

# Chương 1

## GIỚI THIỆU

### 1.1. Đặt vấn đề

Sự bùng nổ của thiết bị di động đặt ra nhiều vấn đề phức tạp cho các nhà nghiên cứu, một trong các vấn đề được nhiều nhà nghiên đặc biệt quan tâm đó là vấn đề năng lượng trên các thiết bị di động. Các nghiên cứu phát triển phần mềm tập trung vào việc nghiên cứu sự ảnh hưởng của phần mềm tới mức tiêu hao năng lượng, từ đó đưa ra các giải pháp nhằm tối ưu phần mềm giúp quá trình thực hiện các tác vụ ít tiêu tốn năng lượng của thiết bị hơn.

Luận án sử dụng các kỹ thuật phân tích chương trình tĩnh để xây dựng mô hình, đề xuất các thuật toán dựa trên mô hình và xây dựng các công cụ hỗ trợ cho đội ngũ phát triển phần mềm. Đối tượng nghiên cứu chính của luận án là các phần mềm chạy trên các thiết bị di động. Cụ thể là mã nguồn của các ứng dụng và sự ảnh hưởng của chúng tới việc tiêu thụ năng lượng của thiết bị di động. Đồng thời, luận án cũng quan tâm tới các mô hình tiêu thụ năng lượng, các thuật toán và ứng dụng của các mô hình này trong việc giải quyết các vấn đề về năng lượng trên thiết bị di động. Từ đó, có sự đánh giá, phân tích, tổng hợp và đề xuất những giải pháp mới.

### 1.2. Các kết quả chính của luận án

Sau quá trình giải quyết bài toán với mục tiêu, đối tượng và nội dung nghiên cứu đã đề ra, luận án có các đóng góp chính sau đây:

- (i) *Đề xuất mô hình tiêu thụ năng lượng trên các thiết bị di động dựa trên phân tích mã nguồn của phần mềm.*
- (ii) *Đề xuất phương pháp ước lượng mức độ tiêu thụ năng lượng dựa trên mô hình.*
- (iii) *Đề xuất phương pháp sinh dữ liệu kiểm thử mức độ tiêu thụ năng lượng tự động.*

(iv) *Đề xuất phương pháp giảm tải mức độ tiêu thụ năng lượng cho thiết bị di động trong mô hình điện toán đám mây di động.*

Các kết quả nghiên cứu của luận án góp phần bổ sung, hoàn thiện các mô hình, phương pháp ước lượng, kiểm thử và giảm thiểu mức độ tiêu thụ năng lượng của thiết bị di động. Bên cạnh đó, các công cụ được xây dựng từ các phương pháp đề xuất trong luận án đã hỗ trợ đội ngũ phát triển phần mềm tối ưu mã nguồn để đảm bảo các tính chất về năng lượng cho thiết bị.

### **1.3. Bố cục luận án**

Luận án bao gồm 6 chương. Trong đó, Chương 1 trình bày về lý do thực hiện đề tài, đối tượng, phạm vi, nội dung nghiên cứu, các đóng góp và cấu trúc của luận án. Chương 2 trình bày về các kiến thức nền tảng được sử dụng trong các chương tiếp theo. Chương 3 trình bày về Mô hình hóa mức độ tiêu thụ năng lượng trên các thiết bị di động dựa trên phân tích mã nguồn. Chương 4 tập trung đề xuất các phương pháp ước lượng mức độ tiêu thụ năng lượng dựa trên mô hình. Chương 5 giới thiệu một định hướng tiếp cận mới trong việc kiểm thử các tính chất năng lượng của phần mềm trên thiết bị di động. Chương 6 đề xuất một phương pháp giảm tải (tối ưu) mức tiêu thụ năng lượng của thiết bị di động khi hoạt động trong một mô hình cụ thể là mô hình điện toán đám mây di động. Cuối cùng là Chương 7 phân tích về các ưu, nhược điểm của từng phương pháp đã đề xuất và so sánh với một số phương pháp nghiên cứu liên quan và thảo luận các hướng nghiên cứu trong tương lai.

## Chương 2

# KIẾN THỨC CƠ SỞ

### 2.1. Phân tích và biểu diễn chương trình

Phân tích chương trình cho phép các nhà nghiên cứu hiểu và phát hiện một số lỗi của chương trình trong giai đoạn lập trình. Các trình biên dịch của các ngôn ngữ lập trình cũng có thể phát hiện và cung cấp các thông tin liên quan đến lỗi cú pháp của chương trình, tuy nhiên chúng chưa thể phát hiện ra các lỗi về năng lượng. Các nhà nghiên cứu về phần mềm sử dụng các kỹ thuật phân tích chương trình làm nền tảng để mở rộng khả năng phân tích đặc tính năng lượng của phần mềm. Phân tích chương trình là phương pháp phổ biến được áp dụng trong các nghiên cứu về tác động của phần mềm đối với mức độ tiêu thụ năng lượng của thiết bị.

Phương pháp phân tích chương trình tĩnh cung cấp nhiều phương pháp luận khác nhau làm nền tảng cho việc kiểm chứng, kiểm thử phần mềm. Các bộ phân tích chương trình tĩnh cho khả năng nắm bắt một cách chính xác và toàn diện mô hình của phần mềm, xác định đầy đủ các đường thực thi của chương trình mà không cần phải thực thi phần mềm. Do sự phù hợp của phương pháp đối với các yêu cầu của bài toán, luận án sử dụng phương pháp phân tích chương trình tĩnh là phương pháp chủ yếu để phát triển các kỹ thuật phân tích đặc tính năng lượng của phần mềm.

### 2.2. Một số vấn đề trong kiểm thử phần mềm

Kiểm thử phần mềm cung cấp cho các bên liên quan một cách nhìn độc lập về phần mềm từ đó cho phép đánh giá các rủi ro trong quá trình triển khai phần mềm. Quá trình kiểm thử phần mềm có thể được thực hiện trong bất kỳ pha nào của quy trình phát triển phần mềm, các phương pháp kiểm thử phần mềm được xây dựng dựa trên các tính chất, đặc trưng của mỗi pha trong quy trình phát triển phần mềm, do đó phương pháp kiểm thử bị chi phối theo một quy trình nhất định. Có nhiều các kỹ thuật kiểm thử khác nhau, mỗi một kỹ thuật kiểm thử được xây dựng để phục

vụ cho quá trình kiểm tra trong quy trình phát triển phần mềm, luận án sử dụng kỹ thuật kiểm thử cấu trúc để thực hiện xây dựng đồ thị kiểm thử và xác định các đường đi hữu ích trong đồ thị.

