

یک الگوریتم جستجوی جدید از روش جستجوی قدم های تصادفی برای شبکه های نظیر به- نظیر با بهره گیری از اتوماتای یادگیر

مهدی قربانی^۱، علی محمد صغیری^۲ و محمد رضا میبدی^۳
دانشگاه آزاد اسلامی واحد قزوین، m.ghorbani@qiau.ac.ir
دانشگاه صنعتی امیرکبیر، a_m_saghiri@aut.ac.ir
دانشگاه صنعتی امیرکبیر، mmeybodi@aut.ac.ir

چکیده - ساختار توزیع شده شبکه های نظیر به نظیر، سبب شده است که یافتن یک شیء در میان انبوهی از داده ها، نیازمند به کارگیری یک روش جستجوی مناسب و هوشمندانه باشد. در روش k -قدم های تصادفی، انتخاب k گره به طور تصادفی و بدون در نظر گرفتن وضعیت شبکه، از بین گره های همسایه، منجر به کاهش میزان موفقیت در جستجو و نیز افزایش سربار شبکه به میزان قابل توجهی می گردد. در این مقاله، یک نسخه جدید و توزیع شده از الگوریتم k -قدم های تصادفی با تاکید بر بهره گیری از اتوماتای یادگیر ارائه می دهیم، به گونه ای که مقدار k برای k -قدم های تصادفی، نه به طور تصادفی بلکه به صورت تطبیقی و هوشمند انتخاب می شود. 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 می کند. تصمیم گیری برای انتخاب گره ها، با توجه به این بردار و با استفاده از رای گیری اکثریت از بین گره های همسایه انجام می شود. به منظور ارزیابی، الگوریتم پیشنهادی با استفاده از نرم افزار شبیه ساز *oversim* شبیه سازی شده است. نتایج شبیه سازی عملکرد مطلوب الگوریتم پیشنهادی، مانند بهبود میزان موفقیت در جستجو و افزایش تعداد اشیاء کشف شده را نشان می دهد.

کلید واژه - اتوماتای یادگیر، جستجو، خود-تطبیق، شبکه های نظیر به نظیر، k -قدم های تصادفی

۱- مقدمه

می گردد. قرارگیری محتویات در شبکه های نظیر به نظیر غیر-ساخت یافته، به گونه ای است که گره ها به سادگی می توانند وارد شبکه شوند یا از آن خارج گردند [۱-۲]. بنابراین، به سادگی، نمی توان مکان شیء را پیدا کرد و استفاده از مکانیسم های جستجوی مناسب ضروری به نظر می رسد. امروزه، اکثر کاربردهای سیستم های نظیر به نظیر، معطوف به شبکه های غیر-ساخت یافته نظیر به نظیر است و طراحی یک روش جستجوی کارا یکی از چالش های مهم در این زمینه محسوب می شود [۳].

روش های جستجو در شبکه های نظیر به نظیر غیرساخت یافته، با توجه به معیارهای متفاوتی دسته بندی می شوند. با توجه به نحوه هدایت درخواست های جستجو، روش های جستجو در دو گروه قطعی و احتمالی قرار می گیرند. در روش های قطعی،

شبکه های نظیر به نظیر در سال های اخیر، به سرعت در حال رشد هستند. بسیاری از کاربردهای شبکه های نظیر به نظیر مانند اشتراک فایل، مدیون ساختار توزیع شده این گونه شبکه ها هستند. شبکه های نظیر به نظیر، معمولاً بدون هیچ کنترل کننده مرکزی به منظور تامین برخی امکانات مانند انعطاف پذیری، مقیاس پذیری و قابلیت اطمینان پیاده سازی می شوند [۱-۲].

این شبکه ها، به دو گروه تقسیم می شوند: ساخت یافته و غیرساخت یافته. در شبکه های نظیر به نظیر ساخت یافته، مکان محتویات شبکه، از طریق جداول درهم سازی توزیع شده (DHT) کنترل می شوند و به همین دلیل سربار کمی به شبکه تحمیل

هوشمندانه انتخاب می کند. الگوریتم پیشنهادی، میزان موفقیت جستجو، تعداد اشیاء یافت شده برای هر درخواست و تعداد پیام های تولید شده برای هر درخواست جستجو را در مقایسه با روش k -قدم های تصادفی بهبود می بخشد. به منظور ارزیابی، از نرم افزار شبیه ساز oversim استفاده شده است و نتایج به دست آمده، برتری الگوریتم پیشنهادی را نشان می دهند.

