

یک روش جدید جستجوی خودتطبیق برای شبکه‌های نظیربه‌نظیر غیرساخت‌یافته با بهره‌گیری از اتوماتای یادگیر

مهدی قربانی¹، علی محمد صغیری²، محمد رضا میبیدی³

¹ دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه آزاد اسلامی واحد قزوین، قزوین
mahdigh13@yahoo.com

² دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران
a_m_saghiri@aut.ac.ir

³ دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر، تهران
mmeybodi@aut.ac.ir

چکیده

به دلیل عدم وجود کنترل مرکزی یا اطلاع کافی از مکان اشیاء در شبکه‌های نظیربه‌نظیر غیرساخت‌یافته، طراحی یک روش جستجوی کارا در این‌گونه شبکه‌ها بسیار مورد توجه است. در روش k -قدم‌های تصادفی، به عنوان یکی از روش‌های جستجو، تعیین مقدار k به طور تصادفی، می‌تواند بر معیارهای کارایی شبکه تاثیر قابل توجهی داشته باشد. در این مقاله، یک روش جستجوی توزیع‌شده خودتطبیق با بهره‌گیری از اتوماتای یادگیر ارائه شده است تا این چالش را برطرف کند. در این روش، نیازی به تعیین مقدار k به طور تصادفی از قبل نیست، بلکه هر نظیر، می‌تواند همسایه خود را به صورت تطبیقی بیابد. با به‌کارگیری اتوماتای یادگیر برای هر گره، همه همسایه‌هایی که دارای بالاترین احتمال جستجوی موفق در مراحل قبلی جستجو هستند، به صورت تطبیقی انتخاب می‌شوند. سابقه جستجو برای هر گره، در جداولی ذخیره می‌شود که در زمان جستجو، بر اساس مقادیر احتمالی موجود در آنها، تصمیم‌گیری برای انتخاب همسایه‌ها انجام خواهد شد. شبیه‌سازی‌ها نشان می‌دهد که روش جستجوی پیشنهادی، برخی ویژگی‌ها را مانند میانگین تعداد همسایه‌ها به ازای هر درخواست، میانگین تعداد پیام‌های تولید شده، تعداد اشیاء کشف‌شده به ازای هر درخواست و همچنین میزان موفقیت در جستجو در مقایسه با روش k -قدم‌های تصادفی، بهبود می‌بخشد.

کلمات کلیدی

نظیربه‌نظیر، جستجو، اتوماتای یادگیر، k -قدم‌های تصادفی

1- مقدمه

گردد. قرارگیری محتویات در شبکه‌های نظیربه‌نظیر غیرساخت‌یافته، به‌گونه‌ای است که گره‌ها به سادگی می‌توانند وارد شبکه شوند یا از آن خارج گردند [1,2]. بنابراین، به سادگی، نمی‌توان مکان شیء را پیدا کرد و استفاده از مکانیسم‌های جستجوی مناسب ضروری به نظر می‌رسد. امروزه، اکثر کاربردهای سیستم‌های نظیربه‌نظیر، معطوف به شبکه‌های غیرساخت‌یافته نظیربه‌نظیر است و طراحی یک روش جستجوی کارا یکی از چالش‌های مهم در این زمینه محسوب می‌شود. روش‌های جستجو در شبکه‌های نظیربه‌نظیر غیرساخت‌یافته به دو دسته تقسیم می‌شوند: کورکورانه و آگاهانه [3,4]. در روش‌های جستجوی کورکورانه، گره‌ها هیچ‌گونه اطلاعاتی در مورد مکان اشیاء ندارند. این روش‌ها، از تکنیک‌های سیل‌آسا برای هدایت درخواست‌ها استفاده می‌کنند. از طرف دیگر، در رویکردهای آگاهانه، گره‌ها از

شبکه‌های نظیربه‌نظیر در سال‌های اخیر، به سرعت در حال رشد هستند. بسیاری از کاربردهای شبکه‌های نظیربه‌نظیر مانند اشتراک فایل، مدیون ساختار توزیع‌شده این‌گونه شبکه‌ها هستند. شبکه‌های نظیربه‌نظیر، معمولاً بدون هیچ کنترل‌کننده مرکزی به منظور تامین برخی امکانات مانند انعطاف‌پذیری، مقیاس‌پذیری و قابلیت اطمینان پیاده‌سازی می‌شوند. این شبکه‌ها، به دو گروه تقسیم می‌شوند: ساخت-یافته و غیرساخت‌یافته. در شبکه‌های نظیربه‌نظیر ساخت‌یافته، مکان محتویات شبکه، از طریق جداول درهم‌سازی توزیع‌شده (DHT) کنترل می‌شوند و به همین دلیل سربار کمی به شبکه تحمیل می‌-



همسایه‌اش که مقدار احتمالی مربوط به آن در شاخص محلی‌اش قرار دارد هدایت می‌کند. مقادیر شاخص، توسط دریافت بازخورد از قدم‌ها، به‌روزرسانی می‌شوند. APS، قابلیت اطمینان را افزایش می‌دهد و میزان مصرف پهنای باند را بهبود می‌بخشد اما سربرابر شبکه، همچنان قابل توجه است.

