CẢI TIẾN THUẬT TOÁN LUỒNG CỰC ĐẠI CÓ GIÁ CỰC TIỂU CHO PHƯƠNG PHÁP MTA

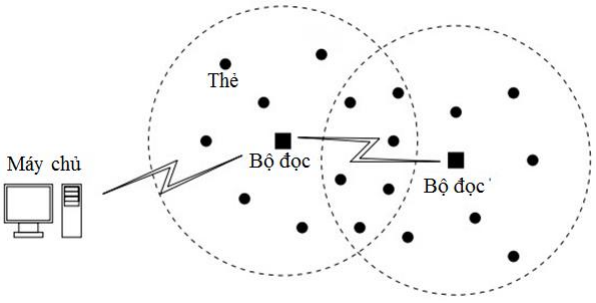
PHAN HOÀNG NAM[*]*, TRẦN HOÀI NHÂN*
Khoa Tin học, Trường Đại học Sư phạm, Đại học Huế
[*]*Email: nam18ph@gmail.com*

Tóm tắt: Radio Frequency Identification (RFID) là công nghệ vô tuyến tầm ngắn để thu thập dữ liệu tự động xuất hiện lần đầu tiên vào thập niên 1960. Có hai phương pháp cân bằng tải cho hệ thống RFID đã được đề xuất vào năm 2007 là phương pháp Min-Max Cost Assignment (MCA) và Min-Max Tag Count Assignment (MTA). Trong đó, phương pháp MTA là trường hợp đặc biệt của MCA. Qunfeng Dong và các cộng sự đã đề xuất các thuật toán cho MTA dựa trên Luồng cực đại (MNF). Trong luận văn, tôi đã đề xuất một hàm mục tiêu với vai trò phân phối lại thẻ dựa trên năng lượng của bộ đọc và sử dụng thuật toán Luồng cực đại có giá thành nhỏ nhất (MCMF) cho MTA. Trong bài báo này, chúng tôi tiếp tục nghiên cứu thuật toán MCMF cho MTA trong việc: Xử lý ràng buộc dương vô cùng trong hàm mục tiêu đã đề xuất trước đây và đề xuất một chu trình đổi luồng mới nhằm giảm giá trị cho hàm mục tiêu đến mức tối đa. Thuật toán MCMF dựa trên chu trình mới vẫn đảm bảo chạy trong thời gian đa thức. Cuối cùng chúng tôi chứng minh đây là một cải tiến so với thuật toán đã trình bày trong luận văn.

Từ khóa: MCMF, MNF, MCA, MTA, RFID.

1. GIỚI THIỆU

RFID là công nghệ nhận dạng đối tượng bằng sóng vô tuyến, cho phép truyền và nhận dữ liệu từ một điểm đến một điểm khác. Công nghệ này đáng tin cậy để phát hiện và giám sát điện tử, một dạng mới của phương pháp truyền thông tin vô tuyến. Bộ đọc quét dữ liệu của thẻ và gửi thông tin đến cơ sở dữ liệu lưu trữ dữ liệu của thẻ. Công nghệ này có ứng dụng lớn trong thực tiễn: Thẻ có thể được đặt trên kính chắn gió xe hơi để hệ thống thu phí đường bộ có thể nhanh chóng nhận dạng và thu tiền trên các tuyến đường. Một hệ thống RFID có ba thành phần cơ bản: Thẻ, Bộ đọc và Máy chủ.



Hình 1. *Hệ thống RFID*

Tạp chí Khoa học, Trường Đại học Sư phạm, Đại học Huế
ISSN 1859-1612, Số 04(48)/2018: tr. 117-127
Ngày nhận bài: 16/11/2018; Hoàn thành phản biện: 10/12/2018; Ngày nhận đăng: 12/12/2018

Việc sử dụng năng lượng ở bộ đọc chủ yếu trong giao tiếp bộ đọc-thẻ. Giảm thiểu năng lượng này được các tác giả trong [1],[2] xem như là vấn đề Min-Max Cost Assignment (MCA), với giả sử rằng bộ đọc có thể sử dụng các mức công suất khác nhau để đọc các thẻ khác nhau, được xác định dựa trên khoảng cách của thẻ với bộ đọc. Các phương pháp sau sẽ xem xét vấn đề cân bằng tải dưới bài toán tối ưu hóa trên đồ thị.

