نظام خاصی به یکدیگر متصل می‌شوند و هیچ یک از ناظرها ویژگی‌های پایداری ندارد. شرایط شبکه دائماً در حال تغییر است و گاهی تغییرات قابل پیش‌بینی نخواهد بود. در صورتی که مکانیزم‌های کنترلی کارآمدی در این شبکه وجود نداشته باشند به سرعت شرایط توپولوژی شبکه به شرایط غیر قابل قبول سوق پیدا خواهد کرد [4]. یکی از مسائلی که در مدیریت توپولوژی شبکه‌های ناظر به ناظر به آن پرداخته می‌شود مسئله عدم تطبیق توپولوژی است. مسئله عدم تطبیق توپولوژی شبکه روگستر با شبکه فیزیکی، که تاثیر وسیعی بر تاخیر و ترافیک شبکه ایجاد میکند، به چالشی جدی پیش روی این شبکه‌ها بدل گشته است. در این بخش به منظور حل مشکل ذکر شده، الگوریتم خود تطبیق جدیدی مبتنی بر اتوماتای یادگیر طراحی می‌گردد. اتوماتای یادگیر یک واحد تصمیم گیر خود تطبیق است [5], [6] که کاربردهای متعددی در شبکه‌های کامپیوتری داشته [9]-[7] و این قابلیت را دارد که در حل مسئله تطبیق توپولوژی به کار گرفته شود. در بخش دوم کارهای مرتبط بررسی می‌شوند. در بخش سوم مروری بر مفاهیم اتوماتای یادگیر ارائه می‌شود. در بخش چهارم شرایط مسئله مورد بررسی قرار میگیرد و در بخش پنجم یک الگوریتم جدید مبتنی بر اتوماتای یادگیر ارائه میگردد. در بخش ششم و هفتم نتایج حاصل از شبیه سازی و نتیجه گیری ارائه خواهد شد.

داخلی خود را اصلاح میکند. اتوماتا تازمانی این رویه را دنبال میکند که عمل بهینه را پیدا کند.

یک نمونه از الگوریتم‌های یادگیری در آtomاتای یادگیر یادگیری خطی است. فرض کنید عمل  $\alpha_i$  در مرحله  $n$  انتخاب شود. در صورتی که پاداش دریافت شود فرمول ۴ برای به روز رسانی بردار احتمال استفاده می‌گردد، در غیر این صورت فرمول ۵ استفاده می‌شود.

$$p_i(n+1) = p_i(n) + a[1 - p_i(n)] \quad (4)$$

$$p_j(n+1) = (1-a)p_j(n)$$

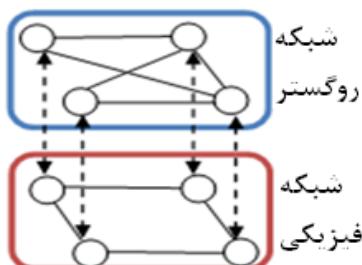
$$p_i(n+1) = (1-b)p_i(n) \quad (5)$$

$$p_j(n+1) = (b/r-1) + (1-b)p_j(n)$$

در روابط (4) و (5)،  $a$  پارامتر پاداش و  $b$  پارامتر جریمه می‌باشد. اگر  $a=b$  باشد، الگوریتم یادگیر  $L_{RP}$  می‌نامند و زمانی که  $b=0$  باشد الگوریتم یادگیر  $L_{RI}$  می‌نامند [۵].

#### ۴- شرایط مسئله تطبیق توپولوژی

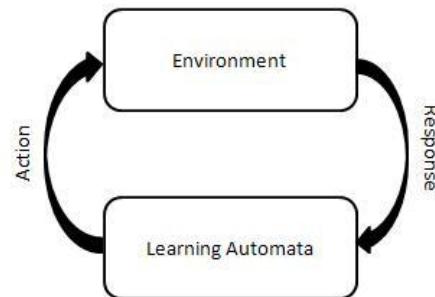
شبکه نظیر به نظری میتواند یک شبکه روگستر دیده شود که بر روی یک شبکه دیگر ایجاد میگردد. شبکه روگستر ایجاد شده ممکن است یک شبکه بهینه نباشد چرا که به راحتی نمیتوان نگاشت نظیرها را از فضای فیزیکی به فضایی که در شبکه روگستر ایجاد میشود را مدیریت کرد.



