FET is open source free software for automatically scheduling the timetable of a school, high-school or university. It uses a fast and efficient timetabling algorithm. It is licensed under the GNU Affero General Public License version 3 or later.

<https://lalescu.ro/liviu/fet/>

<https://lalescu.ro/liviu/fet/doc/>

[**https://www.timetabling.de/manual/FET-manual.en.html#id\_16**](https://www.timetabling.de/manual/FET-manual.en.html#id_16) **-- Look for Constraints discription it can be used Weight based constraints**

**List of Analyze:**

**1. Cost of system**

FET is free software. Please consider donating to this project, to support its present and future development.

**2. Algorithms they use**

**recursive swapping algorithm**

[**https://lalescu.ro/liviu/fet/doc/en/generation-algorithm-description.html**](https://lalescu.ro/liviu/fet/doc/en/generation-algorithm-description.html)

[**https://lalescu.ro/liviu/fet/doc/en/generation-algorithm-details.html**](https://lalescu.ro/liviu/fet/doc/en/generation-algorithm-details.html)

[**https://lalescu.ro/liviu/fet/doc/en/generation-algorithm-references.html**](https://lalescu.ro/liviu/fet/doc/en/generation-algorithm-references.html) **--** References for the idea of recursive swapping algorithm:

**3. OpenSourse?**

open source

**4. Installation**

FET needs a computer that can run [Qt](https://www.timetabling.de/manual/www.qt.io). So it is possible to run it for example with Microsoft Windows, Apple MacOS or Linux. Even it is running with a Raspberry Pi 1 computer (it has got only 0,7 GHz and only 0,5 GB RAM), a faster computer is highly recommended. Also a screen size with at least 1280x800 is useful, because lower values will make it difficult to read the activity planning or some other dialogues. But in fact you can also run FET without a screen by using the command line version *fet-cl*.

**5. Web/Desktop**

Desktop

**6. Configure your own Rules**

Yes

**7. Limitation for scheduling(Rooms, classes, teachers)**

There are indeed maximum limits for the generation algorithm (all these limits can be increased on demand, as a custom version, because this requires a bit more memory).

These limits are:

- Maximum total number of hours (periods) per day: 60;

- Maximum number of working days per week: 35;

- Maximum total number of teachers: 6000

- Maximum total number of sets of students: 30000

- Virtually unlimited number of subjects

- Virtually unlimited number of activity tags

- Maximum number of activities: 30000

- Maximum number of rooms: 6000

- Maximum number of buildings: 6000

- Virtually unlimited number of teachers and students sets for each activity

- Virtually unlimited number of time constraints

- Virtually unlimited number of space constraints

**8. Time spend to schedule**

Depends on hardware and algorithm they use

**9. Integration with other systems(API) - DATA EXCHANGE**

FET uses text files, xml or html or txt or csv (comma separated values - for import/export). The used encoding is UTF-8.

**10. Interface:**

<https://lalescu.ro/liviu/fet/screenshots/english-1/index.html> -- Images

**11. Model they use(Constraint based or what)**

Constraints. They can be: time constraints (referring to the allocated day and hour) or space constraints (referring to rooms allocation). They have a weight percentage, from 0.0% to 100.0%. 100% means that the constraint will always be respected and if this constraint is impossible, FET will not be able to generate a timetable.

**12. Functional abilities:**

* FET is free software, licensed under the GNU Affero General Public License version 3 or later. You can freely use, copy, modify and redistribute it
* Localized to ar (Arabic), ca (Catalan), cs (Czech), da (Danish), de (German), el (Greek), en\_GB (British English), en\_US (US English, default), es (Spanish), eu (Basque), fa (Persian), fr (French), gl (Galician), he (Hebrew), hu (Hungarian), id (Indonesian), it (Italian), ja (Japanese), lt (Lithuanian), mk (Macedonian), ms (Malay), nl (Dutch), pl (Polish), pt\_BR (Brazilian Portuguese), ro (Romanian), ru (Russian), si (Sinhala), sk (Slovak), sq (Albanian), sr (Serbian), tr (Turkish), uk (Ukrainian), uz (Uzbek), vi (Vietnamese), zh\_CN (Chinese Simplified) and zh\_TW (Chinese Traditional) (incompletely for some languages)
* Fully automatic generation algorithm, allowing also semi-automatic or manual allocation
* Platform independent implementation, allowing running on GNU/Linux, Windows, Mac and any system that Qt supports
* Flexible modular XML format for the input file, allowing editing with an XML editor or by hand (besides FET interface)
* Import/export from CSV format
* The resulted timetables are exported into HTML, XML and CSV formats
* Flexible students structure, organized into sets: years, groups and subgroups. FET allows overlapping years and groups and non-overlapping subgroups. You can even define individual students (as separate sets)
* Each constraint has a weight percentage, from 0.0% to 100.0% (but some special constraints are allowed to have only 100% weight percentage)
* Limits for the algorithm (all these limits can be increased on demand, as a custom version, because this would require a bit more memory):  
  + Maximum number of working days per week: 35
  + Maximum total number of hours (periods) per day: 60
  + Maximum total number of teachers: 6000
  + Maximum total number of sets of students: 30000
  + Virtually unlimited number of subjects
  + Virtually unlimited number of activity tags
  + Maximum number of activities: 30000
  + Maximum number of rooms: 6000
  + Maximum number of buildings: 6000
  + Possibility of adding multiple teachers and students sets for each activity. (it is possible also to have no teachers or no students sets for an activity)
  + Virtually unlimited number of time constraints
  + Virtually unlimited number of space constraints
* A large and flexible palette of time constraints:  
  + Break periods
  + For teacher(s):  
    - Not available periods
    - Max/min days per week
    - Max gaps per day/week
    - Max hours daily/continuously
    - Max span per day
    - Min hours daily
    - Max hours daily/continuously with an activity tag
    - Min hours daily with an activity tag
    - Min gaps between an ordered pair of activity tags
    - Respect working in an hourly interval a max number of days per week
    - Min resting hours
  + For students (sets):  
    - Not available periods
    - Max days per week
    - Begins early (specify max allowed beginnings at second hour)
    - Max gaps per day/week
    - Max hours daily/continuously
    - Max span per day
    - Min hours daily
    - Max hours daily/continuously with an activity tag
    - Min hours daily with an activity tag
    - Min gaps between an ordered pair of activity tags
    - Respect working in an hourly interval a max number of days per week
    - Min resting hours
  + For an activity or a set of activities/subactivities:  
    - A single preferred starting time
    - A set of preferred starting times
    - A set of preferred time slots
    - Min/max days between them
    - End(s) students day
    - Same starting time/day/hour
    - Occupy max/min time slots from selection (two complex and flexible constraints, useful in many situations)
    - Consecutive, ordered (and ordered if same day), grouped (for 2 or 3 (sub)activities)
    - Not overlapping (also for activity tags)
    - Max/min simultaneous in selected time slots
    - Min gaps between a set of (sub)activities
* A large and flexible palette of space constraints:  
  + Room not available periods
  + For teacher(s):  
    - Home room(s)
    - Max room/building changes per day/week
    - Min gaps between room/building changes
  + For students (sets):  
    - Home room(s)
    - Max room/building changes per day/week
    - Min gaps between room/building changes
  + Preferred room(s):  
    - For a subject
    - For an activity tag
    - For a subject and an activity tag
    - Individually for a (sub)activity
  + For a set of activities:  
    - Have the same room if they are consecutive
    - Occupy a maximum number of different rooms