

یک الگوریتم جستجوی قدم‌های تصادفی برای شبکه‌های نظریبه-نظیر با بهره‌گیری از اتوماتای یادگیر

مهندی قربانی^۱, علی محمد صغیری^۲ و محمد رضا میدی^۳

^۱دانشگاه آزاد اسلامی واحد قزوین, m.ghorbani@qiau.ac.ir

^۲دانشگاه صنعتی امیرکبیر, a_m_saghiri@aut.ac.ir

^۳دانشگاه صنعتی امیرکبیر, mmeybodi@aut.ac.ir

چکیده - ساختار توزیع شده شبکه‌های نظریبه-نظیر، سبب شده است که یافتن یک شیء در میان انبوھی از داده‌ها، نیازمند به کارگیری یک روش جستجوی مناسب و هوشمندانه باشد. در روش k -قدم‌های تصادفی، انتخاب k گره به طور تصادفی و بدون در نظر گرفتن وضعیت شبکه، از بین گره‌های همسایه، منجر به کاهش میزان موفقیت در جستجو و نیز افزایش سربار شبکه به میزان قابل توجهی می‌گردد. در این مقاله، یک نسخه جدید و توزیع شده از الگوریتم k -قدم‌های تصادفی با تأکید بر بهره‌گیری از اتوماتای یادگیر ارائه می‌دهیم، به گونه‌ای که مقدار k برای k -قدم‌های تصادفی، نه به طور تصادفی بلکه به صورت تطبیقی و هوشمند انتخاب می‌شود. در الگوریتم پیشنهادی، هر گره دارای بردار احتمال انتخاب در جستجو است که در هر مرحله از جستجو، با توجه به بازخورد دریافتی از محیط تغییر می‌کند. تصمیم‌گیری برای انتخاب گره‌ها، با توجه به این بردار و با استفاده از رای گیری اکثریت از بین گره‌های همسایه انجام می‌شود. به منظور ارزیابی، الگوریتم پیشنهادی با استفاده از نرم افزار شبیه‌ساز *oversim* شبیه‌سازی شده است. نتایج شبیه‌سازی عملکرد مطلوب الگوریتم پیشنهادی، مانند بهبود میزان موفقیت در جستجو و افزایش تعداد اشیاء کشف شده را نشان می‌دهد.

کلید واژه - اتوماتای یادگیر، جستجو، خود-تطبیق، شبکه‌های نظریبه-نظیر، k -قدم‌های تصادفی

می‌گردد. قرارگیری محتویات در شبکه‌های نظریبه-نظیر غیر-ساخت یافته، به گونه‌ای است که گره‌ها به سادگی می‌توانند وارد شبکه شوند یا از آن خارج گردد [۱-۲]. بنابراین، به سادگی، نمی‌توان مکان شیء را پیدا کرد و استفاده از مکانیسم‌های جستجوی مناسب ضروری به نظر می‌رسد. امروزه، اکثر کاربردهای سیستم‌های نظریبه-نظیر، معطوف به شبکه‌های غیر-ساخت یافته نظریبه-نظیر است و طراحی یک روش جستجوی کارا یکی از چالش‌های مهم در این زمینه محسوب می‌شود [۳].

۱ - مقدمه

شبکه‌های نظریبه-نظیر در سال‌های اخیر، به سرعت در حال رشد هستند. بسیاری از کاربردهای شبکه‌های نظریبه-نظیر مانند اشتراک فایل، مدیون ساختار توزیع شده این گونه شبکه‌ها هستند. شبکه‌های نظریبه-نظیر، معمولاً بدون هیچ کنترل کننده مرکزی به منظور تامین برخی امکانات مانند انعطاف‌پذیری، مقیاس‌پذیری و قابلیت اطمینان پیاده‌سازی می‌شوند [۱-۲].

این شبکه‌ها، به دو گروه تقسیم می‌شوند: ساخت یافته و غیرساخت یافته. در شبکه‌های نظریبه-نظیر ساخت یافته، مکان محتویات شبکه، از طریق جداول درهم‌سازی توزیع شده (DHT) کنترل می‌شوند و به همین دلیل سربار کمی به شبکه تحمیل

هوشمندانه انتخاب می‌کند. الگوریتم پیشنهادی، میزان موفقیت جستجو، تعداد اشیاء یافت شده برای هر درخواست و تعداد پیام‌های تولید شده برای هر درخواست جستجو را در مقایسه با روش k-قدم‌های تصادفی بهبود می‌بخشد. به منظور ارزیابی، از نرم افزار شبیه ساز oversim استفاده شده است و نتایج به دست آمده، برتری الگوریتم پیشنهادی را نشان می‌دهند.

