Package 'criticalpath'

February 5, 2021

Title An Implementation of the Critical Path Method

Version 0.1.0

Author Rubens Jose Rosa [aut, cre],
Marcos dos Santos [aut],
Thiago Marques [aut]

Maintainer Rubens Jose Rosa <rubens@rubensjoserosa.com>

URL https://rubensjoserosa.com/criticalpath,
https://github.com/rubens2005/criticalpath

BugReports https://github.com/rubens2005/criticalpath/issues

Description An R6 object oriented implementation of the Critical Path Method (CPM).
CPM is a method used to estimate the minimum project duration and determine
the amount of scheduling flexibility on the logical network paths within the
schedule model. The flexibility is in terms of early start, early finish,
late start, late finish, total float and free float. Beside, it permits
to quantify the complexity of network diagram through the analysis of
topological indicators. Finally, it permits to change the activities duration
to perform what-if scenario analysis. The package was built based on following
references: To make topological sorting and other graph operation, we use
Csardi, G. & Nepusz, T. (2005)
<https://www.researchgate.net/publication/221995787_The_Igraph_Software_Package_for_Complex_Network_Research
For schedule concept, the reference was Project Management Institute (2017)
<https://www.pmi.org/pmbok-guide-standards/foundational/pmbok>;
For standards terms, we use Project Management Institute (2017)
<https://www.pmi.org/pmbok-guide-standards/lexicon>;
For algorithms on Critical Path Method development, we use
Vanhoucke, M. (2013) <doi:10.1007/978-3-642-40438-2> and
Vanhoucke, M. (2014) <doi:10.1007/978-3-319-04331-9>;
And, finally, for topological definitions, we use
Vanhoucke, M. (2009) <doi:10.1007/978-1-4419-1014-1>.

License MIT + file LICENSE

Encoding UTF-8

LazyData true

RoxygenNote 7.1.1

Imports igraph, R6

Suggests DiagrammeR, knitr, rmarkdown, testthat

Collate 'Schedule.R' 'criticalpath.R'

VignetteBuilder knitr

NeedsCompilation no

Repository CRAN

Date/Publication 2021-02-05 10:50:02 UTC

R topics documented:

criticalpath 2
Schedule 4

Index **32**

`criticalpath` *criticalpath: Object Oriented Critical Path Method Implementation*

Description

criticalpath package is an object oriented implementation of the Critical Path Method (CPM) in R with R6 library. CPM is a method used to estimate the minimum project duration and determine the amount of scheduling flexibility on the logical network paths within the schedule model. The flexibility is in terms of early start, early finish, late start, late finish, total float and free float. Besides, it permits to quantify the complexity of network diagram through the analysis of topological indicators. Finally, it permits to change the activities duration to perform what-if scenario analysis.

With this package, you can calculate the following CPM parameters:

- Schedule duration
- Early start and finish date of each activity
- Late start and finish date of each activity
- Critical activities
- Critical path
- Total float and free float
- Gantt Matrix
- What-if scenario analysis
- Topological indicators

Author(s)

Rubens Jose Rosa (<rubens@rubensjoserosa.com>), Marcos dos Santos (<marcosdossantos@ime.eb.br>), Thiago Marques (<profestathimarques@gmail.com>)

References

Csardi, G. & Nepusz, T. (2005). The Igraph Software Package for Complex Network Research. *InterJournal*. Complex Systems. 1695.

Project Management Institute (2017) **A Guide to the Project Management Body of Knowledge (PMBOK Guide)**. Sixth Edition.

Project Management Institute (2017) **PMI Lexicon of Project Management Terms:** Version 3.2.

Vanhoucke, M. (2009) **Measuring Time**: Improving Project Performance Using Earned Value Management. Springer-Verlag US.

Vanhoucke, M. (2013) **Project Management with Dynamic Scheduling**: Baseline Scheduling, Risk Analysis and Project Control. Springer-Verlag Berlin Heidelberg.

Vanhoucke, M. (2014) **Integrated Project Management and Control**: First Comes the Theory, then the Practice. Springer International Publishing Switzerland.