سازماندهی ادامه مقاله به این صورت است. در بخش ۲، خلاصه ای از کارهای مرتبط انجام شده با پروتکل پیشنهادی در زمینه جستجو ارائه می شود. در بخش ۳، اتوماتای یادگیر که به عنوان استراتژی اصلی یادگیری در الگوریتم پیشنهادی استفاده شده است، به اختصار شرح داده می شود. الگوریتم پیشنهادی در بخش ۴ و در بخش ۵، نتایج شبیه سازی های انجام شده، ارائه می گردد. در بخش پایانی، نتیجه گیری بیان شده است.

۲- کارهای مرتبط

همانطور که در بخش مقدمه نیز بحث شد، یکی از چالش های مهم در شبکه های نظیر به نظیر، یافتن مکان یک شیء از بین گره های متعدد در یک شبکه توزیع شده است. یافتن یک استراتژی جستجوی کارا، از اهمیت زیادی برخوردار است.

استراتژی های جستجو در شبکه های نظیر به نظیر با توجه به میزان آگاهی گره ها از وضعیت شبکه، به دو دسته کورکورانه و آگاهانه دسته بندی می شوند. در روش های کورکورانه، هر گره، درخواست را به همه همسایگانش هدایت می کند و جستجو زمانی پایان می یابد که "موفقیت" یا "شکست" اتفاق بیفتد یا TTL تمام شود. روش های کورکورانه، پهنای باند شبکه را تلف و سربار زیادی ایجاد می کنند.

در روش k -قدم های تصادفی [۵]، به جای هدایت درخواست ها به همه همسایگان، تنها به برخی از آنها که به طور تصادفی انتخاب می شوند هدایت انجام می گردد. در این روش، پیچیدگی پیام های تولید شده بسیار کم است اما به دلیل انتخاب تصادفی همسایگان برای هدایت درخواست های جستجو، میزان موفقیت متغیر است.

روش سیل آسای محلی با k قدم تصادفی مستقل

اطلاعات قبلی در مسیر جستجو برای مسیریابی مورد استفاده قرار می گیرند و هدایت درخواست ها، از قبل معین است. در روش های احتمالی، درخواست ها هم به صورت احتمالاتی و هم به صورت تصادفی ارسال می شوند [۳]. روش های جستجو بر اساس آگاهی گره ها از محتویات، به دو دسته آگاهانه و کورکورانه تقسیم می شوند [۴-۵]. در روش های آگاهانه، گره ها از وضعیت شبکه و مکان قرارگیری محتویات به وسیله ذخیره کردن تعدادی متاداده از اطلاعات، آگاه هستند. در روش های کورکورانه، گره ها هیچگونه اطلاعی در مورد مکان اشیاء ندارند و از روش های سیل آسا برای هدایت درخواست ها استفاده می کنند. با توجه به دسته بندی فوق، روش های قدم های تصادفی [۵]، نمونه هایی از تکنیک های احتمالی و کورکورانه هستند. یکی از انواع روش های قدم های تصادفی، روش k -قدم های تصادفی است که از عدد قدم تصادفی مستقل استفاده می کند. وقتی جستجو برای یافتن یک شیء ناموفق باشد، k تا گره از بین گره های همسایه به طور تصادفی انتخاب شده و درخواست های جستجو به سمت آنها هدایت می شوند. جستجو، زمانی موفقیت آمیز است که شیء مورد نظر توسط حداقل یکی از قدم های مستقل پیدا شود. در این روش، میزان موفقیت جستجو و تعداد اشیاء یافت شده به دلیل انتخاب تصادفی متغیر است و ممکن است گره های نامناسب بیش از یک بار برای مدت طولانی انتخاب شوند.

با توجه به مطالعات گذشته [۶-۱۲] و [۱۷]، تکنیک های یادگیری تقویتی [۱۳-۱۴] مانند Q-learning [۱۳]، برای بهبود جستجو مورد استفاده قرار گرفته اند. با بهره گیری از مکانیسم های یادگیری در یک شبکه، هر گره می تواند وضعیت شبکه را یاد بگیرد و برای مراحل بعدی جستجو بر اساس بازخوردها، تصمیم بگیرد که درخواست ها را به کدام گره ها هدایت کند. این انتظار می رود که روش های جستجوی مبتنی بر یادگیری، بر روی زمان پاسخ، سربار شبکه و میزان موفقیت جستجو، تاثیر به سزایی داشته باشند.

