

Chạy số liệu minh hoạ bằng thư viện TiLearn

Thực hiện: Nguyễn Chí Bằng

Hướng dẫn: TS. Lê Minh Huy

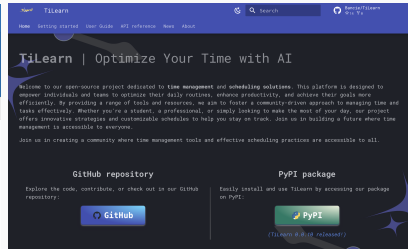
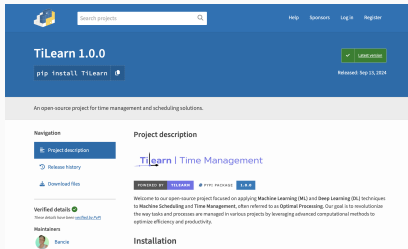
Ngày 15 tháng 10 năm 2024

- 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
```

Một số thư viện hỗ trợ phân tích dữ liệu:

```
1 pip install pandas  
2 pip install ipython  
3 pip install jinja2
```

Giới thiệu về thư viện TiLearn

Ta khai báo thư viện chính sau:

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

Thư viện hỗ trợ phân tích dữ liệu:

```
1 import csv
2 import pandas as pd
3 from IPython.display import display
4 import matplotlib.pyplot as plt
```

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.

Cấu trúc tập tin được thiết lập như sau:

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

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 ]
```

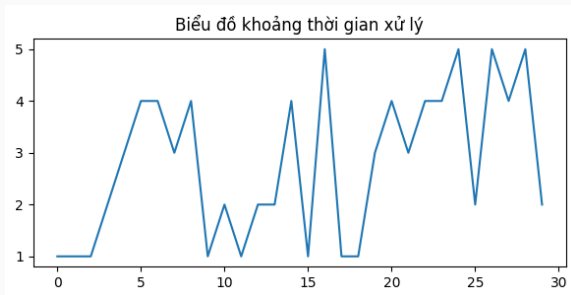

Chạy số liệu minh họa

Ta chạy chương trình bằng lệnh sau:

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

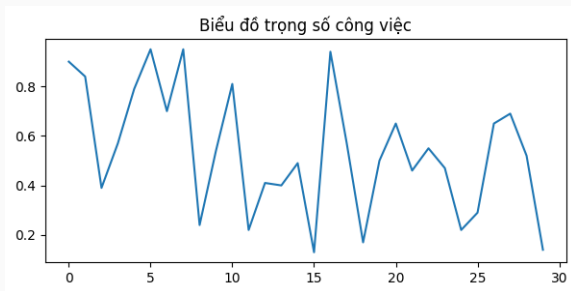
```
['Job 14', 1.0, 0, 100, 0.9, 0.9], ['Job 2', 1.0, 0, 100, 0.84, 0.84], ['Job 12', 1.0, 0, 100, 0.39,
0.39], ['Job 13', 2.0, 0, 100, 0.57, 0.285], ['Job 4', 3.0, 0, 100, 0.79, 0.26333333333333336], ['Job
7', 4.0, 0, 100, 0.95, 0.2375], ['Job 16', 4.0, 0, 100, 0.7, 0.175], ['Job 17', 3.0, 0, 100, 0.95,
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,
0.15333333333333333], ['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]]
```

Trực quan hoá dữ liệu



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

Tài liệu

- [1] E.B. Bajalinov. **Linear-Fractional Programming Theory, Methods, Applications and Software.** Applied Optimization. Springer US, 2013.
- [2] Phan Hoàng Chơn. In: *Giáo trình Đại số tuyến tính* Đại học Sài Gòn (2022).
- [3] M. Conforti, G. Cornuejols, and G. Zambelli. **Integer Programming.** Graduate Texts in Mathematics. Springer International Publishing, 2014. ISBN: 9783319110097. URL: <https://books.google.com.vn/books?id=QdAMswEACAAJ>.
- [4] G.B. Dantzig and M.N. Thapa. **Linear Programming 2: 2: Theory and Extensions.** Linear Programming. Springer, 1997. ISBN: 9780387986135. URL: <https://books.google.com.vn/books?id=qUvXMT00PZwC>.

- [5] George B. Dantzig and Mukund Narain Thapa. **Linear Programming 1: Introduction**. Springer Series in Operations Research and Financial Engineering. New York: Springer, 1997. ISBN: 0387948333 9780387948331 0387986138 9780387986135.
- [6] Arnaud Deza and Elias B. Khalil. “**Machine Learning for Cutting Planes in Integer Programming: A Survey**”. In: *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence*. IJCAI-2023. International Joint Conferences on Artificial Intelligence Organization, Aug. 2023.
- [7] Nguyễn Hữu Điển. In: *Giáo trình Tối ưu tuyến tính và Ứng dụng* NXB ĐHQG Hà Nội ().
- [8] H.A. Eiselt and C.L. Sandblom. **Linear Programming and its Applications**. Springer Berlin Heidelberg, 2007. ISBN: 9783540736714. URL: <https://books.google.com.ec/books?id=pCMCdycl2kMC>.
- [9] Ralph Gomory. “**Outline of an Algorithm for Integer Solutions to Linear Programs and An Algorithm for the Mixed Integer Problem**”. In: Jan. 2010, pp. 77–103. ISBN: 978-3-540-68274-5. DOI: 10.1007/978-3-540-68279-0_4.

- [10] M. Jünger et al. **50 Years of Integer Programming 1958-2008: From the Early Years to the State-of-the-Art**. Springer Berlin Heidelberg, 2009. ISBN: 9783540682790. URL: https://books.google.com.vn/books?id=bUJc_weiYfkC.
- [11] Ailsa H. Land and Alison G. Doig. **“An Automatic Method of Solving Discrete Programming Problems”**. In: *Econometrica* 28 (1960), p. 497. URL: <https://api.semanticscholar.org/CorpusID:35442133>.
- [12] D.C. Lay. **Linear Algebra and Its Applications**. Pearson Education, 2003. ISBN: 9788177583335. URL: <https://books.google.com.vn/books?id=v8Zls26Y0wkC>.
- [13] Đ.N. Nguyễn. **Tối ưu hoá: Quy hoạch tuyến tính và rời rạc**. Giáo dục, 1996. URL: <https://books.google.com.vn/books?id=MNBraQAACAAJ>.
- [14] S.S. Rao. **Engineering Optimization: Theory and Practice**. Wiley, 2009. ISBN: 9780470183526. URL: <https://books.google.com.vn/books?id=YNt34dvnQLEC>.

- [15] Tạ Quang Sơn. In: *Bài giảng Quy hoạch tuyến tính* Đại học Sài Gòn (2023).
- [16] Bùi Thế Tâm. In: *Quy hoạch rời rạc* NXB Hà Nội (2008).

Cảm ơn mọi người đã quan tâm theo dõi!