

GIẢI THUẬT HEURISTIC VÀ DI TRUYỀN GIẢI BÀI TOÁN ĐỊNH TUYẾN ĐA ĐIỂM TRÊN MẠNG CẢM BIẾN KHÔNG DÂY NHIỆM VỤ TUẦN HOÀN

NGUYỄN THÁI DƯƠNG¹, HUỖNH THỊ THANH BÌNH², NGÔ HỒNG SƠN³

Trường Đại học Bách Khoa Hà Nội, Việt Nam

¹*thaiduongnguyen91@gmail.com;*

²*binhht@soict.hust.edu.vn;*

³*sonnh@soict.hust.edu.vn*

Tóm tắt. Bài báo này nghiên cứu về bài toán định tuyến đa điểm (multicast) cho mạng cảm biến không dây nhiệm vụ tuần hoàn (DC-WSN). Đặc điểm của loại mạng cảm biến không dây này là các nút cảm biến hoạt động tuần hoàn theo chu kỳ và không bắt buộc phải hoạt động liên tục. Bài toán này đã được chứng minh thuộc lớp NP-khó. Chúng tôi đề xuất một giải thuật heuristic và một giải thuật di truyền để giải bài toán trên. Các giải thuật đề xuất được thử nghiệm trên bốn dạng đồ thị mạng cảm biến và được so sánh kết quả với giải thuật TCS là giải thuật tốt nhất hiện nay. Kết quả thử nghiệm cho thấy các giải thuật đề xuất đưa ra lời giải tốt hơn giải thuật TCS về mặt tối ưu năng lượng.

Từ khóa. Mạng cảm biến không dây, multicast, tối thiểu năng lượng, giải thuật heuristic, giải thuật di truyền.

Abstract. We study the Minimum-Energy Multicasting problem in Duty-Cycled Wireless Sensor Networks (DC-WSN). In DC-WSN, nodes can switch between active and dormant states to save energy. This problem has proved to be NP-hard. This paper proposes a heuristic algorithm and a genetic algorithm for solving this problem. We compare the proposed algorithms with TCS - the best known algorithm - by means of simulation on four typical WSN topologies. Experimental results show that our algorithms significantly outperform TCS in terms of minimizing the energy cost.

Keywords. Wireless sensor networks, multicast, minimum-energy, heuristic, genetic algorithm.

1. GIỚI THIỆU

Hiện nay, mạng cảm biến không dây đang được sử dụng rộng rãi trong theo dõi môi trường, giám sát đối tượng, cảnh báo nguy cơ cháy rừng,... Mạng cảm biến không dây gồm một tập các nút cảm biến nằm phân tán trên một khu vực, hợp tác với nhau qua mạng để thực hiện nhiệm vụ. Các nút cảm biến thường nhỏ với nguồn năng lượng giới hạn (thường dùng pin), vì vậy chúng khó hoạt động liên tục trong thời gian dài. Do đó, vấn đề tiết kiệm năng lượng hoạt động của mạng cảm biến rất được quan tâm trong những năm gần đây.

Một phương pháp tiết kiệm năng lượng cho mạng cảm biến không dây là cho các nút hoạt động tuần hoàn qua các chu kỳ. Trong từng chu kỳ, mỗi nút có thể luân chuyển giữa hai trạng thái hoạt động và tạm nghỉ. Lịch luân chuyển trạng thái là độc lập đối với từng nút. Từ đây, ta sẽ gọi tên mô hình mạng này là mô hình mạng cảm biến không dây nhiệm vụ tuần hoàn

(DC-WSN: Duty-Cycled Wireless Sensor Networks). Nhờ việc luân chuyển trạng thái hoạt động mà mô hình DC-WSN đã được chứng minh là hiệu quả về mặt năng lượng và đang được áp dụng rất nhiều trong thực tế [10, 11, 12, 13].

Truyền dữ liệu đa điểm (multicast) là quá trình truyền dữ liệu từ một nút nguồn đến một tập các nút đích. Multicast được thực hiện thường xuyên trong hoạt động của mạng, do đó thiết kế một giao thức multicast hiệu quả về mặt năng lượng cho mạng cảm biến không dây là rất cần thiết. Bài toán này (MEM: Minimum-Energy Multicasting) đã được chứng minh thuộc lớp NP-khó và thường được giải quyết bằng thuật toán xấp xỉ đề xuất trong [14, 15, 16, 17]. Tuy nhiên các phương pháp trên chỉ áp dụng cho các mạng mà các nút luôn hoạt động. Các nghiên cứu về bài toán MEM trong DC-WSN được đưa ra trong [3, 6]. Các tác giả trong [6] đề xuất các thuật toán tối ưu giải quyết bài toán MEM trong DC-WSN thu hẹp với các khe thời gian hoạt động của mỗi nút là liên tục, tuy nhiên độ phức tạp của các thuật toán này đều là hàm mũ theo số lượng các nút đích. Han và các tác giả khác [3] đã đề xuất giải thuật TCS để giải quyết bài toán MEM trên mạng DC-WSN tổng quát. Giải thuật này xây dựng một đồ thị mở rộng dựa vào đồ thị mạng ban đầu và lịch hoạt động của các nút, sau đó tìm cây Steiner nhỏ nhất trên đồ thị mở rộng và cuối cùng ánh xạ cây Steiner tìm được thành lời giải cho bài toán. Theo các tác giả, hiện tại TCS là giải thuật xấp xỉ tốt nhất cho bài toán MEM trong DC-WSN. Tuy nhiên, chất lượng lời giải của giải thuật này cũng phụ thuộc nhiều vào độ tốt của cây Steiner tìm được trên đồ thị mở rộng.

Để khắc phục các nhược điểm trên, chúng tôi đề xuất một giải thuật heuristic (HMEM) và một giải thuật di truyền (GAMEM) nhằm mang lại lời giải có mức năng lượng tiêu thụ tốt hơn cho bài toán MEM trên DC-WSN so với các phương pháp trước. Các giải thuật đề xuất được thử nghiệm trên bốn bộ dữ liệu tương tự trong [3] và được so sánh với giải thuật TCS. Kết quả thử nghiệm cho thấy các giải thuật đề xuất đều mang lại lời giải tốt hơn so với TCS xét về mặt tối ưu năng lượng. Giải thuật HMEM có thời gian tính toán ngắn nhất khi so với TCS và GAMEM.

Phần tiếp theo của bài báo được tổ chức như sau. Phần 2 trình bày bài toán MEM trên mô hình DC-WSN. Phần 3 và 4 lần lượt đề xuất giải thuật heuristic HMEM và giải thuật di truyền GAMEM. Phần 5 trình bày về các bộ dữ liệu thử nghiệm và kết quả thử nghiệm của các giải thuật. Kết luận về bài báo và các hướng phát triển sẽ được trình bày ở phần 6.

2. MÔ HÌNH BÀI TOÁN

Phần này tóm tắt mô hình bài toán MEM trong mạng DC-WSN đã được đề cập trong [3].

