Trường Đại Học Bách Khoa Tp. Hồ Chí Minh
Khoa Khoa Học & Kỹ Thuật Máy Tính

MÔ HÌNH HÓA TOÁN HỌC

Bài tập lớn

Hỗ trợ ra quyết định đầu tư chứng khoán

Tutors
HUYNH TUONG Nguyen
NGUYEN An Khuong

Nhóm 7 - Thứ 2 - Tiết 7/8/9 - 301H6

Hoàng Văn Nghĩa
Nguyễn Văn Đại
Phạm Tiến Anh
Trần Thị Cẩm Giang
Trần Minh Mẫn
Nguyễn Đình Ngọc Anh

Mục lục

Decision-making system for investment in stock exchange market
Hệ thống hỗ trợ ra quyết định đầu tư chứng khoán

Nguyen HUYNH TUONG, An Khuong NGUYEN
Faculty of Computer Science & Engineering,
Ho Chi Minh city University of Technology, Vietnam
268 Lý Thường Kiệt, Hồ Chí Minh, Việt Nam
{htnguyen;nakhuong}@hcmut.edu.vn

Ngày 19 tháng 4 năm 2016

Tóm tắt nội dung

Trong phần nghiên cứu này, chúng ta sẽ tìm hiểu về một bài toán đầu tư chứng khoán mà trong đó, mỗi quyết định liên quan đến việc đầu tư mua vào hay bán ra một lượng cổ phiếu trong một tháng. Mục tiêu bài toán là làm sao tìm ra quyết định mỗi tháng (mua/bán cổ phiếu hoặc gửi ngân hàng) để số tiền tổng cuối cùng của nhà đầu tư là lớn nhất. Hai mô hình toán học được đề xuất. Các đánh giá và thực nghiệm cũng được trình bày để so sánh hai mô hình đề xuất.

Keywords: mathematical model, optimization, stock investment.

1 Giới thiệu

Trong bài tập này, chúng ta quan sát một bài toán hỗ trợ ra quyết định đầu tư chứng khoán. Loại cổ phiếu và các thông tin liên quan được xác định thông qua sự hỗ trợ của các chuyên gia tư vấn. Do đó, chúng ta chỉ có thể tìm các quyết định trên một tập con cổ phiếu mà chuyên gia tư vấn mà không cần phải quan sát hết tất cả các mã trên sàn. Ngoài ra, một số thông tin chứng khoán như là giá, khối lượng cổ phiếu có khả năng giao dịch trong tháng, tháng được chia cổ tức bằng tiền, lượng cổ tức bằng tiền) cũng được dự đoán xấp xỉ (theo góp ý của chuyên gia).

Và lượng tiền mặt còn dư trong tháng (không sử dụng để mua cổ phiếu) thì chúng ta cũng có thể gửi ngân hàng để nhận thêm lãi suất. Công việc chính là làm sao tìm ra quyết định mỗi tháng: mua hoặc bán cổ phiếu, hoặc gửi ngân hàng để số tiền tổng cuối cùng của nhà đầu tư là lớn nhất.

1.1 Mô tả bài toán

Một nhà đầu tư có vốn nhàn rỗi đang tìm các quyết định để có thể thu lợi cao nhất thông qua các kênh đầu tư tín dụng.

- Lãi suất ngân hàng: α%/tháng;
- Tiền nhàn rỗi: Θ;
- Hôm nay là ngày 09/10

Một tập các cổ phiếu quan tâm có thể đầu tư được cung cấp các thông tin bao gồm:

- Mã cổ phiếu: tên viết tắt, định danh riêng biệt và duy nhất cho doanh nghiệp trong thị trường chứng khoán.
- Giá trị giao dịch.
- Khả năng giao dịch.

- Và cổ tức dự đoán.

Giả thuyết rằng:

- Không quan tâm đến sàn giao dịch của cổ phiếu và độ chênh lệch phần trăm của giao dịch;
- Các mã cổ phiếu trên có độ rủi ro thấp;
- Các mã cổ phiếu quan sát có tính thanh khoản cao, nghĩa là luôn có nhu cầu giao dịch mua và bán trên thị trường;
- Để được nhận cổ tức, cần mua cổ phiếu của mã tương ứng trước 1 tháng và bán ra sau 1 tháng. Ví dụ, cổ tức với mã X được phát ra vào tháng m_X thì cần mua vào trễ nhất là tháng $m_X - 1$ và để được nhận cổ tức, có thể bán cổ phiếu ra lại thị trường vào tháng $m_X + 1$;
- Mỗi một tháng, các giao dịch chỉ được ra quyết định một lần. Hay nói cách khác, ta có thể xem như quyết định tại một thời điểm trong bài toán này là quyết định của tháng.

Vấn đề chính yếu của bài toán là hỗ trợ ra quyết định đầu tư trong suốt τ tháng liên tục để số tiền thu vào là lớn nhất.

1.2 Nhận xét và giả thiết

Giả thiết rằng:

- Phí giao dịch được xem như là không đáng kể.
- Không xét đến việc chia cổ tức bằng cổ phiếu
- Lãi suất ngân hàng được tính theo tháng, trường hợp gửi ngân hàng nhiều tháng liên tục thì tiền sẽ được tính tăng dần theo từng tháng
- Lượng tiền bán cổ phiếu ở tháng t không được dùng để mua cổ phiếu khác ở tháng t hoặc thu lợi từ ngân hàng trong tháng t
- Cổ tức tại tháng t sẽ không được nhận nếu quyết định bán cổ phiếu vào tháng t hoặc trước đó.

Ta có nhận xét như sau:

- Tại một thời điểm, quyết định vừa mua vào và bán ra cùng một loại cổ phiểu không được xét đến.

2 Mô hình hóa bài toán

Trong phần này, chúng ta sẽ đặc tả rõ bào toán thông qua mô hình hóa lại. Để đơn giản hóa, chúng ta sẽ dời trục thời gian về 0, nghĩa là xét thời điểm đầu tiên ra quyết định là $t = 0$.

- n: số lượng cổ phiếu,
- C_j: giá đơn vị của cổ phiếu j $(j = 1, 2, \dots, n)$
- τ: số tháng cần ra quyết định,
- $D_{j,t}$: cổ tức được chia trong tháng t $(t = 0, 1, \dots, \tau)$ ứng với cổ phiếu j $(j = 1, 2, \dots, n)$,
- Q_j: khối lượng cổ phiếu j giao địch tối đa trong 1 tháng $(j = 1, 2, \dots, n)$,
- α: % lãi suất ngân hàng mỗi tháng,
- θ: tiền nhàn rỗi đầu cơ ban đầu,
- INF: một giá trị cực lớn.

2.1 Constraints

Các ràng buộc cứng (hard-constraints) trong bài toán:

Ràng buộc H1. Ràng buộc về kiểu và giới hạn của các biến

Tại thời điểm bắt đầu $t = 0$:

Ràng buộc H2. không có cổ phiếu j nào được bán ra.

Ràng buộc H3. lượng tiền có thể giao dịch chính là lượng tiền đầu tư ban đầu.

Tại tháng thứ t:

Ràng buộc H4. số lượng cổ phiếu j bán ra không thể vượt quá số lượng cổ phiếu j đang có

Ràng buộc H5. giá trị cổ phiếu mua vào không thể vượt quá lượng tiền mặt có thể giao dịch

Ràng buộc H6. khối lượng cổ phiếu j bán ra hoặc mua vào không thể vượt quá khối lượng giao dịch tối đa tương ứng với cổ phiếu j đang có

Ràng buộc H7. không thể có quyết định vừa mua vừa bán 1 loại cổ phiếu

Ràng buộc H8. lượng cổ phiếu mua (hoặc bán) chỉ có nếu có quyết định mua (hoặc bán)

Ràng buộc H9. lượng tiền mặt hàng tháng phải không âm.

2.2 Objective function

Mục tiêu là tối đa hóa lượng tiền mặt cuối cùng thu được.

2.3 Mô hình 1

2.3.1 Decision variables

Ta đặt các biến ra quyết định của bài toán như sau:

- $x_{i,j,t}$: biến nhị phân biểu diễn quyết định mua ($i = 0$) hoặc bán ($i = 1$) cổ phiếu thứ j tại tháng thứ t (tính từ thời điểm bắt đầu ra quyết định)
- $y_{i,j,t}$: biến nguyên biểu diễn quyết định số cổ phiếu j được mua vào ($i = 0$) hoặc bán ra ($i = 1$) tại tháng thứ t (tính từ thời điểm bắt đầu ra quyết định)

2.3.2 Intermediate variables

Các biến trung gian sử dụng trong mô hình:

- $q_{j,t}$: Số lượng cổ phiếu j đang đầu tư ở tháng thứ t ($t = 0, 1, \ldots, \tau$)

$$q_{j,t} = \sum_{m=0}^{t} y_{0,j,m} - \sum_{m=0}^{t} y_{1,j,m}$$

- θ_t: Lượng tiền có thể giao dịch ở tháng thứ t ($t = 0, 1, \ldots, \tau$)
  Lượng tiền này tính ở thời điểm đầu mỗi tháng, sau khi đã nhận tiền chia cổ tức, bán cổ phiếu từ tháng trước và trước khi thực hiện các giao dịch của tháng này.