Cấu trúc tiếp theo của bài viết như sau: mục 2 trình bày mô hình bài toán RFID và các nghiên cứu liên quan; mục 3 mô tả thuật toán đề xuất; mục 4 mô phỏng đánh giá thuật toán đề xuất và mục 5 là phần kết luận.

2. MÔ HÌNH BÀI TOÁN VÀ CÁC NGHIÊN CỨU LIÊN QUAN

2.1. Mô hình hóa RFID bằng đồ thị

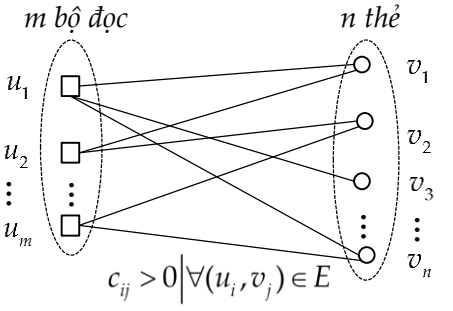
Các tác giả trong [1],[2] đã chuyển bài toán cân bằng tải trong hệ thống RFID thành mô hình đồ thị hai phía có trọng số dương $G=\langle U\cup V,E\rangle$, trong đó:

- $U=\{u_1,u_2,...,u_m\}$ là tập hợp m bộ đọc và $V=\{v_1,v_2,...,v_n\}$ là tập hợp n thẻ.

- $E=\{(u,v)|u\in U,v\in V\}$ là tập cạnh sao cho bộ đọc u và thẻ v có thể giao tiếp với nhau.

- $c_{ij}=c(u_i,v_j)>0$ là năng lượng mỗi lần bộ đọc u_i đọc thẻ v_j.

S



Hình 2. *Đồ thị mô hình hóa hệ thống RFID*

2.2. Phương pháp MCA và MTA

Các tác giả trong [1], [2] tiếp tục đề xuất các phương pháp cân bằng tải cho hệ thống RFID trên đồ thị. Họ đã mô hình hóa phương pháp MCA như sau:

Cho đồ thị $G=\langle U\cup V,E\rangle$, chi phí $c_{ij}:(u_i,v_j)\mapsto Z^+$ và $B_i\in Z^+$. Tìm một phân bổ $\varphi:V\to U$ cho các thẻ v_j vào các bộ đọc sao cho tổng chi phí năng lượng tối đa trên tất cả các bộ đọc được tối thiểu và thỏa ràng buộc trên mỗi bộ đọc $u_i\in U$.

$$\sum_{j\in[1,n],u_i=\varphi(v_j)} c_{ij}\le B_i \qquad (1)$$

Nếu $c_{ij}=1$ phương pháp MCA được chuyển thành phương pháp MTA (Min-Max Tag Count Assignment).

2.3. Đề xuất thuật toán MCMF cho phương pháp MTA

Phương pháp MTA có thể được giải quyết bằng thuật toán Luồng cực đại trên mạng (MNF) [1], [6], [7]. Từ mô hình đồ thị $G=\langle U\cup V,E\rangle$ biểu diễn phương pháp MCA các tác giả đã chuyển thành một *Mạng* như sau:

$\forall(u_i,v_j)\in E$ đều có khả năng thông qua $c_{ij}=c(u_i,v_j)=1$.

$U=U\cup\{s,t\}$, *s* được gọi là đỉnh phát, *t* được gọi là đỉnh thu.

$E=E\cup(s,u_i)$ $\forall u_i\in U$ với khả năng thông qua $c_{s,u_i}=c(s,u_i)=B_i$.

$E=E\cup(v_j,t)$ $\forall v_j\in V$ với khả năng thông qua $c_{v_j,t}=c(v_j,t)=1$.

MTA trở thành bài toán tìm *Luồng cực đại* (Maximum Network Flow - MNF) trên mạng $G=\langle U\cup V,E\rangle$.