سازماندهی ادامه مقاله به این صورت است. در بخش ۲، خلاصه‌ای از کارهای مرتبط انجام شده با پروتکل پیشنهادی در زمینه جستجو ارائه می‌شود. در بخش ۳، اتماناتی یادگیر که به عنوان استراتژی اصلی یادگیری در الگوریتم پیشنهادی استفاده شده است، به اختصار شرح داده می‌شود. الگوریتم پیشنهادی در بخش ۴ و در بخش ۵، نتایج شبیه سازی‌های انجام شده، ارائه می‌گردد. در بخش پایانی، نتیجه گیری بیان شده است.

۲- کارهای مرتبط

همانطور که در بخش مقدمه نیز بحث شد، یکی از چالش‌های مهم در شبکه‌های نظریه‌نظری، یافتن مکان یک شیء از بین گره‌های متعدد در یک شبکه توزیع شده است. یافتن یک استراتژی جستجوی کارا، از اهمیت زیادی برخوردار است.

استراتژی‌های جستجو در شبکه‌های نظریه‌نظری با توجه به میزان آگاهی گره‌ها از وضعیت شبکه، به دو دسته کورکرانه و آگاهانه دسته‌بندی می‌شوند. در روش‌های کورکرانه، هر گره، درخواست را به همه همسایگانش هدایت می‌کند و جستجو زمانی پایان می‌یابد که "موفقیت" یا "شکست" اتفاق بیفتد یا TTL تمام شود. روش‌های کورکرانه، پنهانی باند شبکه را تلف و سربار زیادی ایجاد می‌کنند.

در روش k-قدم‌های تصادفی [۵]، به جای هدایت درخواست‌ها به همه همسایگان، تنها به برخی از آنها که به طور تصادفی انتخاب می‌شوند هدایت انجام می‌گردد. در این روش، پیچیدگی پیام‌های تولید شده بسیار کم است اما به دلیل انتخاب تصادفی همسایگان برای هدایت درخواست‌های جستجو، میزان موفقیت متغیر است.

روش سیل‌آسای محلی با k قدم تصادفی مستقل

اطلاعات قبلی در مسیر جستجو برای مسیریابی مورد استفاده قرار می‌گیرند و هدایت درخواست‌ها، از قبل معین است. در روش‌های احتمالی، درخواست‌ها هم به صورت احتمالاتی و هم به صورت تصادفی ارسال می‌شوند [۳]. روش‌های جستجو بر اساس آگاهی گره‌ها از محتويات، به دو دسته آگاهانه و کورکرانه تقسیم می‌شوند [۴-۵]. در روش‌های آگاهانه، گره‌ها از وضعیت شبکه و مکان قرارگیری محتويات به وسیله ذخیره کردن تعدادی متاداده از اطلاعات، آگاه هستند. در روش‌های ندارند و از روش‌های هیچگونه اطلاعی در مورد مکان اشیاء ندارند و از روش‌های سیل‌آسا برای هدایت درخواست‌ها استفاده می‌کنند. با توجه به دسته‌بندی فوق، روش‌های قدم‌های تصادفی [۵]، نمونه‌هایی از تکنیک‌های احتمالی و کورکرانه هستند. یکی از انواع روش‌های قدم‌های تصادفی، روش k-قدم‌های تصادفی است که از k عدد قدم تصادفی مستقل استفاده می‌کند. وقتی جستجو برای یافتن یک شیء ناموفق باشد، k تا گره از بین گره‌های همسایه به طور تصادفی انتخاب شده و درخواست‌های جستجو به سمت آنها هدایت می‌شوند. جستجو، زمانی موفقیت آمیز است که شیء مورد نظر توسط حداقل یکی از قدم‌های مستقل پیدا شود. در این روش، میزان موفقیت جستجو و تعداد اشیاء یافت شده به دلیل انتخاب تصادفی متغیر است و ممکن است گره‌های نامناسب بیش از یک بار برای مدت طولانی انتخاب شوند. با توجه به مطالعات گذشته [۱۶-۶] و [۱۷]، تکنیک‌های یادگیری تقویتی [۱۴-۱۳] مانند Q-learning [۱۳]، برای بهبود جستجو مورد استفاده قرار گرفته‌اند. با بهره‌گیری از مکانیسم‌های یادگیری در یک شبکه، هر گره می‌تواند وضعیت شبکه را یاد بگیرد و برای مراحل بعدی جستجو بر اساس بازخوردها، تصمیم بگیرد که درخواست‌ها را به کدام گره‌ها هدایت کند. این انتظار می‌رود که روش‌های جستجوی مبتنی بر یادگیری، بر روی زمان پاسخ، سربار شبکه و میزان موفقیت جستجو، تاثیر بهسزایی داشته باشند.

