

یک الگوریتم خوشه‌بندی مبتنی بر اتوماتای یادگیر برای شبکه‌های حسگر

ناصر فرج زاده*، محمد رضا میبیدی[†]

چکیده

خوشه‌بندی گره‌ها در شبکه‌های حسگر یکی از روش‌های موثر در طولانی کردن عمر این شبکه‌ها بشمار می‌آید. در این مقاله الگوریتمی مبتنی بر اتوماتاهای یادگیر جهت خوشه‌بندی گره‌ها (LACA) در شبکه‌های حسگر پیشنهاد می‌گردد. الگوریتم پیشنهادی یک الگوریتم توزیع شده است و مستقل از اندازه و ساختار شبکه حسگر عمل می‌کند. کارایی الگوریتم پیشنهادی از طریق شبیه‌سازی و مقایسه نتایج بدست آمده با نتایج روش‌های پیشین مورد ارزیابی قرار گرفته است. نتایج بدست آمده حاکی از کارایی بالای الگوریتم پیشنهادی است.

کلمات کلیدی

شبکه‌های حسگر، خوشه‌بندی، اتوماتاهای یادگیر.

Learning Automata-based Clustering Algorithm for Sensor Networks

Nacer Farajzadeh, Mohammad Reza Meybodi

Abstract

Clustering sensor nodes is one of the effective ways which extends the life time of sensor networks. In this paper, a learning automata-based clustering algorithm (LACA) for a sensor network is proposed. The proposed algorithm is completely distributed and independent of network size and topology. The performance of the LACA via the computer simulation was evaluated and compared with other clustering algorithms. The simulation results show the high performance of the proposed clustering algorithm.

Keywords

Sensor Networks, Clustering, Learning Automata.

* دانشگاه آزاد اسلامی، واحد اهر، ایران nacer_farajzadeh@yahoo.com

[†] دانشکده مهندسی کامپیوتر و فناوری اطلاعات، دانشگاه صنعتی امیرکبیر تهران، ایران mmeybodi@aut.ac.ir

۱- مقدمه

های همسایه و توان ارسال انتخاب می‌نماید. الگوریتم خوشه‌بندی سریع، بودجه‌ای را برای گره رهبر در نظر می‌گیرد و خوشه‌بندی را بر اساس نحوه‌ی هزینه کردن این بودجه انجام می‌دهد. الگوریتم خوشه‌بندی بر اساس رای گیری بر پایه‌ی امتیازاتی که هر گره از گره‌های همسایه اخذ می‌کند، عمل می‌نماید. الگوریتم خوشه‌بندی سلسله‌مراتبی بر اساس راهکار ایجاد درخت پوشا عمل می‌کند. الگوریتم خوشه‌بندی وفق‌پذیر شبکه را به شش ضلعی‌های منتظم تقسیم می‌کند و نزدیک‌ترین گره به مرکز هر شش ضلعی را به عنوان گره رهبر آن شش ضلعی انتخاب می‌کند. از بین الگوریتم‌های فوق، الگوریتم‌های خوشه‌بندی وفق‌پذیر (ACA)، خوشه‌بندی بر اساس امتیازدهی (VCA) و خوشه‌بندی سریع (RCA) جهت مقایسه با الگوریتم پیشنهادی انتخاب شده‌اند.

در این مقاله الگوریتمی مبتنی بر اتوماتاهای یادگیر جهت خوشه‌بندی گره‌ها (LACA¹) در شبکه‌های حسگر ارائه می‌گردد. الگوریتم پیشنهادی یک الگوریتم توزیع شده است و مستقل از اندازه و ساختار شبکه حسگر عمل می‌کند. کارایی الگوریتم پیشنهادی از طریق شبیه‌سازی و مقایسه نتایج آن با نتایج حاصله از روش‌های خوشه‌بندی وفق‌پذیر (ACA)، خوشه‌بندی بر اساس امتیازدهی (VCA) و خوشه‌بندی سریع (RCA)، مورد ارزیابی قرار می‌گیرد. ادامه مقاله بدین صورت سازماندهی شده است. در بخش ۲ اتوماتاهای یادگیر به اختصار شرح داده می‌شود. در بخش ۳ الگوریتم پیشنهادی و در بخش ۴ نتایج شبیه‌سازی‌ها ارائه می‌گردد. بخش پایانی مقاله نتیجه‌گیری می‌باشد.

۲- اتوماتاهای یادگیر

یک اتوماتای یادگیر یک ماشین با حالات محدود است که می‌تواند تعداد محدودی عمل را انجام دهد. هر عمل انتخاب شده، توسط یک محیط تصادفی ارزیابی شده و پاسخی به اتوماتای یادگیر داده می‌شود. اتوماتای یادگیر از این پاسخ استفاده نموده و عمل خود را برای مرحله بعد انتخاب می‌کند. در طی این فرآیند، اتوماتای یادگیر فرا می‌گیرد که چگونه بهترین عمل را از بین اعمال مجاز خود انتخاب نماید. شکل (۱) ارتباط بین اتوماتای یادگیر و محیط را نشان می‌دهد.