شکل ۱-۴ عدم تطبیق توپولوژی شبکه روگستر با شبکه فیزیکی

در صورتی که  $G$  گراف شبکه فیزیکی باشد و تابع نگاشت  $H$  نقش نگاشت نظیرها را از فضای شبکه فیزیکی به شبکه روگستر به عهده داشته باشد و تابع  $D$  تاخیر انتهای را در توپولوژی شبکه روگستر محاسبه کند هدف الگوریتم‌های تطبیق توپولوژی

نقص به حساب میآید، میتواند یک مزیت در استفاده از اتوماتا در محیط یا کاربردهای ناشناخته باشد.



شکل ۱-۳ اتوماتای یادگیر

یک اتوماتای یادگیر با اقدام‌های محدود توسط چهارتای  $A, R, D$  و محیط توسط  $A, Q, R, L$  تعیین میگردد که از  $A$  توسط مجموعه  $\{\alpha_1, \alpha_2, \dots, \alpha_r\}$  تعریف میگردد که نشان دهنده  $r$  عمل از نوع  $\alpha$  است که از سوی دیگر میتوانند خروجی‌های ما برای محیط باشند.  $R$  محدوده جواب اتوماتا برای محیط است و شامل  $\beta(k)$  هایی است که از یک سو پاسخ‌های محیط و از سوی دیگر ورودی‌های اتوماتا در قبال عمل  $\alpha(k)$  در مرحله  $k$  هستند. نماد  $D$  نیز مجموعه احتمال پاداش هاست که به شکل  $D = \{d_1, d_2, \dots, d_r\}$  تعریف میشود که  $d_i$  در مرحله  $k$  ام به شکل  $d_i(k) = E[\beta(k)|\alpha(k)]$  تعریف میشود بردار احتمالی که اتوماتا هر بار مبتنی بر آن در مرحله  $k$  ام عملی را انتخاب میکند را  $P(k)$  مینامیم.

$$P(K) = [p_1^{(k)}, p_2^{(k)}, \dots, p_r^{(k)}] \quad (2)$$

$Q$  حالت اتوماتاست و در نهایت  $L$  الگوریتم یادگیری و یا همان تابع تقویت است که توسط اتوماتا برای به روز رسانی حالت درونی به شکل زیر مورد استفاده قرار میگیرد.

$$Q(k+1) = L(Q(k), \alpha(k), \beta(k)) \quad (3)$$

اتوماتا در هر مرحله یک عمل را مبتنی بر بردار احتمال انتخاب میکند سپس مبتنی بر بازخورد محیطی و تابع تقویت متغیر های

### ۵- الگوریتم پیشنهادی

در مسئله تطبیق توپولوژی از طریق جابجایی همسایگان پارامتر شعاع همسایگی نقش مهمی ایفا می‌کند. جابجایی همسایگان به شکل موازی در سراسر شبکه انجام می‌شود و شرایط محیطی برای هر نظری پیوسته در حال تغییر است. باید مکانیزمی طراحی شود که این شعاع را به شکل برخط و محلی کنترل کند. در صورتی که شعاع همسایگی صفر شود دیگر همسایه‌ای برای ارزیابی و جابجایی وجود نخواهد داشت و در صورتی که شعاع بسیار بزرگ انتخاب شود در سراسر شبکه سربار بالایی ایجاد می‌شود. به عبارت دیگر، تنظیم این شعاع به نحوی باعث کنترل الگوریتم بهینه ساز می‌گردد. بنابراین، یک اتوماتای یادگیر به ازای هر نظری تعریف شده که دو عمل افزایش و کاهش شعاع همسایگی را خواهد داشت. الگوریتم ارائه شده در شکل ۱-۵ در هر نظری اجرا می‌شود و برای به روز رسانی بردار احتمال اتوماتای یادگیر، از الگوریتم ارائه شده در شکل ۲-۵ استفاده می‌شود. لازم به ذکر است که این الگوریتم کاملاً توزیع شده عمل می‌کند و نیاز به اطلاع از سراسر شبکه ندارد.