در این مقاله، یک الگوریتم جستجوی جدید، هوشمند و توزیع شده از الگوریتم k -قدم های تصادفی ارائه می دهیم که به منظور هدایت درخواست های جستجو به سمت گره های همسایه، از اتوماتای یادگیر بهره می گیرد و همسایه ها را به صورت

محیط را می توان توسط سه تایی $E \equiv \{\alpha, \beta, c\}$ نشان داد که در آن $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه ورودی-ها، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه خروجی ها و $c \equiv \{c_1, c_2, \dots, c_r\}$ مجموعه احتمال های جریمه می باشد. احتمال اینکه عمل c_i نتیجه نامطلوب داشته باشد α_i است. در محیط ایستا، مقادیر c_i بدون تغییر می ماند، حال آنکه در محیط غیرایستا این مقادیر در طی زمان تغییر می کنند. اتوماتاهای یادگیر به دو گروه با ساختار ثابت و با ساختار متغیر تقسیم بندی می شوند. در ادامه، اتوماتای یادگیر با ساختار متغیر معرفی می شود.

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

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

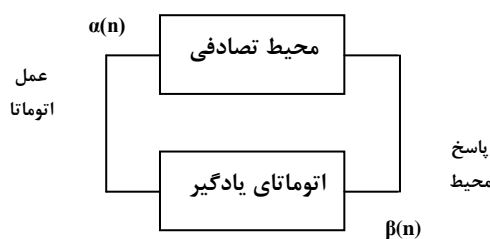
در روابط (۱) و (۲)، a پارامتر پاداش و b پارامتر جریمه می باشند. با توجه به مقادیر a و b سه حالت زیر را می توان در نظر گرفت. زمانی که a و b با هم برابر باشند، الگوریتم را LRP می نامند، زمانی که b از a خیلی کوچکتر باشد، الگوریتم را LRI می نامند و زمانی که b مساوی صفر باشد الگوریتم را LRI می نامند.

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

(LFKIRW) [۴]، موازنه ای از روش های سیل آسا و قدم های تصادفی است. در این تکنیک، جستجو با ارسال سیل آسای همه درخواست ها به سوی k همسایه انتخاب شده انجام می شود. جستجو، زمانی موفق است که یکی از همسایگان، شیء مورد جستجو را داشته باشد، در غیر این صورت هر کدام از k همسایه، مجدداً قدم های تصادفی مستقل را آغاز می کنند. تعداد گام های بیشتر برای جستجو، منجر به سربار بیشتری در شبکه می شود. روش جستجوی احتمالی تطبیقی (APS) [۹-۱۰]، بر اساس استفاده از k قدم مستقل و هدایت درخواست ها به صورت احتمالی کار می کند. انتخاب به صورت احتمالی به جای انتخاب تصادفی، یکی از مزایای این تکنیک محسوب می شود. هر گره میانی، درخواست را به سمت همسایه اش که یک مقدار احتمالی را در شاخص محلی اش ذخیره کرده است هدایت می کند. مقادیر شاخص ها، با دریافت بازخورد از قدم ها به روزرسانی می شوند. APS، قابلیت اعتماد را افزایش داده و میزان مصرف پهنای باند را بهبود می بخشد.

۳- اتوماتای یادگیر

اتوماتای یادگیر [۱۵-۱۶]، ماشینی است که می تواند 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 می شود و اتوماتا از این پاسخ در انتخاب عمل بعدی تاثیر می گیرد. 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 ۱ مشاهده می شود.



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

۴- الگوریتم پیشنهادی

در این مقاله تمرکز بر روی شبکه های نظیر به نظیر 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، $1/k$ 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 اکثریت استفاده 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 این صورت 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 TTL 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 چهار تایی $\{\alpha, \beta, p, T\}$ نشان داد که $\alpha \equiv \{\alpha_1, \alpha_2, \dots, \alpha_r\}$ مجموعه عمل های اتوماتا، $\beta \equiv \{\beta_1, \beta_2, \dots, \beta_m\}$ مجموعه ورودی های اتوماتا، $p = \{p_1, p_2, \dots, p_r\}$ بردار احتمال انتخاب هر یک از عمل ها و $p(n+1) = T[\alpha(n), \beta(n), p(n)]$ الگوریتم یادگیری می باشد. در این اتوماتا، در هر مرحله k عمل انتخاب شده و در محیط اعمال می شوند. $\alpha^h(s) = \{\alpha_1, \dots, \alpha_{r_s}^h\}$ در این اتوماتا، مجموعه عمل های قابل اعمال اتوماتا برای انتخاب عمل h ام در مرحله s است که $\alpha^1(s) = \alpha$ است. در این اتوماتا در مرحله s ، عمل انتخاب شده h ام 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 h ام در مرحله n ام عمل α_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. ماکزیمم زمان برای تحویل پیغام در شبکه (TTL) برابر با ۶ می‌باشد و حداکثر 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
غیرساخت یافته	نوع شبکه
۱۰۰	تعداد اشیاء
۱۵	ماکزیمم تعداد قدم‌ها
۶	TTL