See Also

On vignette package there are more information with examples about:

- Schedule Class Definition: Schedule
- How to create a schedule:
  - Add activities and relations together to an schedule.
  - Add activities to a schedule.
  - Add relations to a schedule.
  - Create a schedule object from data frames.
- How to get schedule information:
  - Title, Reference and Schedule Duration.
- How to get activities properties:
  - Activity Properties.
  - Gantt Matrix.
- How to change activities duration:
  - Change Activities Duration.
- How to get relations properties:
  - Relation Properties
  - Successors and Predecessors.
- How to get topological properties:
  - Topological Indicators.

`Schedule` *R6 Class Representing a Schedule*

Description

This class is a representation of Precedence Diagramming Method (PDM). PDM is a technique used for constructing a schedule model in which activities are represented by nodes and are graphically linked by one or more logical relationships to show the sequence in which the activities are to be performed.

A schedule has activities and relations data-frames. With this class, it is possible to apply critical path method

Active bindings

`title` A project title for identification. It depends on user of the class. Its use are:

- `Schedule$title <-"A title"`
  - sets a title for a project.
- `Schedule$title`
  - gets the title of the project.

`reference` A reference from project origin, for example, a book, a paper, a corporation, or nothing. Its uses are:

- `Schedule$reference <-"A reference"`
  - sets a reference for a project.
- `Schedule$title`
  - gets the reference of the project.

`has_any_activity` A logical value that indicates if the schedule has any activity. A TRUE value means that the schedule has some activity; a FALSE, means that the schedule is empty.

- Usage: `Schedule$has_any_activity`

`nr_activities` Number of activities in a schedule as an integer value.

- Usage: `Schedule$nr_activities`

`activities` Return a data frame with all activities of a schedule in an activity id order. This is the main information calculated by CPM. The data frame is formed by following structure:

- **id:** Activity id.
- **name:** The name of activity.
- **duration:** A number that represents the activity's duration.
- **milestone:** A milestone is an activity with zero duration. This property indicates if an activity is a milestone or not: `TRUE` indicates it is a milestone; `FALSE` indicates it is not.
- **critical:** A critical activity is one with total float minor or equal to zero. This property indicates if an activity is critical: `TRUE` indicates it is critical; `FALSE` indicates it is not critical.
- **ES:** Early Start: is the earliest start period an activity can begin after its predecessors without violating precedence relation.

- **EF:** Early Finish: is the early start plus activity duration.
- **LS:** Late Start: is the late finish minus activity duration.
- **LF:** Late Finish: is the latest finish an activity can finish before their successors without violating precedence relation.
- **total_float:** It is the amount of period an activity can be delayed without violating the project duration. Its formula is: LS - ES or LF - EF.
- **free_float:** It is the amount of period an activity can be delayed without violating the start time of the successors activities.
- **progr_level:** Progressive level is the rank of activities counted from begin. The level of the activities that don't have predecessor is one; the level of the other activities, is one plus the maximal level of their predecessor.
- **regr_level:** Regressive level is the rank of activities counted from the end. The level of the activities that don't have successor is the maximal progressive level; the level of the other activities, is one minus the minimal level of their successor.
- **topo_float:** It is the difference between progressive level and regressive level.
- Usage: `Schedule$activities`

`has_any_relation` A logical value that indicates if the schedule has any relation. A TRUE value means that the schedule has some relation; a FALSE, means that the schedule does not have any relation.

- Usage: `Schedule$has_any_relation`

`nr_relations` Number of relations in a schedule as an integer value.

- Usage: `Schedule$nr_relations`

`relations` Return a data frame with all relations of a schedule in topological order. This is the main information calculated by CPM. The data frame is formed by following structure:

- **from:** Predecessor activity id from a relation.
- **to:** Successor activity id from a relation.
- **type:** The type of relation between activities. Its value may be: FS, FF, SS, SF.
- **lag:** The time period between activity predecessor and activity successor activity
- **critical:** A critical relation formed by two activity critical: predecessor and successor. `TRUE` indicates it is critical; `FALSE` indicates it is not critical.
- **ord:** Indicates de order that the relation was added in the schedule.
- **i_from:** It is the index of predecessor activity in the activities data frame.
- **i_to:** It is the index of successor activity in the activities data frame.
- Usage: `Schedule$relations`

`duration` An integer value that indicates the duration of a schedule.