محیط را می‌توان توسط سه تایی $E \equiv \{a, b, c\}$ نشان داد که در آن $a = \{a_1, a_2, \dots, a_r\}$ مجموعه ورودی‌ها، $b = \{b_1, \dots, b_m\}$ مجموعه خروجی‌ها و $c = \{c_1, \dots, c_p\}$ مجموعه احتمال‌های جریمه است. زمانیکه b_i دو مقداری باشد، محیط را محیطی از نوع P می‌نامیم. در این محیط $b_i(n) = 1$ را بعنوان پاسخ مثبت یا موفقیت فرض می‌کنیم. در یک محیط از نوع Q، b_i می‌تواند بطور گسسته یکی از مقادیر محدود در فاصله $[0, 1]$ را اختیار کند و در مدل S، b_i متغیر تصادفی بین صفر و یک است ($b_i(n) \in [0, 1]$). احتمال اینکه عمل a_i نتیجه نامطلوب داشته باشد، است. در محیط ایستا مقادیر c_i در طی زمان بدون تغییر می‌مانند، حال آنکه در محیط

یک شبکه‌ی حسگر از تعداد زیادی گره‌های حسگر تشکیل شده است. حسگرها که در محیط یک پدیده بطور تصادفی پخش شده‌اند، وظیفه جمع‌آوری اطلاعاتی خاص از محیط، پردازش و در نهایت ارسال آنها به گره‌های اصلی جمع‌آوری‌کننده‌ی اطلاعات^۱ را دارند [1]. بدلیل تراکم بالای گره‌های حسگر در واحد سطح و در نتیجه نزدیکی آنها با یکدیگر، ارتباط‌های چندگامی در این گونه شبکه‌ها مفیدتر و مقرون به صرفه‌تر از ارتباط‌های تک‌گامی هستند. اما با توجه به انرژی محدود هر حسگر و اینکه بیشترین انرژی در حسگرها صرف ایجاد ارتباط با حسگرهای دیگر می‌شود، استفاده از ارتباط‌های چندگامی باعث مصرف انرژی در حسگرها و در نتیجه کاهش عمر شبکه‌ی حسگر می‌گردد. یکی از راه‌حل‌های این مشکل، خوشه‌بندی^۲ گره‌ها است. در فرآیند خوشه‌بندی، تعدادی از گره‌های حسگر با برد رادیویی تعریف شده با همدیگر تشکیل یک خوشه را داده و یکی از گره‌ها را از بین خود بعنوان معرف دسته^۳ انتخاب می‌نمایند که مسئول ارتباط با معرف خوشه‌های دیگر است. خوشه‌های تشکیل شده ممکن است در آینده با یکدیگر برای تشکیل خوشه‌های بزرگتر ادغام شوند. همچنین ممکن است یک یا چند ایستگاه جهت جمع‌آوری اطلاعات خوشه‌ها و ایجاد درجه ارتباطی با سایر شبکه‌های خارجی در نظر گرفته شود [1] و [2]. خوشه‌بندی در شبکه‌های حسگر دارای مزایایی است که می‌توان به تجمیع داده‌ها جهت کاهش انرژی مصرفی توسط حسگرها [3] و [4]، تسهیل عمل جمع‌آوری اطلاعات از شبکه‌ی حسگر [5]، شکل‌دهی ساختاری مناسب جهت مسیریابی توسعه‌پذیر [6] و [7] و انتشار موثر داده‌ها در شبکه [8] اشاره کرد.

در پایان یک الگوریتم خوشه‌بندی، گره‌ها در خوشه‌های مجزا سازماندهی می‌شوند. هر یک از خوشه‌ها شامل یک گره رهبر و چندین گره تابع^۴ که در محدوده‌ی شعاع رادیویی گره رهبر واقع شده‌اند، می‌باشند. هر گره فقط عضو یک خوشه می‌باشد. بدین معنی که حتی با وجود چندین گره رهبر در حوزه، هر یک از گره‌ها تنها یک رهبر را برای خود بر می‌گزینند. هدف یک الگوریتم خوشه‌بندی، پیدا کردن یک خوشه‌بندی با حداقل تعداد خوشه است که در عین پوشش کل شبکه با همدیگر هم‌پوشانی نداشته باشند. با به حداقل رساندن تعداد خوشه‌ها، تعداد گره‌های رهبر در شبکه به حداقل میرسد که این کار باعث کاهش رقابت گره‌های رهبر برای دستیابی به کانال‌های ارتباطی و در نتیجه افزایش کارایی الگوریتم‌هایی مانند مسیریابی و تجمیع داده‌ها که بر اساس خوشه‌بندی عمل می‌نمایند، می‌شود.

الگوریتم‌های متعددی برای خوشه‌بندی از جمله خوشه‌بندی توزیع شده^۵ [9]، خوشه‌بندی وزن دار^۶ [10]، خوشه‌بندی سریع^۷ [11]، خوشه‌بندی بر اساس رای‌گیری^۸ [12]، خوشه‌بندی سلسله‌مراتبی^۹ [13] و خوشه‌بندی وفق‌پذیر^{۱۰} [14] گزارش شده‌اند. الگوریتم خوشه‌بندی وزندار گره‌های رهبر را بر اساس پارامترهایی مانند تعداد گره-

متناظر با اتوماتای یادگیر می‌باشد. بردار P ، بردار احتمال انتخاب اعمال اتوماتای یادگیر می‌باشد که در آن p_i احتمال انتخاب عمل a_i است و یا به عبارتی احتمال انتخاب همسایه i ام بعنوان گره رهبر است.