$$\theta_0 = \theta$$

$$\theta_1 = [\theta_0 - \sum_{j=1}^{n} y_{0,j,0} C_j](\alpha + 1) = [\theta - \sum_{j=1}^{n} y_{0,j,0} C_j](\alpha + 1)$$

$$\theta_2 = [\theta_1 - \sum_{j=1}^{n} y_{0,j,1} C_j](\alpha + 1) + \sum_{j=1}^{n} y_{1,j,1} C_j + \sum_{j=1}^{n} (y_{0,j,0} - y_{1,j,1}) 10 D_{j,1}$$

$$= \theta(\alpha+1)^2 - \sum_{j=1}^{n} y_{0,j,0} C_j (\alpha+1)^2 - \sum_{j=1}^{n} y_{0,j,1} C_j (\alpha+1) + \sum_{j=1}^{n} y_{1,j,1} C_j + \sum_{j=1}^{n} (y_{0,j,0} - y_{1,j,1}) 10 D_{j,1}$$

Do đó:

$$\theta_t = \theta(\alpha+1)^t - \sum_{j=1}^{n}\sum_{m=0}^{t-1} y_{0,j,m}C_j(\alpha+1)^{t-m} + \sum_{j=1}^{n}\sum_{m=1}^{t-1} y_{1,j,m}C_j(\alpha+1)^{t-m-1} + \sum_{j=1}^{n}\sum_{m=1}^{t-1}(q_{j,m-1} - y_{1,j,m})D_{j,m}.10(\alpha+1)^{t-m-1}$$

Thay $q_{j,t}$ vào biểu thức trên ta được biểu thức θ_t theo $y_{i,j,t}$:

$$\theta_t = \theta(\alpha+1)^t - \sum_{j=1}^{n}\sum_{m=0}^{t-1} y_{0,j,m}C_j(\alpha+1)^{t-m} + \sum_{j=1}^{n}\sum_{m=1}^{t-1} y_{1,j,m}C_j(\alpha+1)^{t-m-1} + \sum_{j=1}^{n}\sum_{m=1}^{t-1}\left(\sum_{p=0}^{m-1} y_{0,j,p} - \sum_{p=0}^{m-1} y_{1,j,p} - y_{1,j,m}\right)D_{j,m}.10(\alpha+1)^{t-m-1}$$

Trong công thức trên, cổ tức của cổ phiếu j được chia được ở tháng t (nhận được ở tháng $t+1$) được tính bằng :

$(q_{j,t-1} - y_{1,j,t})D_{j,t}.10$

Điều đó đảm bảo rằng, cổ phiếu j muốn nhận cổ tức ở tháng t thì phải mua chậm nhất vào tháng $t-1$ và bán ra sớm nhất vào tháng $t+1$.

2.3.3 Constraints

(A) Tại thời điểm bắt đầu $t = 0$:

Ràng buộc 1 *Không có cổ phiếu j nào được bán ra.*

$$x_{1,j,0} = 0 \quad \forall j \in [1, 2, ..., n]$$

Ràng buộc 2 *Lượng tiền có thể giao dịch chính là lượng tiền đầu tư ban đầu.*

$$\theta_0 = \theta$$

(B) Tại tháng thứ t:

Ràng buộc 3 *Số lượng cổ phiếu j bán ra không thể vượt quá số lượng cổ phiếu j đang có (tháng 0 được đảm bảo từ ràng buộc 1).*

$$y_{1,j,t} - \sum_{n=0}^{t-1} y_{0,j,n} + \sum_{n=0}^{t-1} y_{1,j,n} \leq 0 \quad \forall j \in [1, 2, ..., n],\ t \in [1, 2, ..., \tau]$$

Ràng buộc 4 *Giá trị cổ phiếu mua vào không thể vượt quá lượng tiền mặt có thể giao dịch.*

$$\sum_{j=1}^{n} y_{0,j,t}C_j \leq \theta_t \quad \forall t \in [0, 1, ..., \tau]$$

Ràng buộc 5 *Khối lượng cổ phiếu j bán ra hoặc mua vào không thể vượt quá khối lượng giao dịch tối đa tương ứng với cổ phiếu j đang có.*

$$y_{i,j,t} \leq Q_j \quad \forall j \in [1, 2, ..., n],\ i \in [0, 1],\ t \in [0, 1, ..., \tau]$$

Ràng buộc 6 *Không thể có quyết định vừa mua vừa bán 1 loại cổ phiếu (t chạy từ 1 do tháng 0 được đảm bảo từ ràng buộc 1).*

$$x_{0,j,t} + x_{1,j,t} \leq 1 \quad \forall j \in [1, 2, ..., n],\ t \in [1, 2, ..., \tau]$$

Ràng buộc 7 *Lượng cổ phiếu mua (hoặc bán) chỉ có nếu có quyết định mua (hoặc bán). Bất đẳng thức dưới đây cũng đảm bảo ràng buộc 5.*

$$y_{i,j,t} \leq x_{i,j,t}Q_j \quad \forall i \in [0, 1],\ j \in [1, 2, ..., n],\ t \in [0, 1, ..., \tau]$$

Ràng buộc 8 *Lượng tiền mặt hàng tháng phải không âm:*

Nếu lượng tiền mặt hàng tháng nếu tính từ đầu tháng - trước khi thực hiện các giao dịch của tháng đó. Chẳng hạn đầu tháng t có 10000 tiền mặt, lượng tiền mặt tháng t như thế không âm. Sau đó tháng t giao dịch hết 11000 tiền mặt, lượng tiền mặt tháng t lúc này là -1000, nhưng điều kiện tiền mặt tháng t đã xét trước khi giao dịch (10000 không âm). Tiếp theo, sang tháng t + 1, chẳng hạn được nhận 2000 tiền chia cổ tức từ cổ phiếu nào đó, lượng tiền mặt tháng t + 1 lúc này lại là 1000 - không âm.

Do đó, ràng buộc tiền mặt tháng t không âm phải xét sau khi tất cả các giao dịch tháng t đã hoàn thành. Khi đó ràng buộc này được đảm bảo từ ràng buộc 4 - nếu tháng t - 1 giao dịch không quá lượng tiền mặt thì tại tháng t, lượng tiền mặt sẽ không âm.

Cuối cùng, cả hai ràng buộc 4 và 8 được đảm bảo bằng duy nhất 1 biểu thức :

$$\sum_{j=1}^{n} y_{0,j,t}C_j \leq \theta_t \quad \forall t \in [0, 1, 2, ...\tau]$$

(C) Ràng buộc về kiểu và giới hạn của các biến

Ràng buộc 9 x_{ijt} *là biến nhị phân.*

$$\forall i \in [0, 1],\ j \in [1, 2, ..., n],\ t \in [0, 1, ...\tau]. \text{ Ta có: } 0 \leq x_{i,j,t} \leq 1,\ x_{i,j,t} \in Z$$

Ràng buộc 10 y_{ijt} *nguyên và có giá trị không âm*

$$\forall i \in [0, 1],\ j \in [1, 2, ..., n],\ t \in [0, 1, ...\tau]. \text{ Ta có: } 0 \leq y_{i,j,t},\ y_{i,j,t} \in Z$$

2.3.4 Objective function

Ta cần tối ưu lượng tiền mặt thu được, chính là lượng tiền mặt có thể giao dịch ở tháng $t + 1$.

$$z = \theta_{t+1}$$

2.4 Mô hình 2

Nhận xét: Với mô hình chỉ có 1 biến $y_{j,t}$ biểu diễn cho cả lượng cổ phiếu mua hoặc bán như dưới đây, ta không thể mô hình hoá bài toán một cách tuyến tính. Để biểu diễn lượng cổ phiếu mua vào hay bán ra phải thông qua biến trung gian $z = x_{j,t}y_{j,t}$, ngoài ra do tính chất "cổ phiếu mua vào trừ tiền ngay, còn cổ phiếu bán ra và cổ tức thì nhận được tiền chậm 1 tháng" nên ngay cả thông qua biến trung gian z thì việc tính lãi xuất ngân hàng khác nhau cho những số hạng này trong công thức θ_t là không thể tuyến tính.

2.4.1 Decision variables

Ta đặt các biến ra quyết định của bài toán như sau:

- $x_{j,t}$: biến nguyên biểu diễn quyết định mua hoặc bán cổ phiếu thứ j tại tháng thứ t (tính từ thời điểm bắt đầu ra quyết định) - có giá trị $= 0$ nếu quyết định không mua và không bán; $= 1$ nếu quyết định mua và $= -1$ nếu quyết định bán.
- $y_{j,t}$: biến nguyên biểu diễn quyết định số lượng cổ phiếu j được mua vào hoặc bán ra tại tháng thứ t (tính từ thời điểm bắt đầu ra quyết định)

2.4.2 Intermediate variables

Các biến trung gian sử dụng trong mô hình:

- θ_t: lượng tiền có thể giao dịch ở tháng thứ t ($t = 0, 1, \ldots, \tau$)
- $q_{j,t}$: số lượng cổ phiếu j đang đầu tư ở tháng thứ t ($t = 0, 1, \ldots, \tau$)

2.4.3 Constraints

(A) Tại thời điểm bắt đầu $t = 0$:

Ràng buộc 11 *không có cổ phiếu j nào được bán ra.*

Ràng buộc 12 *lượng tiền có thể giao dịch chính là lượng tiền đầu tư ban đầu.*

(B) Tại tháng thứ t:

Ràng buộc 13 *số lượng cổ phiếu j bán ra không thể vượt quá số lượng cổ phiếu j đang có.*

Ràng buộc 14 *giá trị cổ phiếu mua vào không thể vượt quá lượng tiền mặt có thể giao dịch.*