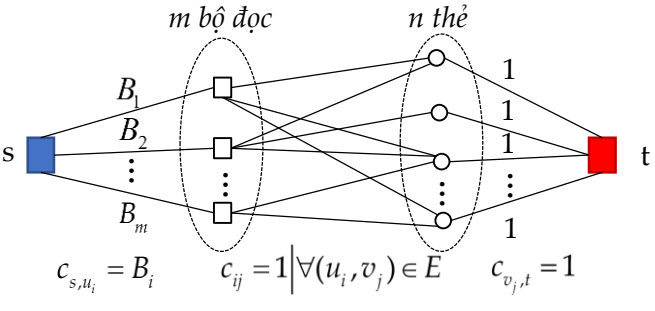
Hàm mục tiêu cho thuật toán MNF

Để hệ thống đạt hiệu quả hơn trong việc sử dụng năng lượng, bằng cách phân bố lại số thẻ dựa trên mức năng lượng của bộ đọc mà không ảnh hưởng đến số lượng tối đa các thẻ đã được phân phối cho các bộ đọc. Tác giả trong [3] đã đề xuất một hàm mục tiêu phân bố như sau: $T=\sum_{u\in U}\frac{c_{s,u}}{f_{s,u}}\rightarrow\min$ (2)

Ràng buộc: $T=\begin{cases}\sum_{u\in U}\frac{c_{s,u}}{f_{s,u}} & \forall u\in U: f_{s,u}>0\\ +\infty & \exists u\in U: f_{s,u}=0\end{cases}$ (3)

T được gọi là *tỉ số giữa năng lượng* của bộ đọc *u* và số thẻ được gán cho bộ đọc đó. Giá trị hàm T cần nhỏ nhất, bài toán tìm *Luồng cực đại trên mạng* trở thành bài toán tìm *Luồng cực đại với giá cực tiểu* (Minimum Cost Maximum Flow - MCMF).

Ví dụ: Chuyển đổi bài toán MTA sang bài toán MNF và MCMF



Hình 3. *Mô hình bài toán tìm Luồng cực đại có giá cực tiểu*

3. CÁC THUẬT TOÁN CẢI TIẾN

3.1. Thuật toán dựa MCMF với chu trình ϕ^1

Như đã đề cập ở trên, tác giả trong [3] đã đề xuất hàm mục tiêu *T* và thêm ràng buộc dương vô cùng nếu tồn tại một bộ đọc chưa được phân phối thẻ. Thuật toán MCMF đã được đề xuất trong [3] thực hiện việc đổi luồng dọc theo chu trình xuất phát từ đỉnh phát *s* qua hai bộ đọc và một thẻ - sau khi đã tìm được luồng cực đại - để giảm giá trị hàm mục tiêu. Trong bài báo này chúng tôi đặt tên chu trình này là ϕ^1 và tên thuật toán đã được trình bày trong [3] là $\phi^1 MCMF$. Tại đây, chúng tôi thêm phần xử lý ràng buộc dương vô cùng vào thuật toán và trình bày lại như sau:

Input: *Mạng* $G = \langle U \cup V, E \rangle$

Output: *Luồng cực đại* f^* *sao cho hàm mục tiêu T nhỏ nhất có thể.*

Function $\phi^1 MCMF$ *()*

Tìm luồng cực đại f *trong mạng* G *;*

Tính giá trị hàm mục tiêu T theo công thức (3) ;

do {

foreach $u_i, u_k \in U$ **do**

if *(*$v \in V$ *sao cho* $(u_i, v), (u_k, v) \in E$ *)* **then**

if *(* $f(u_i, v) = 1$ *và* $f(u_k, v) = 0$ *và (đổi luồng làm giảm hàm mục tiêu T hoặc giảm số cung (s,u) có* $f_{s,u} = 0$ *))* **then** *cập nhật lại luồng* f *;*

if *(* $f(u_i, v) = 0$ *và* $f(u_k, v) = 1$ *và (đổi luồng làm giảm hàm mục tiêu T hoặc giảm số cung (s,u) có* $f_{s,u} = 0$ *))* **then** *cập nhật lại luồng* f *;*