Methods

Public methods:

- `Schedule$new()`
- `Schedule$add_activity()`
- `Schedule$add_activities()`
- `Schedule$get_activity()`

- Schedule$add_relation()
- Schedule$add_relations()
- Schedule$add_act_rel()
- Schedule$print()
- Schedule$all_successors()
- Schedule$all_predecessors()
- Schedule$is_redundant()
- Schedule$change_durations()
- Schedule$gantt_matrix()
- Schedule$xy_gantt_matrix()
- Schedule$topoi_sp()
- Schedule$topoi_ad()
- Schedule$topoi_la()
- Schedule$topoi_tf()
- Schedule$clone()

Method `new()`**:** Make a schedule with activities and relations between activities. The method `Schedule$new(activities,relations)` creates an schedule object from two data frames, one containing activities lists and the other the precedence relations between activities. After creation, it is applied the Critical Path Method (CPM).

It is possible to create a empty schedule, without any activity or relation with the constructor `Schedule$new()`. After that, it is possible to add activity with `add_activity` and relation with `add_relation` methods.

Usage:

```
Schedule$new(activities = NULL, relations = NULL)
```

Arguments:

`activities` Data frame with activities. If it is not informed, the schedule will be created without any activity. Its structure is:

- **id:** Activity id. It is an integer number that must be unique within a schedule.
- **name:** Activity name. It may be empty.
- **duration:** Activity duration. It is integer number without unit time. It may be zero.

`relations` Data frame with precedence relations between activities. If it is informed, the activities has to be informed too. If it is not informed, the schedule will be created without any relation. It is formed by predecessor activity e successor activity. Its structure is:

- **from:** The id of predecessor activity. Must exist an activity with from id.
- **to:** The id of successor activity. Must exist an activity with to id.
- **type:** Specifies the type of relation between activities. The default type is FS and its value may be: FS, FF, SS, SF, that means:
  - **FS:** Finish-Start relation. Activity to_id can only start after the finish of activity from_id.
  - **FF:** Finish-Finish relation. Activity to_id must finish together with activity from_id.
  - **SS:** Start-Start relation. Activity to_id must start together with activity from_id.
  - **SF:** Start-Finish relation. Activity to_id must finish when activity from_id starts.

- **lag:** The time period between activities that the successor activity must be advanced, or lated, after activity from_id. It must be an integer, less than, equal or greater than zero.

Returns: A Schedule object with CPM parameters calculated.

Examples:

```
# An empty schedule.
schedule <- Schedule$new()
schedule$duration
schedule$activities
schedule$relations

# A schedule with activities and relations.
activities <- data.frame(
  id        = 1:17,
  name      = paste("a", as.character(1:17), sep=""),
  duration  = c(1,2,2,4,3,3,3,2,1,1,2,1,1,1,1,2,1)
)

relations <- data.frame(
  from = c(1, 1, 2, 2, 2, 3, 3, 3,  3,  4,  5,  6,
           7,  8,  9, 10, 11, 11, 12, 12, 13, 13, 14, 14, 15, 15),
  to   = c(2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 11, 11,
           12, 13, 14, 15, 16, 17, 16, 17, 16, 17, 16, 17, 16, 17)
)
schedule <- Schedule$new(activities, relations)
schedule$title <- "Project 1: Cost Information System"
schedule$reference <- "VANHOUCKE, Mario.
Integrated project management and control:
  first comes the theory, then the practice.
  Gent: Springer, 2014, p. 6"
schedule$duration
schedule$activities
schedule$relations
```

Method `add_activity()`**:** Add an activity to a schedule.