در این مقاله، یک الگوریتم جستجوی جدید، هوشمند و توزیع شده از الگوریتم k-قدم‌های تصادفی ارائه می‌دهیم که به منظور هدایت درخواست‌های جستجو به سمت گره‌های همسایه، از اتماناتی یادگیر بهره می‌گیرد و همسایه‌ها را به صورت

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 ورودی-ها، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه خروجی‌ها و $c \equiv \{c_1, c_2, \dots, c_r\}$ مجموعه احتمال‌های جریمه می‌باشد. احتمال اینکه عمل i نتیجه نامطلوب داشته باشد α_i است. در محیط ایستا، مقادیر 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 تقسیم بندی می‌شوند. در ادامه، اتوماتای یادگیر با ساختار متغیر معرفی می‌شود.

اتوماتای یادگیر با ساختار متغیر را می‌توان توسط چهارتایی $\{\alpha, \beta, p, T\}$ 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 $T[\alpha(n), \beta(n), p(n)] = p(n+1)$ الگوریتم یادگیری می‌باشد. الگوریتم زیر، یک نمونه از الگوریتم‌های یادگیری خطی است. فرض کنید عمل i در مرحله n انتخاب شود.

پاسخ مطلوب:

$$\begin{aligned} p_i(n+1) &= p_i(n) + a[1 - p_i(n)] \\ p_j(n+1) &= (1 - a)p_j(n) \end{aligned} \quad (1)$$

پاسخ نامطلوب:

$$\begin{aligned} p_i(n+1) &= (1 - b)p_i(n) \\ p_j(n+1) &= (b/r - 1) + (1 - b)p_j(n) \end{aligned} \quad (2)$$

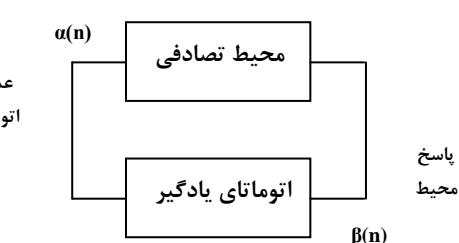
در روابط (1) و (2)، a پارامتر پاداش و b پارامتر جریمه می‌باشند. با توجه به مقادیر a و b سه حالت زیر را می‌توان در نظر گرفت. زمانی که a و b با هم برابر باشند، الگوریتم را L_{RP} می‌نامند، زمانی که b از a خیلی کوچکتر باشد، الگوریتم را L_{RI} می‌نامند و زمانی که b مساوی صفر باشد الگوریتم را L_{ReP} می‌نامند.

روش جستجویی که ما در این مقاله ارائه می‌دهیم به نوع خاصی از اتوماتا نیاز دارد که در آن در هر مرحله بیش از یک عمل اتوماتا انتخاب می‌شود. این اتوماتا که ما آن را اتوماتای

(LFKIRW) [۴]، موارنای از روش‌های سیل‌آسا و قدم‌های تصادفی است. در این تکنیک، جستجو با ارسال سیل‌آسای همه درخواست‌ها به سوی k همسایگان، شیء مورد جستجو، زمانی موفق است که یکی از همسایگان، شیء مورد جستجو را داشته باشد، در غیر اینصورت هر کدام از k همسایه، مجدداً قدم‌های تصادفی مستقل را آغاز می‌کند. تعداد گام‌های بیشتر برای جستجو، منجر به سربار بیشتری در شبکه می‌شود. روش جستجوی احتمالی طبیعی (APS) [۱۰-۹]، بر اساس استفاده از k قدم مستقل و هدایت درخواست‌ها به صورت احتمالی کار می‌کند. انتخاب به صورت احتمالی به جای انتخاب تصادفی، یکی از مزایای این تکنیک محسوب می‌شود. هر گره میانی، درخواست را به سمت همسایه‌اش که یک مقدار احتمالی را در شاخص محلی‌اش ذخیره کرده است هدایت می‌کند. مقادیر شاخص‌ها، با دریافت بازخورد از قدم‌ها به روزرسانی می‌شوند. APS، قابلیت اعتماد را افزایش داده و میزان مصرف پهنای باند را بهبود می‌بخشد.