DST، از روش Q-learning برای انتخاب همسایه‌ها جهت هدایت درخواست‌ها بهره می‌گیرد. در این روش، همه گره‌ها به دو دسته گره-های عادی و قدرتمند تقسیم‌بندی می‌شوند. در ابتدا، k عدد قدم، در شبکه، منتشر می‌شوند و درخواست‌ها به سمت گره‌ها برای یافتن اشیاء هدایت می‌شوند. اگر جستجو ناموفق باشد، انتشار درخواست‌ها از طریق گره‌های عادی و قدرتمند با توجه به تجربیات گذشته در حافظه گره‌ها، انجام می‌شود. آنچه مهم است این است که کارایی جستجو به میزان پیشرفت یادگیری گره‌ها، وابسته است که معمولاً به کندی انجام می‌گیرد.

در روش‌های کورکورانه، هر گره درخواست جستجو را به سمت همه همسایه‌های هدایت می‌کند. جستجو زمانی کامل می‌شود که 'hit' یا 'miss' رخ دهد یا اینکه مقدار TTL تمام شود. در برخی از این روش‌ها مانند k -قدم‌های تصادفی، درخواست جستجو نه به همه گره‌ها، بلکه تنها به برخی از آنها که به صورت تصادفی انتخاب شده‌اند ارسال می‌شود. به‌کارگیری این نوع از روش‌های جستجو، به دلیل هدایت درخواست‌ها به صورت تصادفی و فقدان یک راه حل تطبیقی در زمانی که بار شبکه متغیر است، کارایی شبکه را کاهش می‌دهد.

سازماندهی ادامه مقاله به این صورت است. در بخش 2، اتوماتای یادگیر که به عنوان استراتژی اصلی یادگیری در الگوریتم پیشنهادی استفاده شده است، به اختصار شرح داده می‌شود. الگوریتم پیشنهادی در بخش 3 و در بخش 4، نتایج شبیه‌سازی‌های انجام‌شده ارائه می‌گردد. در بخش پایانی، نتیجه‌گیری بیان شده است.

2- مروری بر اتوماتای یادگیر

اتوماتای یادگیر [14]، ماشینی است که می‌تواند تعدادی متناهی عمل را انجام دهد. هر عمل انتخاب شده توسط یک محیط احتمالی ارزیابی شده و نتیجه ارزیابی در قالب سیگنالی مثبت یا منفی به اتوماتا داده می‌شود و اتوماتا از این پاسخ در انتخاب عمل بعدی تاثیر می‌گیرد. هدف نهایی آن است که اتوماتا یاد بگیرد که از بین اعمال خود بهترین عمل را انتخاب کند. بهترین عمل، عملی است که احتمال دریافت پاداش از محیط را به حداکثر برساند. کارکرد اتوماتای یادگیر در تعامل با محیط، در شکل 1 مشاهده می‌شود.



وضعیت شبکه و مکان محتویات از طریق ذخیره‌سازی متاداده‌هایی از اطلاعات، آگاه هستند. الگوریتم k -قدم‌های تصادفی، نمونه‌ای از روش-های جستجوی کورکورانه هستند. وقتی جستجوی یک شیء ناموفق باشد، k گره از بین همسایه‌ها به طور تصادفی، انتخاب می‌شوند و درخواست‌ها به سمت آنها هدایت می‌شوند. زمانی جستجو موفقیت‌آمیز خواهد بود که یک شیء توسط حداقل یکی از همسایه‌ها پیدا شود [12,13].

مقدار بهینه و قابل تطبیق با اندازه شبکه برای k ، کارایی شبکه را در مواردی مانند بار شبکه و زمان پاسخ بهبود می‌بخشد. انتخاب مقدار k ، بزرگتر از میانگین تعداد همسایه‌ها منجر به ترافیک بیشتر در شبکه می‌شود، زیرا در این حالت، درخواست‌ها به صورت سیل‌آسا ارسال می‌شوند. از طرف دیگر، اگر مقدار k کمتر از میانگین تعداد همسایه‌ها باشد، احتمال انتخاب همسایه‌های غیرمرتبط برای جستجو، افزایش می‌یابد. با فرض انتخاب مقدار مناسب برای k ، اینکه کدام k همسایه باید از بین همه گره‌های شبکه باید انتخاب شوند، نیز مهم است.