Usage:

```
Schedule$add_activity(id, name = "", duration = 0)
```

Arguments:

`id` Activity id that will be used to make relation between activities. It must be unique.

`name` The name of activity. The default is an empty string.

`duration` A number that represents the activity's duration. It must be equal or greater than zero. The default value is zero.

Returns: A Schedule object with an activity added and the critical path calculated.

Examples:

```
schedule <- Schedule$new()
```

```
schedule$add_activity(1, "Task 1", 5)
schedule$add_activity(2, "Task 2", 6)
schedule$add_activity(3, "Task 3", 8)
schedule$add_activity(4, "Task 4", 6)
schedule$add_activity(5, "Task 5", 9)
schedule$add_activity(6, "Task 6", 3)
schedule$add_activity(7, "Task 7", 4)
schedule$duration
schedule$activities
```

Method `add_activities()`**:** Add activities from a data frame to a schedule.

Usage:

```
Schedule$add_activities(activities)
```

Arguments:

`activities` A data frame with the activities to be added.

Returns: A Schedule object with activities added and CPM calculated.

Examples:

```
activities <- data.frame(
  id        = 1:17,
  name      = paste("a", as.character(1:17), sep=""),
  duration  = c(1,2,2,4,3,3,3,2,1,1,2,1,1,1,1,2,1)
)
schedule <- Schedule$new()
schedule$add_activities(activities)
schedule$duration
schedule$activities
```

Method `get_activity()`**:** Gets an activity by id. It returns a data frame with one line about activity.

Usage:

```
Schedule$get_activity(id)
```

Arguments:

`id` An activity id as defined by the user.

Returns: A data frame with one line with the activity, or an error if activity id doesn't exist.

Examples:

```
x <- runif(1)
```

Method `add_relation()`**:** Add a relation to a schedule.

Usage:

```
Schedule$add_relation(from, to, type = "FS", lag = 0)
```

Arguments:

`from` The id of predecessor activity. Must exist an activity with from.

`to` The id of successor activity. Must exist an activity with to.

`type` Specifies the type of relation between activities. The default type is FS and its value may be: FS, FF, SS, SF, that means: If type is not defined, it is assumed to be FS.
FS: Finish-Start relation. Activity 'to' id can only start after the finish of activity 'from' id.
FF: Finish-Finish relation. Activity 'to' id must finish together with activity 'from' id.
SS: Start-Start relation. Activity 'to' id must start together with activity 'from' id.
SF: Start-Finish relation. Activity 'to' id must finish when activity 'from' id starts.

`lag` The time period between activities that the successor activity 'to' must be advanced after activity 'from' has been finished. The value may be negative, in such case, the activity 'to' will be anticipated 'lag' time periods. It must be an integer, less than, equal or greater than zero. If lag is not defined, it is assumed to be zero.

Returns: A Schedule object with CPM parameters calculated.

Examples:

```
# First, create an empty schedule
schedule <- Schedule$new()
schedule$title <- "Project 3: Old Carriage House Renovation"
schedule$reference <-
  "VANHOUCKE, Mario. Integrated project management and control:
  first comes the theory, then the practice. Gent: Springer, 2014, p. 11"

# Second, add activities to it
schedule$add_activity(1, "a1" , 2)
schedule$add_activity(2, "a2" , 2)
schedule$add_activity(3, "a3" , 4)
schedule$add_activity(4, "a4" , 3)
schedule$add_activity(5, "a5" , 4)
schedule$add_activity(6, "a6" , 1)
schedule$add_activity(7, "a7" , 1)
schedule$add_activity(8, "a8" , 1)
schedule$add_activity(9, "a9" , 1)
schedule$add_activity(10, "a10", 1)
schedule$add_activity(11, "a11", 3)
schedule$add_activity(12, "a12", 2)
schedule$add_activity(13, "a13", 1)
schedule$add_activity(14, "a14", 1)
schedule$add_activity(15, "a15", 2)
schedule$add_activity(16, "a16", 1)
schedule$add_activity(17, "a17", 1)

# Finally, add relations to it
schedule$add_relation( 1, 2)
schedule$add_relation( 2, 3)
schedule$add_relation( 3, 4)
schedule$add_relation( 4, 5)
schedule$add_relation( 5, 6)
schedule$add_relation( 6, 7)
schedule$add_relation( 6, 8)
schedule$add_relation( 6, 9)
schedule$add_relation( 7, 10)
```

```
schedule$add_relation( 8, 10)
schedule$add_relation( 9, 10)
schedule$add_relation( 10, 11)
schedule$add_relation( 10, 13)
schedule$add_relation( 11, 12)
schedule$add_relation( 12, 15)
schedule$add_relation( 13, 14)
schedule$add_relation( 14, 15)
schedule$add_relation( 15, 16)
schedule$add_relation( 16, 17)
schedule$duration
schedule$activities
schedule$relations
```