}

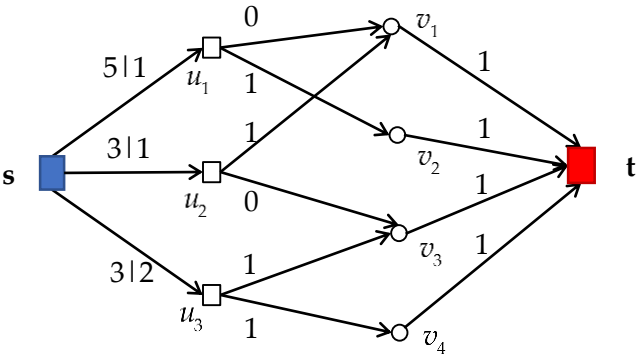
while *(hàm mục tiêu T đã được cập nhật) ;*

return f *;*

Như vậy, thuật toán $\phi^1 MCMF$ có thể giảm hàm mục tiêu T sau khi tìm được luồng cực đại, giúp phân phối thẻ vào các bộ đọc được cân bằng hơn [3]. Tuy nhiên, thuật toán chỉ phân phối lại thẻ cho hai bộ đọc bất kỳ có cùng thẻ; bởi vì, tất cả chu trình thay đổi luồng có dạng:

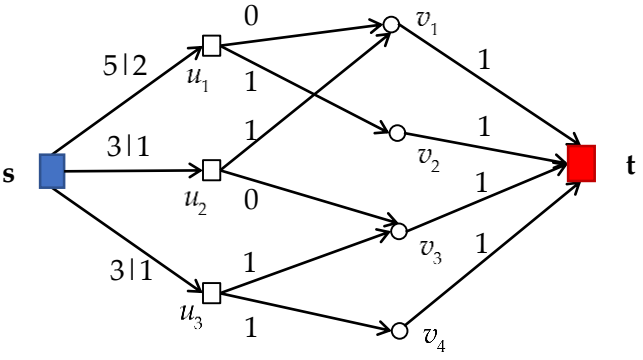
$$\phi^1 = s \xrightarrow{f<c} u_i \xrightarrow{0} v \xrightarrow{1} u_k \xrightarrow{f>1} s \quad (\forall u_i, u_k \in U, \forall v \in V) \tag{4}$$

Nên, nếu rơi vào trường hợp sau, $\phi^1 MCMF$ không thể tiếp tục làm giảm được hàm mục tiêu T. Thật vậy, giả sử sau khi tìm được luồng cực đại ta có $T = 9.5$ (Hình 4):



Hình 4. *Trường hợp thuật toán MCMF không làm giảm hàm mục tiêu T*

Trong ví dụ ở Hình 4, thuật toán $\phi^1 MCMF$ chỉ xét được 2 chu trình: $\phi_1^1 = s \xrightarrow{f<c} u_1 \xrightarrow{0} v_1 \xrightarrow{1} u_2 \xrightarrow{f>1} s$ và $\phi_2^1 = s \xrightarrow{f<c} u_2 \xrightarrow{0} v_3 \xrightarrow{1} u_3 \xrightarrow{f>1} s$, các chủ trình đều không làm giảm hàm T. Nếu xét chu trình qua 2 thẻ như sau $s \xrightarrow{f<c} u_1 \xrightarrow{0} v_1 \xrightarrow{1} u_2 \xrightarrow{0} v_3 \xrightarrow{1} u_3 \xrightarrow{f>1} s$ làm giảm hàm T. Cụ thể, với chu trình mới $T = 8.5$ cũng là giá trị nhỏ nhất (Hình 5)



Hình 5. *Luồng mới có giá trị T đạt cực tiểu*

3.2. Chu trình đổi luồng mới ϕ^k

Từ ví dụ trên, chúng tôi đề xuất thuật toán dựa trên chu trình mới. Chu trình chứa k thẻ với $k \geq 1$ và ký hiệu ϕ^k. Chu trình ϕ^k được định nghĩa như sau:

$$\phi^k = s \xrightarrow{f<c} u_{i_1} \xrightarrow{0} v_{i_1} \xrightarrow{1} \ldots \xrightarrow{0} v_{i_k} \xrightarrow{1} u_{i_{k+1}} \xrightarrow{f>1} s \ (1 \leq k < m) \qquad (5)$$

trong đó:

$u_{i_j} \xrightarrow{0} v_{i_j} : (u_{i_j}, v_{i_j}) \in E,\ f_{u_{i_j}, v_{i_j}} = 0$

$v_{i_j} \xrightarrow{1} u_{i_{j+1}} : (u_{i_{j+1}}, v_{i_j}) \in E,\ f_{u_{i_{j+1}}, v_{i_j}} = 1$

3.3. Thuật toán dựa trên MCMF với chu trình ϕ^k

Trong bài báo này, chúng tôi đặt tên thuật toán mới là $\phi^k MCMF$. Thuật toán được trình bày như sau:

Input: *Mạng* $G = \langle U \cup V, E \rangle$

Output: *Luồng cực đại* f^* *sao cho hàm mục tiêu* T *nhỏ nhất.*

Function $\phi^k MCMF$ *()*

Tìm luồng cực đại f *trên mạng* G *;*
Tính giá trị hàm mục tiêu T theo công thức (3) ;
do {
  foreach $k = [1..n]$ **do**
    if *(*ϕ^k *làm giảm hàm mục tiêu T)* **then** {
      Cập nhật luồng dọc theo ϕ^k *;*
      Cập nhật hàm T ;
    }
    else {
    if *(*$T = +\infty$ *)* **then**
      if (*ϕ^k làm giảm số cung (s,u) có* $f_{s,u} = 0$) **then** *Cập nhật luồng dọc theo* ϕ^k *;*
    }
}
while *(hàm mục tiêu T đã được cập nhật)* ;
return f ;

3.4. Chứng minh các tính chất của chu trình ϕ^k

i. **Chu trình ϕ^k chỉ phân phối lại số lượng thẻ cho 2 bộ đọc đầu và cuối của nó.** Chứng minh: Chu trình ϕ^k có bộ đọc đầu tiên là u_{i_1} và bộ đọc cuối cùng là $u_{i_{k+1}}$; mỗi bộ đọc u_{i_j} còn lại có hai đỉnh lân cận $v_{i_{j-1}}, v_{i_j}$ sao cho $u_j \xrightarrow{f=1} v_{j-1}$ và $u_j \xrightarrow{f=0} v_j$. Nếu cập nhật luồng dọc theo ϕ^k thì ta có $f\left(u_{i_j}, v_{i_{j-1}}\right) = 0$ và $f\left(u_{i_j}, v_{i_j}\right) = 1$, nghĩa là số lượng thẻ đã phân bố cho bộ đọc u_{i_j} không thay đổi. Như vậy, tổng số thẻ mà các bộ đọc trên ϕ^k từ 2 tới k không thay đổi.

ii. **Chu trình ϕ^k không làm thay đổi tổng luồng ra khỏi *s*.** Chứng minh: Với mọi chu trình làm thay đổi hàm *T* thì $f^*_{s,u_i} + f^*_{s,u_k} = f_{s,u_i} + 1 + f_{s,u_k} - 1 = f_{s,u_i} + f_{s,u_k}$ với u_i, u_k là hai bộ đọc đầu và cuối của chu trình. Như vậy, chu trình ϕ^k không làm thay đổi tổng luồng ra khỏi đỉnh phát *s*.

iii. **Mọi chu trình làm giảm hàm T đều có dạng ϕ^k.** Chứng minh: Có hai trường hợp sau khi tìm được luồng cực đại *f*: (1) $\forall u \in U : f_{s,u} = c_{s,u}$ và $\sum_{\forall v \in V} f_{v,t} \le m$; (2) $\exists u \in U : f_{s,u} < c_{s,u}$ và $\sum_{\forall v \in V} f_{v,t} = m$. Đối với trường hợp (1) thuật toán không cần phải tìm ϕ^k vì lúc này hàm *T* có giá trị nhỏ nhất. Đối với trường hợp (2), có thể có luồng cực đại mới làm thay đổi hàm *T* nên cần phải tìm chu trình thay đổi luồng. Lúc này chu trình cần tìm phải xuất phát từ đỉnh phát *s* và không thể đến đỉnh nguồn *t* bởi vì $f_{v,t|\forall v \in V} = 1$. Vậy chu trình cần tìm phải có dạng ϕ^k.