۳- اتوماتای یادگیر

اتوماتای یادگیر [۱۵-۱۶]، 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 حد اکثر برساند. کارکرد اتوماتای یادگیر در تعامل با محیط، در شکل ۱ مشاهده می‌شود.



شکل ۱: ارتباط با اتوماتای یادگیر و محیط [۱۶]

۴- الگوریتم پیشنهادی

در این مقاله تمرکز بر روی شبکه های نظری به نظری غیر-ساخت یافته است و از KSALA برای آموزش گره ها در روال جستجو بهره می گیریم. در ابتدا، ساختار داده الگوریتم پیشنهادی را توصیف می کنیم و سپس الگوریتم پیشنهادی را شرح خواهیم داد.

همانطور که در بخش ۳ گفته شد، مقادیر احتمالی جهت تصمیم گیری در KSALA بسیار مهم هستند و این مقادیر، بر اساس بازخوردهای محیط، و پاداش/جریمه به روز می شوند. در الگوریتم پیشنهادی ما، هر گره، برای انتخاب همسایه ها، از KSALA بهره می گیرند. به این معنی که در هر گره، جدولی وجود دارد که در آن k همسایه و k مقدار احتمالی از انتخاب این همسایه ها، بر اساس مراحل قبلی جستجو موجود است و با توجه به بازخورد از محیط، همه مقادیر احتمالی از گره جاری، به روز می شوند.

در اولین گام از جستجو، برای هر گره، k همسایه به طور تصادفی انتخاب می شوند. برای هر گره همسایه، احتمال انتخاب، I/k است. زمانی که یک گره، درخواستی را به سمت همسایه ای هدایت می کند، یک سطر به جدول گره جاری افزوده می شود. اگر شیء پیدا نشود، جستجو با انتخاب k تا همسایه با بالاترین مقدار احتمال از بین همه همسایگان ادامه پیدا می کند. همسایه های انتخاب شده، پیغام های درخواست را به سوی همسایه های خود منتشر می کنند. برای به روزرسانی مقادیر احتمالی همسایه ها، از تکنیک رای گیری اکثرباده می شود، به این صورت که پس از دریافت بازخوردها از محیط، اگر بیش از $k/2$ از همسایه های انتخاب شده، شیء را پیدا کردند، مقادیر احتمال همسایگان انتخاب شده افزایش می یابد، در غیر اینصورت جریمه بر روی آن گره ها اعمال خواهد شد. به روزرسانی توسط KSALA انجام می شود. جستجو، زمانی پایان می یابد که شیء پیدا شود یا تمام شود.

در روش پیشنهادی، برای به روزرسانی مقادیر احتمالی از روابط (۳) و (۴) در بخش ۳ استفاده می کنیم. شکل های (۲) و (۳) به ترتیب، به روزرسانی بردارهای احتمال و الگوریتم