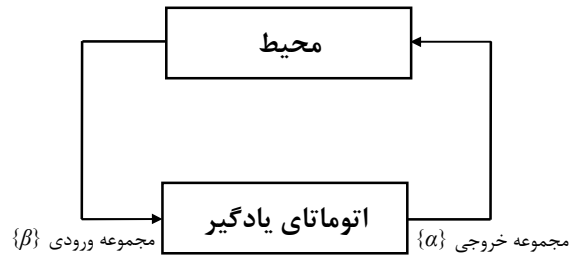
بعد از پراکنده شدن گره‌های حسگر، هر یک از آنها اقدام به شناسایی گره‌های همسایه خود می‌نمایند. این کار با انتشار پیام Hello صورت می‌گیرد؛ هر گره حسگری که پیام Hello را دریافت کرده باشد، با ارسال پیامی که شامل ID خودش است، به پیام رسیده پاسخ می‌دهد. با دریافت پاسخ از هر یک گره‌های همسایه توسط گره منتشر کننده پیام Hello، تعداد همسایه‌های آن گره (r) تعیین می‌گردد. با معلوم شدن تعداد گره‌های همسایه هر یک از گره‌ها حسگر، اتوماتاهای متناظر هر یک از گره‌ها، مقادیر اولیه بردار احتمالات انتخاب اعمال خود را برابر با $p_i(0) = 1/r$ می‌کنند.

در ابتدای مرحله t ، اتوماتای یادگیر هر گره با استفاده از بردار احتمال انتخاب اعمالش (P)، یکی از همسایه‌های خود را به عنوان گره رهبر انتخاب می‌کند و سپس پیامی بنام IEY^{۱۵} را به آن ارسال می‌نماید. اگر گره دریافت کننده پیام IEY عضو هیچ خوشه‌ای نبوده و انرژی باقیمانده‌ی آن بزرگتر یا مساوی مقدار آستانه باشد، آنگاه طبق الگوریتم یادگیری احتمال انتخابش افزایش و در غیر اینصورت کاهش می‌یابد. در الگوریتم پیشنهادی، موارد زیر نیز مورد توجه قرار می‌گیرد.

- اگر گره‌ی همسایه‌ای نداشته باشد (یعنی هیچ پاسخی برای پیام Hello دریافت ننماید)، آنگاه خود را بعنوان گره رهبر اعلام می‌کند.
- اگر گره‌ی یک پیام IEY از طرف یکی از همسایه‌هایش دریافت نماید و قبلاً عضو خوشه‌ای دیگر نباشد، خود را بعنوان گره رهبر اعلام خواهد کرد و در صورتی که عضو خوشه دیگر بوده و یا پیش از این بعنوان رهبر انتخاب شده باشد، پیام IEY دریافت شده را نادیده می‌گیرد.
- اگر گره A داده‌ای (بجز پیام‌ها) از گره B که گره A را بعنوان رهبر خود انتخاب کرده است، دریافت نماید و خود گره A عضو یک خوشه دیگر باشد (یعنی گره رهبر محسوب نشود)، آنگاه داده‌ی دریافت شده از گره B را نادیده می‌گیرد.
- اگر گره A گره B را بعنوان رهبر انتخاب کند و پاسخی برای پیام IEY از آن دریافت ننماید (به دلیل خراب شدن و یا اتمام انرژی)، آنگاه احتمال انتخاب گره B را بعنوان رهبر کاهش می‌دهد.

در آزمایش‌های انجام شده، مقدار آستانه (با توجه به اینکه مقدار اولیه انرژی هر یک از گره‌های حسگر 1.0J در نظر گرفته شده است) برابر با 0.3J تعیین شده است. شکل (۲) شبه کد الگوریتم پیشنهادی را که توسط یک گره حسگر مثل i اجرا می‌شود، نشان می‌دهد.

غیر ایستا این مقادیر در طی زمان تغییر می‌کنند. اتوماتای یادگیر به دو گروه اتوماتاهای یادگیر با ساختار ثابت و اتوماتاهای یادگیر با ساختار متغیر تقسیم می‌شوند. در این مقاله از اتوماتاهای یادگیر با ساختار متغیر استفاده شده است که در ادامه این بخش توضیح داده می‌شود.



شکل (۱): ارتباط بین اتوماتای یادگیر و محیط

اتوماتاهای یادگیر با ساختار متغیر: اتوماتای یادگیر با ساختار متغیر توسط ۴ تایی $\{a, b, p, T\}$ نشان داده می‌شود که در آن $a = \{a_1, a_2, \dots, a_r, \mathbf{K}, a_r\}$ مجموعه عمل‌های اتوماتا، $b = \{b_1, b_2, \dots, b_m, \mathbf{K}, b_m\}$ مجموعه ورودی‌های اتوماتا، $p = \{p_1, p_2, \dots, p_m, \mathbf{K}, p_m\}$ بردار احتمال انتخاب اعمال و $p(n+1) = T[a(n), b(n), p(n)]$ الگوریتم یادگیری هستند. در این نوع از اتوماتاهای یادگیر، اگر عمل a_i در مرحله n ام انجام شود و پاسخ مطلوب از محیط دریافت نماید، احتمال $p_i(n)$ افزایش یافته و سایر احتمال‌ها کاهش می‌یابند. در هر حال، تغییرات به گونه‌ای صورت می‌گیرد تا حاصل جمع $p_i(n)$ ها همواره ثابت و مساوی یک باقی بماند. الگوریتم زیر نمونه‌ای از الگوریتم‌های یادگیر خطی است.