در این الگوریتم، ابتدا شعاع همسایگی توسط اتوماتای یادگیر درون هر نظری تغییر (کاهش/افزایش) می‌یابد، سپس نظری موردنظر همسایگان خود را در راستای کاهش تغییر محلی با استفاده از فرمول ۸ ارزیابی می‌کند. نظری پس از ارزیابی، جای خود را با نظری که بیشترین بهبود را در تاخیر محلی خواهد داشت عوض می‌کند و در محل جدید اتوماتای یادگیر پاسخ خود را از محیط می‌گیرد.

تنظیم پاداش اتوماتای یادگیر در این مسئله در دو فاز انجام شده. در فاز اول، پاداش‌ها با استفاده از اطلاعات محلی و در راستای افزایش شعاع همسایگی تعیین شده‌اند. پس از به ثبات رسیدن ساختار، تاثیر آن در توابع پاداش اتوماتای یادگیر دیده می‌شود و این فاز تمام می‌شود. در فاز دوم، پاداش‌ها با استفاده از اطلاعات محلی و در راستای کاهش شعاع همسایگی تعیین شده‌اند.

جدول ۵-۱: مشخصات اتوماتای یادگیر

Actions	$\{Increase\ radius, Decrease\ radius\}$
Learning Algorithm	$L_{Rp}$

کمینه سازی تابع  $M$  (تاخیر انتهای به انتهای) خواهد بود که در فرمول ۶ به آن اشاره شده است [۱۰].

$$M(G, H) = \sum_{(u,v) \in G} D(H(u), H(v)) \quad (6)$$

برای کمینه سازی این تابع عملگرهای وجود دارد که روال‌های مدیریتی از آنها برای بهینه سازی‌های شبکه روگستر استفاده می‌کنند که در [۱۰][۱۶][۱۲] مورد بررسی قرار گرفته‌اند. به عنوان مثال ارزیابی همسایگان از نظر تاخیر انتهای به انتهای میتواند مبنایی برای جابجایی همسایگان باشد که باعث کاهش تاخیر انتهای به انتهای می‌گردد.

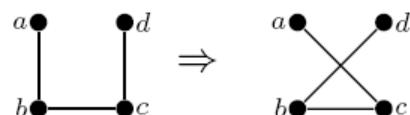
$$M(G, H') = \quad (7)$$

$$M(G, H) - M(b, H) - M(c, H) + M(b, H') + M(c, H')$$

$$K(b, c) = \quad (8) \\ [(M(b, H) + M(c, H)) - (M(b, H') + M(c, H'))]$$

$$M(G, H') = M(G, H) - K(b, c) \quad (9)$$

در صورتی که  $H'$  تابع نگاشت لایه فیزیکی به لایه روگستر پس از جابجایی باشد هر نظری میتواند در شعاع همسایگی خود جدولی از همسایگان ایجاد کرده و توسط ارزیابی همسایگان تابع  $K(b, c)$  را محاسبه کند و در صورتی که جابجایی مثبت ارزیابی شود ( $k(b, c) > 0$ ) تغییر در همسایگی انجام می‌شود.



شکل ۲-۴ جابجایی همسایگان

ولی بایستی شیوه‌ای برای کنترل استفاده از این عملگرهای محیط پویای شبکه نظری به نظری وجود داشته باشد. در صورتی که کنترل درستی بر استفاده از این عملگرهای وجود نداشته باشد شرایط شبکه توسط نظری‌های تازه وارد به سرعت به سمت شرایطی نامناسب تبدیل می‌گردد چرا که یکی از چالش‌هایی که باعث این مشکل می‌شود نرخ بالای ورود و خروج کاربران است [۴].