### **2.3. Thiết bị di động và vấn đề năng lượng**

Một trong những nội dung quan trọng của luận án là nghiên cứu sự ảnh hưởng của mã nguồn tới mức độ tiêu thụ năng lượng của thiết bị. Luận án tiến hành phân tích cấu trúc phần cứng của các thiết bị di động trong đó đặc biệt quan tâm tới vấn đề năng lượng. Luận án cũng tập trung phân tích một số vấn đề liên quan tới phần mềm, các hệ điều hành và các công cụ lập trình ứng dụng trên các thiết bị di động. Vấn đề năng lượng trên các thiết bị di động sẽ được phân tích dựa vào các nghiên cứu đã công bố, từ đó đưa ra bức tranh tổng quát nhất về các mô hình năng lượng trên các thiết bị di động dựa vào việc phân tích mã nguồn.

## Chương 3

# MÔ HÌNH HÓA MỨC ĐỘ TIÊU THỤ NĂNG LƯỢNG CHO CÁC THIẾT BỊ DI ĐỘNG DỰA TRÊN PHÂN TÍCH MÃ NGUỒN

### 3.1. Ảnh hưởng của mã nguồn tới sự tiêu hao năng lượng trên các thiết bị di động

Câu lệnh trong phần mềm đóng vai trò điều khiển hoạt động của phần cứng, làm thay đổi trạng thái hoạt động của phần cứng và làm cho thiết bị di động tiêu thụ năng lượng.

### 3.2. Mô hình hóa các trạng thái năng lượng dựa trên phân tích mã nguồn

Mỗi thành phần phần cứng độc lập sẽ tồn tại hữu hạn các mức độ thụ năng lượng, mỗi khi trạng thái hoạt động của thiết bị thay đổi thì mức độ tiêu thụ năng lượng cũng sẽ thay đổi theo. Từ đây, luận án định nghĩa các trạng thái tiêu thụ năng lượng cho một phần cứng độc lập.

**Định nghĩa 3.1** *Một trạng thái tiêu thụ năng lượng (được viết tắt là Trạng thái năng lượng) của một thành phần phần cứng là mức độ tiêu thụ năng lượng trong một đơn vị thời gian, tương ứng với mức độ hoạt động của phần cứng đó.*

Để mô hình hóa các trạng thái năng lượng cho mỗi phần cứng, luận án sử dụng các Otomat hữu hạn [?] được định nghĩa bởi 5 thành phần cơ bản:

$$A = (Q, \Sigma, \delta, q_0, F)$$

#### 3.2.1. Mô hình trạng thái năng lượng của bộ thu phát âm thanh

$$A_{Audio} = (Q_{Audio}, \Sigma_{Audio}, \delta_{Audio}, q_{0Audio}, F_{Audio})$$

Trong đó:

$$Q_{Audio} = \{off, on\}$$

$$\Sigma_{Audio} = \{ "Start()", "Stop()" \}$$

$$q_{0Audio} = "off"$$

$$F_{Audio} = Q_{Audio}$$

### 3.2.2. Mô hình trạng thái năng lượng của bộ định vị GPS

$$A_{GPS} = (Q_{GPS}, \Sigma_{GPS}, \delta_{GPS}, q_{0GPS}, F_{GPS})$$

Trong đó:

$$Q_{GPS} = \{off, idle, on\}$$

$$q_{0GPS} = "off"$$

$$F_{GPS} = \{ "off" \}$$

$$\Sigma_{GPS} = \{ "PutExtra(String, true)", "PutExtra(String, false)", "RequestLocationUpdates()" \}$$

### 3.2.3. Mô hình trạng thái năng lượng của bộ thu phát âm thanh màn hình LCD

$$A_{LCD} = (Q_{LCD}, \Sigma_{LCD}, \delta_{LCD}, q_{0LCD}, F_{LCD})$$

Trong đó:

$$Q_{LCD} = \{off, on\}$$

$$\Sigma_{LCD} = \{ "LockNow()", "Acquire()" \}$$

$$q_{0LCD} = "off"$$

$$F_{LCD} = Q_{LCD}$$

### 3.2.4. Mô hình trạng thái năng lượng của bộ thu phát 3G

$$A_{3GCellular} = (Q_{3GCellular}, \Sigma_{3GCellular}, \delta_{3GCellular}, q_{03GCellular}, F_{3GCellular})$$

Trong đó:

$$Q_{3GCellular} = \{off, idle, transmitting\}$$

$$\Sigma_{3GCellular} = \{ "SetMobileDataEnabled(true)", "SetMobileDataEnabled(false)", "Execute(httpget)" \}$$

$$q_{03GCellular} = "off"$$

$$F_{3GCellular} = Q_{3GCellular}$$

### 3.2.5. Mô hình trạng thái năng lượng của bộ thu phát Wifi

$$A_{Wifi} = (Q_{Wifi}, \Sigma_{Wifi}, \delta_{Wifi}, q_{0Wifi}, F_{Wifi})$$

Trong đó:

$$Q_{Wifi} = \{off, idle, on\}$$

$$\Sigma_{Wifi} = \{ "SetWifiEnabled(false)", "SetWifiEnabled(true)", "Execute(httpget)" \}$$

$$q_{0Wifi} = "off"$$

$$F_{Wifi} = Q_{Wifi}$$

## 3.3. Mô hình hóa trạng thái năng lượng cho ứng dụng di động

### 3.3.1. Định nghĩa trạng thái năng lượng của ứng dụng

**Định nghĩa 3.2** Một trạng thái biểu diễn mức độ tiêu thụ năng lượng của thiết bị di động tại một thời điểm, dưới sự ảnh hưởng của mã nguồn ứng dụng di động (được viết tắt là Trạng thái năng lượng của ứng dụng, kí hiệu là  $PS_{App}$ ) là sự hợp nhất của các trạng thái năng lượng của các thành phần phân cứng đơn lẻ.

$$PS_{App} = (PS_{LCD}, PS_{GPS}, PS_{Audio}, PS_{wifi}, PS_{3GCellular})$$

### 3.3.2. Sự chuyển trạng thái năng lượng của ứng dụng

### 3.3.3. Otomat biểu diễn trạng thái năng lượng của ứng dụng

Để biểu diễn mô hình năng lượng này, chúng ta sử dụng một Otomat tổng quát  $A_{App}$ :

$$A_{App} = (Q, \Sigma, \delta, q_0, F)$$

Thuật toán 3.1 mô tả quy trình sinh Otomat tổng quát từ các Otomat thành phần.