الف- پاسخ مطلوب

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

$$p_j(n+1) = (1 - a)p_j(n) \quad \forall j \quad j \neq i$$

ب- پاسخ نامطلوب

$$p_i(n+1) = (1 - b)p_i(n)$$

$$p_j(n+1) = \frac{b}{r-1} + (1 - b)p_j(n) \quad \forall j \quad j \neq i$$

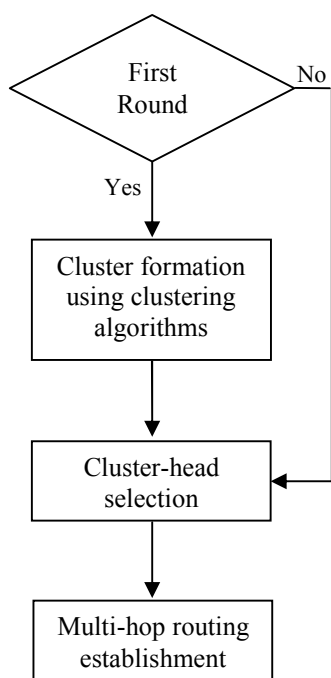
در روابط فوق، a پارامتر پاداش و b پارامتر جریمه است. با توجه به مقادیر a و b سه حالت را می‌توان در نظر گرفت. زمانیکه a و b با هم برابر باشند، الگوریتم را ${}^{11}\text{LRP}$ ، زمانیکه b از a خیلی کوچکتر باشد، الگوریتم را ${}^{12}\text{LRP}$ و زمانیکه b مساوی صفر باشد، الگوریتم را ${}^{14}\text{LRP}$ می‌نامیم. برای اطلاعات بیشتر درباره اتوماتاهای یادگیر و کاربردهای آن می‌توان به [15] مراجعه نمود.

۳- الگوریتم پیشنهادی

در الگوریتم پیشنهادی، هر گره حسگر مجهز به یک اتوماتای یادگیر با r عمل $\{a_1, a_2, \dots, a_r\}$ است که r تعداد همسایه‌های گره

برای مقایسه الگوریتم‌های خوشه‌بندی متریک‌های «انرژی باقیمانده در گرهار»، «تعداد گره‌های زنده» و «تعداد بسته‌های ارسالی» در نظر گرفته شده است. فرضیات و پارامترهای بکار برده شده در شبیه سازی‌ها بشرح زیر می‌باشند:

- شبیه سازی در مساحتی به وسعت 650×650 متر انجام می‌گیرد.
- گره‌های حسگر در طول مدت شبیه سازی ساکن فرض می‌شوند.
- مسیر یابی از هر یک از گره‌های رهبر بطرف گره جمع آوری کننده‌ی اطلاعات بصورت متمرکز و توسط گره جمع آوری کننده‌ی اطلاعات انجام می‌شود.
- گره جمع آوری کننده‌ی اطلاعات در مختصات (۳۲۵, ۳۲۵) قرار گرفته است.
- انرژی اولیه هر یک از گره‌ها 1.0J در نظر گرفته شده است.
- هزینه حس یک اتفاق برای هر یک از گره‌های حسگر 0.001J فرض شده است.
- هزینه ارسال داده 0.002Jm فرض شده است.
- هزینه دریافت داده 0.002Jm فرض شده است.
- شعاع حس هر یک از گره‌های حسگر 40m فرض شده است.
- تاخیر حس هر یک از گره‌های حسگر 0.1 ثانیه فرض شده است.
- شعاع ارسال داده‌های جمع آوری شده توسط هر رهبر به سایر رهبرها، 100m فرض شده است.



شکل (۴): فرآیند فاز برپایی شبکه حسگر

```

    No_of_neighbors = Find_out_neighbors_of(i);
    if(No_of_neighbors == 0)
      Set_as_header(i);
    else
      {
        P = Create_probabilities_vector(No_of_neighbors);
        Initialize_probabilities_vector(P);

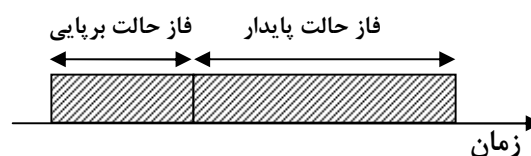
        while (TRUE)
        {
          /* Select a node as a header according to the
            action probabilty vector of learning automata */
          j = Select_header(P);
          Send_IEY_message_to(j);

          response = Wait_for_response_from(j);
          if (response == ACCEPT)
          {
            Increase(P_j); // Revard action j
            Set_as_header(j);
          }
          else
            Decrease(P_j); // Penalize action j
        }
      }
  