ELSE

Update LAP by EQ (5)

شکل ۲-۵ الگوریتم محاسبه بازخورد اتوماتی یادگیر

## ۶- شبیه سازی

برای مطالعه و بررسی الگوریتم طراحی شده از شبیه سازی کامپیوتری استفاده شد. بدین منظور، از شبیه ساز شبکه نظری به نظری [17] OverSim استفاده شد و نتایج با الگوریتم استاندارد جابجایی همسایگان [۱۰، ۱۲] مقایسه گردید. تعداد ۲۰۰ نظری برای شبیه سازی در نظر گرفته شده و پارامترهای تابع توزیع زمان زندگی<sup>۳</sup> به شکل تصادفی مقدار دهی شده اند. به عبارت دیگر به نوعی پویایی درج و حذف نظریه را در این روش دیده شده است. شبیه سازی ۲۰ بار تا ۳۰ مرحله زمانی انجام شد و میانگین نتایج محاسبه شده. منظور از یک مرحله زمانی، اجرای الگوریتم به شکل ترتیبی برای تمام اعضای شبکه است. در آزمایش های اول تا چهارم برای مراحل ۴، ۶، ۸، ۱۰، ۲۰ و ۳۰ شاخص های مورد نظر محاسبه شده اند.

لازم به ذکر است که در نمودارهای رسم شده، الگوریتم ارائه شده با نام *DNCLA* و الگوریتم جابجایی همسایگان با نام *Swap Strategy* مورد اشاره قرار گرفته اند.

\*/ P: Current Peer, R: Neighborhood radius, LAP: Embedded Learning Automata, StopFlag: SF /\*

do

SF = •

Max=•

determine R by LAP

for(i=1 to i=P.neigbrors)

if K(P, P.neigbrors[i])> max

U= P.neigbrors[i]

Max= K(P, P.neigbrors[i])

SF = 1

End if

Exchange(P,U)

Update LAP //figure 2-5

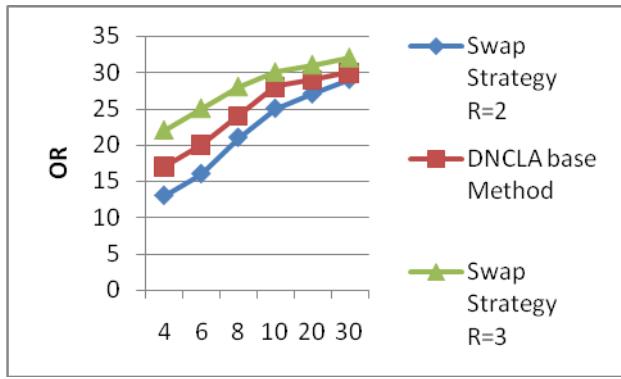
while(SF=• or R>0)

شکل ۱-۵ شبیه کد الگوریتم

\*/ P: Current Peer, IR: Increase Radius, DR: Decrease Radius, LAP: Learning Automaton in Current Peer, K: Number of selected neighbors/\*

IF(LAP.action=DR) and  
major(P.neighbor.LA.action)=DR) OR  
(LAP.action=IR) and  
major(P.neighbors.LA.action)=IR)  
THEN  
Update LAP by EQ (4)