یادگیر با k انتخاب عمل (KSALA) [۱۶] می نامیم، بسیار مشابه اتوماتای یادگیر با ساختار متغیر است، با این تفاوت که در اتوماتای یادگیر با k انتخاب عمل، در هر مرحله، k عمل (متفاوت) از اتوماتا انتخاب می شود و سپس نتیجه هر یک از عمل های انتخابی مشخص شده و احتمالات عمل ها به روز می شود. اتوماتای یادگیر با k انتخاب عمل را می توان توسط چهارتایی $\{\alpha_1, \alpha_2, \dots, \alpha_r\}$ نشان داد که $\alpha \equiv \{\alpha, \beta, p, T\}$ مجموعه عمل های اتوماتا، $\{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودی های اتوماتا، $\{p_1, p_2, \dots, p_r\}$ بردار احتمال انتخاب هر یک از عمل ها و $p = \{p_1, p_2, \dots, p_r\}$ بردار احتمال انتخاب عمل الگوریتم یادگیری می باشد. در این اتوماتا، در هر مرحله k عمل انتخاب شده و در محیط اعمال می شوند. $\alpha^h(s) = \{\alpha_1, \dots, \alpha_{r_s^h}\} \subseteq \alpha, r_s^h \leq r$ در این اتوماتا، مجموعه عمل های قابل اعمال اتوماتا برای انتخاب عمل α^h در مرحله s است که $\alpha^h(s) = \alpha$ است. در این اتوماتا در مرحله s عمل انتخاب شده α^h از مجموعه عمل های قابل اعمال اتوماتا برای انتخاب عمل های بعدی حذف می شود. یکی از الگوریتم های خطی برای این اتوماتا، الگوریتم یادگیری خطی ساده است که توسط روابط (۳) و (۴) بیان می شود. فرض کنید عمل انتخاب شده α^h در مرحله n عمل i باشد.

پاسخ مطلوب:

$$\begin{aligned} p_i^{h+1}(n) &= p_i^h(n) + a[1 - p_i^h(n)] \\ p_j^{h+1}(n) &= (1 - a)p_j^h(n); \forall j; j \neq i \end{aligned} \quad (3)$$

پاسخ نامطلوب:

$$\begin{aligned} p_i^{h+1}(1 - b)p_i^h(n) \\ p_j^{h+1}(n) &= (b/r - 1) + (1 - b)p_j^h(n); \forall j; j \neq i \end{aligned} \quad (4)$$

ارتباط روابط (۳) و (۴) با بردار احتمال اصلی توسط رابطه (۵) بیان می شود.

$$\begin{aligned} p_i^1(n) &= p_i(n); \forall i; p_i \in p(n) \\ p_i(n+1) &= p_i^{k+1}(n); \forall i; p_i \in p(n+1) \end{aligned} \quad (5)$$

در این شبکه، ۱۰۰ شیء متفاوت به طور تصادفی میان گرهها توزیع شده‌اند و هر گره با توجه به ظرفیت حافظه‌ای که برای آن در نظر گرفته شده است، تعدادی از این اشیاء را در خود جای داده است. میزان خرابی گره‌ها به طور متوسط ۲۰ درصد در نظر گرفته شده است تا پویایی شبکه در آزمایشات مختلف مد نظر قرار گرفته شده باشد. ماکریم زمان برای تحويل پیغام‌ها در شبکه (TTL) برابر با ۶ می‌باشد و حداقل تعداد قدم‌ها در ارزیابی شبیه‌سازی ۱۵ عدد است.

جدول زیر، به طور خلاصه پارامترهای شبیه‌سازی را با مقادیر پیش فرض نشان می‌دهد:

جدول (۱) : پارامترهای شبیه‌سازی	
مقادیر اولیه	پارامترها
گراف تصادفی	همبندی شبکه
غیرساخت‌یافته	نوع شبکه
۱۰۰	تعداد اشیاء
۱۵	ماکریم تعداد قدم‌ها
۶	TTL

۲-۵- ارزیابی آزمایشات

در این بخش، کلارایی الگوریتم پیشنهادی توسط آزمایشات مختلف مورد بررسی قرار می‌گیرد. روش پیشنهادی با روش‌های k-قدمهای تصادفی، LFKIRW و APS مورد مقایسه قرار می‌گیرد. آزمایشات به دو گروه معیارهای شبکه‌ای مانند سربار شبکه و نیز معیارهای کیفیت جستجو مانند میزان موفقیت در جستجو تعداد اشیاء پیدا شده، دسته‌بندی می‌شوند. در شبیه‌سازی‌های مختلف، تعداد قدم‌ها از ۱ تا ۱۵ متغیر در نظر گرفته شده است.

همان‌طور که در نمودار ۴، نشان داده شده است، میزان موفقیت جستجو در روش k-قدمهای تصادفی به دلیل انتخاب تصادفی قدم‌ها بسیار پایین است، اما در الگوریتم پیشنهادی، پس از انتخاب ۱۵امین قدم، این میزان موفقیت تا پایان ۱۵امین قدم، به ۸۵٪ می‌رسد. گزینش همسایگان با اولویت بالاتر بر اساس

پیشنهادی را نشان می‌دهند.