در برخی از مطالعات پیشین [5-8][11-13]، تکنیک‌های یادگیری تقویتی [10,11] مانند Q-learning [10] برای بهبود معیارهای جستجو، به‌کار گرفته می‌شوند. با به‌کار بردن روش‌های یادگیری در شبکه، هر گره می‌تواند وضعیت شبکه را یاد بگیرد و برای مراحل بعدی جستجو، بر اساس بازخورد دریافتی از گره‌ها تصمیم‌گیری کند. انتظار می‌رود در رویکردهای جستجوی مبتنی بر یادگیری، زمان پاسخ کوتاه‌تر، بار شبکه کمتر و میزان مصرف پهنای باند بهبود یابد.

در این مقاله، یک الگوریتم جستجوی جدید و تطبیقی کارا مبتنی بر اتوماتای یادگیر [14] برای شبکه‌های نظیربه‌نظیر غیرساخت‌یافته، پیشنهاد می‌شود. در الگوریتم پیشنهادی، چالش مربوط به مقدار غیر-تطبیقی k در الگوریتم k -قدم‌های تصادفی، حل شده است و این مساله از طریق شبیه‌سازی‌های مختلف ارزیابی می‌شود.

همانطور که قبلاً بحث شد، استراتژی‌های جستجو در شبکه‌های نظیربه‌نظیر غیرساخت‌یافته به دو دسته روش‌های کورکورانه و آگاهانه تقسیم می‌شوند.

در روش‌های آگاهانه، هر گره، برخی اطلاعات را در مورد همسایه-هایش و وضعیت شبکه، نگهداری می‌کند. جستجوی سطحی (BFS) [5]، جستجوی احتمالی تطبیقی (APS) [5,7]، تکنیک جستجوی توزیع شده (DST) [12,13]، نمونه‌هایی از روش‌های آگاهانه هستند که در آنها از تجربیات قبلی همسایه‌ها، به منظور جستجو در مراحل آتی استفاده می‌شود.

در BFS هوشمند یا BFS جهتدار [5]، هر گره یک جدول را در حافظه‌اش برای نگهداری اطلاعات درباره درخواست‌ها و شناسه‌های همسایه‌اش استفاده می‌کند. این اطلاعات، بر اساس پاسخ‌های قبلی همسایه‌ها، در مراحل قبلی جستجو هستند. روش BFS هوشمند، از "نظیرهای متخصص"، برای تنظیم دقت جستجو استفاده می‌کند.

APS، از k عدد قدم مستقل و تکنیک هدایت درخواست‌ها به صورت احتمالی استفاده می‌کند. هر گره میانی، درخواست را به



$$N = \{N_1, N_2, \dots | N_i \text{ is } i^{th} \text{ neighbor of } CN, i=1, 2, \dots\} \quad (3)$$

$$P_{vector} = \{P_{1i}, P_{2i}, \dots, P_{ri} | r \text{ is the number of actions for } N_i, \sum P_{ki} = 1\} \quad (4)$$

در الگوریتم پیشنهادی، هر گره از یک الگوریتم اتوماتای یادگیر استفاده می‌کند که یک جدول از اشیاء قابل دسترس را که با AO نشان داده می‌شوند، در ستون خود ذخیره می‌کند. در هر سطر این جدول، شناسه همسایه‌ها (ID) و بردار احتمال مربوط به آن همسایه (P_{vector}) نگهداری می‌شود. همان‌طور که در شکل 2 نشان داده شده است، بردار احتمال متناظر با هر همسایه، دو مقدار احتمالی P_{2i} و P_{1i} دارد که به صورت زیر تعریف می‌شوند:

P_{1i} مقدار احتمالی گره همسایه i ام برای مشارکت بر اساس مراحل قبلی جستجو است، در حالی که P_{2i} مقدار احتمالی i امین همسایه می‌باشد که در مراحل قبلی جستجو، مشارکتی نداشته است. در ابتدای اجرای الگوریتم، مقادیر این بردارهای احتمالی، 0.5 در نظر گرفته می‌شوند.