iv. **Thuật toán $\phi^k MCMF$ không lặp vô hạn.** Chứng minh: Thuật toán $\phi^k MCMF$ dừng tại hai trường hợp là hàm *T* đạt giá trị nhỏ nhất hoặc mọi chu trình ϕ^k được tìm xong. Bây giờ ta chỉ cần chừng minh hàm *T* sẽ giảm tới giá trị nhỏ nhất với số lần giảm là hữu hạn. Gọi $f^*_{s,u|u \in U}$ là giá trị luồng từ *s* tới *u* khi hàm *T* đạt giá trị nhỏ nhất. Mỗi chu trình ϕ^k dịch chuyển $f_{s,u}$ tới gần $f^*_{s,u}$ một đơn vị và do $f_{s,u}$, $f^*_{s,u}$ đều là số tự nhiên nên chắc chắn sau một số hữu hạn lần $f_{s,u} = f^*_{s,u}$.

Tính chất (i), (ii) và (iii) khẳng định tính đúng đắn của thuật toán $\phi^k MCMF$. Tính chất (iv) khẳng định tính dừng của thuật toán $\phi^k MCMF$.

3.5. Độ phức tạp giải thuật tính toán

Thuật toán $\phi^k MCMF$ gồm hai giai đoạn độc lập nhau: (1) Tìm luồng cực đại và (2) Tìm các chu trình ϕ^k cho đến khi hàm *T* không thể giảm.

- Độ phức tạp tính toán của giai đoạn (1) đã được chứng minh là
$$\Theta\left(|U \cup V| * |E|^2\right) = \Theta\left(m^3 n^2 + m^2 n^3\right)$$
- Độ phức tạp tính toán của giai đoạn (2) trong trường hợp xấu nhất: Do không biết trước giá trị nhỏ nhất của hàm *T* nên trường hợp xấu nhất của thuật toán là phải tìm mọi chu trình ϕ^k. Thấy rằng, trong ϕ^{m-1} có chứa m^2 các chu trình có ít thẻ hơn. Mặt khác, từ mỗi chu trình ϕ^{m-1} đã tìm được có thể sinh ra chu trình ϕ^{m-1} mới bằng cách thay thế $m-1$ thẻ cũ thành các $m-1$ thẻ mới. Trường hợp xấu nhất, có $m^2 * n * m$ chu trình ϕ^k cần tìm và việc đổi luồng dọc theo ϕ^k trong trường hợp này là duyệt qua *m* bộ đọc.

Kết luận độ phức tạp tính toán của thuật toán $\phi^k MCMF$:

$$\Theta\left(m^3n^2 + m^2n^3 + m^4n\right) = \Theta\left(m^2n(m+n)^2\right) \quad (6)$$

4. THỰC NGHIỆM VÀ ĐÁNH GIÁ KẾT QUẢ

4.1. Mục tiêu thực nghiệm

Chúng tôi đã chứng minh được mọi chu trình đối luồng làm giảm hàm mục tiêu *T* tại tính chất (iii) và nêu ra ví dụ thuật toán $\phi^1 MCMF$ không làm giảm được hàm mục tiêu. Bây giờ, chúng tôi tiến hành thử nghiệm thực thế để so sánh hai thuật toán với mục tiêu sau:

- Đánh giá khả năng làm giảm giá trị hàm mục tiêu của hai thuật toán $\phi^1 MCMF$ và $\phi^k MCMF$.
- So sánh tốc độc chạy chương trình viết theo hai thuật toán: $\phi^1 MCMF$ *(1)* và $\phi^k MCMF$ *(2)*. Công thức so sánh thời gian:

$$S = \frac{\text{Thời gian chạy của (1) trên dữ liệu i - Thời gian chạy của (2) trên dữ liệu i}}{100} * 100\%$$

- Nếu $S < 0$: Thời gian chạy của chương trình 1 nhanh hơn 2,
- Nếu $S = 0$: Thời gian chạy của hai chương trình bằng nhau,
- Nếu $S > 0$: Thời gian chạy của chương trình 2 nhanh hơn 1.