۵-۲- ارزیابی آزمایشات

در این بخش، کارایی 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: به‌روزرسانی 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 ۵-۱، پارامترهای شبیه‌سازی و مقادیر اولیه برای پیاده‌سازی بیان می‌شود و در زیر بخش ۵-۲، نتایج حاصل از شبیه‌سازی را نشان می‌دهیم.

۵-۱- فرضیات شبیه‌سازی

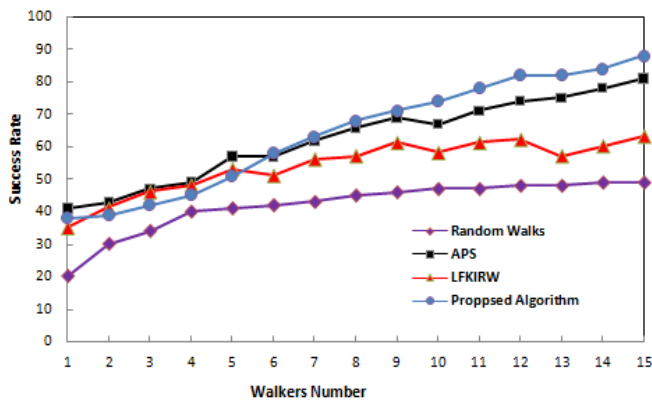
برای شبیه‌سازی الگوریتم جستجوی پیشنهادی از شبیه‌ساز oversim [۱۸] استفاده شده است. شبکه مورد استفاده، یک گراف تصادفی [۷] با ۱۰۰۰ گره است که در آن برای هر گره حداقل یک لینک و حداکثر ۳۰ لینک در نظر گرفته شده است.

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

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-قدم های تصادفی پیدا می کند اما این میزان تا حدود زیادی به روش APS نزدیک است. از آنجایی که در روش پیشنهادی، قدم های انتخاب شده از بین همسایگان با مقدار احتمالی بالا هستند، شانس کشف اشیاء در زمان انتشار درخواست ها، افزایش می یابد.

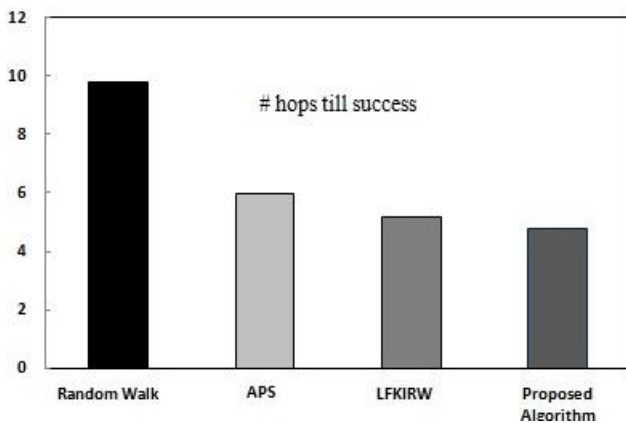
شکل ۶، تعداد پیغام های تولید شده به ازای درخواست ها را در چهار روش مورد بررسی، نشان می دهد. در الگوریتم k-قدم های تصادفی، تقریباً ۷۰٪ از قدم ها، پیغام های TTL را به هدر می دهند. در مقایسه با دیگر الگوریتم ها، الگوریتم پیشنهادی، تعداد پیام کمتری را به ازای هر درخواست جستجو تولید می کند و این بدین دلیل است که انتخاب قدم ها در حین جستجو، هوشمندانه انجام می شود و TTL به مقدار درستی تنظیم می گردد.