```

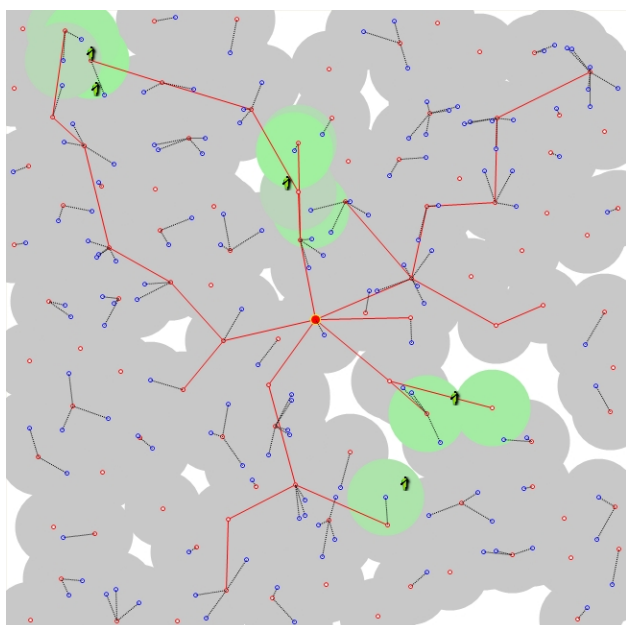
شکل (۲): شبیه کد الگوریتم LACA

۴- نتایج شبیه سازی‌ها

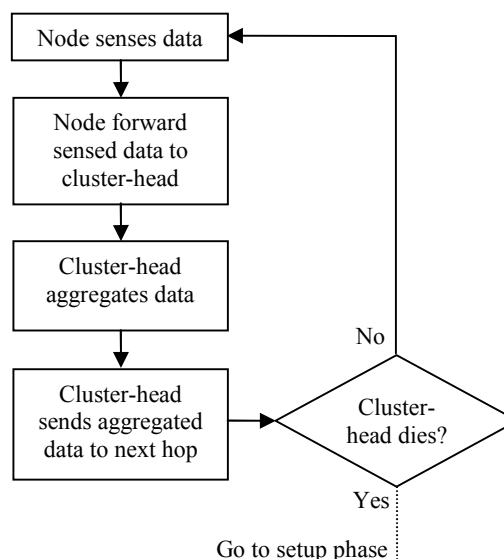
برای ارزیابی الگوریتم پیشنهادی، سیستمی با خصوصیات زیر در نظر گرفته شده است [16]. عملیات اصلی گره‌های حسگر در این سیستم به دوره‌هایی^{۱۶} تقسیم شده است (شکل ۳). هر دوره شامل فاز برپایی و فاز «حالت پایا»^{۱۷} است. در فاز برپایی بر طبق الگوریتم‌های خوشه‌بندی، شبکه به خوشه‌های مختلف تقسیم شده و مسیر یابی متمرکز بر اساس هزینه و تنها با در نظر گرفتن گره‌های رهبر بوسیله ایستگاه اصلی جمع آوری کننده‌ی اطلاعات انجام می‌گیرد (شکل ۴). در فاز ارسال داده (حالت پایا)، گره‌های حسگر، داده‌های حس شده را تولید و به گره رهبر خوشه‌ای که به آن تعلق دارند، ارسال می‌کنند. هر یک از گره‌های رهبر، داده‌های دریافت شده از گره‌های خوشه را به همراه داده‌ی خود جمع کرده و بر اساس مسیری که توسط ایستگاه اصلی جمع آوری کننده‌ی اطلاعات در فاز برپایی ایجاد شده، به طرف آن ارسال می‌نمایند. هر زمان که رهبر خوشه‌ای از کار بیفتد، «حالت پایدار» به پایان خواهد رسید و به این ترتیب سیستم به فاز برپایی برخورد گشت. سیستم این چرخه را تا زمانی که همه گره‌های حسگر در شبکه از کار بیفتند (یا مدت زمان شبیه سازی باتمام برسد)، ادامه می‌دهد (شکل ۵).



شکل (۳): دوره عملکرد شبکه حسگر



شکل (۶): نمونه ای از اجرای شبیه‌سازی توسط نرم افزار SNCS



شکل (۵): فرآیند فاز «حالت پایدار» شبکه حسگر

شکل (۷) نتایج بدست آمده برای الگوریتم‌های دسته بندی LACA، VCA، ACA و RCA را در آزمایش ۱ ($N = 200, S = 3$) نشان می‌دهد. در شکل (۷-الف) مشاهده می‌شود شبکه‌ای که با استفاده از الگوریتم پیشنهادی عمل دسته بندی را انجام داده است تقریباً از ثانیه ۲۰۰ام به بعد و با یادگیری بهتر خصوصیات محیط، کمتر از سایر شبکه‌ها انرژی مصرف کرده است و از اینرو گره‌های بیشتری در پایان شبیه سازی فعال باقی مانده‌اند (شکل ۷-ب).

شکل (۸) نتایج بدست آمده برای الگوریتم‌های دسته بندی LACA، VCA، ACA و RCA را در آزمایش ۲ ($N = 300, S = 4$) نشان می‌دهد. در شکل (۸-الف) بوضوح مشاهده می‌شود که الگوریتم خوشه‌بندی پیشنهادی تا ثانیه ۱۹۰ام بهتر از الگوریتم RCA عمل کرده و مصرف انرژی آن تقریباً نزدیک به الگوریتم‌های VCA و ACA است ولی رفته رفته و با یادگیری خصوصیات محیط، از ثانیه تقریباً ۲۲۰ام، مصرف انرژی آن بطور محسوسی کاهش پیدا کرده است. همچنین شکل (۸-ب) حاکی از فعال بودن گره‌های بیشتری در الگوریتم خوشه‌بندی پیشنهادی در مقایسه با سایر الگوریتم‌ها است.

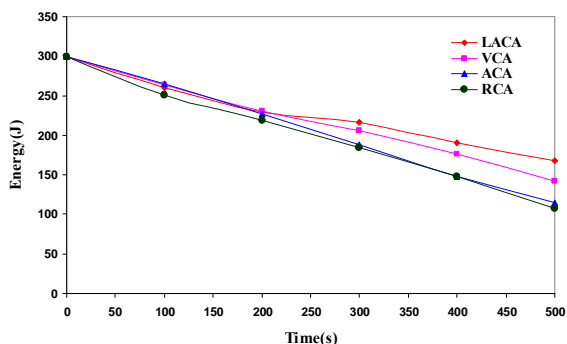
شکل (۹) نتایج بدست آمده برای الگوریتم‌های خوشه‌بندی LACA، VCA، ACA و RCA را در آزمایش ۳ ($N = 400, S = 5$) نشان می‌دهد. طبق نتایج ارائه شده در شکل (۹-الف) می‌توان گفت که الگوریتم LACA تا ثانیه ۲۴۰ انرژی زیادی در مقایسه با دو الگوریتم VCA و ACA مصرف می‌کند. این مصرف انرژی بخاطر کثرت تعداد گره‌هایی است که در همسایگی هر گره می‌باشند. اما همانطور که در شکل (۹-الف) ملاحظه می‌شود، الگوریتم پیشنهادی تقریباً از ثانیه ۳۳۰ام به بعد عملکرد خود را بهبود می‌بخشد و مصرف انرژی آن بطور چشم‌گیری در مقایسه با سایر الگوریتم‌ها کاهش می‌یابد. شکل (۹-ب) نیز نشان دهنده‌ی این واقعیت می‌باشد.