Method `add_relations()`**:** Add relations between activities from a data frame to a schedule.

Usage:

```
Schedule$add_relations(relations)
```

Arguments:

`relations` A data frame with the relations to be added.

Returns: A Schedule object with relations added and CPM calculated.

Examples:

```
# A schedule with activities and relations.
activities <- data.frame(
  id        = 1:17,
  name      = paste("a", as.character(1:17), sep=""),
  duration  = c(1,2,2,4,3,3,3,2,1,1,2,1,1,1,1,2,1)
)

relations <- data.frame(
  from = c(1, 1, 2, 2, 2, 3, 3, 3,  3,  4,  5,  6,
           7,  8,  9, 10, 11, 11, 12, 12, 13, 13, 14, 14, 15, 15),
  to   = c(2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 11, 11,
           12, 13, 14, 15, 16, 17, 16, 17, 16, 17, 16, 17, 16, 17)
)
schedule <- Schedule$new(activities)
schedule$title <- "Project 1: Cost Information System"
schedule$reference <- "VANHOUCKE, Mario.
Integrated project management and control:
  first comes the theory, then the practice.
  Gent: Springer, 2014, p. 6"
schedule$relations # Empty

schedule$add_relations(relations)
schedule$relations # Not empty
```

Method `add_act_rel()`**:** Add an activity and her relations to a schedule.

Usage:

```
Schedule$add_act_rel(
  id,
  name,
  duration,
  relations_id = c(),
  direction = "succ"
)
```

Arguments:

`id` Activity id. The id will be used to make relation between activities.

`name` The name of activity.

`duration` A number that represents the activity's duration. It must be equal or greater than zero.

`relations_id` A vector of ids such that will be linked with activity id. It may be relations of successor or predecessors.

`direction` Direction of relations_id: It may be "succ" or "pred". If dir="succ" the relations_id will be the successor of the activity. If dir="pred" the relations_id will be the predecessor of the activity.

Returns: A Schedule object.

Examples:

```
# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"

# Add activities and relations to it.
schedule$add_act_rel(  1, "a1" , 0, c(2,3,4))
schedule$add_act_rel(  2, "a2" , 4, c(5))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)
schedule$duration
schedule$activities
schedule$relations
```

Method `print()`**:** Print a description of the class

Usage:

```
Schedule$print(...)
```

Arguments:

`...` Variable parameters

Returns: A String .

Examples:

```
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"
schedule
```

Method `all_successors()`**:** List all successors from an activity: direct and indirect successors.

Usage:

```
Schedule$all_successors(id, ign_to = NULL)
```

Arguments:

`id` Activity id to be listed.

`ign_to` A relation to be ignored: id -> ign_to. Activities from this relation will be ignored.

Returns: A vector whith all activities ids.

Examples:

```
# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"

# Add activities and relations to it.
schedule$add_act_rel(  2, "a2" , 4, c(5, 12))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)

schedule$all_successors(2) # 5, 9, 12
schedule$all_successors(7) # 8, 11, 12
schedule$all_successors(10) # 12
```

Method `all_predecessors()`**:** List all predecessors from an activity: direct or indirect predecessors.