Ràng buộc 15 *khối lượng cổ phiếu j bán ra hoặc mua vào không thể vượt quá khối lượng giao dịch tối đa tương ứng với cổ phiếu j đang có.*

Ràng buộc 16 *lượng cổ phiếu mua (hoặc bán) chỉ có nếu có quyết định mua (hoặc bán).*

Ràng buộc 17 *lượng tiền mặt hàng tháng phải không âm:*

(C) Ràng buộc về kiểu và giới hạn của các biến

Ràng buộc 18 *x_{ijt} là biến nguyên có giá trị thuộc tập $\{-1, 0, 1\}$.*

Ràng buộc 19 *y_{ijt} có giá trị không âm.*

2.4.4 Objective function

3 Định dạng nhập - xuất và giải pháp đề xuất

3.1 Ví dụ 1

3.1.1 Dữ liệu đầu vào

Xét trường hợp chúng ta có thông tin như sau:

- Lãi suất ngân hàng: 0.45%/tháng;
- Tiền nhàn rỗi: 100,000;
- Hôm nay là ngày 09/10

Một số cổ phiếu quan tâm có thể đầu tư:

Mã cổ phiếu	Giá trị giao dịch	Khả năng giao dịch	Cổ tức dự đoán
DPM	32	800	5(15%) + 9 (15%)
MCP	13.5	5,000	5(6%) + 11(9%)
REE	25.5	2,000	2(16%)
TDC	9.6	10,000	4(5%)
SRC	25.7	1,000	6(15%)
KHP	12.3	20,000	7(9%) + 11 (5%)
CSM	30.4	15,000	7(25%)

Mẫu ví dụ số (input) tương ứng được định nghĩa trong một tập tin nhập theo dạng như trong Hình 1.

```
1  10  // tháng hiện tại
2  100000  // số tiền đầu cơ
3  0.45    // lãi suất ngân hàng/tháng
4  12  // khoảng thời gian đầu tư tính từ tháng hiện tại
5
6  DPM 32 800 5(15) + 9(15)
7  MCP 13.5 5000 5(6) + 11(9)
8  REE 25.5 2000 2(16)
9  TDC 9.6 10000 4(5)
10 SRC 25.7 1000 6(15)
11 KHP 12.3 20000 7(9) + 11 (5)
12 CSM 30.4 15000 7(25)
13
```

Hình 1: Mẫu ví dụ về định dạng thông tin ngõ nhập

Trong Hình 1, dòng đầu tiên cho biết tháng hiện tại; dòng thứ hai cung cấp thông tin về lượng tiền mặt nhàn rỗi có thể dùng để đầu tư; dòng 3 cho biết phần trăm lãi suất ngân hàng thu được nếu gửi hàng tháng; dòng 4 cho biết khoảng thời gian không cần dùng lượng tiền nhàn rỗi và có thể dùng đầu tư.

Các dòng từ dòng 6 về sau cho biết danh mục chứng khoán có thể đầu tư. Mỗi hàng này chứa thông tin tương ứng với một loại cổ phiếu, bao gồm: mã cổ phiếu, giá trị giao dịch của mỗi cổ phiếu (đơn vị là nghìn đồng), khối lượng giao dịch tối đa trong 1 tháng, tháng chia cổ tức bằng tiền trong năm và giá trị cổ tức được nhận trên mỗi cổ phiếu.

Ví dụ, dòng 6 cho biết thông tin về cổ phiếu có mã DPM: giá trị giao dịch là 32 nghìn đồng/cổ phiếu; số lượng cổ phiếu tối đa có thể giao dịch (cả mua lẫn bán) trong một tháng ước lượng khoảng 800 phiếu; trong một năm có 2 lần chia cổ tức bằng tiền: tháng 5 (15 %) và tháng 9 (15%).

3.1.2 Xuất kết quả

Từ dữ liệu đầu vào được đặc tả như Hình 1, nếu quyết định là không đầu tư gì cả, chỉ gửi ngân hàng thì kết quả sẽ thu được như Hình 2.

Mỗi dòng trong các dòng từ 4-16 ghi nhận thông tin cho từng tháng, lượng tiền mặt có được.

Một kết quả đề xuất khác có thể được xác định bằng cách ra quyết định mua hoặc bán một số lượng cổ phiếu theo từng tháng. Ví dụ, ta có thể quyết định mua 1000 cổ phiếu MCP tại tháng đầu tiên (tháng 10) và lần lượt mua 800 cổ phiếu DPM trong các tháng 2,3, và 4 của năm sau. Sau đó, vào tháng 6 của năm sau, ta lại bán 800 cổ phiếu DPM và 1000 cổ phiếu MCP. Tương tự, vào tháng 8 và tháng 10 của năm sau, ta lần lượt bán 800 cổ phiếu DPM.

Kết quả thu nhận được trình bày trong Hình 3.

Lúc này, mỗi dòng trong các dòng từ 4-16 ghi nhận thông tin cho từng tháng, bao gồm lượng tiền mặt có được và quyết định mua/bán cổ phiếu tại tháng đó.

Chi tiết tính toán lượng tiền mặt như sau:

- Tháng 10 (dòng 4): tiền nhàn rỗi ban đầu là 100 000. Mua 1 000 cổ phiếu MCP tương ứng với số tiền là 13 500 nghìn đồng, ta còn lại: 86 500 nghìn đồng.
- Tháng 11 (dòng 5): từ số tiền tháng trước: 86 500 nghìn đồng, ta gửi ngân hàng và sau một tháng, lãi gộp vốn là 86 889 nghìn đồng.
- Tháng 12 (dòng 6): lãi gộp vốn là 87 280 nghìn đồng khi lấy tất cả số tiền tháng trước đi gửi ngân hàng, gộp thêm với cổ tức của MCP (0,9%), ta thu được 88 180 nghìn đồng.
- Tháng 1 (dòng 7): tương tự, gửi ngân hàng hết, ta thu được 88 577 nghìn đồng.
- Tháng 2 (dòng 8): tiếp tục gửi ngân hàng, ta thu được 88975 nghìn đồng. Và để mua vào 800 cổ phiếu DPM, cần số tiền là 25 600 nghìn đồng, phần còn lại gửi ngân hàng là 63 375 nghìn đồng.
- Tháng 3 (dòng 9): từ việc lãi gộp vốn, ta có 63 660 nghìn đồng. Mua 800 cổ phiếu DPM, còn lại 38 060 nghìn đồng để gửi ngân hàng.

```
1000s // thời gian tính toán
106010

T10 M0 100000
T11 M1 100450
T12 M2 100902
T01 M3 101356
T02 M4 101812
T03 M5 102270
T04 M6 102730
T05 M7 103192
T06 M8 103657
T07 M9 104123
T08 M10 104592
T09 M11 105062
T10 M12 105535
```

Hình 2: Quyết định không mua bán gì cả

```
1  1000s // thời gian tính toán
2  109891 // số tiền thu được
3
4  T10 M0 100000 1000 MCP
5  T11 M1 86889
6  T12 M2 88180
7  T01 M3 88577
8  T02 M4 88975 800 DPM
9  T03 M5 63660 800 DPM
10 T04 M6 38232 800 DPM
11 T05 M7 12688
12 T06 M8 16946 -800 DPM -1000 MCP
13 T07 M9 56122 -800 DPM
14 T08 M10 81974
15 T09 M11 82343
16 T10 M12 83914 -800 DPM
17
```

Hình 3: Mẫu ví dụ về định dạng thông tin ngõ xuất

- Tháng 4 (dòng 10): tiếp tục việc lãi gộp vốn, ta được 38 232 nghìn đồng. Và tiếp tục mua 800 cổ phiếu DPM, phần còn lại gửi ngân hàng là 12 632 nghìn đồng.
- Tháng 5 (dòng 11): nhờ lãi gộp vốn, ta được 12 688 nghìn đồng, tiếp tục để ở ngân hàng. Và cuối tháng này ta nhận được cổ tức từ 2400 cổ phiếu DPM và 1000 cổ phiếu MCP.
- Tháng 6 (dòng 12): đầu tháng, ta thu được 16 946 nghìn đồng (lãi gộp vốn tháng trước và nhận cổ tức bằng tiền của DPM và MCP). Ta lại bán ra800 cổ phiếu DPM và 1000 cổ phiếu MCP.
- Tháng 7 (dòng 13): cuối cùng thu được 56 122 nghìn đồng (17022 nghìn đồng từ lãi và vốn + 32x800 + 13.5x 1000). Ta lại tiếp tục bán 800 cổ phiếu DPM.
- Tháng 8 (dòng 14): nhờ đó, ta thu được vào đầu tháng: 81 974 nghìn đồng. Số tiền này được gửi ngân hàng trong cả tháng.
- Tháng 9 (dòng 15): lãi gộp vốn là 82 343 nghìn đồng. Tháng này, ta lại được nhận cổ tức của DPM.
- Tháng 10 (dòng 16): Kết quả là 83 914 nghìn đồng và ta lại bán 800 cổ phiếu DPM còn lại.
- Cuối tháng, thu lãi gộp vốn từ ngân hàng, số tiền thu được là 109 891 nghìn đồng.

Với quyết định này (mặc dù tương đối ngẫu nhiên), nhưng ta cũng có thể thu được lợi nhuận hơn 50% so với quyết định không đầu tư chứng khoán mà chỉ để ngân hàng.

