

Chạy số liệu minh hoạ bằng thư viện TiLearn

Thực hiện: Nguyễn Chí Bằng

Hướng dẫn: TS. Lê Minh Huy

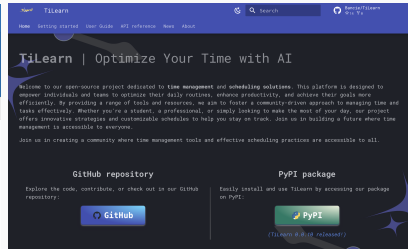
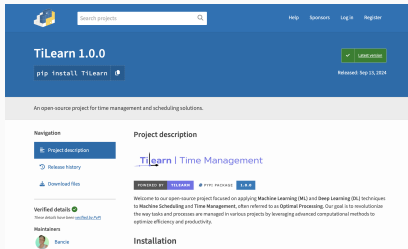
Ngày 18 tháng 10 năm 2024

- 1 Giới thiệu về thư viện
- 2 Chạy số liệu minh hoạ
- 3 Ứng dụng thực tế

Giới thiệu về thư viện

Giới thiệu về thư viện TiLearn

TiLearn là sự kết hợp của Time và Machine Learning (học máy), thư viện được xây dựng với mục đích ban đầu hướng đến mục tiêu tự động hoá quá trình đánh giá trọng số cho công việc (w_j) và cải thiện quá trình phân nhánh (Learning to Branch) bằng phương pháp học máy.



Mã nguồn hiện đã được tải lên **GitHub**, quét mã bên dưới để truy cập:



Hình 1: Mã nguồn GitHub.

Giới thiệu về thư viện TiLearn

Thư viện cũng đã được công bố trên trang **PyPI** giúp dễ dàng tải và sử dụng, quét mã bên dưới để truy cập:



Hình 2: Trang thư viện PyPI.

Giới thiệu về thư viện TiLearn

Bên cạnh đó, **web document** cũng được tạo để cung cấp tài liệu hướng dẫn cách sử dụng, cách chạy cũng như cách thư viện hoạt động, quét mã bên dưới để truy cập:



Hình 3: Web document.

Hiện thư viện đã hoàn thiện các tính năng sau:

- Hàm chạy thuật toán EDD.
- Hàm chạy thuật toán WSPT.
- Hàm giúp xử lý bài toán $1||\sum C_j$.
- Hàm giúp xử lý bài toán $1||\sum w_j C_j$.
- Hàm giúp xử lý bài toán $1|prec|\sum C_j$.
- Hàm giúp xử lý bài toán $1|prec|\sum w_j C_j$.
- Hàm giúp xử lý hỗn hợp 2 bài toán $1|prec|\sum w_j C_j$ và $1||\sum w_j C_j$.

Giới thiệu về thư viện TiLearn

- Để sử dụng thư viện, ta tải thư viện bằng lệnh sau:

```
1 pip install tilearn
```

- Thực hiện khai báo thư viện như sau:

```
1 import tilearn as tl  
2 from tilearn import _plat as pl
```

Giới thiệu về thư viện TiLearn

- Để sử dụng thư viện, ta tải thư viện bằng lệnh sau:


```
1 pip install tilearn
```

- Thực hiện khai báo thư viện như sau:

```
1 import tilearn as tl  
2 from tilearn import _plat as pl
```

Chạy số liệu minh hoạ

Giả sử ta cần xử lý 3 danh sách công việc sau:



Name	p	r	d	w
Job 1	4	0	100	0.65
Job 2	1	0	100	0.84
Job 3	3	0	100	0.46
Job 4	3	0	100	0.79
Job 5	1	0	100	0.17
Job 6	3	0	100	0.50
Job 7	4	0	100	0.95
Job 8	2	0	100	0.14
Job 9	5	0	100	0.52
Job 10	2	0	100	0.40
Job 11	4	0	100	0.55
Job 12	1	0	100	0.39
Job 13	2	0	100	0.57
Job 14	1	0	100	0.90
Job 15	1	0	100	0.22


Trong đó, danh sách 1 các công việc độc lập và **không tồn tại** ràng buộc thứ tự, tức thuộc dạng bài toán $1||\sum w_j C_j$.