	AO _i
N ₁	{P _{1,1} , P _{2,1} }
N ₂	{P _{1,2} , P _{2,2} }
N ₃	{P _{1,3} , P _{2,3} }
⋮	⋮
N _r	{P _{1,r} , P _{2,r} }

شکل (2): نمونه‌ای از ساختار جدول ذخیره‌شده در گره جاری

CN با مقادیر احتمالی

برای آموزش گره‌ها، از سه نوع جدول ذخیره‌شده استفاده می‌شود: جدول Neighbor-hiti-LA، شامل آن دسته از گره‌های همسایه است که قبلاً درخواست جستجو به سمت آنها هدایت شده است و نتیجه جستجو با در نظر گرفتن مراحل قبلی جستجو، منجر به hit شده است. جدول دوم، جدول Neighbor-miss-LA است که گره‌های همسایه را که در جستجوهای قبلی، مشارکت داشته‌اند اما نتیجه جستجو، miss بوده است در خود ذخیره می‌کند. قالب ذخیره‌سازی این جداول، مانند جدول Neighbor-hit-LA می‌باشد. دسته سوم از همسایه‌ها، یعنی همسایه‌هایی که در مراحل قبلی جستجو مشارکتی نداشته‌اند، در جدولی به نام Query-LA ذخیره می‌شوند. بدیهی است که در این حالت، هیچ شیء قابل دسترسی در این جدول وجود نخواهد داشت.

در ابتدا، یک درخواست جستجو به یک گره دلخواه ارسال می‌شود. اگر شیء مربوطه در گره جاری پیدا نشد، مکان‌یابی شیء مورد نظر از طریق جداول پیگیری می‌شود. با ارسال هر درخواست، یک سطر جدید به جدول آن گره جاری افزوده خواهد شد. در آغاز، جدول Neighbor-

محیط را می‌توان 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 جرمیه می‌باشد. احتمال اینکه عمل C_i نتیجه نامطلوب داشته باشد α_i است. در محیط ایستا، مقادیر C_i بدون تغییر می‌مانند، حال آنکه در محیط غیرایستا این مقادیر در طی زمان تغییر می‌کنند. اتوماتاهای یادگیر به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم‌بندی می‌شوند. در ادامه، اتوماتای یادگیر با ساختار متغیر معرفی می‌گردد.

اتوماتای یادگیر با ساختار متغیر را می‌توان توسط چهار تایی $\{\alpha, \beta, p, T\}$ نشان داد که $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عمل‌های اتوماتا، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودی‌های اتوماتا، $p = \{p_1, p_2, \dots, p_r\}$ بردار احتمال انتخاب هر یک از عمل‌ها و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می‌باشد. الگوریتم زیر، یک نمونه از الگوریتم‌های یادگیری خطی است. فرض کنید عمل α_i در مرحله n ام انتخاب شود.

پاسخ مطلوب:

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

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

پاسخ نامطلوب:

$$p_i(n+1) = (1 - b)p_i(n) \quad (2)$$

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

در روابط (1) و (2)، a پارامتر پاداش و b پارامتر جریمه می‌باشند. با توجه به مقادیر a و b سه حالت زیر را می‌توان در نظر گرفت. زمانی که a و b با هم برابر باشند، الگوریتم را LRP می‌نامند، زمانی که b از a خیلی کوچکتر باشد، الگوریتم را LREP می‌نامند و زمانی که b مساوی صفر باشد الگوریتم را LRI می‌نامند.

3- الگوریتم جستجوی پیشنهادی

تمرکز اصلی بر روی شبکه‌های نظیربه‌نظیر غیرساخت‌یافته و به-کارگیری اتوماتای یادگیر برای انتخاب گره‌ها جهت شرکت در جستجو است. در این مقاله، از محیط مدل S استفاده می‌کنیم و اتوماتای یادگیر LRP می‌باشد. در ابتدا، ساختار داده الگوریتم پیشنهادی را شرح می‌دهیم و سپس الگوریتم جستجو را توضیح خواهیم داد. همان‌طور که در بخش 3 بیان شد، در اتوماتای یادگیر، مقادیر احتمالی که با توجه به بازخورد محیط به‌روزرسانی می‌شوند، برای تصمیم‌گیری بسیار مهم هستند.

برای به‌کارگیری اتوماتای یادگیر در الگوریتم پیشنهادی، به دو تعریف ساده نیاز داریم. فرض کنید، گره جاری که درخواست جستجو به سمت آن ارسال می‌شود، با CN نشان داده شود.



نداشته است و همه همسایه‌هایش در یافتن شیء مربوطه شکست خورده‌اند، به عبارتی پیام miss داده‌اند، به آن گره پاداش داده می‌شود. زیرا با این کار، به گره جاری کمک می‌کنیم تا در مراحل بعدی جستجو برای درخواست شیء متفاوت، شانس مشارکت در جستجو را داشته باشد و فراموش نشود. اما در صورتی که حداقل یکی از همسایه‌ها، شیء مورد نظر را بیابد، برای کاهش احتمال انتخاب همسایه‌های غیرمفید، گره جاری را جریمه می‌کنیم.