/N: Number of all the neighbors , K: Number of selected neighbors/

IF the feedback of $[(k/2)+1]$ neighbors is “HIT” THEN
the neighbors with “HIT” feedback updated by EQ (3)
and
the others updated by EQ (4)
ELSE
all k selected neighbors updated by EQ (4)

شکل ۲: به روزرسانی بردار احتمال بر اساس بازخورد

*/CN: Current node for search, Q: Query Keyword
K: The number of selected neighbors, P_{Vector} : The set of probability values/*

1. User submits a query
2. Search query node for Q
3. If Q is not in CN
 - 3.1. Select K neighbors with the highest P_{vector} among all neighbors
 - Generate K query messages
 - Search starts with k neighbors
4. If a hit occurs, sent back results on the reverse path
5. All nodes on the path update appropriate $P_{vectors}$ with KSALA

شکل ۳: الگوریتم جستجوی پیشنهادی

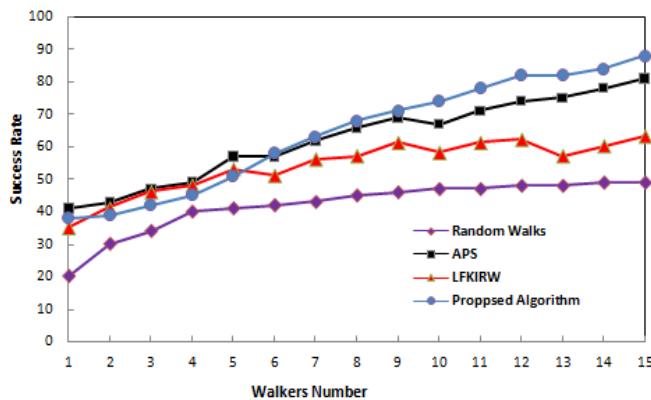
۵- شبیه‌سازی

در این بخش، محیط شبیه‌سازی را توصیف خواهیم کرد و نتایج حاصل از ارزیابی‌های مختلف را خواهیم گفت. در زیر بخش ۱-۱، پارامترهای شبیه‌سازی و مقادیر اولیه برای پیاده‌سازی بیان می‌شود و در زیر بخش ۲-۵، نتایج حاصل از شبیه‌سازی را نشان می‌دهیم.

۱-۵- فرضیات شبیه‌سازی

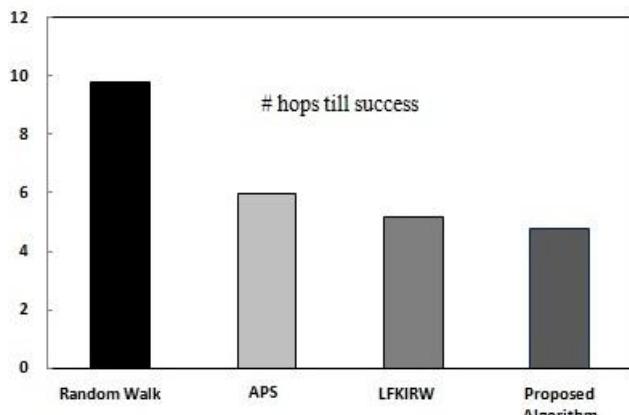
برای شبیه‌سازی الگوریتم جستجوی پیشنهادی از شبیه‌ساز oversim [۱۸] استفاده شده است. شبکه مورد استفاده، یک گراف تصادفی [۷] با ۱۰۰۰ گره است که در آن برای هر گره حداقل یک لینک و حداقل ۳۰ لینک در نظر گرفته شده است.

شکل ۵: تعداد اشیاء یافت شده الگوریتم پیشنهادی در مقایسه با دیگر روش‌ها



شکل ۶: میزان سربار ناشی از تولید پیام برای چهار روش مورد بررسی

در شکل ۷، میزان تاخیر جستجو در چهار الگوریتم مورد توجه قرار گرفته شده است. تاخیر، بر اساس میانگین تعداد گام‌های ملاقات شده برای یک جستجو تعريف می‌شود. تعداد گام کمتر، به معنی میزان تاخیر کمتر و سرعت بیشتر برای کشف اشیاست. با توجه به نمودار زیر، الگوریتم پیشنهادی، تاخیر کمتری نسبت به دیگر الگوریتم‌ها دارد و این متأثر از انتخاب‌های هوشمندانه قدم‌ها در هنگام جستجو می‌باشد.