### 3.3.4. Tối ưu Otomat trạng thái năng lượng dựa trên phân tích mã nguồn

Otomat tổng quát biểu diễn các trạng thái năng lượng bao gồm 108 trạng thái, tuy nhiên trong quá trình phân tích mã nguồn một chương trình cụ thể, luận án thấy rằng một số trạng thái không xuất hiện.

---

**Thuật toán 3.1** Thuật toán hợp nhất 5 Otomat thành phần

---

**Input:**

$$A_{audio} = (Q_{audio}, \Sigma_{audio}, \delta_{audio}, q0_{audio}, F_{audio})$$

$$A_{GPS} = (Q_{GPS}, \Sigma_{GPS}, \delta_{GPS}, q0_{GPS}, F_{GPS})$$

$$A_{LCD} = (Q_{LCD}, \Sigma_{LCD}, \delta_{LCD}, q0_{LCD}, F_{LCD})$$

$$A_{cellular} = (Q_{cellular}, \Sigma_{cellular}, \delta_{cellular}, q0_{cellular}, F_{cellular})$$

$$A_{wifi} = (Q_{wifi}, \Sigma_{wifi}, \delta_{wifi}, q0_{wifi}, F_{wifi})$$

**Output:**

$$A_{App} = (Q, \Sigma, \delta, q_0, F)$$

$$1: Q = \{q | q = (q_{audio}, q_{GPS}, q_{LCD}, q_{cellular}, q_{wifi})\}$$

$$2: \Sigma = \Sigma_{audio} \cup \Sigma_{GPS} \cup \Sigma_{LCD} \cup \Sigma_{Celluar} \cup \Sigma_{wifi}$$

$$3: q_0 = (q0_{audio}, q0_{GPS}, q0_{LCD}, q0_{cellular}, q0_{wifi})$$

$$4: F = Q$$

 $\delta$  được tính theo thuật toán dưới đây:

$$5: \text{foreach } \{q_{audio} \rightarrow aq_{1\_audio}\} \in \delta_{audio} \text{ do}$$

6:  $\underline{\hspace{1cm}}$ 

$$\text{foreach } (q_{audio}, q_{GPS}, q_{LCD}, q_{cellular}, q_{wifi}) \in Q \text{ do}$$

7:  $\underline{\hspace{1cm}}$ 

$$\text{foreach } (q_{1\_audio}, q_{GPS}, q_{LCD}, q_{cellular}, q_{wifi}) \in Q \text{ do}$$

8:  $\underline{\hspace{1cm}}$ 

$$\delta = \delta \cup \{(q_{audio}, q_{GPS}, q_{LCD}, q_{cellular}, q_{wifi}) \rightarrow a(q_{1\_audio}, q_{GPS}, q_{LCD}, q_{cellular}, q_{wifi})\}$$

---

Để tối ưu các Otomat thành phần, luận án định nghĩa các trạng thái hữu ích trong Otomat tổng quát:

**Định nghĩa 3.3** Một trạng thái năng lượng  $q$  của ứng dụng  $P$  được gọi là hữu ích nếu trong Otomat tổng quát  $A_{App}$  tồn tại một tập các kí hiệu vào  $\Sigma' = \{a\}$  thỏa mãn  $q_0 \rightarrow^n aq$ .

Để giữ lại các trạng thái hữu ích và loại bỏ đi các trạng thái vô ích, luận án đề xuất thuật toán alg:optimize.

### 3.4. Công cụ trực quan hóa mức độ tiêu thụ năng lượng dựa trên phân tích mã nguồn

Để hỗ trợ cho người lập trình trực quan hóa các trạng thái năng lượng của ứng dụng trong quá trình lập trình, luận án xây dựng một bộ công cụ có tên PSA (viết tắt của Power Consumption Analyzer). Bằng cách mô phỏng các trạng thái năng lượng dưới dạng các đồ thị chuyển trạng thái, PSA cho phép người lập trình theo dõi các trạng thái năng lượng của ứng



---

**Thuật toán 3.2** Tối ưu Otomat bằng cách giữ lại các trạng thái hữu ích

---

**Input:** $A = (Q, \Sigma, \delta, q_0, F)$  |  $A$  là một Otomat thành phần $\Sigma' = \{a \mid a \text{ là một phương thức trong mã nguồn}\}$ **Output:** $A' = (Q', \Sigma', \delta', q'_0, F')$ 

```
1:  $q'_0 = q_0$ 
2:  $F' = q'_0$ 
3:  $Q' = \emptyset$ 
4:  $newQ = q_0 \cup \{q \mid \{q_0 \rightarrow aq\} \in \delta, \forall a \in \Sigma'\}$ 
5:  $\delta' = \{q_0 \rightarrow aq, \{q_0 \rightarrow aq\} \in \delta, \forall a \in \Sigma'\}$ 
6: while ( $Q' \neq newQ$ ) do
7:  $Q' = newQ$ 
8: foreach  $q_1 \in Q'$  do
9:   foreach  $a \in \Sigma'$  do
10:    if  $\{q_1 \rightarrow aq\} \in \delta$  then
11:       $newQ = newQ \cup \{q\}$ 
12:       $\delta' = \delta' \cup \{q_1 \rightarrow aq\}$ 
```

---

dụng với mã nguồn hiện tại. Từ đây, các lập trình viên có thể điều chỉnh mã nguồn và quan sát sự thay đổi trạng thái năng lượng từ việc điều chỉnh mã nguồn đó. PSA cho phép cài đặt vào các công cụ lập trình phổ biến như Android Studio và IntelliJ dưới dạng trình cài cắm (Plug-in).

Kết quả nghiên cứu của chương này đã được công bố tại Hội nghị *International Conference on Context-Aware Systems and Applications (IC-CASA 2017)*.

## Chương 4

# ƯỚC LƯỢNG MỨC ĐỘ TIÊU THỤ NĂNG LƯỢNG CỦA CÁC PHẦN MỀM DI ĐỘNG BẰNG PHƯƠNG PHÁP PHÂN TÍCH CHƯƠNG TRÌNH TÍNH

Các nghiên cứu về tính toán mức độ tiêu thụ năng lượng cho thiết bị di động chia thành hai nhóm cơ bản:

- (1) **Phương pháp ước lượng (Estimate):** Phương pháp này sử dụng các công cụ toán học để xây dựng các mô hình biểu diễn mức tiêu thụ năng lượng và sử dụng mô hình để thực hiện các thuật toán ước lượng.
- (2) **Phương pháp đo lường (Measure):** Phương pháp sử dụng các thiết bị vật lý chuyên dụng để đo lường mức tiêu hao năng lượng trên thiết bị.