2.1. Mô hình mạng cảm biến không dây nhiệm vụ tuần hoàn

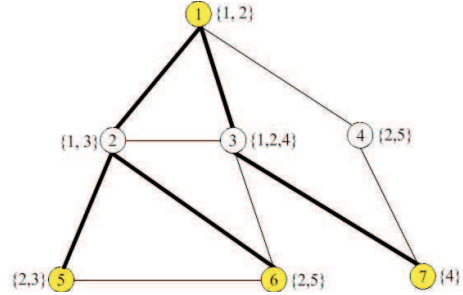
Một mạng cảm biến không dây được biểu diễn bởi một đồ thị vô hướng, không trọng số $G = (V, E)$, trong đó V là tập các nút, E là tập liên kết giữa các nút. Các nút trong V phân bố trên mặt phẳng và tồn tại liên kết giữa hai nút nếu chúng nằm trong phạm vi truyền tin của nhau. Năng lượng ban đầu của các nút được giả thiết là như nhau. Trong mạng DC-WSN, các nút đều hoạt động tuần hoàn qua các chu kỳ, mỗi chu kỳ được chia thành K khe thời gian như nhau. Để tiết kiệm năng lượng, trong từng chu kỳ mỗi nút $u \in V$ chỉ hoạt động trong các khe thời gian thuộc tập $\Gamma(u) \subset \{1, 2, \dots, K\}$ ($\Gamma(u) \neq \emptyset, \forall u \in V$). Giả thiết mỗi nút u đều có thể thức giấc để truyền tin tại bất cứ khe thời gian nào nhưng chỉ nhận được tin trong các

khe thời gian thuộc $\Gamma(u)$.

2.2. Multicast trên mạng cảm biến không dây nhiệm vụ tuần hoàn

Cho một tập các nút terminal $M \subset V$, trong quá trình thực hiện multicast các gói dữ liệu được gửi từ nút nguồn $s \in M$ đến tất cả các nút thuộc $M \setminus \{s\}$. Với T là một cây con bất kì của đồ thị $G = (V, E)$, ta kí hiệu $N(T)$ và $E(T)$ lần lượt là tập các đỉnh và tập các cạnh của T . Trong trường hợp T là cây có gốc, có thêm các kí hiệu:

- $nl(T)$ là tập các nút không phải là lá (non-leaf) của cây T .
- $child(u, T)$ là tập các nút con của nút u trên cây T .



Cây T được gọi là cây multicast của G nếu nó là cây con gốc s của G , đồng thời mọi nút terminal trong M đều thuộc T . Hình 1 mô tả một đồ thị mạng DC-WSN với tập các nút terminal $M = \{1, 5, 6, 7\}$ được bôi vàng trong đó đỉnh nguồn $s = 1$. Tập các khe thời gian hoạt động được ghi bên cạnh mỗi nút u . Một cây multicast với các cạnh được bôi đậm được chỉ ra trong Hình 1.

Hình 1: Mạng cảm biến không dây nhiệm vụ tuần hoàn $G = (V, E)$ và một cây multicast.

Định nghĩa 2.1 (Hitting set [3, 7]) Cho họ tập $C = \{A_1, A_2, \dots, A_n\}$, tập $F \subset A_1 \cup A_2 \cup \dots \cup A_n$ được gọi là *hitting set* của C nếu trong F chứa ít nhất một phần tử của mỗi tập con có trong C , nghĩa là $F \cap A_i \neq \emptyset, \forall i = \overline{1, n}$. Hitting set có số lượng phần tử ít nhất được gọi là *minimum hitting set* của C và được ký hiệu là $MHS(C)$ ¹.

Định nghĩa 2.2 (Lịch truyền khả thi [3]) Với một cây multicast T của G , một hàm $B : nl(T) \rightarrow 2^{\{1,2,\dots,K\}}$ được gọi là lịch truyền khả thi (gọi tắt là lịch truyền) của T nếu với $\forall u \in nl(T)$ thì $B(u)$ là hitting set của họ tập $\{\Gamma(v) | v \in child(u, T)\}$.

Gọi e_s và e_r lần lượt là năng lượng truyền và năng lượng nhận một gói tin của mỗi nút thuộc V ($e_s, e_r > 0$). Năng lượng tiêu tốn trong một phiên multicast gồm hai thành phần: năng lượng truyền tin và năng lượng nhận tin. Từ định nghĩa 2, tổng năng lượng truyền tin của một phiên multicast theo một lịch truyền B trên một cây multicast T bằng: $\sum_{\forall u \in nl(T)} |B(u)| \cdot e_s$.

Ngoài ra, mọi nút trên cây T ngoại trừ nút nguồn s đều được nhận tin, nên năng lượng truyền tin trong mỗi phiên multicast chỉ phụ thuộc vào số lượng nút trên cây và bằng: $(|N(T)| - 1) \cdot e_r$.

Bài toán định tuyến multicast trên mạng cảm biến không dây nhiệm vụ tuần hoàn (MEM DC-WSN) [3]:

Cho mạng cảm biến không dây nhiệm vụ tuần hoàn $G = (V, E)$, tập các nút terminal $M \subset V$ và nút nguồn Bài toán yêu cầu tìm một bộ gồm một cây multicast T_{opt} của G và một lịch truyền B_{opt} trên cây này sao cho tổng năng lượng tiêu tốn trong mỗi phiên multicast: $\Pi(T_{opt}, B_{opt}) = \sum_{\forall u \in nl(T_{opt})} |B_{opt}(u)| \cdot e_s + (|N(T_{opt})| - 1) \cdot e_r$ là nhỏ nhất.

¹Quy ước: với $C = \emptyset$ thì $MHS(C) = \emptyset$.

Đối với cây multicast T có các cạnh bôi đậm trong Hình 1, có thể lấy ví dụ một lịch truyền B_1 cho cây: $B_1(1) = \{2, 3\}$, $B_1(2) = \{3, 5\}$, $B_1(3) = \{4\}$. Với $e_s = 100$, $e_r = 15$, nếu sử dụng lịch truyền B_1 , tổng năng lượng tiêu tốn trong một phiên multicast sẽ là: $5.100 + 5.15 = 575$. Cũng với cây T này, nếu áp dụng lịch truyền B_2 : $B_2(1) = \{1\}$, $B_2(2) = \{2\}$, $B_2(3) = \{4\}$ thì tổng năng lượng tiêu tốn trong mỗi phiên multicast là: $3.100 + 5.15 = 375$. Cây multicast T và lịch truyền B_2 này là một lời giải tối ưu cho mỗi phiên multicast từ nút 1 đến các nút terminal khác.