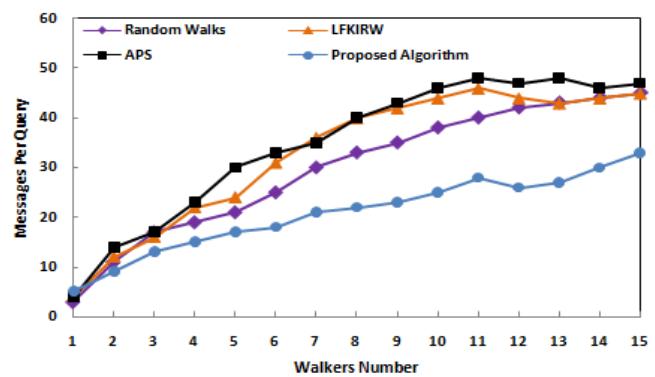


شکل ۷: مقایسه میزان تاخیر جستجو برای چهار الگوریتم مورد بررسی

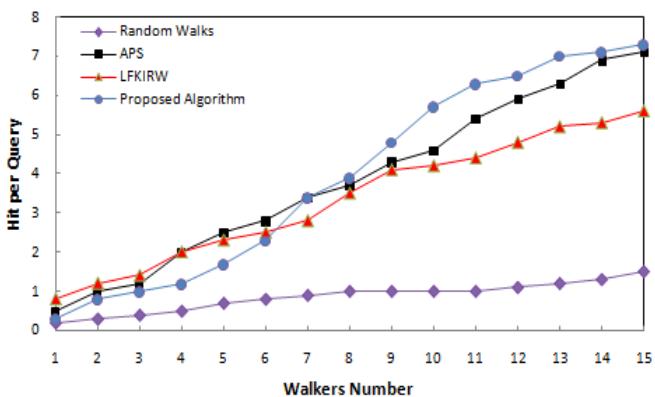
مراحل قبلی جستجو، مهمترین دلیل برای میزان موفقیت بالا برای الگوریتم پیشنهادی محسوب می‌شود. میانگین نرخ موفقیت جستجو برای الگوریتم پیشنهادی، ۶۵٪ است در حالی که برای APS و LFKIRW به ترتیب ۵۳٪ و ۶۰٪ می‌باشد.

تعداد اشیاء کشف شده در هر خواست در شکل ۵ نشان داده شده است. الگوریتم پیشنهادی ما، تعداد اشیاء بیشتری را نسبت به روش LFKIRW و k-قدم‌های تصادفی پیدا می‌کند اما این میزان تا حدود زیادی به روش APS نزدیک است. از آنجایی که در روش پیشنهادی، قدم‌های انتخاب شده از بین همسایگان با مقدار احتمالی بالا هستند، شناس کشف اشیاء در زمان انتشار در خواست‌ها، افزایش می‌یابد.

شکل ۶، تعداد پیغام‌های تولید شده به ازای درخواست‌ها را در چهار روش مورد بررسی، نشان می‌دهد. در الگوریتم k-قدم‌های تصادفی، تقریباً ۷۰٪ از قدم‌ها، پیغام‌های TTL را به هدر می‌دهند. در مقایسه با دیگر الگوریتم‌ها، الگوریتم پیشنهادی تعداد پیام کمتری را به ازای هر درخواست جستجو تولید می‌کند و این بدین دلیل است که انتخاب قدم‌ها در حین جستجو، هوشمندانه انجام می‌شود و TTL به مقدار درستی تنظیم می‌گردد.