Luận án này đề xuất giải pháp kết hợp giữa hai phương pháp tính toán trên để giải quyết một số vấn đề trong việc tính toán mức độ tiêu thụ năng lượng của thiết bị di động.

### 4.1. Đặc tả hình thức cho các UC

Để hỗ trợ cho những người phát triển phần mềm có thể đặc tả và thực thi UC một cách hình thức, luận án tiến hành phân nhóm các câu lệnh trong mã nguồn đồng thời đặt lại các kí hiệu nhằm đơn giản hóa việc thiết kế các UC.

### 4.2. Bộ đếm thời gian trong mô hình trạng thái năng lượng

Để lưu trữ thời gian tồn tại của mỗi trạng thái năng lượng, luận án đề xuất cải tiến Otomat trạng thái tổng quát  $A_{App}$  bằng cách bổ sung thêm

Câu lệnh hình thức	Giải thích ý nghĩa của các câu lệnh
Audio: Start	Trạng thái bộ phát âm thanh chuyển từ off sang on
Audio: Stop	Trạng thái bộ phát âm thanh chuyển từ on sang off
GPS: TurnOn	Trạng thái bộ định vị GPS chuyển từ off sang idle
GPS: TurnOff	Trạng thái bộ định vị GPS từ idle hoặc on sang off
GPS: GetLocation	Trạng thái bộ định vị GPS từ idle sang on
Display: Open	Trạng thái màn hình từ off sang on
Display: Lock	Trạng thái màn hình từ on sang off
3G: TurnOn	Trạng thái bộ thu phát 3G từ off sang on
3G: TurnOff	Trạng thái bộ thu phát 3G từ idle hoặc transmitting sang off
3G: Transfer	Trạng thái bộ thu phát 3G từ idel sang transmitting
Wifi: TurnOn	Trạng thái bộ thu phát Wifi từ off sang on
Wifi: TurnOff	Trạng thái bộ thu phát Wifi từ idle hoặc transmitting sang off
Wifi: Transfer	Trạng thái bộ thu phát Wifi từ idel sang transmitting

Bảng 4.1: Các câu lệnh hình thức trong đặc tả UC

bộ đếm thời gian Timer.

$$A_{app} = (Q, \Sigma, \delta, q_0, F, T)$$

Trong đó:  $T = Q \times \mathbb{N}$  là Timer, một ánh xạ từ tập  $Q$  sang tập số tự nhiên  $\mathbb{N}$  để ghi nhận thời gian tồn tại trạng thái trong mỗi UC.

### 4.3. Xác định hệ số tiêu thụ năng lượng tại mỗi trạng thái

Các trạng thái năng lượng của ứng dụng trong Otomat  $A_{App}$  được hợp thành từ các trạng thái thành phần:

$$PS_{App} = (PS_{LCD}, PS_{GPS}, PS_{Audio}, PS_{wifi}, PS_{3GCellular})$$

Vì vậy, muốn tính toán được mức tiêu thụ năng lượng cho mỗi trạng thái trong Otomat  $PS_{App}$  ta cần tính được mức tiêu thụ năng lượng cho mỗi trạng thái thành phần:  $PS_{LCD}, PS_{GPS}, PS_{Audio}, PS_{wifi}, PS_{3GCellular}$  và  $PS_{CPU}$ .

Gọi  $C_q$  là hệ số tiêu thụ năng lượng của ứng dụng tại trạng thái  $q$ , ta có:

$$C_q = C_{qAudio} + C_{qLCD} + C_{qGPS} + C_{q3G} + C_{qWifi} + C_{CPU}$$

Thành phần  $C$  (Coefficient) được bổ sung vào Otomat tổng quát  $A_{app}$  để xác định mức tiêu thụ năng lượng cho từng trạng thái cụ thể trong Otomat tổng quát  $A_{app}$ , khi đó:

$$A_{app} = (Q, \Sigma, \delta, q_0, F, T, C)$$

#### 4.4. Công thức ước lượng mức độ tiêu thụ năng lượng cho UC

Giả sử một UC thực hiện trong  $n$  đơn vị thời gian và chuyển qua  $k$  trạng thái, ta sẽ có công thức tính mức tiêu thụ năng lượng  $P$  như sau:

$$P = \sum_{i=1}^k C_i * t_i \quad (4.1)$$

Trong đó:  $C_i$  là hệ số tiêu thụ năng lượng tại trạng thái thứ  $i$ ,  $t_i$  là thời gian tồn tại trong trạng thái  $i$  và  $\sum_{i=1}^k t_i = n$ .

#### 4.5. Công cụ ước lượng dựa mô hình

Dựa trên mô đã đề xuất, luận án phát triển công cụ PCE (Power Consumption Estimator) hỗ trợ người lập trình ước lượng mức độ tiêu thụ năng lượng cho các UC.

Kết quả nghiên cứu của chương này đã được công bố trên Tạp chí *Mobile Networks and Applications* (2019).

## Chương 5

# KIỂM THỬ MỨC ĐỘ TIÊU THỤ NĂNG LƯỢNG CỦA CÁC PHẦN MỀM TRÊN THIẾT BỊ DI ĐỘNG

### 5.1. Giới thiệu

Kiểm thử tính chất năng lượng của chương trình có những đặc trưng khác với các loại kiểm thử các tính chất khác. Việc kiểm thử tính chất năng lượng phải được thực hiện trên thiết bị thật, thử nghiệm trong một khoảng thời gian nhất định và có sự tính toán, đo lường chính xác. Do đó, thời gian thực hiện việc kiểm thử cho mỗi Testcase lớn hơn rất nhiều so với các loại kiểm thử khác.

Đề xuất giải pháp sinh Testcase, tối ưu số lượng Testcase trong TestSuite và sắp xếp độ ưu tiên cho các Testcase là ba vấn đề chính được luận án giải quyết trong chương này.