Để kiểm chứng kết quả, các bạn có thể tham khảo đoạn code trong "output.cpp" hoặc "compute.cpp".

3.2 Giải pháp đề xuất - Mô hình 3

3.2.1 Decision variables

Ta đặt các biến ra quyết định của bài toán như sau:

- $x_{j,t}$: biến nhị phân biểu diễn quyết định mua hoặc bán cổ phiếu thứ j tại tháng thứ t (tính từ thời điểm bắt đầu ra quyết định) - có giá trị = 0 nếu quyết định mua và = 1 nếu quyết định bán. Trong trường hợp không mua không bán biến x có thể có giá trị 0 tương ứng với "không bán ra và mua vào là 0", hoặc x có thể có giá trị 1 tương ứng với "không mua vào, và bán ra là 0". Như vậy, trong trường hợp đó, ta không cần quan tâm tới giá trị biến x.
- $y_{i,j,t}$: biến nguyên biểu diễn quyết định số lượng cổ phiếu j được mua vào (i = 0) hoặc bán ra (i = 1) tại tháng thứ t (tính từ thời điểm bắt đầu ra quyết định)

3.2.2 Intermediate variables

Các biến trung gian sử dụng trong mô hình:

- $q_{j,t}$: Số lượng cổ phiếu j đang đầu tư ở tháng thứ t $(t = 0, 1, \ldots, \tau)$

$$q_{j,t} = \sum_{m=0}^{t} y_{0,j,m} - \sum_{m=0}^{t} y_{1,j,m}$$

- θ_t: Lượng tiền có thể giao dịch ở tháng thứ t $(t = 0, 1, \ldots, \tau)$

  Lượng tiền này tính ở thời điểm đầu mỗi tháng, sau khi đã nhận tiền chia cổ tức, bán cổ phiếu từ tháng trước và trước khi thực hiện các giao dịch của tháng này. Tương tự mô hình 1 :

$$\theta_t = \theta(\alpha+1)^t - \sum_{j=1}^{n}\sum_{m=0}^{t-1} y_{0,j,m}C_j(\alpha+1)^{t-m} + \sum_{j=1}^{n}\sum_{m=1}^{t-1} y_{1,j,m}C_j(\alpha+1)^{t-m-1} + \sum_{j=1}^{n}\sum_{m=1}^{t-1}\Big(\sum_{p=0}^{m-1} y_{0,j,p} - \sum_{p=0}^{m-1} y_{1,j,p} - y_{1,j,m}\Big)D_{j,m}.10(\alpha+1)^{t-m-1}$$

3.2.3 Constraints

(A) Tại thời điểm bắt đầu $t = 0$:

Ràng buộc 20 *Không có cổ phiếu j nào được bán ra.*

$$x_{j,0} = 0 \quad \forall j \in [1, 2, ..., n]$$

(B) Tại tháng thứ t:

Ràng buộc 21 *Số lượng cổ phiếu j bán ra không thể vượt quá số lượng cổ phiếu j đang có.*

$$y_{1,j,t} - \sum_{n=0}^{t-1} y_{0,j,n} + \sum_{n=0}^{t-1} y_{1,j,n} \leq 0 \quad \forall j \in [1, 2, ..., n],\ t \in [1, 2, ..., \tau]$$

Ràng buộc 22 *Giá trị cổ phiếu mua vào không thể vượt quá lượng tiền mặt có thể giao dịch (ràng buộc này cũng đảm bảo tiền mặt các tháng đều không âm).*

$$\sum_{j=1}^{n} y_{0,j,t} C_j \leq \theta_t \quad \forall t \in [0, 1, ..., \tau]$$

Ràng buộc 23 *Lượng cổ phiếu mua (hoặc bán) chỉ có nếu có quyết định mua (hoặc bán) và lượng cổ phiếu j bán ra hoặc mua vào không vượt quá khối lượng giao dịch tối đa tương ứng với cổ phiếu j đang có.*

$$y_{1,j,t} \leq x_{j,t} Q_j \quad \forall j \in [1, 2, ..., n],\ t \in [1, 2, ..., \tau]$$
$$y_{0,j,t} \leq (1 - x_{j,t}) Q_j \quad \forall j \in [1, 2, ..., n],\ t \in [1, 2, ..., \tau]$$

(C) Ràng buộc về kiểu và giới hạn của các biến

Ràng buộc 24 *$x_{j,t}$ là biến nhị phân.*

$$0 \leq x_{j,t} \leq 1,\ x \in Z \quad \forall j \in [1, 2, ..., n], t \in [0, 1, ..., \tau]$$

Ràng buộc 25 *$y_{i,j,t}$ nguyên và có giá trị không âm.*

$$y_{i,j,t} \geq 0,\ y_{i,j,t} \in Z \quad \forall i \in [0, 1],\ j \in [1, 2, ..., n],\ t \in [0, 1, ..., \tau]$$

3.2.4 Objective function

$$z = \theta_{t+1}$$

3.3 Giải pháp đề xuất - Mô hình 4

Trước tiên, nếu ta gọi bài toán mà chúng ta đang xét là bài toán A. Xét bài toán B là bài toán A bỏ đi giả thiết "không có quyết định vừa mua vừa bán 1 loại cổ phiếu trong 1 tháng", bài toán B có thể coi là bài toán A nới lỏng điều kiện. Ta sẽ chứng minh rằng 2 bài toán này có cùng đáp số và thiết lập 1 mô hình cho bài toán B, mục đích của việc làm này sẽ được giải thích khi so sánh các mô hình.

Ở bài toán B, giả sử tồn tại 1 đầu vào mà trong đáp án của nó tồn tại $y_{1,j,t} > 0$ và $y_{0,j,t} > 0$ (nghĩa là tồn tại 1 tháng t mà ở đó, cổ phiếu j vừa được mua, vừa được bán). Ta gọi phương án giao dịch cổ phiếu j này là P. Với mỗi trường hợp xét sau đây, ta chỉ ra phương án Q tương ứng thoả mãn *"phương án Q không ảnh hưởng tới việc giao dịch các cổ phiếu khác, phương án Q cho cùng số lượng cổ phiếu j so với P, và Q mang lại lợi nhuận tốt hơn P"*:

- Nếu $y_{1,j,t} = y_{0,j,t}$: Phương án Q là $y_{1,j,t} = y_{0,j,t} = 0$, gửi ngân hàng lượng tiền mà P đã dùng mua cổ phiếu j.
- Nếu $y_{1,j,t} - y_{0,j,t} = \lambda > 0$: Phương án Q là bán ra λ cổ phiếu j, gửi ngân hàng lượng tiền mà P dùng mua cổ phiếu j.
- Nếu $y_{0,j,t} - y_{1,j,t} = \lambda > 0$: Phương án Q là mua vào λ cổ phiếu j, gửi ngân hàng lượng tiền $y_{1,j,t}C_j$.

Trong tất cả các trường hợp, ta chứng minh được phương án P không tối ưu. Do đó, với tất cả các đầu vào, bài toán B không cho một kết quả nào mà ở đó, tồn tại 1 quyết định vừa mua vừa bán 1 loại cổ phiếu trong 1 tháng. Nói cách khác, với cùng một đầu vào thì bài toán B và bài toán A cho kết quả như nhau. Mô hình sau đây mô tả bài toán B, và có thể dùng nó để giải bài toán A.

3.3.1 Decision variables

Ta đặt các biến ra quyết định của bài toán như sau:

- $y_{i,j,t}$: biến nguyên biểu diễn quyết định số lượng cổ phiếu j được mua vào (i = 0) hoặc bán ra (i = 1) tại tháng thứ t (tính từ thời điểm bắt đầu ra quyết định)

3.3.2 Intermediate variables

Các biến trung gian sử dụng trong mô hình:

- θ_t: Lượng tiền có thể giao dịch ở tháng thứ t $(t = 0, 1, \dots, \tau)$ - tương tự mô hình 1 và 3.

3.3.3 Constraints

(A) Tại thời điểm bắt đầu $t = 0$:

Ràng buộc 26 *Không có cổ phiếu j nào được bán ra.*

$$y_{1,j,0} = 0 \quad \forall j \in [1, 2, ..., n]$$

(B) Tại tháng thứ t:

Ràng buộc 27 *Số lượng cổ phiếu j bán ra không thể vượt quá số lượng cổ phiếu j đang có.*

$$y_{1,j,t} - \sum_{n=0}^{t-1} y_{0,j,n} + \sum_{n=0}^{t-1} y_{1,j,n} \leq 0 \quad \forall j \in [1, 2, ..., n],\ t \in [1, 2, ..., \tau]$$

Ràng buộc 28 *Giá trị cổ phiếu mua vào không thể vượt quá lượng tiền mặt có thể giao dịch (ràng buộc này cũng đảm bảo tiền mặt các tháng đều không âm).*

$$\sum_{j=1}^{n} y_{0,j,t}C_j \leq \theta_t \quad \forall t \in [0, 1, ..., \tau]$$

Ràng buộc 29 *Lượng cổ phiếu mua (hoặc bán) không vượt quá khối lượng giao dịch tối đa tương ứng với cổ phiếu j.*

$$y_{i,j,t} \leq Q_j \quad \forall i \in [0, 1],\ j \in [1, 2, ..., n],\ t \in [1, 2, ..., \tau]$$

(C) Ràng buộc về kiểu và giới hạn của các biến

Ràng buộc 30 $y_{i,j,t}$ *nguyên và có giá trị không âm.*

$$y_{i,j,t} \geq 0,\ y_{i,j,t} \in Z \quad \forall i \in [0, 1],\ j \in [1, 2, ..., n],\ t \in [0, 1, ..., \tau]$$

3.3.4 Objective function

$$z = \theta_{t+1}$$

4 Đánh giá kết quả

4.1 Solver sử dụng

Mô hình 1 và mô hình 3 được thực hiện trên 3 solver GLPK, SimplexSolver và GurobiSolver-plugin nằm trong gói Microsoft Solver Foundation. Trong đó 2 solver GLPK và SimplexSolver (nằm trong bản Solver Foundation Academic) sử dụng được với mô hình lớn, phức tạp. Plugin GurobiSolver giới hạn số ràng buộc và số biến mỗi loại ở mức 500.