شکل 4، شبه‌کد نحوه تخصیص پاداش و جریمه به گره‌های شبکه را نشان می‌دهد.

```
* /CN ≡ Current node selected for search, N ≡ Neighbor of CN
Pvector ≡ Set of probability values for N /*

IF CN participates in search
  IF the feedback of Ns is "HIT"
    THEN EQ(1) updates the Pvector
  ELSE EQ(2) update the Pvector
ELSE
  IF all the replies of Ns is "MISS"
    THEN EQ(1) updates the Pvector
  ELSE EQ(2) updates the Pvector
```

شکل (4): نحوه به‌روزرسانی گره‌های شبکه

4- شبیه‌سازی

در این بخش، محیط شبیه‌سازی را توصیف خواهیم کرد و نتایج حاصل از ارزیابی‌های مختلف را خواهیم گفت. در زیر بخش 5-1، پارامترهای شبیه‌سازی و مقادیر اولیه برای پیاده‌سازی بیان می‌شود و در زیر بخش 5-2، نتایج حاصل از شبیه‌سازی را نشان می‌دهیم.

4-1- فرضیات شبیه‌سازی

برای شبیه‌سازی الگوریتم جستجوی پیشنهادی از شبیه‌ساز oversim [15] استفاده شده است. مدل شبکه‌ای که در شبیه‌سازی مورد استفاده قرار گرفته است، یک گراف تصادفی [7] با تعداد 1000 گره می‌باشد. میانگین درجه خروجی هر گره، 3 در نظر گرفته شده است. 100 شیء متفاوت به طور تصادفی در گره‌های مختلف شبکه، توزیع شده‌اند و هر گره با توجه به میزان حافظه‌ای که برای آن در نظر گرفته شده است، تعدادی از این اشیاء را در خود نگهداری می‌کند. ماکزیمم زمان لازم برای دریافت پیام‌ها در شبکه یا به عبارتی، TTL، 6 در نظر گرفته شده است. میزان خرابی گره‌ها، که نسبت گره‌های غیر فعال به همه گره‌ها را نشان می‌دهد، 20 درصد فرض شده است. شبیه‌سازی در 20 دور که هر دور شامل تعداد متفاوتی همسایه است، انجام می‌شود. "یک دور"، بدین معنی است که الگوریتم برای همه گره‌های شبکه در آن زمان، اجرا شود. در روش پیشنهادی، از مدل استاندارد اتوماتای یادگیر استفاده شده است که در این حالت، محیط، مقادیر صفر و یک را به عنوان خروجی اختیار می‌کند. صفر به معنای پاداش یا موفقیت در جستجو و یک به معنای شکست یا عدم موفقیت در فرآیند جستجو

hit-LA مورد بررسی قرار می‌گیرد که آیا شیء مربوطه در آن قرار دارد یا خیر. اگر در این حالت، شیء مربوطه پیدا شد، همه همسایه‌هایی که بیشترین مقدار احتمالی شرکت در جستجو با توجه به سابقه موجود در P_{vector} خود دارند، انتخاب شده و به تعداد آنها، پیام درخواست تولید می‌شود و جستجو با آنها تکرار می‌گردد. اگر جستجوی شیء مورد نظر در Neighbor-hit-LA با شکست روبرو شود، از جدول Query-LA همه همسایه‌های گره جاری با بالاترین مقدار احتمالی، انتخاب می‌شوند تا جستجو از طریق آنها ادامه پیدا کند. پس از انتخاب این همسایه‌ها، به تعدادشان پیام درخواست، تولید شده و به سمت آنها ارسال می‌شود. در این حالت نیز، مقادیر احتمالی موجود در بردار احتمالی هر یک از همسایه‌ها، با توجه به سابقه پاداش و جریمه آنها در جستجوهای قبلی، محاسبه شده است. در نهایت، اگر جستجو با توجه به دو جدول ذکر شده، موفقیت‌آمیز نبود، گره‌های موجود در جدول Neighbor-miss-LA که دارای بالاترین مقدار احتمالی برای شرکت در جستجو هستند، برای ادامه جستجو انتخاب می‌شوند.

اگر جستجو، منجر به یافتن شیء مربوطه شد، نتایج بر روی مسیر معکوس ارسال می‌شوند و همه گره‌های موجود در مسیر با توجه به عملکردشان و اتوماتای یادگیر، بردار احتمال خود را به‌روزرسانی می‌کنند. به طور کلی، جستجو زمانی پایان می‌یابد که یا شیء مورد نظر پیدا شود یا مقدار TTL به پایان برسد.