### 5.2. Phương pháp kiểm thử mức độ tiêu thụ năng lượng của ứng dụng

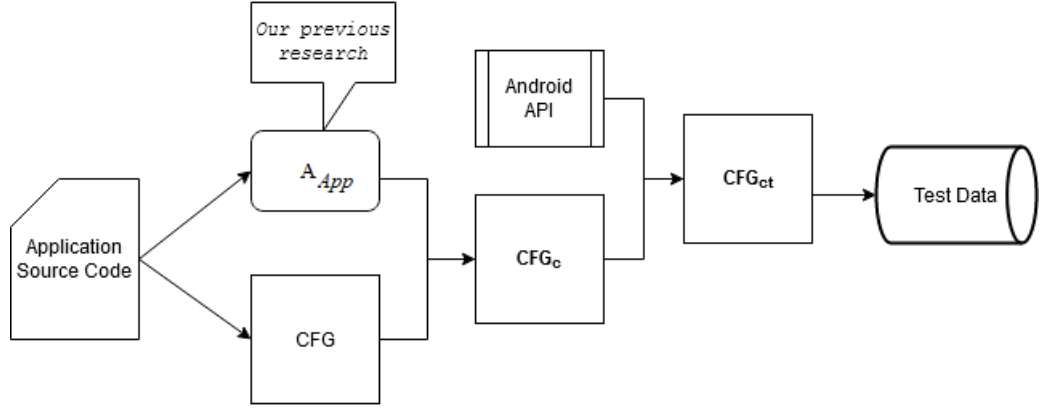
Luận án đề xuất việc kết hợp giữa Otomat năng lượng  $A_{App}$  với đồ thị  $CFG$  để tạo ra một đồ thị  $CFG_c$ , trong đó mỗi nút được tính toán một hệ số tiêu thụ năng lượng bằng cách sử dụng  $A_{App}$ . Sau đó, luận án tiến hành phân loại các câu lệnh trong chương trình, đánh dấu các câu lệnh có phụ thuộc vào thời gian và đề xuất phát triển đồ thị  $CFG_c$  thành đồ thị  $CFG_{ct}$  bằng cách bổ sung vào các nút đặc tính thời gian.

#### 5.2.1. Thiết lập mức tiêu thụ năng lượng cho CFG

Để thiết lập được mức tiêu thụ năng lượng cho mỗi nút, ta sử dụng thuật toán 5.1.

Như vậy, để tính được  $Calc(v_i)$  ta cần xác định được đường dẫn  $p$  có chứa  $v_i$ . Cụ thể:

$$p = \{v_1, v_2, \dots, v_i, \dots, v_n\}$$



Hình 5.1: Tổng quan về phương pháp sinh dữ liệu kiểm thử mức độ tiêu thụ năng lượng

Sau đó, lần lượt đưa các câu lệnh trong  $v_1, v_2, v_3, \dots, v_i$  vào Otomat  $A_{app}$  để chuyển các trạng thái tương ứng. Nếu  $c_i$  là hệ số tiêu thụ năng lượng tại trạng thái  $state_i$  của  $A_{app}$  thì  $Calc(v_i) = c_i$ .

### 5.2.2. Bổ sung đặc tính thời gian cho $CFG_c$

Các nút trên đồ thị sẽ chia làm hai loại: Phụ thuộc vào thời gian và không phụ thuộc vào thời gian. Cải tiến đồ thị  $CFG_c$  thành một đồ thị  $CFG_{ct}$ . Trong đó, nếu  $v_c$  là một nút trong  $CFG_c$  thì  $v_{ct} = \{v_c, blank\} | \{v_c, *\}$  với  $v_{ct}$  là nút trong đồ thị  $CFG_{ct}$ :

- $v_{ct} = \{v_c, *\}$  nếu nút  $v$  chứa câu lệnh phụ thuộc vào thời gian.
- $v_{ct} = \{v_c, blank\}$  nếu nút  $v$  không chứa câu lệnh phụ thuộc vào thời gian

Xây dựng đồ thị  $CFG_{ct}$  bằng thuật toán 5.2.

## 5.3. Phương pháp sinh dữ liệu kiểm thử cho các ứng dụng di động

Để có thể sinh được dữ liệu kiểm thử cho các ứng dụng, luận án phân tích các đường đi trong đồ thị  $CFG_{ct}$ , trong đó mỗi đường đi  $p = \{v_{ct1}, v_{ct2}, \dots, v_{ctm}\}$  và các dữ liệu kiểm thử được sinh ra từ tập các đường đi  $P = \{p_1, p_2, \dots, p_n\}$ .

---

**Thuật toán 5.1** Thiết lập mức độ tiêu thụ năng lượng cho các nút trong CFG

---

**Input:**

Mã nguồn chương trình

**Output:**

Đồ thị luồng điều khiển được bổ sung mức độ tiêu thụ năng lượng:  $CFG_c$

- 1: Sinh đồ thị  $CFG$  từ mã nguồn.
  - 2: Sinh tập các đường đi hoàn chỉnh và độc lập  $P$  từ  $CFG$ :  $P = \{p_1, p_2, \dots, p_n\}$
  - 3: Sinh Otomat năng lượng  $PCA_{app}$ .
  - 4: Khởi tạo  $CFG_c = CFG$  và  $v_c = \{v, null\} \forall v_c$
  - 5: **foreach** Đường đi  $p$  trong  $P$  **do**
  - 6: **foreach** Nút  $v_i$  trong  $p$  **do**
    - 7:  $c = CalC(v_i)$
    - 8:  $v_c = \{v, c\}$
- 

Luận án đề xuất giải pháp tối ưu tập các đường đi trong  $CFG_{ct}$  bằng cách loại bỏ một số đường đi ít cho khả năng ảnh hưởng tới năng lượng của thiết bị.

Giả sử  $\epsilon$  là hệ số năng lượng cho phép với mỗi nút trong  $CFG_{ct}$ , ta định nghĩa một nút có mức độ tiêu thụ năng lượng lớn như sau.

**Định nghĩa 5.1** Một nút trong đồ thị  $CFG_{ct}$  được gọi là nút có mức độ tiêu thụ năng lượng lớn nếu mức tiêu thụ năng lượng tại nút đó lớn hơn  $\epsilon$ .

Hệ số  $\epsilon$  có thể được đặt một giá trị cụ thể do người thiết kế các Testcase đặc tả chúng. Nếu hệ số  $\epsilon$  được thiết lập nhỏ hơn 0 thì mọi nút trong đồ thị  $CFG_{ct}$  đều được coi là nút có mức tiêu thụ năng lượng lớn. Trong luận án này, hệ số  $\epsilon$  được đề xuất bằng cách tính trung bình chung mức tiêu thụ năng lượng của tất cả các nút trong  $CFG_{ct}$ .