Bảng 1: Danh sách 1.

Name	p	r	d	w
Job 16	4	0	100	0.70
Job 17	3	0	100	0.95
Job 18	4	0	100	0.49
Job 19	1	0	100	0.13
Job 20	5	0	100	0.94
Job 21	1	0	100	0.57
Job 22	4	0	100	0.79

Trong đó, danh sách 2 **tồn tại** ràng buộc thứ tự, tức thuộc dạng bài toán $1|prec|\sum w_j C_j$.

Bảng 2: Danh sách 2.



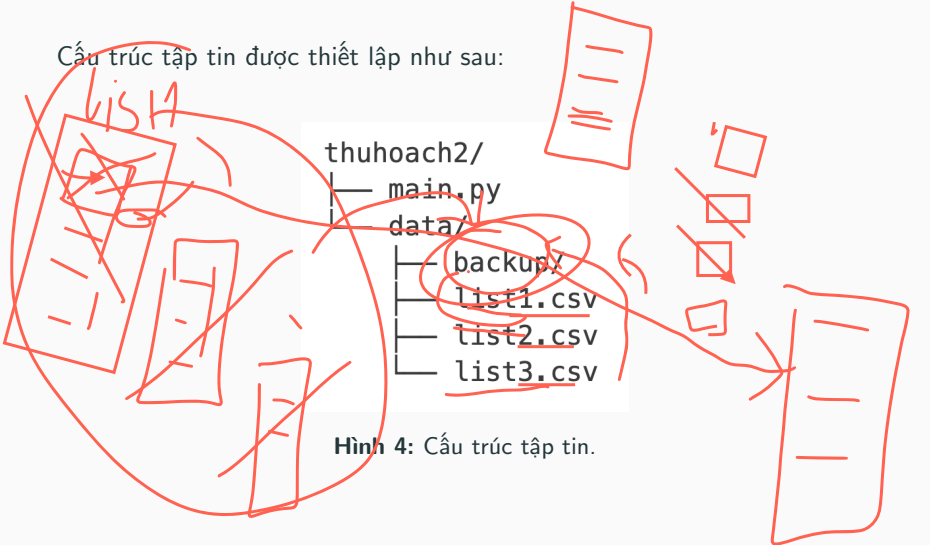
Name	p	r	d	w
Job 23	4	0	100	0.24
Job 24	1	0	100	0.54
Job 25	2	0	100	0.81
Job 26	2	0	100	0.41
Job 27	5	0	100	0.22
Job 28	2	0	100	0.29
Job 29	5	0	100	0.65
Job 30	4	0	100	0.69

Tương tự, danh sách 3 **tồn tại** ràng buộc thứ tự, tức thuộc dạng bài toán $1|prec| \sum w_j C_j$.

Bảng 3: Danh sách 3.

Chạy số liệu minh họa

Cấu trúc tập tin được thiết lập như sau:



thuhoach2/
├── main.py
├── data/
│ ├── backup/
│ │ ├── list1.csv
│ │ ├── list2.csv
│ │ └── list3.csv
└──

The diagram includes several hand-drawn red annotations: a large circle on the left containing several boxes with horizontal lines; a box at the top right with three horizontal lines; a vertical sequence of four boxes on the right, with the first three crossed out by red 'X' marks and an arrow pointing from the 'data/' directory to the first box; and a large box at the bottom right with three horizontal lines, with an arrow pointing to it from the 'list3.csv' file.

Hình 4: Cấu trúc tập tin.