3. GIẢI THUẬT HEURISTIC GIẢI BÀI TOÁN MEM DC-WSN

Trong phần này chúng tôi đề xuất giải thuật heuristic HMEM nhằm giải quyết bài toán MEM DC-WSN. Bên cạnh đó, chúng tôi đề xuất một phương pháp tìm kiếm địa phương giúp nâng cao chất lượng lời giải cho bài toán MEM DC-WSN. Phương pháp này được áp dụng sau khi giải thuật HMEM đưa ra lời giải cho bài toán MEM DC-WSN.

3.1. Giải thuật heuristic HMEM

Giải thuật HMEM gồm hai bước: Bước 1, tìm một cây multicast T có gốc s trên đồ thị G . Bước 2, tìm một lịch truyền khả thi trên cây tìm được ở bước trước. Để thực hiện bước 1, giải thuật HMEM xây dựng từng phần cây T bằng cách lần lượt tìm đường đi ngắn nhất từ các nút terminal về nút nguồn s trên đồ thị G' có hướng và có trọng số phụ thuộc vào thành phần cây T hiện tại. Trong bước 2, giải thuật HMEM tìm Minimum Hitting Set (MHS) để thu được lịch truyền khả thi. Chi tiết về giải thuật (Algorithm 1) được trình bày sau đây:

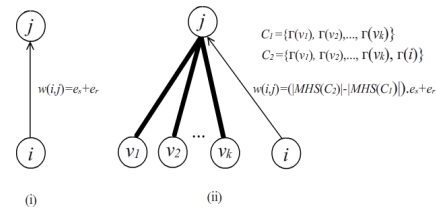
Bước 1 (dòng 1-9): Dòng 1 khởi tạo cây

T ban đầu chỉ gồm đỉnh nguồn s . Dòng 2-4 thực hiện tìm đường đi ngắn nhất từ mỗi đỉnh terminal $u \in M \setminus \{s\}$ đến thành phần cây T hiện tại, quá trình tìm đường đi ngắn nhất này được thực hiện trên đồ thị có hướng, có trọng số $G' = (V', E')$, trong đó $V' = V$ và $E' = E \setminus E(T)$. Trọng số mỗi cạnh $(i, j) \in E'$ có ý nghĩa là phần năng lượng tiêu tốn thêm khi bổ sung cạnh (j, i) vào thành phần cây multicast hiện tại (vì quá trình thực hiện tìm đường đi ngắn nhất được xuất phát từ u đến s , nên trọng số cạnh được định hướng theo chiều ngược lại). Hình 2 mô tả cách tính trọng số cạnh (i, j) của G' . Cụ thể:

(i) Với $j \notin N(T)$, việc bổ sung cạnh (j, i) sẽ tăng thêm năng lượng truyền tin từ nút j và năng lượng nhận tin của nút i nên trọng số cạnh $w(i, j) = e_s + e_r$.

(ii) Với $j \in N(T)$, việc bổ sung cạnh (j, i) sẽ tăng thêm năng lượng nhận tin của nút i , năng lượng truyền tin có tăng thêm hay không phụ thuộc vào việc: nếu bổ sung thêm i làm con của j thì số lượt truyền tin của j (chính là lực lượng của nó tăng hay không). Gọi $C_1 = \{\Gamma(v) | v \in \text{child}(j, T)\}$; $C_2 = \{\Gamma(v) | v \in \text{child}(j, T)\} \cup \{\Gamma(i)\}$. Trọng số cạnh $w(i, j)$ trong trường hợp này được gán bằng: $(|MHS(C_2)| - |MHS(C_1)|) \cdot e_s + e_r$.

Dòng 5-8 bổ sung các đỉnh và cạnh thuộc đường đi ngắn nhất tìm được vào cây hiện tại.



Hình 2: Trọng số cạnh của đồ thị $G' = (V', E')$.

(i) Trường hợp j không thuộc cây hiện tại. (ii) Trường hợp j thuộc cây hiện tại. Các đỉnh v_1, v_2, \dots, v_k là các nút con của j trong cây hiện tại

Bước 2 (dòng 10-12): Đưa ra lịch truyền cho cây multicast T bằng cách tìm minimum hitting set. Tìm minimum hitting set là bài toán thuộc lớp NP-khó và đã được chứng minh là tương đương với bài toán phủ tập (set cover problem) [5, 7, 18], nên chúng tôi sử dụng thuật toán tham lam được trình bày trong [5] để thực hiện.

Algorithm 1: $HMEM(G = (V, E), \Gamma(\cdot)M, s)$

Input: Đồ thị $G = (V, E)$; tập các đỉnh terminal M và nút nguồn s
Tập các khe thời gian hoạt động $\Gamma(\cdot)$ của các nút thuộc V

Output: Cây multicast T và lịch truyền B

begin

1. $T \leftarrow (\{s\}, \emptyset)$
 2. for each $u \in M \setminus \{s\}$
 3. Xây dựng đồ thị có hướng $G' = (V, E \setminus E(T))$, trọng số các cạnh của đồ thị G' đã được chỉ ra trong hai trường hợp (i) và (ii) ở trên
 4. Tìm P_u là đường đi ngắn nhất từ đỉnh u đến một đỉnh $v \in N(T)$
 5. for each $(i, j) \in P_u$
 6. $E(T) \leftarrow E(T) \cup \{(j, i)\}$
 7. $N(T) \leftarrow N(T) \cup \{j\} \cup \{i\}$
 8. end for
 9. end for
 10. for each $u \in nl(T)$
 11. $B(u) \leftarrow MHS(\{\Gamma(v) | v \in child(u, T)\})$
 12. end for
 13. return (T, B)
- end

Độ phức tạp của thuật toán HMEM

Trong giải thuật HMEM, việc tìm đường đi ngắn nhất từ mỗi đỉnh terminal u đến cây hiện tại có thể thực hiện bằng giải thuật Dijkstra với cấu trúc Fibonacci heap [8], thao tác này có độ phức tạp $O(|V| \cdot \log |V| + |E|)$. Vì vậy độ phức tạp của quá trình tìm đường đi cho tất cả các đỉnh terminal là $O(|M| \cdot (|V| \cdot \log |V| + |E|))$. Nếu coi độ dài chu kỳ làm việc K của mỗi nút là hằng số thì thao tác xây dựng đồ thị G' ở mỗi bước lặp không vượt quá $O(|V| + |E|)$, thao tác tìm lịch truyền B cho các nút không vượt quá $O(|V|)$. Tóm lại độ phức tạp của thuật toán HMEM là $O(|M| \cdot (|V| \cdot \log |V| + |E|))$.