<sup>۳</sup> Life time



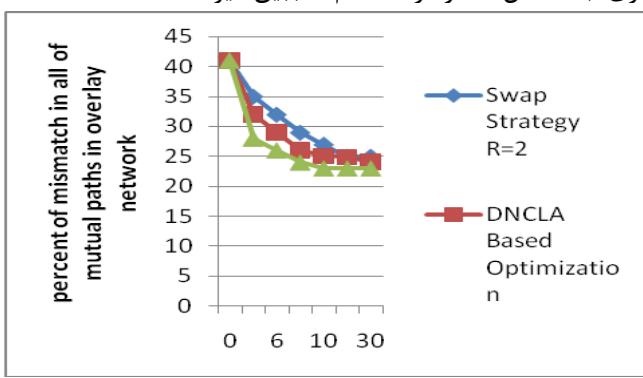
شکل ۲-۶ پیشرفت در بهینه سازی تاخیر انتها به انتها

### ۱-۶- آزمایش اول

هدف این آزمایش بررسی الگوریتم ارائه شده از دیدگاه تنظیم تطبیقی شعاع همسایگی است. برای این آزمایش در هر مرحله از شبیه سازی میانگین شعاع همسایگی تمام نظیرها محاسبه شده. طراحی الگوریتم مورد نظر دو فاز در نظر گرفته شد. در فاز اول هدف افزایش شعاع همسایگی بود که ساختار مناسب حاصل شود، و در فاز دوم که ساختار بدست آمده، هدف کاهش شعاع همسایگی بود که دیگر بهینه سازی متوقف شود. این روند در شبیه سازی نیز مشاهده شد که نشان دهنده، درستی روش تنظیم توابع تقویت اتوماتای یادگیر، در الگوریتم ارائه شده است.

### ۳- آزمایش سوم

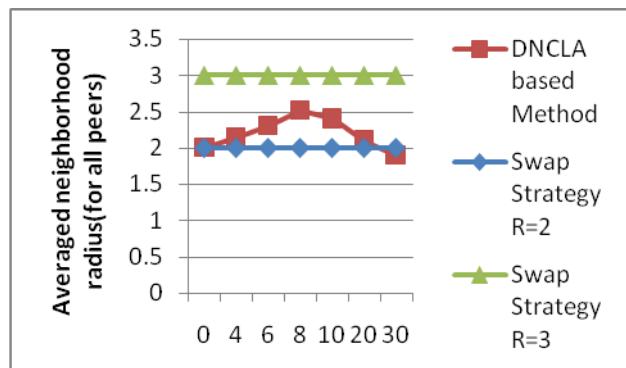
هدف این آزمایش بررسی الگوریتم ارائه شده از دیدگاه کاهش مسیرهای دارای عدم تطبیق است. در این آزمایش برای تمام یالها در توپولوژی شبکه روگستر تعداد مسیرهای دارای عدم تطبیق محاسبه می‌شوند و این تعداد به تعداد کل یال‌های شبکه روگستر تقسیم می‌شود. این معیار نیز نشان میدهد که الگوریتم ارائه شده با تنظیم تطبیقی شعاع همسایگی با سرعت مناسب تری به حداقل مقدار درصد عدم تطبیق میرسد.



شکل ۳-۶ درصد مسیرهای دارای عدم تطبیق در شبکه روگستر

### ۴- آزمایش چهارم

هدف این آزمایش بررسی سربار الگوریتم ارائه شده است. سربار در این آزمایش مبتنی بر محاسبه پیام‌های ارزیابی تعریف شده است. درصد پیام‌هایی که برای ارزیابی دریافت می‌کند نسبت به تمام پیام‌های دریافتی، برای هر نظیر محاسبه می‌گردد و میانگین این مقدار برای تمامی نظیرها محاسبه می‌شود.