Chạy số liệu minh hoạ

Ta thực hiện hai báo đường dẫn dữ liệu như sau:

```
1 original = 'thuhoach2/data'
2 backup = 'thuhoach2/data/backup'
3 list1 = 'thuhoach2/data/list1.csv'
4 list2 = 'thuhoach2/data/list2.csv'
5 list3 = 'thuhoach2/data/list3.csv'
```

Tiếp theo, ta chọn dạng bài toán cho từng danh sách bằng cách thiết lập tham số `prec` như sau:

- `prec=0` cho bài toán không tồn tại ràng buộc thứ tự.
- `prec=1` cho bài toán tồn tại ràng buộc thứ tự.

```
1 lists = [
2     pl.List(file_path=list1, prec=0),
3     pl.List(file_path=list2, prec=1),
4     pl.List(file_path=list3, prec=1),
5 ]
```


Chạy số liệu minh hoạ

Ta thực hiện hai báo đường dẫn dữ liệu như sau:

```
1 original = 'thuhoach2/data'  
2 backup = 'thuhoach2/data/backup'  
3 list1 = 'thuhoach2/data/list1.csv'  
4 list2 = 'thuhoach2/data/list2.csv'  
5 list3 = 'thuhoach2/data/list3.csv'
```

Tiếp theo, ta chọn dạng bài toán cho từng danh sách bằng cách thiết lập tham số **prec** như sau:

- **prec=0** cho bài toán không tồn tại ràng buộc thứ tự.
- **prec=1** cho bài toán tồn tại ràng buộc thứ tự.

```
1 lists = [  
2     pl.List(file_path=list1, prec=0),  
3     pl.List(file_path=list2, prec=1),  
4     pl.List(file_path=list3, prec=1),  
5 ]
```

Chạy số liệu minh họa

Ta chạy chương trình bằng lệnh sau:

```
1 schedule = tl.optimal_list(lists, original, backup)
2 print(schedule)
```

```
[[ 'Job 14', 1.0, 0, 100, 0.9, 0.9], [ 'Job 2', 1.0, 0, 100, 0.84, 0.84], [ 'Job 12', 1.0, 0, 100, 0.39,
0.39], [ 'Job 13', 2.0, 0, 100, 0.57, 0.285], [ 'Job 4', 3.0, 0, 100, 0.79, 0.26333333333333336], [ 'Job
7', 4.0, 0, 100, 0.95, 0.2375], [ 'Job 16', 4.0, 0, 100, 0.7, 0.175], [ 'Job 17', 3.0, 0, 100, 0.95,
0.2357142857142857], [ 'Job 23', 4.0, 0, 100, 0.24, 0.06], [ 'Job 24', 1.0, 0, 100, 0.54, 0.156], [ 'Job
25', 2.0, 0, 100, 0.81, 0.22714285714285715], [ 'Job 15', 1.0, 0, 100, 0.22, 0.22], [ 'Job 26', 2.0, 0,
100, 0.41, 0.205], [ 'Job 10', 2.0, 0, 100, 0.4, 0.2], [ 'Job 18', 4.0, 0, 100, 0.49, 0.1225], [ 'Job 19',
1.0, 0, 100, 0.13, 0.124], [ 'Job 20', 5.0, 0, 100, 0.94, 0.156], [ 'Job 21', 1.0, 0, 100, 0.57,
0.19363636363636363], [ 'Job 5', 1.0, 0, 100, 0.17, 0.17], [ 'Job 6', 3.0, 0, 100, 0.5,
0.16666666666666666], [ 'Job 1', 4.0, 0, 100, 0.65, 0.1625], [ 'Job 3', 3.0, 0, 100, 0.46,
0.15333333333333335], [ 'Job 11', 4.0, 0, 100, 0.55, 0.1375], [ 'Job 22', 4.0, 0, 100, 0.47, 0.1175],
[ 'Job 27', 5.0, 0, 100, 0.22, 0.044], [ 'Job 28', 2.0, 0, 100, 0.29, 0.07285714285714286], [ 'Job 29',
5.0, 0, 100, 0.65, 0.096666666666666668], [ 'Job 30', 4.0, 0, 100, 0.69, 0.115625], [ 'Job 9', 5.0, 0,
100, 0.52, 0.10400000000000001], [ 'Job 8', 2.0, 0, 100, 0.14, 0.07]]
```