Usage:

```
Schedule$all_predecessors(id, ign_from = NULL)
```

Arguments:

`id` Activity id to be listed.

`ign_from` A relation to be ignored: ign_from -> id. Activities from this relation will be ignored.

Returns: A vector with all activities ids.

Examples:

```
# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"

# Add activities and relations to it.
schedule$add_act_rel(  2, "a2" , 4, c(5, 12))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)

schedule$all_predecessors(2) # nothing
schedule$all_predecessors(7) # 6, 4
schedule$all_predecessors(10) # 3
```

Method `is_redundant()`**:** Verify if a relation between two activities is redundant. A relation A->C is redundant if there are A->C, A->B, B->C relations.

Usage:

```
Schedule$is_redundant(id_from, id_to)
```

Arguments:

`id_from` From activity id.

`id_to` To activity id.

Returns: A logical `TRUE` if an arc is redundant; `FALSE` if it is not.

Examples:

```
# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"

# Add activities and relations to it.
schedule$add_act_rel(  2, "a2" , 4, c(5, 12))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)

schedule$is_redundant(2, 5)  #FALSE
schedule$is_redundant(2, 12) #TRUE
```

Method `change_durations()`**:** Change activities duration and calculate critical path. This way is faster than creating a new schedule with new durations.

Usage:

```
Schedule$change_durations(new_durations)
```

Arguments:

`new_durations` A vector with new activities' duration.

Returns: A Schedule object.

Examples:

```
activities <- data.frame(
  id        = 1:17,
  name      = paste("a", as.character(1:17), sep=""),
  duration  = c(1,1,3,2, 2,2,2,1, 4,5,3,3, 4,5,1,5,2)
)

relations <- data.frame(
  from = c(1, 2, 3, 3, 4, 5, 6, 7, 8,  8,  8,
    8,  8,  9, 10, 11, 12, 13, 13, 14, 14, 15, 15),
  to   = c(2, 3, 4, 6, 5, 8, 7, 8, 9, 10, 11,
   12, 13, 14, 14, 14, 14, 14, 15, 16, 17, 16, 17)
)

schedule <- Schedule$new(activities, relations)
schedule$title <- "Project 2: Patient Transport System"
```

```
schedule$reference <-
  "VANHOUCKE, Mario. Integrated project management and control:
  first comes the theory, then the practice. Gent: Springer, 2014, p. 9"
#Project duration
schedule$duration # 25
#Activities duration
schedule$activities$duration

# Now, change activities duration
new_durations <- c(1,2,5, 4,3, 2,1, 5, 3,5,5,3,4, 2,1, 2,4)
schedule$change_durations(new_durations)

#Project duration
schedule$duration # 31
#Activities duration
schedule$activities$duration
```

Method `gantt_matrix()`**:** Create a matrix that represents a Gantt chart, a matrix where "1" indicates that an activity is planned to be in execution.

In this matrix, the rows represent activities, whereas the columns represents the activity execution period. So, the number of columns is equal to project duration.

Usage:

```
Schedule$gantt_matrix()
```

Returns: A matrix where "1" indicates that an activity is in execution.

Examples:

```
activities <- data.frame(
  id        = c( 1,   2,   3,   4 ),
  name      = c("A", "B", "C", "D"),
  duration  = c( 2,   3,   1,   2 )
)
relations <- data.frame(
  from = c(1, 2, 4, 4),
  to   = c(3, 3, 1, 2)
)
schedule <- Schedule$new(activities, relations)
gantt <- schedule$gantt_matrix()
gantt
# What is the effort by time period?
colSums(gantt) # 1 1 2 2 1 1
# What is the duration by activities?
rowSums(gantt) # 2 3 1 2
# what is the S curve
cumsum(colSums(gantt))
plot(cumsum(colSums(gantt)), type="l", lwd=3)
```

Method `xy_gantt_matrix()`**:** Transform a Gantt matrix in x, y coordinates and the weight one. Each point greater than zero in a Gantt matrix becomes a x, y coordinate.