3.2. Nâng cao chất lượng lời giải bằng phương pháp tìm kiếm địa phương

Chúng tôi đề xuất phương pháp tìm kiếm địa phương để nâng cao chất lượng lời giải cho HMEM. Từ một cây lời giải, ta sẽ tiến hành tối ưu cục bộ cho các cây con của nó. Nếu tìm được cây con mới có lịch truyền tốt hơn lịch truyền của cây con ban đầu thì nó sẽ được đưa vào cây lời giải hiện tại. Cụ thể, giả sử (T, B) là một bộ gồm cây lời giải và lịch truyền. Lần lượt xét các đỉnh i của cây T , gọi T_i là cây con gốc i của T (T_i bao gồm tất cả các đỉnh và

cạnh thuộc các đường đi trên T xuất phát từ i và đi xuống các nút lá); B_i là lịch truyền của các nút (không phải là lá) thuộc cây T_i ; M_i là tập những đỉnh terminal có trên cây T_i . Ta áp dụng chính giải thuật HMEM để tìm một cây con \bar{T}_i và một lịch truyền \bar{B}_i mới cho bài toán MEM DC-WSN từ nút nguồn i đến các đỉnh terminal thuộc M_i . Nếu năng lượng multicast lịch truyền \bar{B}_i nhỏ hơn theo lịch truyền B_i thì cây T_i và lịch truyền B_i sẽ được thay thế bằng cây \bar{T}_i và lịch truyền \bar{B}_i .

Độ phức tạp của thuật toán tìm kiếm địa phương

Giải thuật tìm kiếm địa phương sử dụng thuật toán HMEM để tìm cây lời giải và lịch truyền mới cho các cây con gốc i . Độ phức tạp của thuật toán HMEM đã được phân tích ở trên là $O(|M|. (|V|. \log |V| + |E|))$. Số lượng đỉnh của cây T không vượt quá $|V|$, nên quá trình tìm kiếm địa phương được thực hiện trên $O(|V|)$ cây con. Vì vậy độ phức tạp của thuật toán tìm kiếm địa phương đề xuất là $O(|V|. |M|. (|V|. \log |V| + |E|))$.

4. GIẢI THUẬT DI TRUYỀN GIẢI BÀI TOÁN MEM DC-WSN

Trong phần này chúng tôi đề xuất giải thuật di truyền GAMEM nhằm mang lại lời giải tốt hơn cho bài toán MEM DC-WSN. Giải thuật heuristic HMEM trình bày trong phần 3 sẽ được sử dụng trong quá trình xây dựng các phép toán lai ghép và đột biến cho GAMEM.

4.1. Mã hóa nhiễm sắc thể

Mỗi nhiễm sắc thể x biểu diễn lời giải cho bài toán được xác định bởi cặp $(x.T, x.B)$. Trong đó $x.T = \{(u_1, v_1), (u_2, v_2), \dots, (u_n, v_n)\}$, $((u_i, v_i) \in E(T), \forall i = \overline{1, n})$ là tập các cạnh của cây multicast T ; $x.B = \{(a_1, b_1), (a_2, b_2), \dots, (a_m, b_m)\}$ biểu diễn cho lịch truyền B của cây T , trong đó mỗi $(a_i, b_i) \in x.B$ chỉ ra rằng nút a_i của cây T sẽ truyền tin trong khe thời gian b_i .

4.2. Hàm thích nghi

Độ thích nghi của mỗi nhiễm sắc thể phụ thuộc vào độ tốt của lời giải mà nó biểu diễn. Với nhiễm sắc thể x biểu diễn cho cây multicast T và lịch truyền B , năng lượng tiêu thụ cho mỗi phiên multicast đã được định nghĩa là $\Pi(T, B)$. Khi đó ta định nghĩa hàm thích nghi của nhiễm sắc thể x là: $F(x) = \frac{1}{\Pi(T, B)^2}$. Cây và lịch truyền được biểu diễn bởi nhiễm sắc thể có độ thích nghi cao nhất được chọn làm lời giải cuối cùng cho bài toán.

4.3. Khởi tạo quần thể

Nhằm đảm bảo sự đa dạng, khi khởi tạo quần thể ban đầu mỗi nhiễm sắc thể sẽ biểu diễn cho một lời giải “ngẫu nhiên”. Chúng chỉ cần thỏa mãn các điều kiện ràng buộc của bài toán (theo định nghĩa cây multicast và lịch truyền khả thi) mà chưa cần xét đến việc tối ưu năng lượng. Việc khởi tạo được thực hiện như trong Algorithm 2. Cây multicast T của nhiễm sắc thể được tạo ra bằng cách gán trọng số ngẫu nhiên cho các cạnh của đồ thị (dòng 3-6) rồi sau đó tìm cây đường đi ngắn nhất từ s đến các đỉnh terminal (dòng 7, 8). Lịch truyền B trên cây T được xây dựng bằng cách với mỗi nút chuyển tiếp u của T , lần lượt xét các nút con v của

u để lấy ngẫu nhiên ra một khe thời gian thuộc $\Gamma(v)$ (dòng 9-15). Lịch truyền $B(u)$ của nút u chính là tập các khe thời gian đã được lấy ra ngẫu nhiên. Việc khởi tạo nhiễm sắc thể bằng thủ tục INITIAL-CHROMO có độ phức tạp $O(|V|. \log |V| + |E|)$.

Algorithm 2: INITIAL-CHROMO($G = (V, E), \Gamma(\cdot), M, s$)

Input: Đồ thị $G = (V, E)$; tập các đỉnh terminal M và nút nguồn s
Tập các khe thời gian hoạt động $\Gamma(\cdot)$ của các nút thuộc V

Output: Nhiễm sắc thể x biểu diễn một lời giải của bài toán

Begin

1. $V' \leftarrow V$
2. $E' \leftarrow E$
3. for each $(u, v) \in E'$
4. $w(u, v) \leftarrow \text{random_weight}()$
5. end for
6. $G' = (V', E')$
7. Tìm đường đi ngắn nhất từ đỉnh s đến các đỉnh terminal trên đồ thị G'
8. T là cây đường đi ngắn nhất gốc s tìm được ở bước 7
9. for each $u \in nl(T)$
10. $B(u) \leftarrow \emptyset$
11. for each $v \in \text{child}(u, T)$
12. $b \leftarrow$ Một khe thời gian ngẫu nhiên trong tập $\Gamma(v)$
13. $B(u) \leftarrow B(u) \cup \{ b \}$
14. end for
15. end for
16. Cho $x.T$ biểu diễn tập các cạnh của cây T
17. Cho $x.B$ biểu diễn lịch truyền B tìm được ở dòng 9-15
18. return $x = (x.T, x.B)$

End

4.4. Lai ghép

Phép toán lai ghép giữa hai nhiễm sắc thể x và y được thực hiện như sau (Algorithm 3): Kết hợp hai cây multicast được biểu diễn bởi x và y để tạo thành đồ thị $G' = (V', E')$ (dòng 1-5), đối với mỗi đỉnh v của G' ta lấy các khe thời gian mà tại đó nó có thể nhận được gói tin từ nút cha để đưa vào tập các khe thời gian hoạt động mới $\Gamma(v)$ (dòng 6-10). Sau đó ta tiến hành tìm lời giải mới trên đồ thị G' . Bài báo đề xuất hai kiểu lai ghép: “*lai ghép ngẫu nhiên*” và “*lai ghép heuristic*”. Lai ghép ngẫu nhiên sử dụng chính thủ tục khởi tạo nhiễm sắc thể nêu trên để tạo ra nhiễm sắc thể con z_1 (dòng 11). Lai ghép heuristic sử dụng chính thuật toán heuristic HMEM để tạo ra nhiễm sắc thể con z_2 (dòng 12, 13). Các cây được biểu diễn bởi các nhiễm sắc thể con z_1 và z_2 đều là cây multicast gốc s của đồ thị G' , nên cũng đều là cây multicast của G . Độ phức tạp của phép toán lai ghép là $O(|M|.|V|. \log |V|)$. Phép toán này được áp dụng với xác suất P_c . Hai nhiễm sắc thể con z_1 và z_2 sẽ thay thế cha mẹ chúng trong các thế hệ tiếp theo.