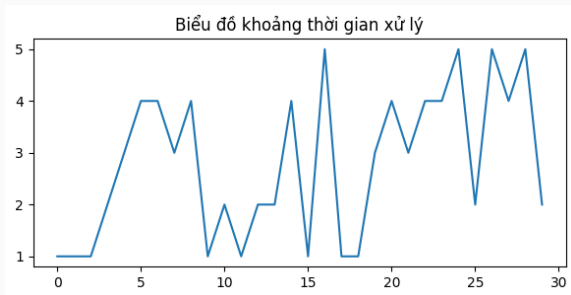
Chạy số liệu minh họa

Ta chạy chương trình bằng lệnh sau:

```
1 schedule = tl.optimal_list(lists, original, backup)
2 print(schedule)
```

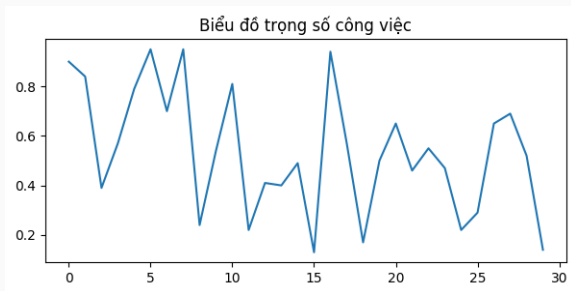
```
['Job 14', 1.0, 0, 100, 0.9, 0.9], ['Job 2', 1.0, 0, 100, 0.84, 0.84], ['Job 12', 1.0, 0, 100, 0.39,
0.39], ['Job 13', 2.0, 0, 100, 0.57, 0.285], ['Job 4', 3.0, 0, 100, 0.79, 0.26333333333333336], ['Job
7', 4.0, 0, 100, 0.95, 0.2375], ['Job 16', 4.0, 0, 100, 0.7, 0.175], ['Job 17', 3.0, 0, 100, 0.95,
0.2357142857142857], ['Job 23', 4.0, 0, 100, 0.24, 0.06], ['Job 24', 1.0, 0, 100, 0.54, 0.156], ['Job
25', 2.0, 0, 100, 0.81, 0.22714285714285715], ['Job 15', 1.0, 0, 100, 0.22, 0.22], ['Job 26', 2.0, 0,
100, 0.41, 0.205], ['Job 10', 2.0, 0, 100, 0.4, 0.2], ['Job 18', 4.0, 0, 100, 0.49, 0.1225], ['Job 19',
1.0, 0, 100, 0.13, 0.124], ['Job 20', 5.0, 0, 100, 0.94, 0.156], ['Job 21', 1.0, 0, 100, 0.57,
0.19363636363636363], ['Job 5', 1.0, 0, 100, 0.17, 0.17], ['Job 6', 3.0, 0, 100, 0.5,
0.16666666666666666], ['Job 1', 4.0, 0, 100, 0.65, 0.1625], ['Job 3', 3.0, 0, 100, 0.46,
0.15333333333333335], ['Job 11', 4.0, 0, 100, 0.55, 0.1375], ['Job 22', 4.0, 0, 100, 0.47, 0.1175],
['Job 27', 5.0, 0, 100, 0.22, 0.044], ['Job 28', 2.0, 0, 100, 0.29, 0.07285714285714286], ['Job 29',
5.0, 0, 100, 0.65, 0.096666666666666668], ['Job 30', 4.0, 0, 100, 0.69, 0.115625], ['Job 9', 5.0, 0,
100, 0.52, 0.10400000000000001], ['Job 8', 2.0, 0, 100, 0.14, 0.07]]
```

Dữ liệu trực quan



Hình 5: Biểu đồ khoảng thời gian xử lý.

Dựa vào biểu đồ trên, có thể thấy rằng sau khi tối ưu hóa danh sách công việc, thời gian xử lý của từng công việc có **xu hướng tăng** từ công việc đầu tiên đến công việc cuối cùng.



Hình 6: Biểu đồ trọng số công việc.

Dựa vào biểu đồ trên, ta có thể thấy rằng sau khi tối ưu hóa danh sách công việc, trọng số của từng công việc có **xu hướng giảm** từ công việc đầu tiên đến công việc cuối cùng.