- **GLPK v4.59 C++**: Các hàm được sử dụng bao gồm :
  - *glp_prob *glp_create_prob(void)* : Trả về biến có kiểu glp_prob * chứa các biến, ràng buộc của bài toán LP, MIP.
  - *int glp_add_cols(glp_prob *lp, int ncs) : Tạo ncs cột - biến mới dùng để mô tả các biến của bài toán, giá trị kiểu int trả về là chỉ số của cột đầu tiên được được thêm vào.*
  - *void glp_set_col_bnds(glp_prob *lp, int i, int type, double lb, double ub)* : Đưa vào cận trên ub, cận dưới lb của cột - biến có chỉ số i. Biến type mô tả các trường hợp như chỉ có cận trên, chỉ có cận dưới, có cả cận trên cận dưới...
  - *void glp_set_col_kind(glp_prob *mip, int j, int kind)* : Chỉ ra cột - biến có chỉ số j thuộc loại nào, biến kind có thể mô tả biến integer - GLP_IV hay biến binary - GLP_BV, continous - GLP_CV.
  - *void glp_set_col_name(glp_prob *lp, int j, const char *name)* : Đặt name làm tên cho cột - biến có chỉ số là j.
  - *int glp_find_col(glp_prob *lp, const char *name)* : Trả về chỉ số của cột - biến có tên là name, trả về 0 trong trường hợp không tìm thấy.
  - *int glp_add_rows(glp_prob *lp, int nrs)* : Tạo nrs hàng - ràng buộc mô tả bài toán, trả về chỉ số của hàng đầu tiên vừa được thêm vào.
  - *void glp_set_row_bnds(glp_prob *lp, int i, int type, double lb, double ub)* : Đưa vào cận trên ub, cận dưới lb của hàng - ràng buộc thứ i. Biến type mô tả các trường hợp như chỉ có cận trên, chỉ có cận dưới, có cả cận trên cận dưới...
  - *void glp_set_row_name(glp_prob *lp, int j, const char *name)* : Đặt name làm tê cho hàng - ràng buộc có chỉ số j.
  - *int glp_find_row(glp_prob *lp, const char *name)* : Trả về chỉ số của hàng - ràng buộc có tên là name.
  - *void glp_set_mat_row(glp_prob *lp, int i, int len, const int ind[], const double val[])* : Gán hệ số cho "len" biến, ở hàng - ràng buộc có chỉ số i. Các biến này có chỉ số mô tả trong mảng ind, và hệ số tương ứng nằm trong mảng val. Ô đầu tiên trong 2 mảng này được bỏ qua.
  - *void glp_set_obj_coef(glp_prob *lp, int j, double coef)* : Gán coef làm hệ số của cột - biến có chỉ số j cho hàm mục tiêu. Trong trường hợp j bằng 0, coef là hệ số tự do của hàm mục tiêu.
  - *void glp_set_obj_dir(glp_prob *lp, int dir)* : Biến dir quy định bài toán min hoặc max qua 2 giá trị tương ứng GLP_MIN hoặc GLP_MAX.
  - *int glp_intopt(glp_prob *mip, const glp_iocp *parm)* : Giải bài toán mô tả bởi mip.
  - *double glp_mip_obj_val(glp_prob *mip)* : Lấy giá trị của hàm mục tiêu sau khi đã giải bài toán.
  - *double glp_mip_col_val(glp_prob *mip, int i)* : Lấy giá trị của cột - biến có chỉ số i sau khi đã giải bài toán.
  - *double glp_mip_row_val(glp_prob *mip, int i)* : Lấy giá trị của hàng có chỉ số i sau khi đã giải bài toán.
  - *int glp_get_num_rows(glp_prob *lp)* : Trả về số hàng - ràng buộc của bài toán.
- **SimplexSolver và GurobiSolver Plugin - Solver Foundation Academic v3.1 C#:**
  Hai Solver này dùng chung hầu hết các hàm, phương thức. Tất cả các hàm đề cập phía trên của GLPK đều có hàm tương tự trong Solver Foundation, do đó xin không tiếp tục đề cập ở đây.

4.2 Branch And Cut

Integer Linear Problem, Mixed Integer Linear Problem (ILP, MILP) có thể được giải quyết bằng giải thuật Branch And Cut (BAC) - giải thuật được thực hiện trong hầu hết các Solver trong đó có các Solver được dùng trong báo cáo này.
Về cơ bản, "Branch And Cut = Branch And Bound (BAB) + Cutting Planes", thủ tục "Cut" có thể cải thiện BAB bằng cách giảm thiểu kích thước của cây phân nhánh.

- **Branch** : Trong thủ tục này, MILP gỡ bỏ điều kiện nguyên ta được bài toán bài toán nới lỏng (Mixed Integer Relaxation Prpblem - MIRLP), bài toán lới nỏng này có thể được giải quyết bằng Simplex Method hoặc 1 thuật toán khác. Sau khi có phương án tối ưu của bài toán nới lỏng, nếu phương án tìm được có tồn tại biến không nguyên, việc chia nhánh sẽ được tiến hành ở một trong số các nghiệm không nguyên này. Bài toán ban đầu được chia thành 2 bài toán nhỏ, ở đó tập xác định của nghiệm được giới hạn nhỏ lại so với bài toán ban đầu.
- **Bound** : Giá trị hàm mục tiêu của bài toán nới lỏng luôn tốt hơn giá trị của bài toán toán nguyên ban đầu. Giá trị hàm mục tiêu của 2 nhánh được chia bởi thủ tục Branch luôn không được tối ưu bằng kết quả của bài toán gốc.
  Do đó, thủ tục này tiến hành dừng việc chia nhánh nếu giá trị hàm mục tiêu của bài toán nới lỏng nhánh hiện tại không tối ưu bằng phương án nguyên tốt nhất đã tìm thấy. Thủ tục này giảm bớt việc chia nhánh những nhánh ở đó phương án tối ưu chắc chắn không xuất hiện.
- **Cutting** : Có nhiều thủ tục "Cut" được đưa ra nhưng nội dung chủ đạo của thủ tục này là tiến hành đưa thêm ràng buộc, điều chỉnh ràng buộc để thu gọn vùng nghiệm khả dĩ của bàn toán. Trong phạm vi báo cáo này, xin không được đề cập tới nội dung phương án trên.

4.3 Vấn đề xuất kết quả và một số kết quả mẫu

Do việc làm tròn lượng tiền sau giao dịch hàng tháng thành nguyên có thể dẫn tới trường hợp một tháng nào đó, lượng tiền do làm tròn sẽ có giá trị âm mặc dù kết quả tính toán hợp lý. Ví dụ :

```
21s
131291

T10 M0 100000 5000 MCP 2642 KHP
T11 M1 3.4153
T12 M2 5824.43 -5000 MCP 228 REE -2642 KHP
T01 M3 100007 2000 REE
T02 M4 49227.6
T03 M5 53013.9 -2000 REE 5522 TDC
T04 M6 51002.7 -228 REE
T05 M7 59807.3 -5522 TDC
T06 M8 113088 2 KHP 3719 CSM
T07 M9 5.41528
T08 M10 9304.74 -2 KHP -3719 CSM
T09 M11 122429 4091 MCP
T10 M12 67502.7 5000 MCP
T11 M13 2.72458
T12 M14 8184.64 -5000 MCP
T01 M15 75721.5 -4091 MCP
```

Hình 4: Output có lượng tiền là số thực

```
10s
131283

T10 M0 100000 5000 MCP 2642 KHP
T11 M1 3
T12 M2 5824 -5000 MCP 228 REE -2642 KHP
T01 M3 100006 2000 REE
T02 M4 49226
T03 M5 53012 -2000 REE 5522 TDC
T04 M6 51000 -228 REE
T05 M7 59804 -5522 TDC
T06 M8 113084 2 KHP 3719 CSM
T07 M9 1
T08 M10 9300 -2 KHP -3719 CSM
T09 M11 122424 4091 MCP
T10 M12 67497 5000 MCP
T11 M13 -3
T12 M14 8178 -5000 MCP
T01 M15 75714 -4091 MCP
```

Hình 5: Output có lượng tiền dạng số nguyên, tương tự output Hình 4

Vì thế một số output dưới đây, lượng tiền mặt hàng tháng được xuất dưới dạng số thực để đảm bảo tính hợp lý. Với input có nội dung tương tự Hình 1, số tháng đầu tư thay đổi từ 10 tới 12 tháng kết quả output tương ứng là:

```
0s
122429

T10 M0 100000 5000 MCP 2642 KHP
T11 M1 3.4153
T12 M2 5824.43 -5000 MCP 228 REE -2642 KHP
T01 M3 100007 2000 REE
T02 M4 49227.6
T03 M5 53013.9 -2000 REE 5522 TDC
T04 M6 51002.7 -228 REE
T05 M7 59807.3 -5522 TDC
T06 M8 113088 2 KHP 3719 CSM
T07 M9 5.41528
T08 M10 9304.74 -2 KHP -3719 CSM
```