Như vậy, hệ số  $\epsilon$  được tính theo công thức dưới đây:

$$\epsilon = (c_1 + c_2 + \dots + c_n)/n$$

---

**Thuật toán 5.2** Đánh dấu các nút trên đồ thị có phụ thuộc vào thời gian

---

**Input:**

$CFG_c$

$LA$  - Danh mục các câu lệnh phụ thuộc thời gian

**Output:**

$CFG_{ct}$

```
1: Thiết lập  $CFG_{ct} = CFG_c$  và  $v_{ct} = \{v_c, blank\} \forall v_{ct}$ 
2: foreach Nút  $v_c$  trong  $CFG_c$  do
3:   Gọi  $s$  là câu lệnh được biểu diễn bởi  $v_c$ 
4:   if  $s \in LA$  then
5:      $v_{ct} = \{v_c, *\}$ 
6:   else
7:      $v_{ct} = \{v_c, blank\}$ 
```

---

### 5.3.1. Tìm kiếm các đường đi hữu ích

**Định nghĩa 5.2** Một đường đi trên đồ thị  $CFG_{ct}$  được gọi là đường đi hữu ích nếu nó bao gồm ít nhất một nút có mức tiêu thụ năng lượng lớn. Ngược lại, một đường đi gọi là đường đi vô ích nếu nó không chứa bất cứ nút nào có mức tiêu thụ năng lượng lớn.

Thuật toán 5.3 dưới đây sẽ mô tả quá trình tìm kiếm các đường đi hữu ích.

### 5.3.2. Sắp xếp mức độ ưu tiên của các đường đi

Để tính toán được độ ưu tiên, ta tính toán tổng mức tiêu thụ năng lượng trên một đường đi trong  $P_e$ . Gọi  $C_i$  là tổng mức tiêu thụ năng lượng của đường đi thứ  $i$  trong  $P_e$ , ta có:

$$C_i = \sum_{j=1}^n c_j$$

Áp dụng công thức trên cho toàn bộ các đường đi trong  $P_e$ , ta có được tập hợp các mức tiêu thụ năng lượng cho tất cả các đường đi trong  $P_e$ , đặt tên là  $C$ , cụ thể:

$$C = \{C_1, C_2, \dots, C_m\}$$



---

### Thuật toán 5.3 Tìm kiếm đường đi hữu ích

---

**Input:** $CFG_{ct}$  $\epsilon$ **Output:** $P_e$  – Tập các đường đi hữu ích

```
1: Gọi  $P$  tập các đường đi của  $CFG_{ct}$ 
2: foreach  $p$  in  $P$  do
3: foreach  $v_{ct}$  in  $p$  do
   4: Gọi  $c$  là mức tiêu thụ năng lượng tại nút  $v_{ct}$ 
   5: if  $c \geq \epsilon$  then
     6:  $P_e = P_e \cup p$ 
     7: break
```

---

Trong đó:

- $C$  là tập biểu diễn mức tiêu thụ năng lượng cho tất cả các đường đi trong  $P_e$ .
- $m$  là tổng số đường đi có trong  $P_e$ .

Như vậy mỗi đường đi trong  $P_e$  đều có một mức tiêu thụ năng lượng  $C_i$  nhất định, để sắp xếp thứ tự ưu tiên cho các đường đi trong  $P_e$  ta chỉ cần sắp xếp các đường đi theo mức tiêu thụ năng lượng  $C_i$  của mỗi đường.

#### 5.3.3. Công cụ sinh dữ liệu kiểm thử mức độ tiêu thụ năng lượng

Từ các thuật toán đã đề xuất, luận án xây dựng một công cụ cho phép tự động sinh dữ liệu phục vụ cho việc thiết kế các Testcase. Công cụ được đặt tên là PCT (Power Consumption Tester) cho phép phân tích các đường đi hữu ích đối với việc kiểm thử mức độ tiêu thụ năng lượng trong chương trình.

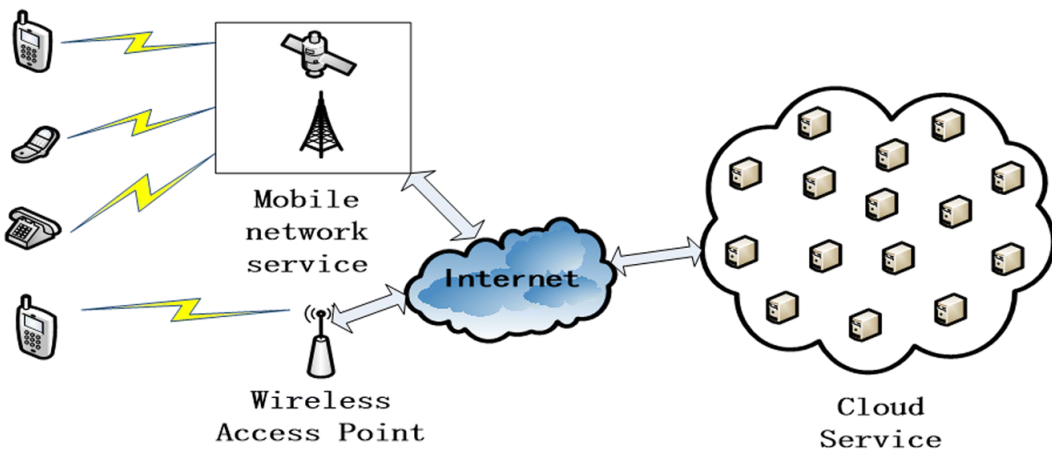
Kết quả nghiên cứu của chương này đã được công bố tại hội nghị *Nafosted Conference on Information and Computer Science (NICS - 2020)*.

## Chương 6

# GIẢM TẢI MỨC ĐỘ TIÊU THỤ NĂNG LƯỢNG CHO THIẾT BỊ DI ĐỘNG TRONG MÔ HÌNH ĐIỆN TOÁN ĐÁM MÂY DI ĐỘNG

### 6.1. Giới thiệu

#### 6.1.1. Mô hình điện toán đám mây di động



Hình 6.1: Kiến trúc của mô hình điện toán đám mây di động - MCC

Hình 6.1 mô tả kiến trúc của mô hình điện toán đám mây di động, trong đó các thiết bị di động được kết nối với các dịch vụ điện toán đám mây từ đó tận dụng sức mạnh tính toán, lưu trữ từ các dịch vụ trên đám mây.