شکل ۴: میزان موفقیت الگوریتم پیشنهادی در مقایسه با دیگر روش‌ها



- [5] D. Tsoumakos, and N. Roussopoulos, "Analysis and comparison of p2p search methods," 1st Int. Conf. Scalable Information Systems, Article no. 25, 2006.
- [6] R. Dorrigiv, A. L'opez-Ortiz and P. Pralat, "Search algorithms for unstructured peer-to-peer networks", Proc. Of 32nd IEEE Conference on Local Computer Networks, pp. 343-349, 2007.
- [7] G. H. Fletcher, H. A. Sheth and K. Borner, "Unstructured peer-to-peer networks: topological properties and search performance," Lecture Notes in Computer Science- Agent and Peer-to-Peer Computing, Springer Berlin/ Heidelberg, pp. 14-27, 2005.
- [8] C. Gkantsidis, M. Mihail, and A. Saberi, "Random walks in peer-to-peer networks," In INFOCOM 2004, Hong Kong, vol. 1, pp. 120-130, 2004.
- [9] D. Tsoumakos and N. Roussopoulos, "Adaptive probabilistic search for peer-to-peer networks," 3rd Int. Conf. P2P Computing, pp. 102-109, 2003.
- [10] D. Tsoumakos and N. Rossopoulos, "Probabilistic knowledge discovery and management for p2p networks," P2P Journal, 2003.
- [11] S. M. Thampi and C. K. Sekaran, "Collaborative load-balancing scheme for improving search performance in unstructured p2p networks," Proc. Of the first Int. Conf. Contemporary Computing, pp. 161-169, 2008.
- [12] S. M. Thampi, and C. K. Sekaran, "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] R. S. Sutton and A. G. Barto, "Reinforcement learning: introduction," in Proceeding of the MIT Pres, 1998.
- [14] E. Mance, and S. H. Stephanie, "Reinforcement learning: A tutorial," in Proceeding of the Wright Laboratory, 1996.
- [15] K. Najim, and A. S. Poznyak, "Learning automata: theory and application," in Proceeding of the Tarrytown, New York, Elsevier Science Publishing Ltd., 1994.
- [16] S. M. Abolhasani, M. M. Meybodi, "LADIT: Learning Automata Based Protocol for Routing in Sensor Networks," 2th Conference on Sensor Networks, Yazd, pp. 20-33, 2008.
- [17] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti, "A local search mechanism for peer-to-peer networks," 11th Conf. Information and knowledge management, pp. 300-307, 2002.
- [18] I. Baumgart, and B. Heep, Oversim community site, [Online], Available: <http://www.oversim.org/wiki>

۶- نتیجه‌گیری

در این مقاله، یک الگوریتم جستجوی جدید و تطبیقی از روش k-قدم‌های تصادفی برای شبکه‌های نظریه‌نظری معرفی شد که برای انتخاب همسایه‌های مناسب، از نوع خاصی از اتماتاتی یادگیر، به نام KSALA استفاده گردید. در روش پیشنهادی، هر گره بازخوردهای محیط را دریافت می‌کند و سپس تصمیم می‌گیرد که کدام گره‌ها را برای مشارکت در جستجو انتخاب کند. با به کارگیری اتماتاتی یادگیر برای هر مجموعه کتابی از گره‌ها، همه همسایگانی که بیشترین احتمال را برای موفقیت در جستجو بر اساس دفعات قبلی جستجو دارند، انتخاب می‌شوند. الگوریتم پیشنهادی توسط شبیه‌سازی با سه روش k-قدم‌های تصادفی، LFKIRW و APS مقایسه شد و نتایج بدستآمده از شبیه‌سازی‌ها، نشان می‌دهد که به دلیل استفاده از جداول سابقه برای هر گره، گره‌هایی که به احتمال بیشتری شیء مورد نظر را درآوردند، انتخاب می‌شوند و به این شکل 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. در ضمن، تعداد گام‌های ملاقات‌شده برای رسیدن به گره‌های انتخاب‌شده نیز کاهش می‌یابد و الگوریتم پیشنهادی با تاخیر کمتری، اشیاء را خواهد یافت. بنابراین، الگوریتم پیشنهادی نسبت به روش‌های مذکور در معیارهای سربار شبکه، میزان موفقیت جستجو و میزان تاخیر، به مراتب بهتر عمل می‌کند.

مراجع

- [1] S. Androulidakis, and D. Spinellis, "A survey of peer-to-peer content distribution technologies," ACM Computing Surveys, vol. 36, no. 4, pp. 335-371, December 2004.
- [2] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network scheme," IEEE Communication Survey and Tutorial, March 2004.
- [3] X. Li, and J. Wu, "Searching techniques in peer-to-peer networks," Handbook of Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks, CRC Press, Boca Raton, FL, 2005.
- [4] S. M. Thampi, C. K. Sekaran, "Survey of search and replication schemes in unstructured p2p networks," Network Protocols and Algorithms, vol 2, no. 1, pp. 93-131, 2010.