Từ đây, ta dễ dàng nhận thấy rằng các công việc có **thời gian xử lý ngắn** và **trọng số (mức độ ưu tiên) cao** sẽ được đẩy lên đầu, trong khi những công việc có thời gian xử lý dài hơn và mức độ ưu tiên thấp hơn sẽ được xếp sau.

Ứng dụng thực tế

Lập kế hoạch gia công thiết kế

Giới thiệu về quy trình thiết kế

Giới thiệu về quy trình thiết kế



- Trong quá trình xử lý, studio thường phải đối mặt với việc phân bổ thời gian và quản lý **hiều dự án đồng thời**, dựa trên sự ưu tiên, độ phức tạp của từng dự án cũng như thời gian thi công.
- Việc áp dụng các **kỹ thuật lập lịch** giúp quản lý dữ liệu và lập kế hoạch hiệu quả hơn, đồng thời hỗ trợ theo dõi tiến độ và điều chỉnh khi cần thiết.

Giới thiệu về quy trình thiết kế



- Trong quá trình xử lý, studio thường phải đối mặt với việc phân bổ thời gian và quản lý **hiều dự án đồng thời**, dựa trên sự ưu tiên, độ phức tạp của từng dự án cũng như thời gian thi công.
- Việc áp dụng các **kỹ thuật lập lịch** giúp quản lý dữ liệu và lập kế hoạch hiệu quả hơn, đồng thời hỗ trợ theo dõi tiến độ và điều chỉnh khi cần thiết.

Giới thiệu về quy trình thiết kế

Thông thường, một dự án thiết kế nội thất sẽ trải qua các giai đoạn chính sau:

- Phát triển ý tưởng.
- Quy hoạch không gian.
- Phát triển thiết kế.
- Lựa chọn vật liệu.
- Tính toán chi phí.
- Xây dựng.
- Lắp đặt.
- Decor.
- Nghiệm thu.

Giới thiệu về quy trình thiết kế

- Các công đoạn trong một dự án cần được thực hiện theo thứ tự (hay **ràng buộc thứ tự**). Do đó, bài toán có thể được mô hình hóa dưới dạng bài toán $1|prec|\sum w_j C_j$.
- Ngoài ra, studio còn phải thực hiện các công việc khảo sát hiện trạng cho nhiều dự án khác nhau mà không yêu cầu thứ tự thực hiện (hay **không tồn tại ràng buộc thứ tự**). Trường hợp này có thể được mô hình hóa thành bài toán $1||\sum w_j C_j$.

Giới thiệu về quy trình thiết kế

- Các công đoạn trong một dự án cần được thực hiện theo thứ tự (hay **ràng buộc thứ tự**). Do đó, bài toán có thể được mô hình hóa dưới dạng bài toán $1|prec|\sum w_j C_j$.
- Ngoài ra, studio còn phải thực hiện các công việc khảo sát hiện trạng cho nhiều dự án khác nhau mà không yêu cầu thứ tự thực hiện (hay **không tồn tại ràng buộc thứ tự**). Trường hợp này có thể được mô hình hóa thành bài toán $1||\sum w_j C_j$.

Ví dụ 6.1

Giả sử studio đang trong giai đoạn thực hiện 3 dự án (công trình) sau:

- **SJC Tower** - Dự án A.
- **Nguyễn Cư Trinh Centre** - Dự án B.
- **Tòa nhà phức hợp Amigo** - Dự án C.

Cũng trong quá trình đó, studio cần thực hiện khảo sát hiện trạng cho các công trình sau:

- **Tòa nhà văn phòng Fideco** - Khảo sát hiện trạng 1.
- **Công trình tại đường Tôn Đức Thắng** - Khảo sát hiện trạng 2.
- **Dự án căn hộ và văn phòng tại đường Nguyễn Hữu Cảnh** - Khảo sát hiện trạng 3.
- **Dự án cầu đường Nguyễn Khoái** - Khảo sát hiện trạng 4.
- **Công trình nhà dân dụng và công nghiệp tại quận 1** - Khảo sát hiện trạng 5.
- **IFC One Saigon** - Khảo sát hiện trạng 6.
- **One Central Saigon** - Khảo sát hiện trạng 7.