برای ارزیابی الگوریتم پیشنهادی، چهار آزمایش مختلف طبق جدول (۱) انجام گرفته شده است. در این آزمایشات، مساله تشخیص حرکت سربازان دشمن در جبهه‌ی جنگ شبیه سازی شده است [17]. از اینرو، یکی از پارامترها در آزمایشات، تعداد سربازان در حال حرکتی است که در هر لحظه در میدان جنگ حضور دارند. سرعت و شتاب هر یک از این سربازها دارای توزیع نرمال بوده و جهت حرکت آنها از شمال به شرق و یا از غرب به جنوب تنظیم شده است. جهت انجام شبیه سازی‌ها، از نرم افزاری بنام SNCS^{۱۸} که توسعه یافته نرم افزار WSNS [18] می‌باشد، استفاده شده است. شکل (۶) نمونه ای از اجرای شبیه سازی با استفاده از نرم افزار SNCS را نشان می‌دهد. در این شکل، دایره‌های سفید رنگ با حاشیه قرمز بعنوان گره‌های رهبر، دایره‌های سفید رنگ با حاشیه آبی بعنوان گره‌های خوشه، دایره قرمز رنگ با حاشیه زرد معرف گره اصلی جمع آوری کننده‌ی اطلاعات، خطوط سیاه رنگ نقطه چین معرف وابستگی هر یک از گره‌های خوشه به گره رهبر و خطوط قرمز رنگ معرف انتقال داده بین دو گره متناظر است.

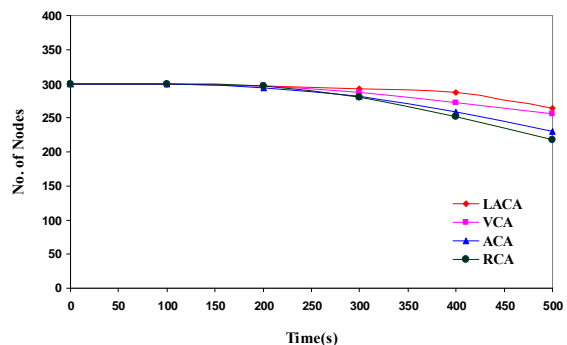
جدول (۱): مشخصات آزمایشات انجام شده

شماره آزمایش	تعداد گره‌های حسگر (N)	مدت زمان شبیه‌سازی (ثانیه)	تعداد سربازان در حال حرکت (S)
۱	۲۰۰	۵۰۰	۳
۲	۳۰۰	۵۰۰	۴
۳	۴۰۰	۵۰۰	۵
۴	۵۰۰	۱۵۰۰	۶

نتایج آزمایش‌ها در شکل‌های (۷) الی (۱۱) نشان داده شده است. نتایج ارائه شده، میانگین اجرای ۵۰ بار شبیه سازی برای هر یک از آزمایش‌ها می‌باشد. الگوریتم پیشنهادی از اتوماتای یادگیر LRP با پارامترهای پاداش و جریمه برابر با 0.1 استفاده می‌کند.

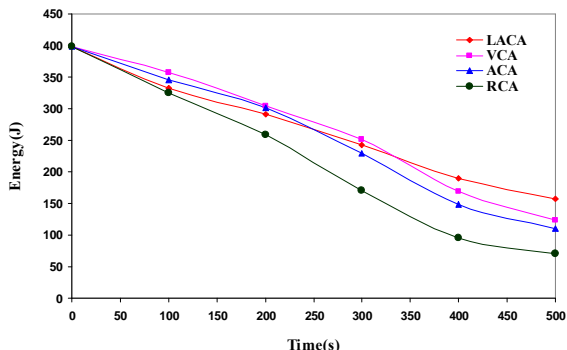


شکل (۸-الف)

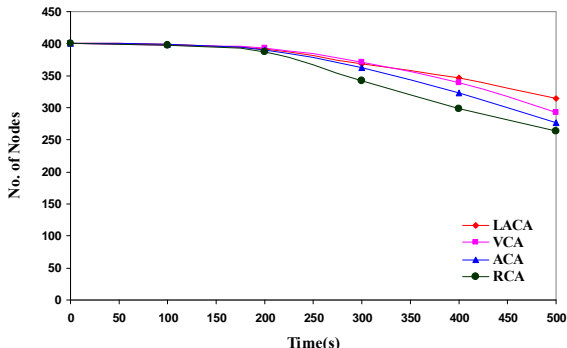


شکل (۸-ب)

شکل (۸): مقایسه الگوریتم‌های دسته بندی LACA، VCA، ACA و RCA ($N = 300, S = 4$). (الف) انرژی باقی مانده‌ی گره‌ها، (ب) تعداد گره‌های فعال



شکل (۹-الف)

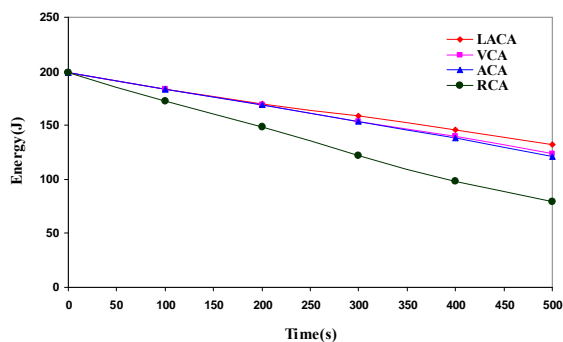


شکل (۹-ب)