شکل ۱-۶ میانگین شعاع همسایگی نظیرها

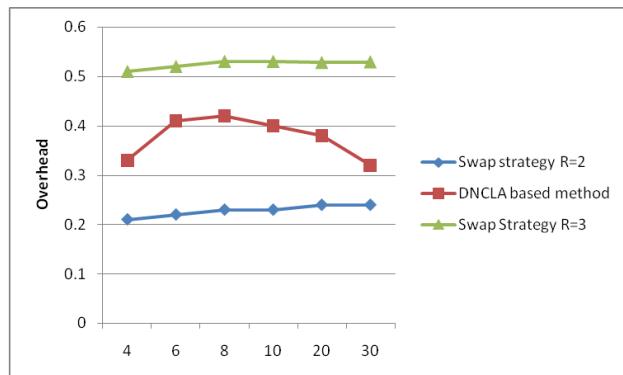
### ۲- آزمایش دوم

هدف این آزمایش بررسی الگوریتم ارائه شده از دیدگاه بهینه سازی تابع تاخیر انتها به انتها است. معیار  $OR$  توسط فرمول ۱۰ محاسبه می‌شود و پیشرفت بهینه سازی را نشان می‌دهد. این معیار هنگامی به ۱۰۰ میرسد که تاخیرسراسری به صفر برسد (که امکان صفر شدن به واسطه ساختار شبکه فیزیکی وجود ندارد) و به هنگام کاهش سراسری تاخیر نیز، افزایش می‌یابد. این معیار نشان می‌دهد که با شعاع بیشتر نیز میتوان به میزان تاخیر قابل قبولی دست یافت ولی بدینه است با شعاع همسایگی بیشتر سربار بیشتری خواهیم داشت. با شعاع همسایگی کمتر هم بسیار کند به تاخیر مناسب خواهیم رسید.

$$OR(t = current) = \frac{End to End delay(t = .) - End to End delay(t = current)}{End to End delay(t = .)} * 100 \quad (10)$$

## ۷- نتیجه گیری

در این مقاله یک الگوریتم جدید مبتنی بر اتوماتاتی یادگیر برای مسئله تطبیق توپولوژی ارائه شد و با یک الگوریتم استاندارد مقایسه گردید. در الگوریتم ارائه شده، جابجایی همسایگان با تغییر شعاع همسایگی به شکل برخط کنترل شد و هر نظیر با استفاده از اتوماتاتی یادگیر شعاع همسایگی خود را به شکل تطبیقی و با توجه به شرایط محیطی تنظیم نمود. تنظیم تطبیقی شعاع همسایگی تاثیری مثبت بر بهینه سازیتابع تاخیر انتها به انتها و درصد تطبیق مسیرها داشت. لازم به ذکر است که در الگوریتم ارائه شده سربار نیز در طی زمان کاهش یافته که یک پارامتر مهم در طراحی الگوریتم‌های مدیریتی در شبکه است.



شکل ۴-۶ سربار الگوریتم

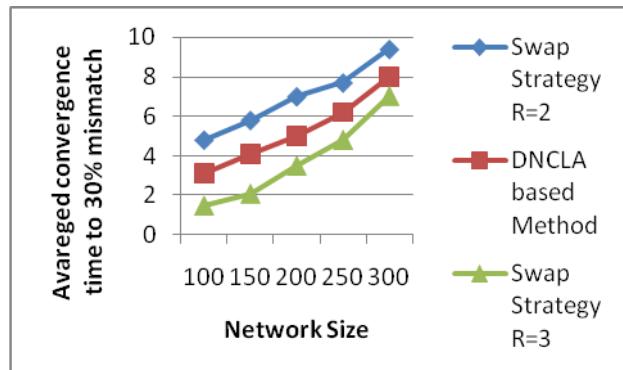
همانگونه که مشاهده میشود سربار این الگوریتم نیز پس از ثبات ساختار کاهش یافته است، گرچه در ابتدا سربار زیادی به شبکه تحمیل میکند. برای تمامی الگوریتم‌های یادگیر فاز یادگیری همواره هزینه بر بوده است که در این الگوریتم نیز این پدیده مشاهده میشود.