Hình 6: Kết quả đầu tư 10 tháng

```
1s
122980

T10 M0 100000 5000 MCP 2642 KHP
T11 M1 3.4153
T12 M2 5824.43 -5000 MCP 228 REE -2642 KHP
T01 M3 100007 2000 REE
T02 M4 49227.6
T03 M5 53013.9 -2000 REE 5522 TDC
T04 M6 51002.7 -228 REE
T05 M7 59807.3 -5522 TDC
T06 M8 113088 2 KHP 3719 CSM
T07 M9 5.41528
T08 M10 9304.74 -2 KHP -3719 CSM
T09 M11 122429
```

Hình 7: Kết quả đầu tư 11 tháng

```
1s
123986

T10 M0 100000 5000 MCP 2642 KHP
T11 M1 3.4153
T12 M2 5824.43 -5000 MCP 228 REE -2642 KHP
T01 M3 100007 2000 REE
T02 M4 49227.6
T03 M5 53013.9 -2000 REE 5522 TDC
T04 M6 51002.7 -228 REE
T05 M7 59807.3 -5522 TDC 650 SRC
T06 M8 96307.4 3168 CSM
T07 M9 975.22 -650 SRC
T08 M10 25604.6 800 DMP -3168 CSM
T09 M11 96311.8
T10 M12 97945.2 -800 DMP
```

Hình 8: Kết quả đầu tư 12 tháng

4.4 So sánh các Solver và mô hình

4.4.1 Đánh giá các Solver sử dụng

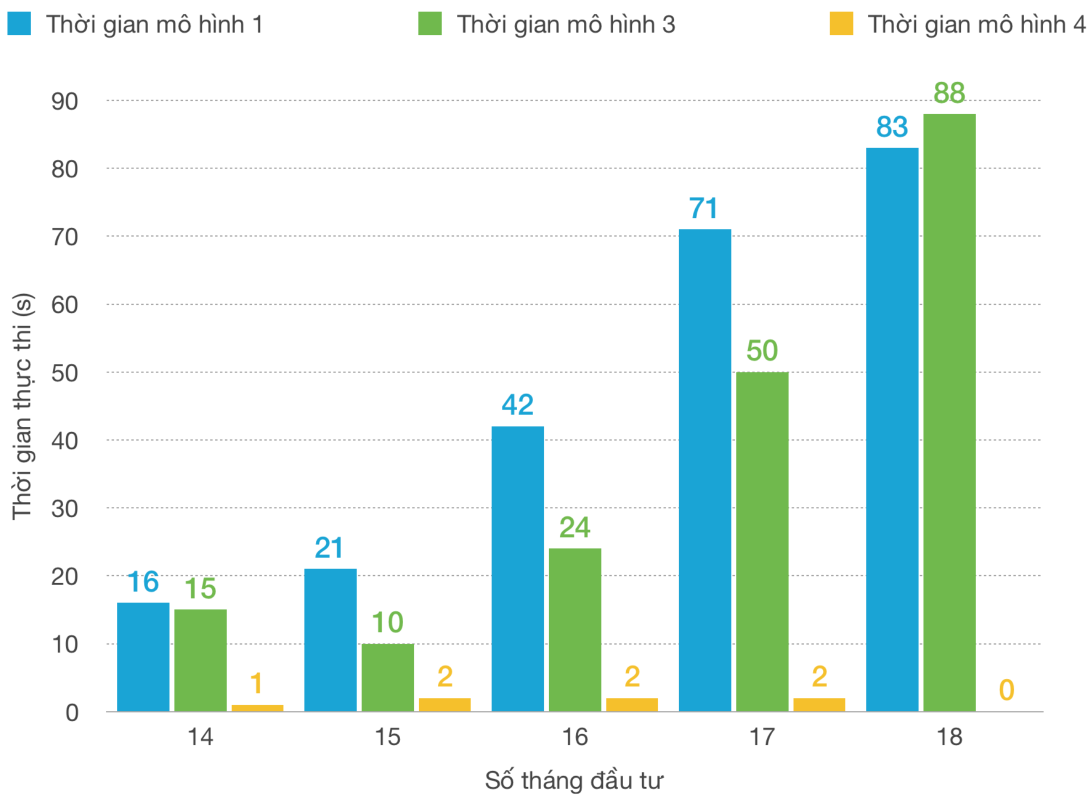
- GLPK chạy ở mức chấp nhận được, hỗ trợ mô tả từng ràng buộc qua code, hoặc đọc bài toán từ các định dạng file *.lp, hoặc *.mps ... các định dạng file này có thể sử dụng ở hầu hết các Solver khác. Ngoài ra, khi đã mô tả các ràng buộc bài toán, GLPK có thể ghi bài toán này ra các định dạng file như đã nói ở trên. Việc sử dụng các file này giải bằng các Solver khác rất dễ dàng.
  Solver này miễn phí, không giới hạn số ràng buộc, biến (1 số input khoảng 1000 biến vẫn dùng được Solver này nhưng thời gian thực thi trương trường hợp này quá lâu).
- SimplexSolver - Solver Foundation Academic : Solver này cũng có thể xử lý bài toán nhiều ràng buộc, nhiều biến nhưng tốc độ xử lý rất chậm. Với input như Hình 1 trong khi GLPK và Gurobi Plugin cho kết quả ngay lập tức thì Solver này thực thi trong thời gian $>$ 1800s !
- GurobiSolver - Plugin nằm trong Solver Foundation Academic : Solver này giới hạn bài toán ở mức 500 ràng buộc, 500 biến. Tuy nhiên Solver này cho tốc độ giải rất nhanh. Với input như Hình 1 - số tháng đầu tư từ 10 tới 15 thì solver này cho kết quả ngay lập tức.

4.4.2 So sánh 2 mô hình thực hiện trên Solver GLPK

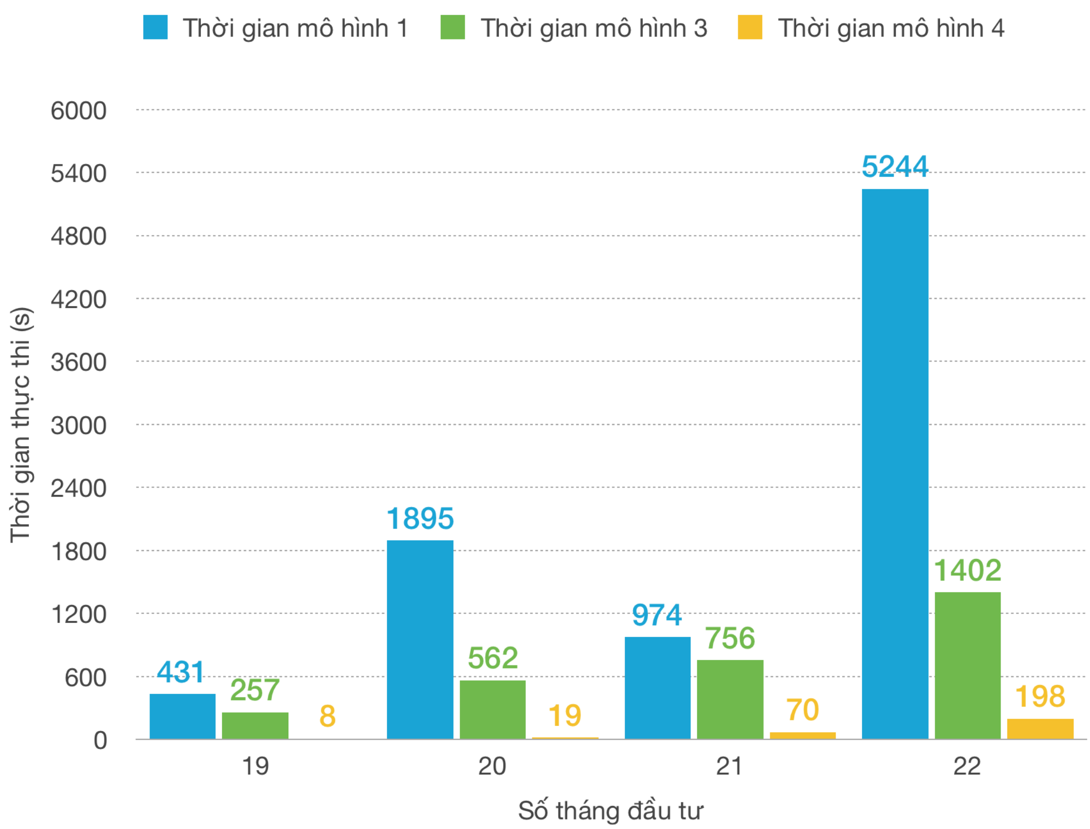
Để tiện cho việc so sánh 2 mô hình - các đầu vào có số lượng biến lớn, thời gian thực thi ở mức chấp nhận được, kết quả dưới đây dùng Solver GLPK.
Vẫn sử dụng input như mô tả ở Hình 1, thay đổi số tháng đầu tư từ 14 tới 25. Thời gian thực thi của 2 mô hình thu được như bảng dưới đây:

Số tháng đầu tư	14	15	16	17	18	19	20	21	22	23	24	25	Tổng
Thời gian mô hình 1	16	21	42	71	83	431	1895	974	5244	17111	22131	61438	109457
Thời gian mô hình 3	15	10	24	50	88	257	562	756	1402	2727	2697	4792	13380
Thời gian mô hình 4	1	2	2	2	0	8	19	70	198	19	966	143	1430

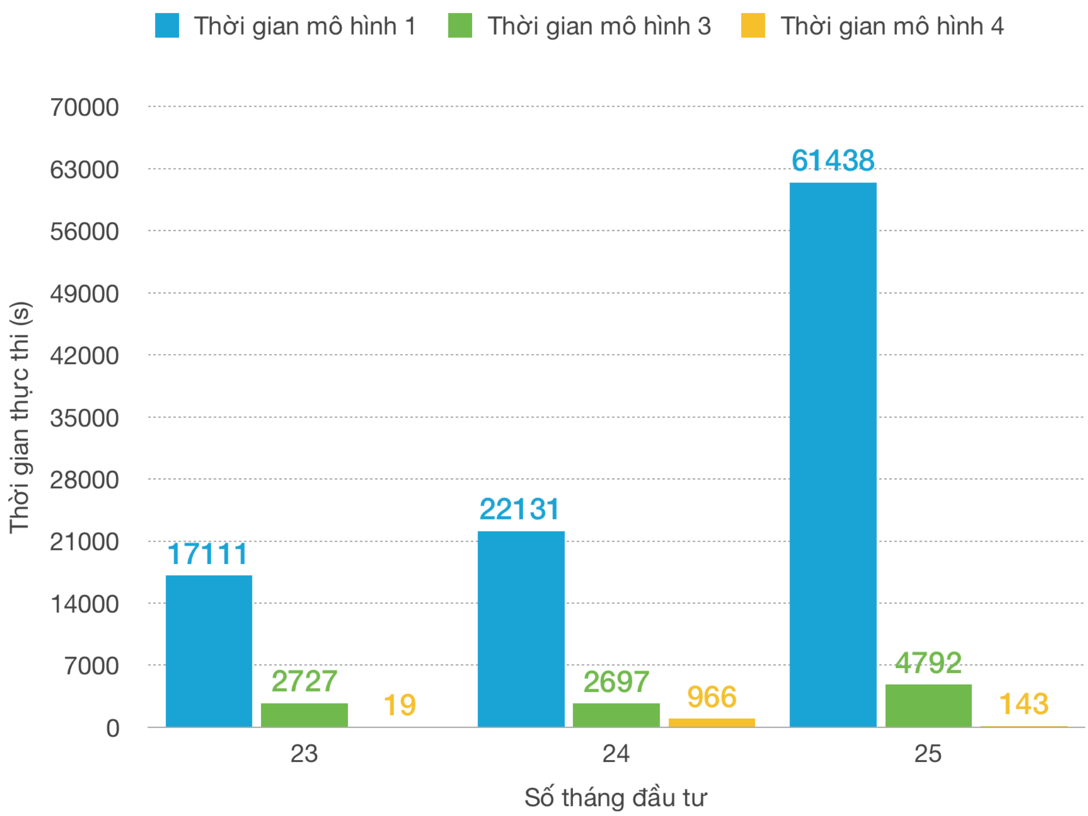
Hình 9: Thời gian thực thi của 3 mô hình (s)



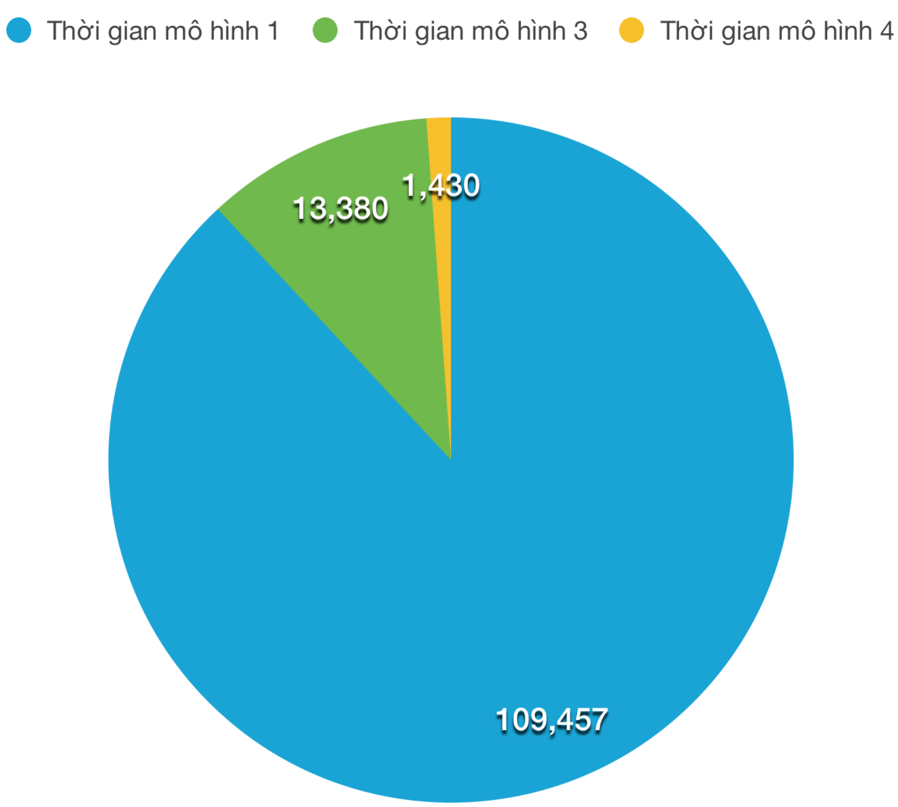
Hình 10: Thời gian thực thi của 3 mô hình với số tháng đầu tư từ 14 tới 18



Hình 11: Thời gian thực thi của 3 mô hình với số tháng đầu tư từ 19 tới 22



Hình 12: Thời gian thực thi của 3 mô hình với số tháng đầu tư từ 23 tới 25



Hình 13: Tổng thời gian thực thi từ 14 tới 25 tháng đầu tư

Với một input khác như sau :

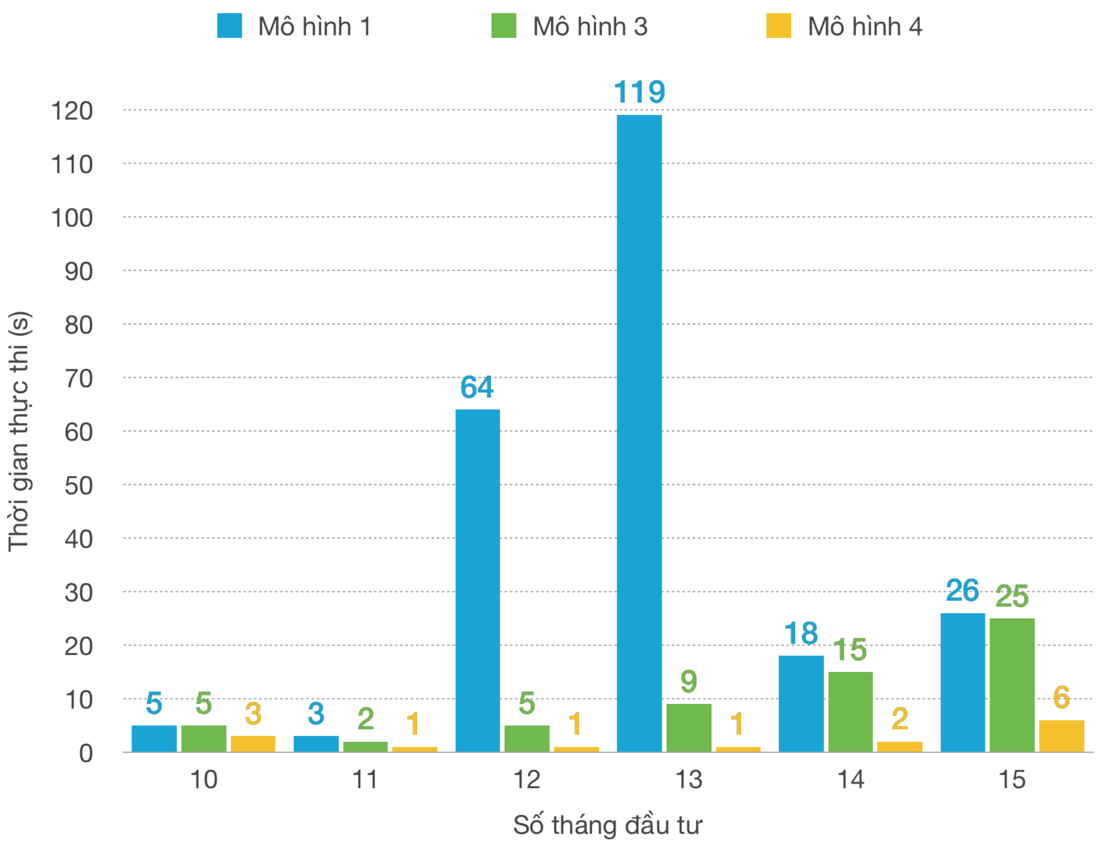
```
5     //thang hien tai
40000 //so tien du tu
0.45    //lai xuat
10    //thoi gian dau tu

CSE 7 11000 6(5) + 12(7)
HUS 15 1500 1(10) + 7(7)
SSA 40 2000 3(20) + 9(22)
C++ 20 10000 10(14)
LAA 17 9000 12(15)
BAA 11.5 5000 2(10)
CAA 9 3000 3(11)
MHH 10.5 500 1(12) + 7(9)
JQK 40 4000 3(22) + 10(20)
```