Algorithm 3: Crossover (x, y)

Input: Hai nhiễm sắc thể cha mẹ x, y

Output: Hai nhiễm sắc thể con z_1, z_2

begin

1. x biểu diễn cây multicast T_1 và lịch truyền B_1
2. y biểu diễn cây multicast T_2 và lịch truyền B_2

```

3.  $V' \leftarrow N(T_1) \cup N(T_2)$ 
4.  $E' \leftarrow E(T_1) \cup E(T_2)$ 
5.  $G' = (V', E')$ 
6. for each  $v \in V' \setminus \{s\}$ 
7.    $u_1$  là nút cha của  $v$  trên cây  $T_1$ 
8.    $u_2$  là nút cha của  $v$  trên cây  $T_2$ 
9.    $\Gamma'(v) \leftarrow \Gamma(v) \cap (B_1(u_1) \cup B_2(u_2))$ 
10. end for
11.  $z_1 \leftarrow \text{INITIAL-CHROMO}(G' = (V', E'), \Gamma', M, s)$ 
12.  $(T', B') \leftarrow \text{HMEM}(G' = (V', E'), \Gamma', M, s)$ 
13. Cho  $z_2$  biểu diễn cây multicast  $T'$  và lịch truyền  $B'$ 
    tìm được ở bước 12
14. return  $z_1, z_2$ 
end

```

4.5. Đột biến

Phép toán đột biến được áp dụng với xác suất P_m . Nhiễm sắc thể x được đột biến bằng cách thêm ngẫu nhiên một số đỉnh và các cạnh kề của các đỉnh đó vào cây lời giải được biểu diễn bởi x (Algorithm 4, dòng 9-16), sau đó tiến hành tìm lời giải mới trên đồ thị thu được bằng chính giải thuật HMEM (dòng 17). Xác suất thêm đỉnh và các cạnh kề phụ thuộc vào tham số P_s . Độ phức tạp của phép toán đột biến là $O(|M| \cdot (|V| \cdot \log|V| + |E|))$.

Algorithm 4: Mutation(x, P_s)

```

Input:  Nhiễm sắc thể  $x$  cần được đột biến
Tham số  $P_s$  là xác suất để thêm một đỉnh và các cạnh kề
Output: Nhiễm sắc thể  $x$  sau đột biến
begin
1.  $x$  biểu diễn cây multicast  $T$  và lịch truyền  $B$ 
2.  $V' \leftarrow N(T)$ 
3.  $E' \leftarrow E(T)$ 
4.  $G' = (V', E')$ 
5. for each  $v \in V' \setminus \{s\}$ 
6.    $u$  là nút cha của  $v$  trên cây  $T$ 
7.    $\Gamma'(v) \leftarrow \Gamma(v) \cap B(u)$ 
8. end for
9. for each  $v \in V$ 
10.   $r \leftarrow \text{random\_number}(0, 1)$ 
11.  if ( $r \leq P_s$ )
12.     $V' \leftarrow V' \cup \{v\}$ 
13.     $E' \leftarrow E' \cup \{\text{các cạnh kề của đỉnh } v \text{ trên } G\}$ 
14.     $\Gamma'(v) \leftarrow \Gamma(v)$ 
15.  end if
16. end for
17.  $(T', B') \leftarrow \text{HMEM}(G' = (V', E'), \Gamma', M, s)$ 
18. Cho  $x$  biểu diễn cây multicast  $T'$  và lịch truyền  $B'$ 
19. return  $x$ 
end

```


5. KẾT QUẢ THỬ NGHIỆM VÀ NHẬN XÉT

Giải thuật heuristic HMEM (kết hợp với tìm kiếm địa phương) và giải thuật di truyền GAMEM được thử nghiệm trên bốn bộ dữ liệu và được so sánh với giải thuật TCS [3]. Trong quá trình cài đặt thử nghiệm, các tác giả bài viết [3] sử dụng giải thuật trong [4] để tìm cây Steiner nên TCS có độ phức tạp $O(|M| \cdot |V|^2)$. Các chương trình được cài đặt trên C++ (GCC 4.4.1) và thực hiện trên máy PC CPU Core i3 3.06Hz, 2GB RAM.

5.1. Xây dựng các bộ dữ liệu

Các bộ dữ liệu thử nghiệm được sinh theo phương pháp có trong bài viết [3] như sau. Đồ thị mạng cảm biến gồm N nút, vị trí được sinh ngẫu nhiên trong mặt phẳng $1000 \text{ m} \times 1000 \text{ m}$. Phạm vi truyền tin của mỗi nút có bán kính 300 m. Thông thường năng lượng truyền tin lớn hơn nhiều so với năng lượng nhận tin, nên không mất tính tổng quát ta lấy $e_s = 100$ và $e_r = 15$ (đơn vị năng lượng). Chu kỳ của mỗi nút có $K = 20$ khe thời gian, trong đó mỗi nút sẽ chọn ngẫu nhiên một vài khe thời gian hoạt động. Tỷ lệ số khe thời gian hoạt động và số các nút terminal thay đổi theo từng bộ dữ liệu.

Bộ dữ liệu 1, 2, 3 với lần lượt $|V| = 100, 200, 300$ nút. Số lượng khe thời gian hoạt động trong một chu kỳ làm việc của mỗi nút trên ba bộ dữ liệu này là cố định, chiếm 25% trên tổng số $K = 20$ khe thời gian. Số lượng các nút terminal trên tổng số các nút thay đổi trong khoảng 5-100% (với bước nhảy 5%), ứng với 20 test trên mỗi bộ dữ liệu. Ứng với mỗi test dữ liệu, sau khi xác định được số lượng (dựa theo tỷ lệ phần trăm), các nút terminal sẽ được lựa chọn ngẫu nhiên từ tập đỉnh V . Bộ dữ liệu 4 với $|V| = 200$ nút. Số lượng các nút terminal là cố định, chiếm 50% tổng số nút toàn mạng. Số các khe thời gian hoạt động trên tổng số $K = 20$ khe thời gian trong mỗi chu kỳ làm việc của mỗi nút thay đổi trong khoảng 5-60% (với bước nhảy 5%), ứng với 12 test dữ liệu.