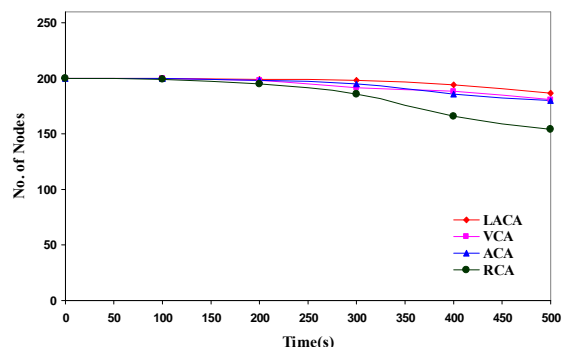
شکل (۹): مقایسه الگوریتم‌های دسته بندی LACA، VCA، ACA و RCA ($N = 400, S = 5$). (الف) انرژی باقی مانده‌ی گره‌ها، (ب) تعداد گره‌های فعال

شکل (۱۰) نتایج بدست آمده برای الگوریتم‌های خوشه‌بندی LACA، VCA، ACA و RCA را در آزمایش ۴ ($N = 500, S = 6$) نشان می‌دهد. در شکل (۱۰-الف) مشاهده می‌شود شبکه‌ای که با استفاده از الگوریتم پیشنهادی عمل دسته بندی را انجام داده است تقریباً از ثانیه ۳۰۰م به بعد و با یادگیری بهتر خصوصیات محیط کمتر از سایر شبکه‌ها انرژی مصرف کرده است و از اینرو گره‌های بیشتری (تقریباً دو برابر) در پایان شبیه سازی فعال باقی مانده‌اند (شکل ۱۰-ب). با توجه به اینکه در این آزمایش مدت زمان شبیه سازی خیلی طولانی (۱۵۰۰ ثانیه) در نظر گرفته شده است اما همچنان ملاحظه می‌شود که سطح مصرف انرژی الگوریتم پیشنهادی تا پایان شبیه‌سازی در مقایسه با سایر الگوریتم‌ها کمتر باقی مانده است.

شکل (۱۱) الگوریتم‌های خوشه‌بندی LACA، VCA، ACA و RCA را از نظر تعداد بسته‌های دریافت شده توسط گره اصلی جمع‌آوری کننده اطلاعات در چهار آزمایش انجام شده، مقایسه میکند. همانطور که مشاهده می‌شود تعداد بسته‌های دریافت شده توسط گره اصلی جمع‌آوری کننده اطلاعات در شبکه‌ای که از الگوریتم پیشنهادی جهت خوشه‌بندی استفاده می‌کند، بیشتر از شبکه‌هایی است که از سایر الگوریتم‌ها استفاده می‌کنند. دلیل این امر در توازن مصرف انرژی شبکه‌ای است که از LACA برای عمل دسته‌بندی استفاده میکند.



شکل (۷-الف)



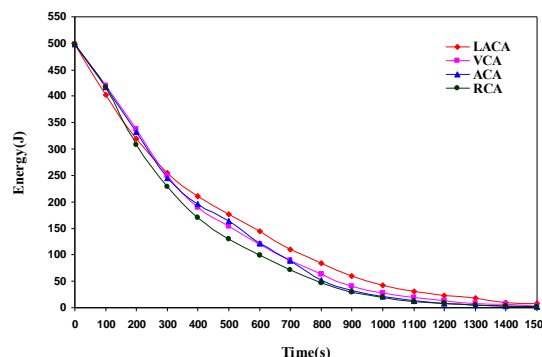
شکل (۷-ب)

شکل (۷): مقایسه الگوریتم‌های دسته بندی LACA، VCA، ACA و RCA ($N = 200, S = 3$). (الف) انرژی باقی مانده‌ی گره‌ها، (ب) تعداد گره‌های فعال

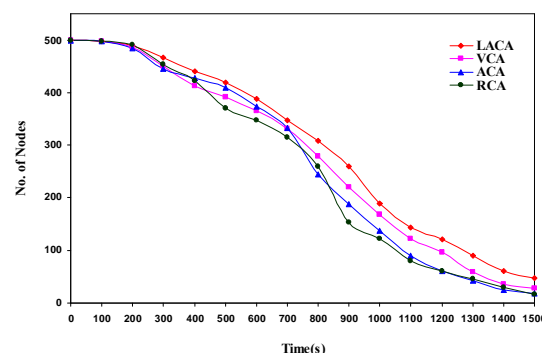
برخوردار است. همچنین نشان داده شد که الگوریتم پیشنهادی از نظر توازن مصرف انرژی در کل شبکه بهتر از سایر الگوریتم‌ها عمل کرده است. از اینرو تعداد بسته‌های ارسال شده به گره اصلی جمع آوری کننده اطلاعات در شبکه‌ای که از این الگوریتم استفاده کرده است بیشتر از سایر شبکه‌هایی است که از الگوریتم‌های پیشین استفاده کرده‌اند.