Hình 14: Input mẫu

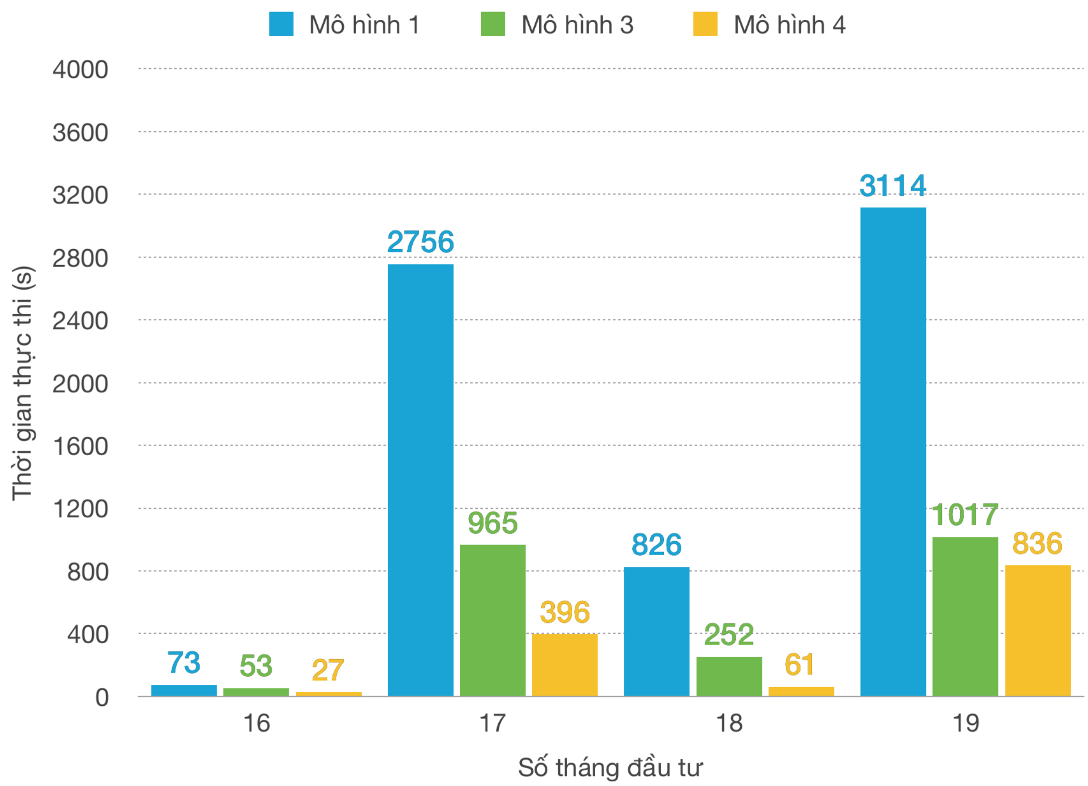
Dùng 2 mô hình tính toán với lượng thời gian đầu tư thay đổi từ 10 tới 20 tháng ta thu được kết quả :

Số tháng đầu tư	10	11	12	13	14	15	16	17	18	19	Tổng
Mô hình 1	5	3	64	119	18	26	73	2756	826	3114	7004
Mô hình 3	5	2	5	9	15	25	53	965	252	1017	2348
Mô hình 4	3	1	1	1	2	6	27	396	61	836	1334

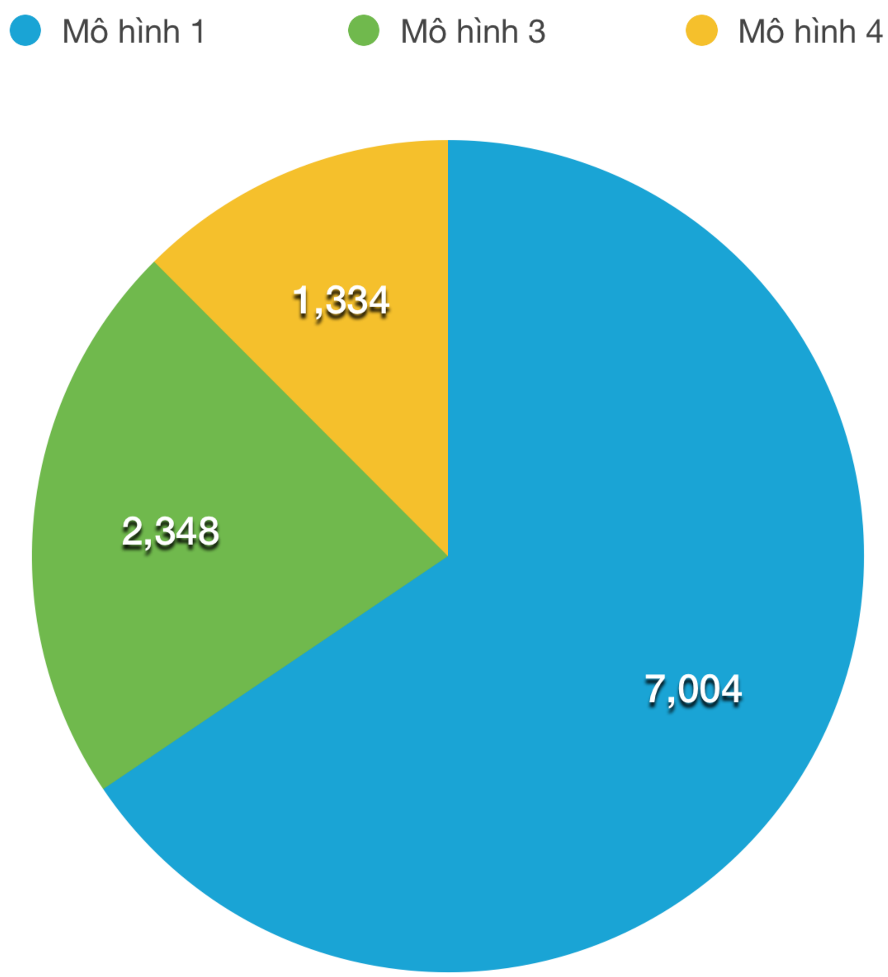
Hình 15: Thời gian thực thi của 3 mô hình (s)



Hình 16: Thời gian thực thi của 3 mô hình với số tháng đầu tư từ 10 tới 16



Hình 17: Thời gian thực thi của 2 mô hình với số tháng đầu tư từ 17 tới 20



Hình 18: Tổng thời gian thực thi từ 10 tới 20 tháng

Ta có thể nhận thấy, mô hình 3 cho thời gian chạy tốt hơn mô hình 1 trong hầu hết các trường hợp, còn mô hình 4 tốt hơn mô hình 3. Kết quả đó có thể giải thích bởi số biến của các mô hình. Với mỗi đầu vào τ tháng, n loại cổ phiếu thì mô hình 1 có $4n\tau$ biến, mô hình 3 có $3n\tau$ biến còn mô hình 4 có $2n\tau$ biến.
Trong khi đó, như đã đề cập tại 4.2, thuật toán các Solver thực hiện là Branch And Cut - số nhánh có thể chia của thủ tục Branch là 2^n với n là số biến của bài toán. Do đó, ngay cả khi bài toán MIRLP giải bằng Simplex Method có chi phí chênh lệch không nhiều khi số lượng biến thay đổi thì chi phí cho bài toán MIP giải bằng thuật toán Branch And Cut vẫn tăng theo hàm mũ của số lượng biến.

5 Kết luận

Đây là một bài toán ví dụ trong số các bài toán tối ưu chung quanh chúng ta. Nếu chúng ta có thể xác định được các bài toán này, và đề xuất được các thuật giải/giải pháp tìm ra đáp án tốt cho bài toán, điều này sẽ giúp cho các công việc hàng ngày của chúng ta sẽ được thực hiện trôi chảy và hiệu quả hơn. Hy vọng thông qua việc tìm hiểu và giải bài toán này, chúng ta sẽ hiểu hơn về các thuật toán ứng dụng thực tiễn quanh ta; và hy vọng trong một tương lai gần, các ban có cơ hội và có thể đề xuất các giải pháp tốt cho các bài toán hỗ trợ ra quyết định.
Mặc dù đã rất cố gắng nhưng do kiến thức hạn chế nên báo cáo này không tránh khỏi việc xuất hiệt sai xót vì thế nhóm rất mong nhận được nhận xét góp ý.

Tài liệu

[1] GLPK Documentation - Reference Manual
https://www.cs.unb.ca/bremner/docs/glpk/glpk.pdf

[2] Developing with Solver Foundation Services (SFS)
https://msdn.microsoft.com/en-us/library/ff524508(v=vs.93).aspx

[3] Cutting Planes in Mixed Integer Programming
http://www2.isye.gatech.edu/sdey30/IntroCuttingPlanes.pdf

[4] Branch-and-Cut Algorithms for Combinatorial Optimization Problems
http://homepages.rpi.edu/mitchj/papers/bc_hao.pdf

[5] Using GLPK in XCode
http://people.anu.edu.au/timothy.kam/play/computing/glpk_xcode/

[6] ISE 418 : Integer Programming
http://coral.ie.lehigh.edu/ted/teaching/ie418/