5.2. Các tham số và cấu hình thử nghiệm

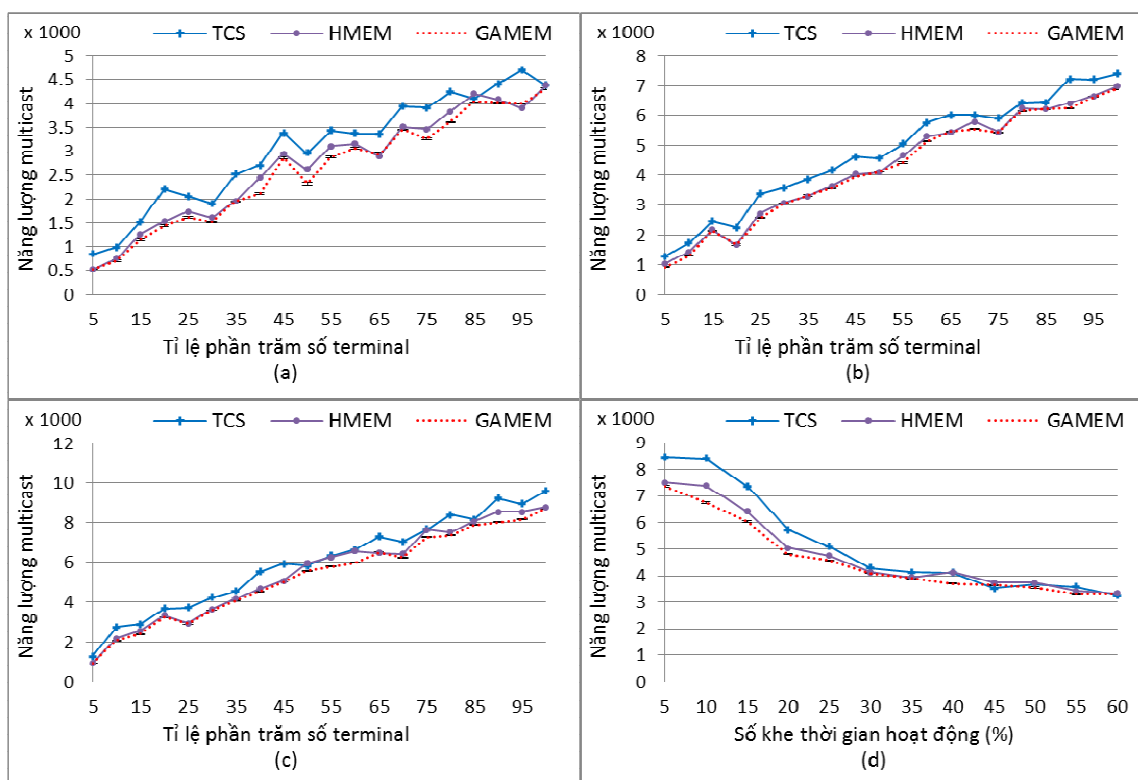
Giải thuật heuristic HMEM được kết hợp với phương pháp tìm kiếm địa phương đã nêu để đưa ra lời giải cho bài toán. Giải thuật di truyền GAMEM sử dụng 200 thế hệ quần thể, mỗi quần thể bao gồm 200 cá thể và sử dụng các tham số xác suất: $P_c = 0.2$; $P_m = 0.05$; $P_s = 0.3$. Trong mỗi test thử nghiệm, giải thuật di truyền do có tính chất ngẫu nhiên nên được tiến hành chạy 50 lần và lấy kết quả trung bình.

5.3. Kết quả thử nghiệm

Các giải thuật được đánh giá theo các tiêu chuẩn: tổng năng lượng tiêu tốn trong một phiên multicast, độ trễ truyền tin và thời gian chạy chương trình². Hình 3(a)-(d) cho biết tổng năng lượng tiêu tốn trong mỗi phiên multicast của các giải thuật TCS, HMEM và tổng năng lượng tiêu tốn tính trung bình trong 50 lượt chạy của giải thuật GAMEM, cùng với khoảng tin cậy 95%. Các giá trị trong hình (theo trục tung) được tính dựa theo công thức tính tổng năng lượng tiêu tốn trong mỗi phiên multicast đã được trình bày ở phần 2.2. Với mục đích

²Trong bài viết [3], các tác giả đã coi năng lượng nhận tin là không đáng kể nên chỉ xét đến tổng số lượt truyền tin của các nút trong một phiên multicast. Tuy nhiên, để giữ cho bài toán tổng quát chúng tôi xét cả năng lượng nhận tin.

chỉ so sánh tương quan về độ tốt của các giải thuật, chúng tôi không sử dụng đến đơn vị tính năng lượng cụ thể. Mỗi giá trị trên trục hoành ứng với mỗi test dữ liệu của hình 3(a)-(c) cho biết tỉ lệ phần trăm các nút terminal so với tổng số $|V|$ nút và của hình 3(d) cho biết tỉ lệ phần trăm các khe thời gian hoạt động so với K khe thời gian của mỗi nút. Dựa vào các hình 3(a)-(d) có thể thấy rằng, giải thuật GAMEM và giải thuật HMEM có hiệu quả tốt hơn so với giải thuật TCS về mặt tối ưu năng lượng, đặc biệt là giải thuật GAMEM cho lời giải có năng lượng tiêu tốn nhỏ nhất trên hầu hết các test của bộ dữ liệu 1-4.



Hình 3: Năng lượng multicast của các giải thuật trên bốn bộ dữ liệu. (a) Bộ dữ liệu 1, $|V| = 100$. (b) Bộ dữ liệu 2, $|V| = 200$. (c) Bộ dữ liệu 3, $|V| = 300$. (d) Bộ dữ liệu 4, $|V| = 200$.

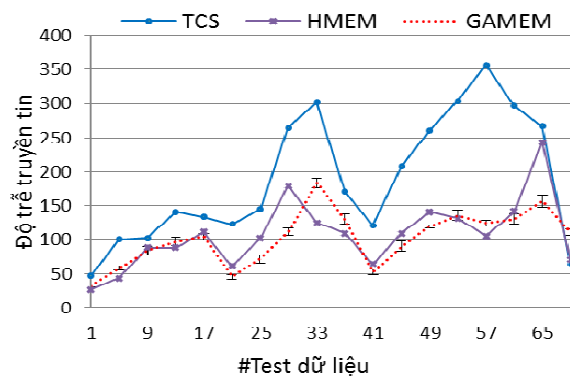
Kết quả thử nghiệm của các giải thuật được giải thích như sau. Giải thuật TCS không tiến hành tìm lời giải trực tiếp trên đồ thị gốc mà dựa trên đồ thị mở rộng [3]. Đồ thị này có cấu trúc đặc biệt, các đỉnh của nó bao gồm các đỉnh gốc trên đồ thị G cùng với các đỉnh vệ tinh được tạo thêm. Trong đó, mỗi đỉnh vệ tinh biểu diễn tương ứng một lượt truyền trong một khe thời gian nào đó của một đỉnh trên đồ thị gốc. Sau khi xây dựng đồ thị mở rộng, giải thuật TCS tiến hành lựa chọn một tập các nút vệ tinh C (bằng một giải thuật tham lam) sao cho mỗi đỉnh terminal đều kề với ít nhất một đỉnh trong C . Sau đó giải thuật tiến hành tìm một cây Steiner nhỏ nhất bao phủ các nút của C và từ cây Steiner này ánh xạ ra lời giải cho bài toán. Vì vậy độ tốt của TCS phụ thuộc nhiều vào bước lựa chọn các đỉnh cho tập C và độ tốt của giải thuật tìm cây Steiner được áp dụng (giải thuật trong [4]). Việc lựa chọn tập C (theo tiêu chí lựa chọn của phương pháp tham lam có trong [3]) được tiến hành riêng