مراجع

- [1] F. Akyldiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, "Wireless Sensor Networks: A Survey", Computer Networks, No. 4, Vol. 38, pp. 393-422, 2002.
- [2] D. Estrin, L. Girod, G. Pottie, and M. Srivastava, "Instrumenting the World with Wireless Sensor Networks", Proceedings of International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001), 2001.
- [3] W. Rabiner Heinzelman, A. Chandrakasan and Hari Balakrishnan. "Energy-Efficient Communication Protocol for Wireless Microsensor Networks", Proceedings of the 33rd International Conference on System Sciences (HICSS '00), 2000.
- [4] S. Bandyopadhyay and E. Coyle, "An Energy-Efficient Hierarchical Clustering Algorithm for Wireless Sensor Networks", Proceedings of IEEE INFOCOM, 2003.
- [5] M. Demirbas and H. Ferhatosmanoglu, "Peer-to-Peer Spatial Queries in Sensor Networks", Proceeding of 3rd IEEE International Conference on Peer-to-Peer Computing (P2P '03), pp.32-39, 2003.
- [6] S. Banerjee and S. Khuller, "A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks", Proceedings of IEEE INFOCOM, 2001.
- [7] C. R. Lin and M. Gerla, "Adaptive Clustering for Mobile Wireless Networks", IEEE Journal Selected Areas Communication, 1997.
- [8] Wei-Peng Chen, Jennifer C. Hou and Lui Sha, "Dynamic Clustering for Acoustic Target Tracking in Wireless Sensor Networks", Proceedings of the 11th IEEE International Conference on Network Protocols, pp. 284-294, 2003.
- [9] S. Basagni, "Distributed Clustering Algorithm for Ad-hoc Networks", Proceedings of International Symposium on Parallel Architectures, Algorithms, and Networks (ISPAN), 1999.
- [10] M. Chatterjee, S. K. Das and D. Turgut, "WCA: A Weighted Clustering Algorithm for Mobile Ad hoc Networks", Journal of Cluster Computing (Special Issue on Mobile Ad hoc Networks), Vol. 5, No. 2, pp.193-204, 2002.
- [11] R. Krishnan and, D. Starobinski, "Efficient Clustering Algorithms for Self-organizing Wireless Sensor Networks", Ad Hoc Networks, vol. 4, pp. 36-59, 2006.
- [12] M. Qin and R. Zimmermann, "An Energy-efficient Voting-based Clustering Algorithm for Sensor Networks", Proceedings of the Sixth International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing and First ACIS International Workshop on Self-Assembling Wireless Networks (SNPD/SAWN'05), pp. 444-451, 2005.
- [13] S. Banerjee and S. Khuller, "A Clustering Scheme for Hierarchical Control in Multi-hop Wireless Networks", Proceedings of IEEE INFOCOM, 2001.

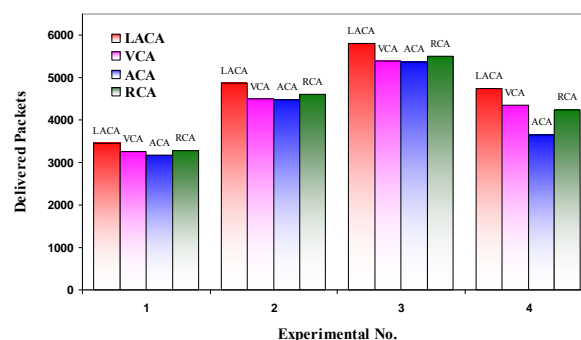


شکل (۱۰-الف)



شکل (۱۰-ب)

شکل (۱۰): مقایسه الگوریتم‌های دسته بندی LACA، VCA، ACA و RCA (N = 500, S = 6). (الف) انرژی باقی مانده ی گره‌ها، (ب) تعداد گره‌های فعال



شکل (۱۱): مقایسه الگوریتم‌های دسته بندی LACA، VCA، ACA و RCA از نظر تعداد بسته‌های دریافت شده توسط گره اصلی جمع آوری کننده اطلاعات

۵- نتیجه گیری

در این مقاله الگوریتمی مبتنی بر اتوماتاهای یادگیر جهت خوشه‌بندی گره‌ها در شبکه‌های حسگر پیشنهاد گردید. کارایی الگوریتم پیشنهادی از طریق شبیه سازی و مقایسه نتایج آن با نتایج روش‌های پیشین مورد ارزیابی قرار گرفت. نشان داده شد که شبکه‌ای که از LACA برای دسته بندی گره‌های حسگر استفاده میکند، هم از نظر میزان انرژی مصرفی و هم از نظر تعداد گره‌های فعال باقی مانده در انتهای شبیه سازی ها، از کارایی بالایی در مقایسه با روش‌های گزارش شده

- [14] D. V. Paruchuri, "Adaptive Clustering Protocol for Sensor Networks", Aerospace, 2005 IEEE Conf., pp. 1-8, 2005.
- [15] K. S. Narendra and M.A.L. Thathachar, *Learning Automata: An Introduction*, Prentice Hall, Englewood Cliffs, NJ, 1989.
- [16] P. Tillapart, T. Thumthawatworn, P. Pakdeepinit, T. Yeophantong, S. Charoenvikrom and J. Daengdey, "Method for Cluster Heads Selection in Wireless Sensor Network", Proceedings of the 2004 IEEE Aerospace Con., Big Sky, Montana, USA, 2004.
- [17] A. Arora et al, "A Line in the Sand: A Wireless Sensor Network for Target Detection, Classification, and Tracking", Computer Networks Journal, pp. 605-634, 2004.
- [18] Wireless Sensor Network Simulator:
<http://www.djstein.com/projects/WirelessSensorNetworkSimulator.html>.

زیر نویس‌ها

- ¹ Sink
- ² Clustering
- ³ Cluster Head
- ⁴ Followers
- ⁵ Distributed Clustering Algorithm
- ⁶ Weighted Clustering Algorithm
- ⁷ Rapid Clustering Algorithm
- ⁸ Voting-based Clustering Algorithm
- ⁹ Hierarchical Clustering Algorithm
- ¹⁰ Adaptive Clustering Algorithm
- ¹¹ Learning Automata-based Clustering Algorithm
- ¹² Linear Reward Penalty
- ¹³ Linear Reward Epsilon Penalty
- ¹⁴ Linear Reward Inaction
- ¹⁵ I Elect You
- ¹⁶ Rounds
- ¹⁷ Steady-state phase
- ¹⁸ Sensor Network Clustering Simulator