Usage:

```
Schedule$xy_gantt_matrix(gantt = NULL)
```

Arguments:

`gantt` A Gantt Matrix. If it is not informed, it will use `gantt_matrix()` before this function.

Returns: A matrix x, y and weight.

Examples:

```
activities <- data.frame(
  id        = c( 1,   2,   3,   4 ),
  name      = c("A", "B", "C", "D"),
  duration  = c( 2,   3,   1,   2 )
)
relations <- data.frame(
  from = c(1, 2, 4, 4),
  to   = c(3, 3, 1, 2)
)
schedule <- Schedule$new(activities, relations)
gantt <- schedule$gantt_matrix()
xyw <- schedule$xy_gantt_matrix()
xyw
plot(xyw[, 1:2])
```

Method `topoi_sp()`**: SP Serial or Parallel Topological Indicator:** It shows the closeness of a network to a serial or parallel graph. As the network becomes serial, the SP increase, until one, when the network totally serial.

Usage:

```
Schedule$topoi_sp()
```

Returns: A number between 0 and 1, inclusive.

Examples:

```
# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"

# Add activities and relations to it.
schedule$add_act_rel(  1, "a1" , 0, c(2,3,4))
schedule$add_act_rel(  2, "a2" , 4, c(5))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
```

```
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)

schedule$topoi_sp()
```

Method `topoi_ad()`**: AD Activity Distribution Topological Indicator:** Measures the distribution of the activities over the levels. If AD is approximately equal zero, each level has same numbers of activities. Otherwise, if AD is equal one, the quantity of each level is not uniformly distributed.

Usage:

```
Schedule$topoi_ad()
```

Returns: A number between 0 and 1, inclusive.

Examples:

```
# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"

# Add activities and relations to it.
schedule$add_act_rel(  1, "a1" , 0, c(2,3,4))
schedule$add_act_rel(  2, "a2" , 4, c(5))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)

schedule$topoi_ad()
```

Method `topoi_la()`**: LA Length of Arcs Topological Indicator:** Measures the presence of long arcs based on the difference between the progressive level of the end activity and the start node of each relation. If LA is approximately equal zero, the progressive level between activities is as far as possible. Otherwise, if LA is equal one, the relation distance are one.

Usage:

```
Schedule$topoi_la()
```

Returns: A number between 0 and 1, inclusive.

Examples:

```
# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"

# Add activities and relations to it.
schedule$add_act_rel(  1, "a1" , 0, c(2,3,4))
schedule$add_act_rel(  2, "a2" , 4, c(5))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)

schedule$topoi_la()
```

Method `topoi_tf()`**: TF Topological Float Indicator:** Measures the topological float of each activity. If TF = 0, there is no float between activities. If TF = 1, there is float between activities and they be shift without affecting other activities.

Usage:

```
Schedule$topoi_tf()
```

Returns: A number between 0 and 1, inclusive.

Examples:

```
# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"

# Add activities and relations to it.
schedule$add_act_rel(  1, "a1" , 0, c(2,3,4))
schedule$add_act_rel(  2, "a2" , 4, c(5))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
```

```
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)

schedule$topoi_tf()
```

Method `clone()`**:** The objects of this class are cloneable with this method.

Usage:

```
Schedule$clone(deep = FALSE)
```

Arguments:

`deep` Whether to make a deep clone.

Author(s)

Rubens Jose Rosa (<rubens@rubensjoserosa.com>), Marcos dos Santos (<marcosdossantos@ime.eb.br>), Thiago Marques (<profestathimarques@gmail.com>)

References

Csardi, G. & Nepusz, T. (2005). The Igraph Software Package for Complex Network Research. *InterJournal*. Complex Systems. 1695.

Project Management Institute (2017) **A Guide to the Project Management Body of Knowledge (PMBOK Guide)**. Sixth Edition.

Project Management Institute (2017) **PMI Lexicon of Project Management Terms:** Version 3.2.

Vanhoucke, M. (2009) **Measuring Time**: Improving Project Performance Using Earned Value Management. Springer-Verlag US.