rẽ và cố định từ trước sau đó mới tìm cây Steiner dẫn tới việc cây Steiner thu được có trọng số lớn trong nhiều trường hợp. Kết quả thử nghiệm cho thấy chất lượng lời giải xét về mặt tối ưu năng lượng của TCS còn thấp so với các thuật toán đề xuất. Giải thuật HMEM trực tiếp đưa ra cây lời giải cho bài toán sau quá trình tìm đường đi ngắn nhất trên đồ thị G' . Đồ thị này được xây dựng dựa trên cấu trúc đồ thị mạng ban đầu và điều kiện ràng buộc về các khe thời gian hoạt động của các nút. Ngoài ra, HMEM được nâng cao chất lượng lời giải bằng phương pháp tìm kiếm địa phương đề xuất nên cho kết quả tốt. Giải thuật GAMEM có các phép toán di truyền hiệu quả, trong số đó có phép toán lai ghép (cụ thể là lai ghép heuristic) và phép toán đột biến được áp dụng giải thuật HMEM trong quá trình xây dựng. Lý do này khiến cho giải thuật GAMEM có kết quả tốt hơn giải thuật HMEM về mặt tối ưu năng lượng trong hầu hết các test dữ liệu.

Hình 3(a)-(c) còn cho thấy khi số lượng các terminal tăng thì năng lượng multicast của các giải thuật cũng tăng. Điều này là hiển nhiên vì khi đó số lượng nút trong cây lời giải cũng tăng theo, dẫn tới năng lượng dùng để truyền và nhận tin trên cây này cũng tăng. Hình 3(d) cho thấy, trong bộ dữ liệu 4 giải thuật TCS kém hiệu quả đối với các test có tỉ lệ số khe thời gian hoạt động thấp (dưới 30%). Bên cạnh đó, khi tỉ lệ số khe thời gian hoạt động tăng thì năng lượng multicast của các giải thuật giảm. Điều đó là do khi tỉ lệ số khe thời gian hoạt động của các nút tăng lên thì khả năng nhiều nút có chung khe thời gian hoạt động cũng tăng, dẫn tới việc các nút chuyển tiếp của cây multicast lời giải chỉ cần truyền tin tại một số ít khe thời gian là đã có thể truyền tin được tới tất cả các nút con của nó, dẫn tới năng lượng multicast có xu hướng giảm xuống.

So sánh các giải thuật về độ trễ.

Độ trễ truyền tin được định nghĩa là khoảng thời gian tính từ khi nút gốc s bắt đầu gửi gói tin đến khi mọi nút terminal đều nhận được gói tin đó (tính theo đơn vị số khe thời gian). Để tính được độ trễ cho một phiên multicast ta phải áp dụng một lịch truyền không có xung đột nhằm tránh trường hợp xung đột có thể xảy ra giữa các lượt truyền tại cùng một thời điểm. Thực tế, bài toán cực tiểu hóa độ trễ multicast trên mạng cảm biến không dây là bài toán NP-khó, thậm chí đối với mạng cảm biến không dây mà các nút luôn hoạt động. Trong nghiên cứu này, chúng tôi áp dụng mô hình lịch truyền collision-free trong [3] để tính độ trễ truyền tin và so sánh các giải thuật. Hình 4 đưa ra độ trễ truyền tin trên cây multicast của các giải thuật trên các test dữ liệu. Các test dữ liệu được đánh chỉ số từ 1-72 (lần lượt theo các bộ dữ liệu 1-4). Giá trị trên trục hoành cho biết chỉ số của các test dữ liệu. Giá trị theo trục tung được tính theo đơn vị số khe thời gian. Đối với giải thuật GAMEM, trên mỗi test dữ liệu, độ trễ truyền tin được tính trung bình trên 50 cây lời giải (của 50 lượt chạy) cùng với khoảng tin cậy 95%. Thấy rằng, giải thuật HMEM từng bước thực hiện quá trình tìm đường đi ngắn nhất trên đồ thị G' để tạo ra cây multicast. Trong khi đó, giải thuật TCS xây dựng cây multicast bằng cách sử dụng cây Steiner tìm được trên đồ thị



Hình 4: Độ trễ truyền tin trên cây multicast của các giải thuật (tính theo đơn vị số khe thời gian).

mở rộng, vì thế khoảng cách từ nút gốc s đến các nút terminal trên cây multicast của HMEM thường nhỏ hơn so với TCS. Điều này dẫn tới độ trễ truyền tin của HMEM nhỏ hơn TCS. Giải thuật GAMEM có sử dụng phương pháp tìm đường đi ngắn nhất của giải thuật HMEM trong một số phép toán di truyền, nên hai giải thuật này có độ trễ truyền tin gần như tương đương.

Bảng 1 cho biết thời gian chạy trung bình trên mỗi test của các bộ dữ liệu 1-4 của các giải thuật. Dễ thấy thời gian tính toán của các giải thuật phụ thuộc vào kích thước mạng (số lượng nút). Giải thuật TCS tiến hành tìm lời giải dựa trên đồ thị mở rộng, đồ thị này có số đỉnh lớn hơn nhiều so với đồ thị gốc ban đầu nên có thời gian tính lớn hơn giải thuật HMEM. Giải thuật GAMEM có thời gian tính lớn nhất trong ba giải thuật do đặc điểm của giải thuật di truyền cần phải thực hiện các phép toán di truyền trên nhiều nhiễm sắc thể và qua nhiều thế hệ quần thể, tuy nhiên thời gian tính này vẫn ở mức chấp nhận được (trung bình cỡ 35 s đối với bộ dữ liệu có số lượng đỉnh lớn nhất).

Bộ dữ liệu	TCS	HMEM	GAMEM
1	108.95	59.15	2662.92
2	422.25	271.05	13459.57
3	868.25	526.55	35211.17
4	394.08	179.50	11109.78

Bảng 1: Thời gian tính trung bình của các giải thuật trên mỗi test của bốn bộ dữ liệu (đơn vị millisecond).