شکل 3، شبه‌کد الگوریتم جستجوی پیشنهادی را نشان می‌دهد.

* / Q ≡ Query Keyword P_c ≡ Probability values of participation in search, M ≡ Total number of neighbors with the highest P_c /*

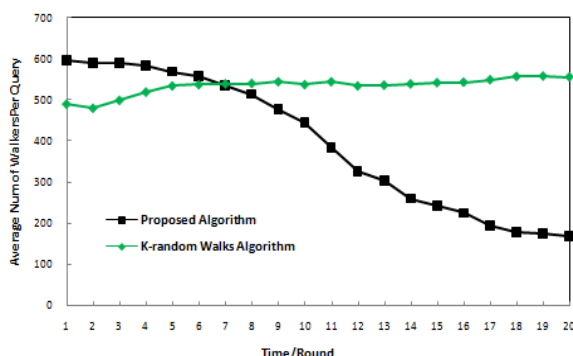
```
1. User submits a query
2. Search query node for Q
3. If Q is not in CN
  3.1. Search for Q in the Neighbor-hit-LA-table
  3.1.1. If Q is found
    Select all neighbors with the highest Pc
    Generate M query messages
    Search starts with M neighbors
  Else
    Select M neighbors in the Query-LA-table
    Generate M query messages
    Search starts with M neighbors
  Else
    Select M neighbors in the Neighbor-miss-LA-table
    Generate M query messages
    Search starts with M neighbors
4. If a hit occurs, sent back results on the reverse path
5. All nodes on the path update appropriate LA-table with learning automata algorithm
```

شکل (3): الگوریتم جستجوی پیشنهادی

در صورتی که گره جاری قبلاً در جستجو شرکت کرده باشد و حداقل یکی از همسایه‌هایش شیء مورد نظر را بیابد، به عبارتی hit برگرداند، بردار احتمال گره جاری با توجه به رابطه (1) پاداش دریافت می‌کند، در غیر این صورت بر اساس رابطه (2)، جریمه می‌شود. در مورد گره‌ای که با توجه به سابقه‌اش، مشارکتی در جستجوهای قبلی

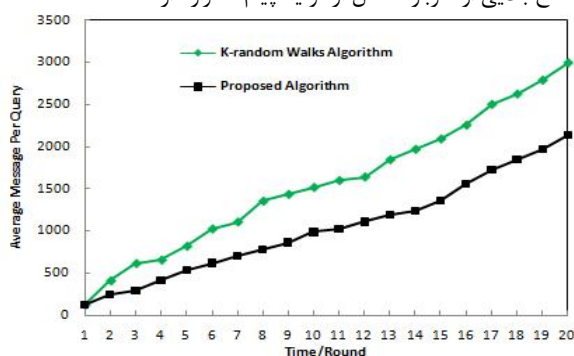


همسایه‌های انتخابی در دورهای اولیه، زیاد باشد. نتایج شبیه‌سازی در شکل 6، نشان می‌دهد که بعد از چهارمین دور، این میانگین کاهش می‌یابد تا اینکه به یک مقدار مشخص همگرا گردد. در روش k -قدم‌های تصادفی، اجرای بیشتر دورهای الگوریتم، میانگین تعداد همسایه‌های انتخاب‌شده را افزایش می‌دهد اما این نرخ افزایش، به کندی صورت می‌گیرد.



شکل (6): مقایسه میانگین تعداد همسایه‌های انتخاب‌شده

شکل 7، مقایسه‌ای بین الگوریتم پیشنهادی و روش k -قدم‌های تصادفی، با توجه به سربار حاصل از تولید پیام‌های ناشی از درخواست را نشان می‌دهد. مقدار TTL شش است و تعداد پیام‌های تولیدی را کنترل می‌کند. همان‌طور که در نمودار دیده می‌شود در الگوریتم پیشنهادی، میانگین تعداد پیام‌های تولید شده به ازای هر درخواست کم است، زیرا هیچ‌گونه پیام و درخواست اضافی برای انجام این کار در شبکه ارسال نمی‌گردد. این روش، فضای اندکی برای ذخیره‌سازی جداول و نگهداری مقادیر موجود در جداول نیاز دارد. روش k -قدم‌های تصادفی نیز به دلیل ارسال تصادفی درخواست‌ها، در سطح بالایی از سربار حاصل از تولید پیام‌ها قرار دارد.