#### 6.1.2. Bài toán giảm tải trong mô hình điện toán đám mây di động

Giảm tải trong mô hình MCC nhằm mục tiêu tiết kiệm năng lượng và tăng hiệu suất cho ứng dụng bằng cách tận dụng sức mạnh tính toán của máy chủ đám mây. Bài toán giảm tải trong mô hình MCC thường được mô tả như sau:

Values (Giá trị)	Definition (Định nghĩa/diễn giải)
$E_{mobile}^t$	Tổng năng lượng tiêu thụ khi thực thi tác vụ $t$ trên mobile
$E_{cloud}^t$	Tổng năng lượng tiêu thụ khi thực thi tác vụ $t$ trên cloud
$E_{send}^t$	Tổng năng lượng tiêu thụ khi gửi dữ liệu của tác vụ $t$ tới cloud
$E_{recieve}^t$	Tổng năng lượng tiêu thụ khi nhận dữ liệu của tác vụ $t$ từ cloud
$T_{mobile}^t$	Tổng thời gian thực hiện tác vụ $t$ trên mobile
$T_{cloud}^t$	Tổng thời gian thực hiện tác vụ $t$ cloud
$T_{send}^t$	Thời gian gửi dữ liệu tới cloud cho tác vụ $t$
$T_{recieve}^t$	Thời gian nhận dữ liệu đầu từ cloud cho tác vụ $t$
$Tasks$	Tập các tác vụ
$E_{total}$	Tổng mức năng lượng để thực thi toàn bộ tác vụ trong $Tasks$
$T_{total}$	Tổng thời gian để thực thi toàn bộ tác vụ trong $Tasks$
$f(x)$	Giá trị hàm mục tiêu khi thực thi toàn bộ các tác vụ trong $Tasks$

Bảng 6.1: Các yếu tố tác động tới quá trình ra quyết định giảm tải

Thiết bị di động cần xử lý  $n$  tác vụ độc lập, bài toán đặt ra là cần tính toán để xử lý tác vụ nào ở thiết bị di động và đưa tác vụ nào lên đám mây để xử lý?

## 6.2. Mô hình giảm tải cho các tác vụ trong mô hình điện toán đám mây di động

Trong bài toán xử lý  $n$  tác vụ trong mô hình điện toán đám mây di động, các tác vụ trên thiết bị di động (Mobile) sẽ được giảm tải bằng cách đưa lên đám mây (Cloud) để xử lý. Việc ra quyết định xử lý một tác vụ phụ thuộc vào thời gian xử lý và mức năng lượng mà thiết bị tiêu thụ.

## 6.3. Phương pháp giảm tải cho các tác vụ trong mô hình điện toán đám mây di động

Để ra quyết định giảm tải cho các tác vụ, luận án đề xuất phương pháp tính toán bao gồm các bước: Tính toán thời gian thực thi, Tính toán mức độ tiêu thụ năng lượng, Xác định hàm mục tiêu và Ra quyết định giảm tải.

Các yếu tố tác động tới quá trình ra quyết định giảm tải được mô tả trong bảng 6.1 :

### 6.3.1. Tính toán thời gian thực thi cho mỗi tác vụ

Thời gian thực thi tác vụ  $t$  trên thiết bị di động được tính bằng tổng số câu lệnh cần xử lý nhân với tốc độ xử lý của thiết bị di động, cụ thể:

$$T_{mobile}^t = I_{mobile}^t * \mu_{mobile} \quad (6.1)$$

Thời gian thực thi tác vụ trên máy chủ điện toán đám mây được tính bằng tổng thời gian của quá trình chuyển dữ liệu từ thiết bị di động tới máy chủ, thời gian xử lý tác vụ trên máy chủ và thời gian chuyển kết quả đầu ra từ máy chủ tới thiết bị di động:

$$T_{cloud}^t = D_{input}^t * S_{send} + I_{cloud}^t * \mu_{cloud} + D_{output}^t * S_{recieve} \quad (6.2)$$

### 6.3.2. Tính toán mức độ tiêu thụ năng lượng cho mỗi tác vụ

Để tính toán mức độ tiêu thụ năng lượng trong trường hợp thực thi tại thiết bị di động, ta sử dụng mô hình mô hình Otomat năng lượng kết hợp phương pháp ước lượng mức độ tiêu thụ năng lượng đã đề xuất:

$$P = \sum_{i=1}^k C_i * t_i \quad (6.3)$$

Mức độ tiêu thụ năng lượng của thiết bị di động trong trường hợp các tác vụ thực hiện trên máy chủ đám mây:

$$E_{cloud}^t = E_{send}^t + E_{recieve}^t \quad (6.4)$$

### 6.3.3. Tính toán tổng chi phí cho mỗi trường hợp thực thi tác vụ

Để có thể ra được quyết định giảm tải tổng thể, ta cần tính được chi phí năng lượng và thời gian cho mỗi trường hợp thực thi. Đối với chi phí năng lượng, ta tính tổng chi phí năng lượng của các tác vụ:

$$E_{total} = \sum_{t=1}^n \{E_{mobile}^t(label = 0) | E_{cloud}^t(label = 1)\} \quad (6.5)$$

Trong một trường hợp thực thi cụ thể, nếu gọi  $\sum T_{mobile}^t$  là tổng thời gian thực thi tất cả các tác vụ trên mobile (có nhãn là 0) và  $\sum T_{cloud}^t$  là tổng thời gian thực thi tất cả các tác vụ trên cloud (có nhãn là 1), khi đó chi phí thời gian được tính theo công thức:

$$T_{total} = \max(\sum T_{mobile}^t, \sum T_{cloud}^t) \quad (6.6)$$

#### 6.3.4. Xác định hàm mục tiêu

Như vậy, đối với mỗi trường hợp thực thi cụ thể, ta có thể tính toán được tổng chi phí về năng lượng  $E_{total}$  và tổng chi phí về thời gian  $T_{total}$ . Công thức tính toán tỷ suất lợi nhuận chi phí thời gian như sau:

$$\gamma_T = \frac{T_{total}^x - T_{total}^0}{T_{total}^0} \quad (6.7)$$

Xác định tỷ suất lợi nhuận về mặt năng lượng  $\gamma_E$  theo công thức:

$$\gamma_E = \frac{E_{total}^x - E_{total}^0}{E_{total}^0} \quad (6.8)$$

Luận án đề xuất hai hệ số ưu tiên về mặt thời gian và năng lượng lần lượt là  $\alpha$  và  $\beta$ . Hai hệ số này biểu diễn cho tỷ lệ ưu tiên về mặt thời gian và năng lượng khi đó ta có hàm mục tiêu được biểu diễn dưới dạng công thức toán học, phụ thuộc vào  $\alpha$  và  $\beta$ :

$$f(x) = \alpha * \gamma_T + \beta * \gamma_E \rightarrow Min \quad (6.9)$$

Trong đó  $f(x)$  là giá trị của hàm mục tiêu trong trường hợp thực thi  $x$  và  $\alpha + \beta = 1$ .