5.4. Nhận xét về khả năng triển khai

Giải thuật GAMEM đòi hỏi thông tin toàn cục và tính toán tập trung nên phù hợp với những phiên multicast diễn ra thường xuyên với tập terminal và nút nguồn xác định trước. Khi đó cần có một nút tập trung (thường là trạm gốc) để thu thập thông tin trong mạng, tính toán ra cây multicast và lịch truyền trước rồi truyền thông tin này cho các nút cảm biến để áp dụng cho những phiên truyền multicast này.

Với cách tiếp cận phân tán cho những yêu cầu multicast đến ngẫu nhiên, giải thuật TCS có độ phức tạp tính toán là $O(|D| \cdot |V|)$ với D là đường kính của đồ thị mạng cảm biến, độ phức tạp về số lượng gói tin trao đổi là $O(|M| \cdot |V|)$ [3]. Độ phức tạp của giải thuật HMEM chủ yếu phụ thuộc vào quá trình tìm $|M|$ đường đi ngắn nhất trên đồ thị G' (dòng 2-9 của Algorithm 1). Giả sử sử dụng giải thuật phân tán trong [19] để tìm đường đi ngắn nhất và giả sử phạm vi truyền tin của các nút là hằng số thì HMEM sẽ có độ phức tạp tính toán là $O(|M| \cdot |V|^{1+\varepsilon})$ và độ phức tạp về số lượng gói tin trao đổi là $O(|M| \cdot |E|^{1+\varepsilon})$, trong đó $\varepsilon = O(1) / \sqrt[4]{\log |V|}$.

6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Bài báo này đề xuất một giải thuật heuristic (HMEM) và một giải thuật di truyền (GAMEM) nhằm giải quyết bài toán định tuyến đa điểm trên mạng cảm biến không dây nhiệm vụ tuần hoàn với mục tiêu tiết kiệm năng lượng truyền và nhận dữ liệu. Các giải thuật đề xuất được đánh giá trên bốn bộ dữ liệu và được so sánh kết quả với giải thuật TCS. Kết quả thử nghiệm cho thấy hai giải thuật đề xuất mang lại lời giải tốt hơn giải thuật TCS về mặt tối ưu năng lượng và độ trễ truyền tin. Trong thời gian tới, chúng tôi sẽ phát triển thêm các giải pháp cho bài toán này trong việc tối thiểu hóa độ trễ với các ràng buộc tránh xung đột. Ngoài ra, chúng tôi sẽ mở rộng mô hình để giải quyết bài toán multicast với độ tin cậy cao cho các trường hợp các nút cảm biến hoạt động thiếu ổn định.

LỜI CẢM ƠN

Đề tài nghị định thư hợp tác với Nhật Bản – Nghiên cứu, phát triển mạng quang đô thị bền vững - mã số 12/2012/HD-NĐT, đã hỗ trợ nhóm tác giả để hoàn thành bài báo này.

TÀI LIỆU THAM KHẢO

- [1] F. Wang, and J. Liu, “Duty-cycle-aware broadcast in wireless sensor networks”, in *Proc. IEEE INFOCOM*, pp. 468–476, 2009.
- [2] S. Xiong, J. Li, M. Li, J. Wang, and Y. Liu, “Multiple task scheduling for low-duty-cycled wireless sensor networks”, in *Proc. IEEE INFOCOM*, pp. 1323–1331, 2011.
- [3] K. Han, Y. Liu, and J. Luo, “Duty-Cycle-Aware Minimum-Energy Multicasting in Wireless Sensor Networks,” *IEEE/ACM Transactions on Networking*, vol. 21, no. 3, pp. 910–923, 2013.
- [4] L. Kou, G. Markowsky, and L. Berman, “A fast algorithm for Steiner trees”, *Acta Informatica*, vol. 15, no. 2, pp. 141–145, 1981.
- [5] D. S. Johnson, “Approximation algorithms for combinatorial problems”, in *Proc. ACM STOC*, pp. 38–49, 1973.
- [6] L. Su, B. Ding, Y. Yang, T. F. Abdelzaher, G. Cao, and J. C. Hou, “ocast: Optimal multicast routing protocol for wireless sensor networks”, in *Proc. IEEE ICNP*, pp. 151–160, 2009.
- [7] M. R. Garey, and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*, W. H. Freeman, New York, 1979.
- [8] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, Third edition, The MIT Press, Cambridge, 2009.
- [9] S. Guo, Y. Gu, B. Jiang, and T. He, “Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links,” in *Proc. ACM MobiCom*, pp. 133–144, 2009.
- [10] G. Anastasi, M. Conti, M. D. Francesco, and A. Passarella, “Energy conservation in wireless sensor networks: A survey,” *Ad Hoc Networks*, vol. 7, no. 3, pp. 537–568, 2009.
- [11] C. Gui, and P. Mohapatra, “Power conservation and quality of surveillance in target tracking sensor networks”, in *Proc. ACM MobiCom*, pp. 129–143, 2004.
- [12] T. He, P. Vicaire, T. Yan, Q. Cao, G. Zhou, L. Gu, L. Luo, R. Stoleru, J. A. Stankovic, and T. F. Abdelzaher, “Achieving long-term surveillance in vigilnet,” in *Proc. IEEE INFOCOM*, pp. 1–12, 2006.
- [13] L. Mo, Y. He, Y. Liu, J. Zhao, S. Tang, X.-Y. Li, and G. Dai, “Canopy closure estimates with greenorbs: sustainable sensing in the forest”, in *Proc. ACM SenSys*, pp. 99–112, 2009.

- [14] J. Wieselthier, G. Nguyen, and A. Ephremides, “On the construction of energy-efficient broadcast and multicast trees in wireless networks”, in *Proc. IEEE INFOCOM*, pp. 585–594, 2000.
- [15] P.-J. Wan, G. Calinescu, and C.-W. Yi, “Minimum-power multicast routing in static ad hoc wireless networks”, *IEEE/ACM Trans. Netw.*, vol. 12, no. 3, pp. 507–514, 2004.
- [16] W. Liang, “Approximate minimum-energy multicasting in wireless ad hoc networks,” *IEEE Trans. Mobile Comput.*, vol. 5, no. 4, pp. 377–387, 2006.
- [17] D. Li, Q. Liu, X. Hu, and X. Jia, “Energy efficient multicast routing in ad hoc wireless networks”, *Computer Communications*, vol. 30, no. 18, pp. 3746–3756, 2007.
- [18] G. Ausiello, A. D’Atri, and M. Protasi, “Structure preserving reductions among convex optimization problems,” *Journal of Computer and System Sciences*, vol. 21, no. 1, pp. 136–153, 1980.
- [19] B. Awerbuch, “Distributed Shortest Paths Algorithms (Extended Abstract)”, in *Proc. ACM STOC*, pp. 490–500, 1989.

Ngày nhận bài 26 – 9 – 2013
Nhận lại sau sửa 26 – 7 – 2014