شکل (7): میزان سربار ناشی از تولید پیام به ازای درخواست

جستجو

میانگین تعداد اشیاء کشف‌شده در جستجو به ازای هر درخواست، قابل محاسبه است. شکل 8، نشان می‌دهد که در روش k -قدم‌های تصادفی، به دلیل ارسال درخواست جستجو بدون هیچ الگوی مشخص و به صورت کاملاً تصادفی به سمت گره‌ها، میانگین تعداد کشف‌شده، کم است و تا دور بیستم از شبیه‌سازی، شیب نسبتاً ملایمی را در نمودار از خود نشان می‌دهد. در دورهای ابتدایی اجرای الگوریتم

می‌باشد. در جدول 1، به طور خلاصه، فرضیات شبیه‌سازی به همراه مقادیر اولیه آنها آمده است.

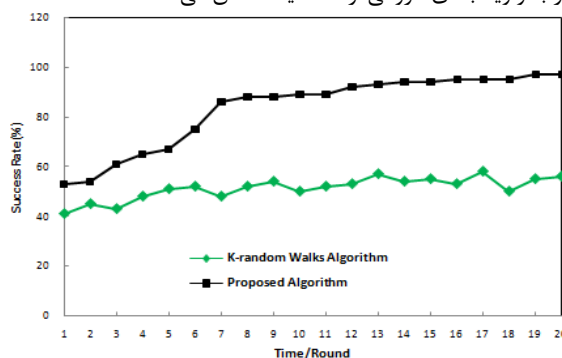
جدول (1): پارامترهای شبیه‌سازی

پارامترها	مقادیر اولیه
همبندی شبکه	گراف تصادفی
نوع شبکه	غیرساخت یافته
اتوماتای یادگیر	LRP
تعداد اشیاء	100
میزان خرابی گره‌ها	20%
TTL	6

4-2- ارزیابی آزمایشات

در این بخش، کارایی الگوریتم پیشنهادی را در مقایسه با روش k -قدم‌های تصادفی، توسط آزمایشات مختلف مورد بررسی قرار می‌دهیم. این آزمایشات هم جنبه‌های شبکه‌ای و هم جنبه‌های کیفیت جستجو را در نظر می‌گیرند.

در شکل 5، میزان موفقیت جستجوی الگوریتم پیشنهادی با روش k -قدم‌های تصادفی مورد مقایسه قرار گرفته است. همان‌طور که نشان داده شده است، نرخ موفقیت الگوریتم پیشنهادی، پس از پنجمین دور اجرا، افزایش می‌یابد. میانگین نرخ موفقیت برای الگوریتم ما، تقریباً 85 درصد است، در حالی که این میزان در روش k -قدم‌های تصادفی، به دلیل انتخاب تصادفی همسایه‌ها به 52 درصد می‌رسد. دلیل اصلی نرخ موفقیت بالا در الگوریتم ما، استفاده از جداولی است که انتخاب همسایه‌ها را با اولویت‌بندی انجام می‌دهد. در بدترین حالت، همه همسایه‌ها برای اجرای جستجو، انتخاب می‌شوند اما با این وجود، این کار با اولویت‌بندی گروهی از همسایه‌ها اتفاق می‌افتد.



شکل (5): میزان موفقیت الگوریتم پیشنهادی در مقایسه با روش

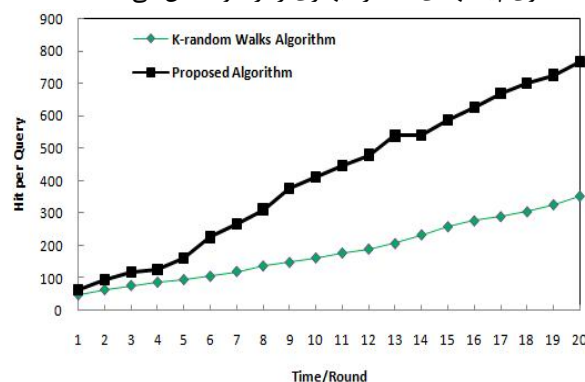
k -قدم‌های تصادفی

در هر دور از اجرای الگوریتم پیشنهادی، می‌توان میانگین تعداد همسایه‌های انتخاب‌شده را محاسبه کرد. در دورهای ابتدایی این الگوریتم، همسایه‌ها به طور تصادفی انتخاب می‌شوند، زیرا مرحله یادگیری آهسته پیش می‌رود. بنابراین، بدیهی است که میانگین تعداد

scheme, IEEE Communication Survey and Tutorial, March 2004.