#### 6.3.5. Thuật toán ra quyết định giảm tải

Dưới đây là thuật toán để ra quyết định giảm tải cho thiết bị di động:

---

**Thuật toán 6.1** Thuật toán ra quyết định giảm tải cho thiết bị di động

---

**Input:**

Application Source Code  
Cloud Server Information  
Mobile Device Information  
 $\alpha$  and  $\beta$

**Output:**

$x = \{t_1, t_2, t_3, \dots, t_n\}$  where  $t_i = 0|1, i = 1..n$  and  $f(x) \rightarrow Min$

```
1:  $A = \emptyset$ 
2:  $setTimer(timer)$ 
3: while  $timer > 0$  do
4:  $A+ = Genetic()$ 
5:  $timer --$ 
6:  $minf = f(a_0)$  where  $a_0 = \{t_1, t_2, t_3, \dots, t_n\}, t_i = 0 \forall i$ 
7: foreach  $a \in A$  do
8:  $f(a) = \alpha * \gamma_T + \beta * \gamma_E$ 
9: if  $f(a) < minf$  then
10:  $minf = f(a)$ 
11:  $x = a$ 
12:  $Execute(x)$ 
```

---

# Chương 7

## KẾT LUẬN

### 7.1. Kết luận

Luận án đã đạt được một số kết quả sau đây:

- (i) *Đề xuất mô hình tiêu thụ năng lượng trên các thiết bị di động dựa trên phân tích mã nguồn của phần mềm.* Dựa trên các kỹ thuật phân tích chương trình tĩnh, luận án đã đề xuất một mô hình biểu diễn mức độ tiêu thụ năng lượng của ứng dụng dưới dạng một Otomat trạng thái.
- (ii) *Đề xuất phương pháp ước lượng mức độ tiêu thụ năng lượng dựa trên phân tích mã nguồn.* Dựa vào mô hình Otomat trạng thái năng lượng đã đề xuất, luận án tiếp đưa ra các phương pháp và các thuật toán ước lượng mức độ tiêu thụ năng lượng cho các trường hợp sử dụng.
- (iii) *Đề xuất phương pháp sinh dữ liệu kiểm thử mức độ tiêu thụ năng lượng.* Dựa trên mô hình Otomat năng lượng, luận án đã đưa ra phương pháp cải tiến đồ thị *CFG* trở thành một đồ thị mới phù hợp với bài toán kiểm thử mức độ tiêu thụ năng lượng cùng các thuật toán trên đồ thị.
- (iv) *Đề xuất phương pháp giảm tải mức độ tiêu thụ năng lượng cho thiết bị trong mô hình điện toán đám mây di động.* Luận án đề xuất một mô hình giảm tải cho tác vụ, một hàm mục tiêu thỏa mãn điều kiện năng lượng và thời gian và thuật toán để giảm tải cho các tác vụ nhằm tối ưu mức tiêu thụ năng lượng và thời gian thực thi.

### 7.2. Hướng phát triển

Nghiên cứu vẫn còn một số hướng có thể xem xét và phát triển. Luận án đề xuất một số hướng tiếp theo bao gồm:

- (i) *Hoàn thiện mô hình năng lượng phù hợp với thiết bị trong thực tế.*  
Việc nghiên cứu mô hình trạng thái của các phần cứng này một cách

đầy đủ và đưa vào Otomat tổng quát sẽ giúp cho mô hình trạng thái năng lượng phù hợp với thực tế hơn.

- (ii) *Mở rộng khả năng phân tích của Otomat trạng thái năng lượng.* Nghiên cứu tập các thư viện lập trình để mở rộng tập các kí hiệu nhằm nâng cao khả năng phân tích mã nguồn của Otomat trạng thái, tăng khả năng thích nghi của Otomat trạng thái với nhiều loại mã nguồn khác nhau.
- (iii) *Cải tiến độ chính xác của phương pháp ước lượng dựa trên mô hình.* Việc nghiên cứu cải thiện khả năng đo lường mức tiêu thụ năng lượng cho từng trạng thái của phần cứng sẽ giúp giảm thiểu các sai số trong quá trình ước lượng trên mô hình tổng quát.
- (iv) *Ước lượng mức độ tiêu thụ năng lượng đa ứng dụng.* Mở rộng nghiên cứu cho phép ước lượng mức tiêu thụ năng lượng cho thiết bị dưới sự tác động bởi nhiều ứng dụng sẽ đem tới một giải pháp thực tiễn hơn trong việc đánh giá mức tiêu thụ năng lượng của thiết bị.
- (v) *Sinh các Testcase kiểm thử mức tiêu thụ năng lượng một cách tự động.* Sử dụng phương pháp thực thi tượng trưng (Symbolic Execution) để tự động sinh các Testcase. Hướng nghiên cứu này có thể giúp cho phương pháp kiểm thử mà luận án đã đề xuất có giá trị thực tiễn hơn.
- (vi) *Cải tiến hệ số  $\epsilon$  trong kiểm thử mức tiêu thụ năng lượng.* Sử dụng học máy bằng cách học hỏi các hệ số  $\epsilon$  được thiết lập bởi người thiết kế kiểm thử, từ đó đưa ra gợi ý giá trị  $\epsilon$  một cách tự động cho các mã nguồn khác nhau.
- (vii) *Cải tiến mô hình giảm tải mức tiêu thụ năng lượng.* Cải tiến mô hình cho phép biểu diễn sự phụ thuộc các tác vụ, thay đổi công thức tính toán thời gian và cải tiến thuật toán ra quyết định giảm tải.
- (viii) *Xác định hệ số ưu tiên thời gian và năng lượng một cách tự động.* Có thể sử dụng học máy để thiết lập hai hệ số ưu tiên thời gian ( $\alpha$ ) và năng lượng ( $\beta$ ) một cách tự động.