4.2. Cách thức tiến hành

- Chúng tôi cài đặt chương trình cho cả hai thuật toán cần so sánh.
- Thực hiện chạy chương trình trên cùng một hệ thống máy tính.
- Các bộ dữ liệu dùng so sánh được chúng tôi tạo ra với như sau:

Bảng 1. *Mô tả các bộ dữ liệu dùng để so sánh*

	Kích thước	**Phân bố thẻ** (Số thẻ mỗi bộ đọc có thể giao tiếp)	**Năng lượng của mỗi bộ đọc**
1	R=3,T=4	$N_1 = 2, N_2 = 2, N_3 = 2$	$B_1 = 5,\ B_2 = 3,\ B_3 = 3$
2	R=4,T=5	$N_1 = 2, N_2 = 2, N_3 = 2, N_4 = 2$	$B_1 = 6, B_2 = 5, B_3 = 3, B_4 = 3$
3	R=6,T=10	$N_1 = 2, N_2 = 2, B_3 = 2, N_4 = 2,$ $N_5 = 3, N_6 = 2$	$B_1 = 10, B_2 = 9, B_3 = 9, B_4 = 4,$ $B_5 = 4, B_6 = 3$
4	R=6,T=9	$N_1 = 2, N_2 = 2, B_3 = 2, N_4 = 2,$ $N_5 = 2, N_6 = 2$	$B_1 = 10, B_2 = 9, B_3 = 9, B_4 = 4,$ $B_5 = 4, B_6 = 3$
5	R=7,T=22	Ngẫu nhiên	Ngẫu nhiên
6	R=8,T=24	Ngẫu nhiên	Ngẫu nhiên
7	R=10,T=50	Ngẫu nhiên	Ngẫu nhiên
8	R=50,T=71	Ngẫu nhiên	Ngẫu nhiên

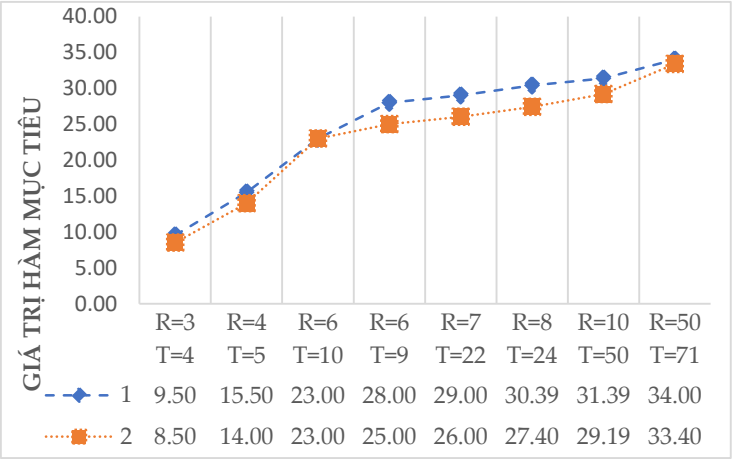
5. KẾT QUẢ

Bảng 2. *Số thẻ tối đa được phân phối cho các bộ đọc*

	Dữ liệu 1	**Dữ liệu 2**	**Dữ liệu 3**	**Dữ liệu 4**	**Dữ liệu 5**	**Dữ liệu 6**	**Dữ liệu 7**
$\phi^1 MCMF$	4	5	10	9	22	44	71
$\phi^k MCMF$	4	5	10	9	22	44	71
Đạt mức	100%	100%	100%	100%	100%	88%	100%

Bảng 3. *Tốc độ chạy chương trình của hai thuật toán*

		Thời gian chạy của thuật toán 1 trên các bộ dữ liệu							
		0.054	0.0740	0.075	0.0810	0.155	0.130	0.132	0.224
Thời gian chạy của thuật toán 2 trên các bộ dữ liệu	0.060	-0.01%							
	0.070		0.00%						
	0.083			-0.01%					
	0.069				0.01%				
	0.132					0.02%			
	0.161						-0.03%		
	0.173							-0.04%	
	0.230								-0.01%



Biểu đồ 1. *Giá trị hàm mục tiêu T đạt được của thuật toán* $1: \phi^1 MCMF$, $2: \phi^k MCMF$

Phân tích kết quả:

- Biểu đồ 1: Cho thấy giá trị hàm mục tiêu đạt được bởi thuật toán 2 luôn bé hơn hoặc bằng thuật toán 1.
- Bảng 2: Cho thấy số lượng thẻ phân phối cho các bộ đọc của hai thuật toán đều bằng nhau.