## مراجع

- [1] M. Goff, *Network distributed computing: fitscapes and fallacies*. Prentice Hall Professional Technical Reference, 2003.
- [2] S. Androuloutsos-Theotokis and D. Spinellis, “A survey of peer-to-peer content distribution technologies,” *ACM Computing Surveys (CSUR)*, vol. 36, no. 4, pp. 335–371, 2004.
- [3] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, “A survey and comparison of peer-to-peer overlay network schemes,” *IEEE Communications Surveys and Tutorials*, vol. 7, no. 2, pp. 72–93, 2005.
- [4] Y. K. Kwok, *Peer-to-Peer Computing: Applications, Architecture, Protocols, and Challenges*. CRC Press, 2011.
- [5] K. S. Narendra and M. A. L. Thathachar, *Learning automata: an introduction*. Prentice-Hall, Inc., 1989.
- [6] M. Thathachar and P. S. Sastry, *Networks of learning automata: Techniques for online stochastic optimization*. Kluwer Academic Publishers, 2004.
- [7] M. Esnaashari and M. R. Meybodi, “A cellular learning automata-based deployment strategy for mobile wireless sensor networks,” *Journal of Parallel and Distributed Computing*, vol. 71, pp. 988–1001, 2011.
- [8] M. Esnaashari and M. R. Meybodi, “Data aggregation in sensor networks using learning automata,” *Wireless Networks*, vol. 16, no. 3, pp. 687–699, 2010.

## ۵-۶- آزمایش پنجم

هدف این آزمایش بررسی تاثیر مقیاس شبکه بر کارایی الگوریتم است. برای مقیاس‌های متفاوت میانگین تعداد مراحل برای رسیدن به ۳۰ درصد عدم تطبیق محاسبه شده است. بدیهی است که با افزایش تعداد نظیرها زمان همگرایی نیز بیشتر میشود، ولی افزایش مقیاس شبکه تاثیر منفی بر سرعت همگرایی الگوریتم مورد نظر نداشته است زیرا الگوریتم از اطلاعات محلی استفاده می‌کند.



شکل ۵-۶ تاثیر مقیاس شبکه بر کارایی الگوریتم

- [9] H. Beigy and M. R. Meybodi, "A self-organizing channel assignment algorithm: A cellular learning automata approach," *Intelligent Data Engineering and Automated Learning*, Springer Verlag, vol. 14, pp. 119–126, 2003.
- [10] H. Rostami and J. Habibi, "A mathematical foundation for topology awareness of p2p overlay networks," in *Grid and Cooperative Computing*, Heidelberg, 2005, vol. 3, pp. 906–918.
- [11] Y. Liu, "A two-hop solution to solving topology mismatch," *IEEE Transactions on Parallel and Distributed Systems*, vol. 19, pp. 1591–1600, 2008.
- [12] T. Qiu, E. Chan, M. Ye, G. Chen, and B. Y. Zhao, "Peer-exchange schemes to handle mismatch in peer-to-peer systems," *The Journal of Supercomputing*, vol. 48, no. 1, pp. 15–42, 2009.
- [13] H. Rostami and J. Habibi, "Topology awareness of overlay P2P networks," *Concurrency and Computation: Practice and Experience*, vol. 19, no. 7, pp. 999–1021, 2007.
- [14] Y. Liu, Z. Zhuang, L. Xiao, and L. M. Ni, "A distributed approach to solving overlay mismatching problem," in *24th International Conference on Distributed Computing Systems*, Tokyo , Japan, 2004, pp. 132–139.
- [15] Y. Liu, L. Xiao, X. Liu, L. M. Ni, and X. Zhang, "Location awareness in unstructured peer-to-peer systems," *IEEE Transactions on Parallel and Distributed Systems*, vol. 16, no. 2, pp. 163–174, 2005.
- [16] P. Mahlmann and C. Schindelhauer, "Random graphs for peer-to-peer overlays," in *Proceedings of the Dynamically Evolving, Large Scale Information Systems*, Barcelona, pp. 1–22.
- [17] I. Baumgart, B. Heep, and S. Krause, "OverSim: A scalable and flexible overlay framework for simulation and real network applications," in *Peer-to-Peer Computing*, Seattle, Washington, USA, 2009, pp. 87–88.