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

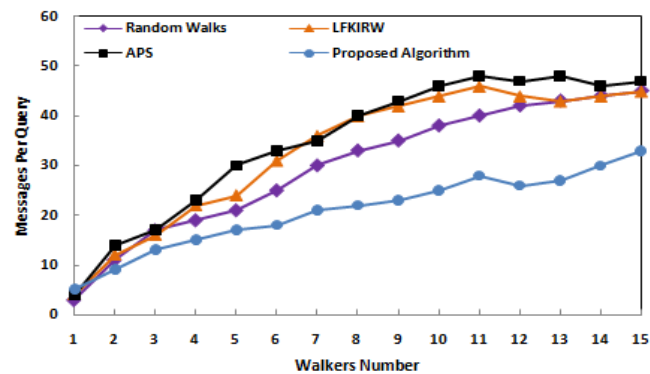


شکل ۶: میزان سربار ناشی از تولید پیام برای چهار روش مورد بررسی

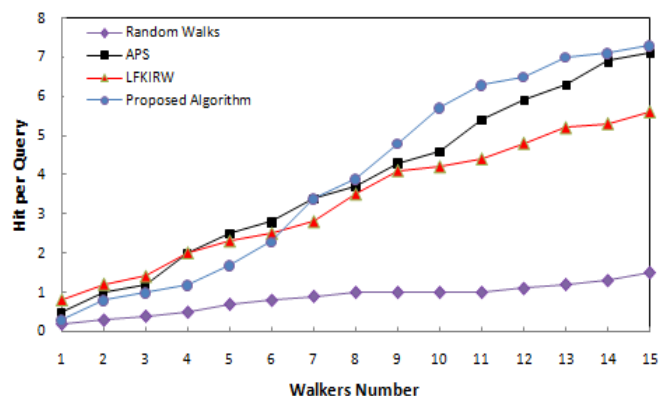
در شکل ۷، میزان تاخیر جستجو در چهار الگوریتم مورد توجه قرار گرفته شده است. تاخیر، بر اساس میانگین تعداد گام های ملاقات شده برای یک جستجو تعریف می شود. تعداد گام کمتر، به معنی میزان تاخیر کمتر و سرعت بیشتر برای کشف اشیاء است. با توجه به نمودار زیر، الگوریتم پیشنهادی، تاخیر کمتری نسبت به دیگر الگوریتم ها دارد و این متاثر از انتخاب های هوشمندانه قدم ها در هنگام جستجو می باشد.



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



شکل ۴: میزان موفقیت الگوریتم پیشنهادی در مقایسه با دیگر روش ها



۶- نتیجه گیری

در این مقاله، یک الگوریتم جستجوی جدید و تطبیقی از روش k-قدم های تصادفی برای شبکه های نظیر به نظیر معرفی شد که برای انتخاب همسایه های مناسب، از نوع خاصی از اتوماتای یادگیر، به نام KSALA استفاده گردید. در روش پیشنهادی، هر گره بازخوردهای محیط را دریافت می کند و سپس تصمیم می گیرد که کدام گره ها را برای مشارکت در جستجو انتخاب کند. با به کارگیری اتوماتای یادگیر برای هر مجموعه کتایی از گره ها، همه همسایگانی که بیشترین احتمال را برای موفقیت در جستجو بر اساس دفعات قبلی جستجو دارند، انتخاب می شوند. الگوریتم پیشنهادی توسط شبیه سازی با سه روش k-قدم های تصادفی، LFKIRW و APS مقایسه شد و نتایج به دست آمده از شبیه سازی ها، نشان می دهد که به دلیل استفاده از جداول سابقه برای هر گره، گره هایی که به احتمال بیشتری شیء مورد نظر را دارند، انتخاب می شوند و به این شکل نرخ موفقیت افزایش می یابد. از طرفی هوشمندی الگوریتم پیشنهادی، منجر به ارسال درخواست های کمتر و در نتیجه تولید پیام های کمتری در شبکه می شود و سربار را کاهش می دهد. در ضمن، تعداد گام های ملاقات شده برای رسیدن به گره های انتخاب شده نیز کاهش می یابد و الگوریتم پیشنهادی با تاخیر کمتری، اشیاء را خواهد یافت. بنابراین، الگوریتم پیشنهادی نسبت به روش های مذکور در معیارهای سربار شبکه، میزان موفقیت جستجو و میزان تاخیر، به مراتب بهتر عمل می کند.

مراجع

- [5] D. Tsumakos, and N. Roussopoulos, "Analysis and comparison of p2p search methods," 1st Int. Conf. Scalable Information Systems, Article no. 25, 2006.
- [6] R. Dorrigin, 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. Tsumakos and N. Roussopoulos, "Adaptive probabilistic search for peer-to-peer networks," 3rd Int. Conf. P2P Computing, pp. 102-109, 2003.
- [10] D. Tsumakos 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 Press, 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>

- [1] S. Androutesellis, 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.