- Bảng 3: Cho thấy tốc thực tế của hai chương trình không chệnh lệch nhau. Từ đây chúng tôi có thể khẳng định, trong trường hợp số lượng thẻ và số lượng bộ đọc lớn hơn (thực tế số lượng thẻ luôn lớn hơn rất nhiều số bộ đọc) hai chương trình tốc độ chạy tương đương nhau, bởi vì độ phức tạp tính toán cũa hai thuật toán đều bằng với độ phức tạp tính toán của bước tìm luồng cực đại.

4. KẾT LUẬN

Dựa vào phản ví dụ được thể hiện tại Hình 4, các tính chất của thuật toán $\phi^k MCMF$ đã được chứng minh và kết quả so sánh thực nghiệm theo hai thuật toán tại Biểu đồ 1, chúng tôi kết luận rằng: Thuật toán $\phi^k MCMF$ tối ưu được hàm mục tiêu *T* hơn so với thuật toán $\phi^1 MCMF$ mà vẫn phân phối được tối đa số thẻ cho các bộ đọc.

Chúng tôi sẽ tiếp tục nghiên cứu thay đổi hàm mục tiêu *T* để có thể giảm được ràng buộc dương vô cùng trong trường hợp tồn tại $f_{s,u} = 0$ và cải tiến thuật toán $\phi^k MCMF$ nhằm giảm độ phức tạp tính toán của nó.

TÀI LIỆU THAM KHẢO

[1] Qunfeng Dong et al. Load Balancing in Large-Scale RFID Systems. *0743-166X/07 © 2007 IEEE*.

[2] Qunfeng Dong et al. Load Balancing in Large-Scale RFID Systems. *This is an extended version of a paper that appeared in IEEE Infocom 2007: Minisymposium on Wireless Networks*, Anchorage, Alaska, USA, May 2007.

[3] Phan Hoàng Nam. *Tìm hiểu phương pháp cân bằng tải trong hệ thống RFID*. Luận văn Thạc sĩ, Chuyên ngành Công nghệ thông tin, Trường Đại học Khoa học, Đại học Huế 2017.

[4] Vijayakumar G. Dhas et al. Effective Load Balancing with Power Conservation in RFID. *International Journal of UbiComp (IJU)*, Vol.1, No.4, October 2010.

[5] Vijayakumar G. Dhas et al. Optimal Solution for RFID Load Balancing. N. Meghanathan et al. (Eds.): NeCoM, WiMoN, and WeST 2010, CCIS 90, pp. 41–49, 2010 © Springer-Verlag Berlin Heidelberg 2010.

[6] J. Erickson. *Maxflow Extensions Lecture Notes*. UIUC, Fall 2013.

[7] L.R. Ford and D.R. Fulkerson. *Maximal Flow Through a Network*. Canadian Journal of Mathematics 8.3 (1956): 399-404.

Tilte: IMPROVE ALGORITHM WITH MINIMUM COST AND MAXIMUM FLOW FOR METHOD MTA

Abstract: Radio Frequency Identification (RFID) which has reseached since 1960 is a short-range radio technology to collect data automatically. Two solutions were proposed in 2007 to balance load problem in RFID system were Min-Max Cost Assignment (MCA) and Min-Max Tag Count Assignment (MTA) [1]. MTA was a special case of MCA. Qunfeng Dong et al

proposed an algorithm based on the Maximum network flow (MNF) to solve the MTA. In my dissertation, I proposed an objective function to redistribute tags based on the energy of readers and used the Minimum-Cost Maximum-Flow (MCMF) algorithm. In this paper, we continue to research MCMF algorithm for the MTA problem and proposed a new cycle for changing flow to reduce value of the objective function to minimum, and the computational complexity of this algorithm is polynomial. Finally, we prove this algorithm is an improvement over the algorithm presented in my dissertation.

Keywords: MCMF, MNF, MCA, MTA, RFID.