Xử lý số liệu bằng TiLearn

Name	p	r	d	w
Project A - Phát triển ý tưởng	4	0	200	0.85
Project A - Quy hoạch không gian	3	0	200	0.80
Project A - Phát triển thiết kế	5	0	200	0.75
Project A - Lựa chọn vật liệu	2	0	200	0.70
Project A - Tính toán chi phí	2	0	200	0.60
Project A - Xây dựng	10	0	200	0.95
Project A - Lắp đặt	4	0	200	0.70
Project A - Decor	3	0	200	0.85
Project A - Bàn giao	1	0	200	1.00

Bảng 4: Project A - SJC Tower.

Name	p	r	d	w
Project B - Phát triển ý tưởng	8	0	200	0.85
Project B - Quy hoạch không gian	7	0	200	0.80
Project B - Phát triển thiết kế	20	0	200	0.95
Project B - Lựa chọn vật liệu	3	0	200	0.70
Project B - Tính toán chi phí	2	0	200	0.60
Project B - Xây dựng	40	0	200	0.95
Project B - Lắp đặt	15	0	200	0.70
Project B - Decor	9	0	200	0.85
Project B - Bàn giao	1	0	200	1.00

Bảng 5: Project B - Nguyễn Cư Trinh Centre.

Name	p	r	d	w
Project C - Phát triển ý tưởng	5	0	200	0.85
Project C - Quy hoạch không gian	2	0	200	0.80
Project C - Phát triển thiết kế	4	0	200	0.75
Project C - Lựa chọn vật liệu	6	0	200	0.70
Project C - Tính toán chi phí	1	0	200	0.60
Project C - Xây dựng	15	0	200	0.95
Project C - Lắp đặt	4	0	200	0.70
Project C - Decor	4	0	200	0.85
Project C - Bàn giao	1	0	200	1.00

Bảng 6: Project C - Tòa nhà phức hợp Amigo.

Name	p	r	d	w
Khảo sát hiện trạng 1	2	0	200	0.33
Khảo sát hiện trạng 2	6	0	200	0.67
Khảo sát hiện trạng 3	5	0	200	0.45
Khảo sát hiện trạng 4	15	0	200	0.85
Khảo sát hiện trạng 5	4	0	200	0.10
Khảo sát hiện trạng 6	4	0	200	0.60
Khảo sát hiện trạng 7	20	0	200	0.70

Bảng 7: Danh sách khảo sát hiện trạng.

Cấu trúc tập tin được thiết lập như sau:

```
thuhoach2/  
├── main.py  
└── data_project/  
    ├── backup/  
    ├── Project A-Project A.csv  
    ├── Project B-Project B.csv  
    ├── Project C-Project C.csv  
    └── Survey.csv
```

Hình 7: Cấu trúc tập tin.

Ta thực hiện khai báo đường dẫn dữ liệu như sau:

```
1 original_project = 'thuhoach2/data_project'
2 backup_project = 'thuhoach2/data_project/backup'
3 project_A = 'thuhoach2/data_project/Project A-Project A.
4 csv'
5 project_B = 'thuhoach2/data_project/Project B-Project B.
6 csv'
7 project_C = 'thuhoach2/data_project/Project C-Project C.
8 csv'
9 survey = 'thuhoach2/data_project/Survey.csv'
```

Tiếp theo, ta chọn dạng bài toán cho từng danh sách bằng cách thiết lập tham số **prec** như sau:

- **prec=0** cho bài toán không tồn tại ràng buộc thứ tự.
- **prec=1** cho bài toán tồn tại ràng buộc thứ tự.

```
1 lists_project = [  
2     pl.List(file_path=project_A, prec=1),  
3     pl.List(file_path=project_B, prec=1),  
4     pl.List(file_path=project_C, prec=1),  
5     pl.List(file_path=survey, prec=0),  
6 ]
```