- [3] Tsoumakos D., and Roussopoulos N., *Analysis and comparison of p2p search methods*, 1st Int. Conf. Scalable Information Systems, Article no. 25, 2006.
- [4] Thampi S. M., Sekaran C. K., *Survey of search and replication schemes in unstructured p2p networks*, Network Protocols and Algorithms, vol 2, no. 1, pp. 93-131, 2010.
- [5] Dorriv R., L'opez-Ortiz A. and Pralat P., *Search algorithms for unstructured peer-to-peer networks*, Proc. Of 32nd IEEE Conference on Local Computer Networks, pp. 343-349, 2007.
- [6] Gkantsidis C., Mihail M., and Saberi A., *Random walks in peer-to-peer networks*, In INFOCOM 2004, Hong Kong, vol. 1, pp. 120-130, 2004.
- [7] Tsoumakos D. and Roussopoulos N., *Adaptive probabilistic search for peer-to-peer networks*, 3rd Int. Conf. P2P Computing, pp. 102-109, 2003.
- [8] Tsoumakos D. and Rossopoulos N., *Probabilistic knowledge discovery and management for p2p networks*, P2P Journal, 2003.
- [9] Sutton R. S., Barto A. G., *Reinforcement learning: introduction*, in Proceeding of the MIT Press, 1996.
- [10] Mance E. and Stephanie S. H., *Reinforcement learning: A tutorial*, in Proceeding of the Wright Laboratory, 1996.
- [11] Thampi S. M., and Sekaran C. K., *Collaborative load-balancing scheme for improving search performance in unstructured p2p networks*, Proc. Of the first Int. Conf. Contemporary Computing, pp. 161-169, August 2008.
- [12] Thampi S. M., and Sekaran C. K., *An efficient distributed search technique for unstructured peer-to-peer networks*, Int. Jou. Computer and Network Security, vol. 8, no. 1, pp. 128-135, January 2008.
- [13] Torabmostaedi F., Meybodi M. R., *An Intelligent Search Algorithm Based on Learning Automata for Peer to Peer Networks*, Int. Conf. Contemporary Issues in Computer and Information Sciences, Zanjan, pp. 495-500, June 2011.
- [14] Watkins C. J. C. H., Dayan P. (1992) *Machine learning*, vol. 8, Springer, pp. 279-292.
- [15] Baumgart I., and Heep B., *Oversim community site*, [Online], Available: <http://www.oversim.org/wiki>

پیشنهادی بر روی شبکه، تعداد اشیایی که در جستجو یافت می‌شوند، به دلیل نرخ یادگیری پایین برای یافتن همسایه‌های مناسب، کم است اما به مرور زمان که میزان یادگیری گره‌ها افزایش می‌یابد، روش جستجوی پیشنهادی عملکرد بهتری را از خود نشان می‌دهد.



شکل (8): میانگین تعداد اشیاء پیدا شده در دو الگوریتم

5- نتیجه‌گیری

در این مقاله، یک الگوریتم جستجوی خودتطبیق برای شبکه‌های نظیربه‌نظیر غیرساخت‌یافته با استفاده از اتوماتای یادگیر معرفی شد. روش پیشنهادی، مبتنی بر انتخاب تطبیقی همسایه‌ها بر اساس بازخورد هر گره می‌باشد. در این روش، مساله انتخاب مقدار k در الگوریتم جستجوی k -قدم‌های تصادفی حل شد. با به‌کارگیری اتوماتای یادگیر برای هر گره، همه همسایه‌هایی که دارای بالاترین احتمال جستجوی موفق در مراحل قبلی جستجو هستند، به صورت تطبیقی انتخاب می‌شوند. سابقه جستجو برای هر گره، در جداولی ذخیره می‌شود که در زمان جستجو، بر اساس مقادیر احتمالی موجود در آنها، تصمیم‌گیری برای انتخاب همسایه‌ها انجام خواهد شد. الگوریتم پیشنهادی توسط شبیه‌سازی با روش k -قدم‌های تصادفی مقایسه شد و نتایج به دست آمده از شبیه‌سازی‌ها، نشان می‌دهد که به دلیل استفاده از جداول سابقه برای هر گره، گره‌هایی که به احتمال بیشتری شیء مورد نظر را دارند، انتخاب می‌شوند و به همین دلیل نرخ موفقیت افزایش می‌یابد. از طرفی، هوشمندی الگوریتم پیشنهادی، منجر به ارسال درخواست‌های کمتر و در نتیجه تولید پیام‌های کمتری در شبکه می‌شود و سربار را کاهش می‌دهد. بنابراین الگوریتم پیشنهادی، نسبت به روش k -قدم‌های تصادفی، به مراتب بهتر عمل می‌کند.

مراجع

- [1] C.S. Wallace, "A Suggestion for a Fast Multiplier", *IEEE Trans. Comput.*, vol. 22, no. 12, pp. 1261-1262, Dec 1973.
- [2] Lua E. K., Crowcroft J., Pias M., Sharma R., and Lim S., *A survey and comparison of peer-to-peer overlay network*