Chạy số liệu minh họa

Ta chạy chương trình bằng lệnh sau:

```
1 schedule = tl.optimal_list(lists_project,
2                             original_project,
3                             backup_project)
4 print(schedule)
```

```
[['Project A - Phát triển ý tưởng', 4.0, 0, 200, 0.85, 0.2125], ['Project A - Quy hoạch không gian',
3.0, 0, 200, 0.8, 0.2357142857142857], ['Project C - Phát triển ý tưởng', 5.0, 0, 200, 0.85,
0.16999999999999998], ['Project C - Quy hoạch không gian', 2.0, 0, 200, 0.8,
0.2357142857142857], ['Project A - Phát triển thiết kế', 5.0, 0, 200, 0.75, 0.15], ['Project A - Lựa
chọn vật liệu', 2.0, 0, 200, 0.7, 0.20714285714285713], ['Project A - Tính toán chi phí', 2.0, 0,
200, 0.6, 0.22777777777777775], ['Project A - Xây dựng', 10.0, 0, 200, 0.95, 0.095], ['Project A -
Lắp đặt', 4.0, 0, 200, 0.7, 0.11785714285714285], ['Project A - Decor', 3.0, 0, 200, 0.85,
0.14705882352941177], ['Project A - Bàn giao', 1.0, 0, 200, 1.0, 0.19444444444444445], ['Project
C - Phát triển thiết kế', 4.0, 0, 200, 0.75, 0.1875], ['Project C - Lựa chọn vật liệu', 6.0, 0, 200,
0.7, 0.11666666666666665], ['Project C - Tính toán chi phí', 1.0, 0, 200, 0.6,
0.1857142857142857], ['Khảo sát hiện trạng 1', 2.0, 0, 200, 0.33, 0.165], ['Khảo sát hiện trạng 6',
```

Chạy số liệu minh họa

Ta chạy chương trình bằng lệnh sau:

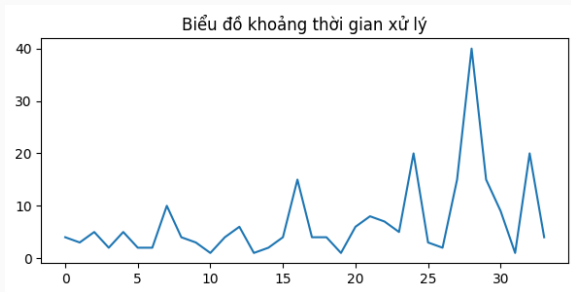
```
1 schedule = tl.optimal_list(lists_project ,
2                             original_project ,
3                             backup_project)
4 print(schedule)
```

```
['Project A - Phát triển ý tưởng', 4.0, 0, 200, 0.85, 0.2125], ['Project A - Quy hoạch không gian',
3.0, 0, 200, 0.8, 0.2357142857142857], ['Project C - Phát triển ý tưởng', 5.0, 0, 200, 0.85,
0.16999999999999998], ['Project C - Quy hoạch không gian', 2.0, 0, 200, 0.8,
0.2357142857142857], ['Project A - Phát triển thiết kế', 5.0, 0, 200, 0.75, 0.15], ['Project A - Lựa
chọn vật liệu', 2.0, 0, 200, 0.7, 0.20714285714285713], ['Project A - Tính toán chi phí', 2.0, 0,
200, 0.6, 0.22777777777777775], ['Project A - Xây dựng', 10.0, 0, 200, 0.95, 0.095], ['Project A -
Lắp đặt', 4.0, 0, 200, 0.7, 0.11785714285714285], ['Project A - Decor', 3.0, 0, 200, 0.85,
0.14705882352941177], ['Project A - Bàn giao', 1.0, 0, 200, 1.0, 0.19444444444444445], ['Project
C - Phát triển thiết kế', 4.0, 0, 200, 0.75, 0.1875], ['Project C - Lựa chọn vật liệu', 6.0, 0, 200,
0.7, 0.11666666666666665], ['Project C - Tính toán chi phí', 1.0, 0, 200, 0.6,
0.1857142857142857], ['Khảo sát hiện trạng 1', 2.0, 0, 200, 0.33, 0.165], ['Khảo sát hiện trạng 6',
```

Chạy số liệu minh họa

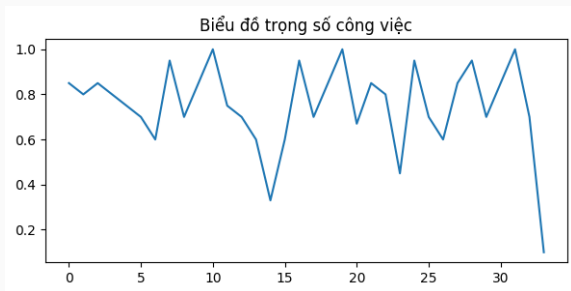
4.0, 0, 200, 0.6, 0.15], ['Project C - Xây dựng', 15.0, 0, 200, 0.95, 0.063333333333333332], ['Project C - Lắp đặt', 4.0, 0, 200, 0.7, 0.08684210526315789], ['Project C - Decor', 4.0, 0, 200, 0.85, 0.10869565217391304], ['Project C - Bàn giao', 1.0, 0, 200, 1.0, 0.14583333333333334], ['Khảo sát hiện trạng 2', 6.0, 0, 200, 0.67, 0.111666666666666668], ['Project B - Phát triển ý tưởng', 8.0, 0, 200, 0.85, 0.10625], ['Project B - Quy hoạch không gian', 7.0, 0, 200, 0.8, 0.11], ['Khảo sát hiện trạng 3', 5.0, 0, 200, 0.45, 0.09], ['Project B - Phát triển thiết kế', 20.0, 0, 200, 0.95, 0.0475], ['Project B - Lựa chọn vật liệu', 3.0, 0, 200, 0.7, 0.07173913043478261], ['Project B - Tính toán chi phí', 2.0, 0, 200, 0.6, 0.09], ['Khảo sát hiện trạng 4', 15.0, 0, 200, 0.85, 0.056666666666666664], ['Project B - Xây dựng', 40.0, 0, 200, 0.95, 0.02375], ['Project B - Lắp đặt', 15.0, 0, 200, 0.7, 0.03], ['Project B - Decor', 9.0, 0, 200, 0.85, 0.0390625], ['Project B - Bàn giao', 1.0, 0, 200, 1.0, 0.05384615384615385], ['Khảo sát hiện trạng 7', 20.0, 0, 200, 0.7, 0.034999999999999996], ['Khảo sát hiện trạng 5', 4.0, 0, 200, 0.1, 0.025]]

Dữ liệu trực quan



Hình 8: Biểu đồ khoảng thời gian xử lý.

Dựa vào biểu đồ trên, có thể thấy rằng sau khi tối ưu hóa danh sách công việc, thời gian xử lý của từng công việc có **xu hướng tăng** từ công việc đầu tiên đến công việc cuối cùng.



Hình 9: Biểu đồ trọng số công việc.

Dựa vào biểu đồ trên, ta có thể thấy rằng sau khi tối ưu hóa danh sách công việc, trọng số của từng công việc có **xu hướng giảm** từ công việc đầu tiên đến công việc cuối cùng.

Từ đây, ta dễ dàng nhận thấy rằng các công việc có **thời gian xử lý ngắn** và **trọng số (mức độ ưu tiên) cao** sẽ được đẩy lên đầu, trong khi những công việc có thời gian xử lý dài hơn và mức độ ưu tiên thấp hơn sẽ được xếp sau.

Từ đó giúp tối ưu hoá quy trình dự án một cách hiệu quả.

Tài liệu

- [1] Le Minh Huy. **“Single Machine Scheduling”**. In.
- [2] M.L. Pinedo. **Planning and Scheduling in Manufacturing and Services**. Springer series in operations research. Springer New York, 2009.
- [3] Michael Pinedo. **“Scheduling: Theory, Algorithms, And Systems”**. In: Jan. 2008.
- [4] PMI, ed. **A Guide to the Project Management Body of Knowledge (PMBOK Guide)**. 5th ed. Newtown Square, PA: Project Management Institute, 2013.
- [5] Krzysztof Postek et al. **Hands-On Mathematical Optimization with AMPL in Python**. Online, 2024.

Thanks for listening!