Vanhoucke, M. (2013) **Project Management with Dynamic Scheduling**: Baseline Scheduling, Risk Analysis and Project Control. Springer-Verlag Berlin Heidelberg.

Vanhoucke, M. (2014) **Integrated Project Management and Control**: First Comes the Theory, then the Practice. Springer International Publishing Switzerland.

See Also

On vignette package there is more information with examples about:

- Critical Path Method Package criticalpath.
- How to create a schedule:
  - Add activities and relations together to an schedule.
  - Add activities to a schedule.
  - Add relations to a schedule.

    - Create a schedule object from data frames.
  - How to get schedule information:
    - Title, Reference and Schedule Duration.
  - How to get activities properties:
    - Activity Properties.
    - Gantt Matrix.
  - How to change activities duration:
    - Change Activities Duration.
  - How to get relations properties:
    - Relation Properties
    - Successors and Predecessors.
  - How to get topological properties:
    - Topological Indicators.

Examples

```
## ------------------------------------------------
## Property `Schedule$title`
## ------------------------------------------------
# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"
schedule$title

## ------------------------------------------------
## Property `Schedule$reference`
## ------------------------------------------------
# Create a schedule
schedule <- Schedule$new()
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"
schedule$reference

## ------------------------------------------------
## Property `Schedule$duration`
## ------------------------------------------------
# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"

# Add activities and relations to it.
```

```
schedule$add_act_rel(  1, "a1" , 0, c(2,3,4))
schedule$add_act_rel(  2, "a2" , 4, c(5))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)
schedule$duration

## ------------------------------------------------
## Property `Schedule$has_any_activity`
## ------------------------------------------------
# Create a schedule
schedule <- Schedule$new()
schedule$has_any_activity  # FALSE

# Add one activity.
schedule$add_activity( 1, "a1" , 0)
schedule$has_any_activity  # TRUE

## ------------------------------------------------
## Property `Schedule$nr_activities`
## ------------------------------------------------
# Create a schedule
schedule <- Schedule$new()
schedule$nr_activities     # 0

# Add one activity.
schedule$add_activity( 1, "a1" , 0)
schedule$nr_activities     # 1

## ------------------------------------------------
## Property `Schedule$activities`
## ------------------------------------------------
# Create a schedule
schedule <- Schedule$new()

# Add activities and relations to it.
schedule$add_act_rel(  1, "a1" , 0, c(2,3,4))
schedule$add_act_rel(  2, "a2" , 4, c(5))
schedule$add_act_rel(  3, "a3" , 9, c(6))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(6))
schedule$add_act_rel(  6, "a6" , 5)
schedule$activities

## ------------------------------------------------
## Property `Schedule$get_activity(id)`
```

```
## ------------------------------------------------
# Create a schedule
schedule <- Schedule$new()

# Add activities and relations to it.
schedule$add_act_rel(  1, "a1" , 0, c(2,3,4))
schedule$add_act_rel(  2, "a2" , 4, c(5))
schedule$add_act_rel(  3, "a3" , 9, c(6))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(6))
schedule$add_act_rel(  6, "a6" , 5)
schedule$get_activity(4)
schedule$get_activity(6)

## ------------------------------------------------
## Property `Schedule$has_any_relation`
## ------------------------------------------------
# Create a schedule
schedule <- Schedule$new()
schedule$has_any_relation   #FALSE

# Add activities and relations to it.
schedule$add_act_rel(1, "a1" , 0, c(2,3,4))
schedule$add_act_rel(2, "a2" , 4)
schedule$add_act_rel(3, "a3" , 9)
schedule$add_act_rel(4, "a4" , 1)
schedule$has_any_relation   # TRUE

## ------------------------------------------------
## Property `Schedule$nr_relations`
## ------------------------------------------------
# Create a schedule
schedule <- Schedule$new()
schedule$nr_relations   # 0

# Add activities and relations to it.
schedule$add_act_rel(1, "a1" , 0, c(2,3,4))
schedule$add_act_rel(2, "a2" , 4)
schedule$add_act_rel(3, "a3" , 9)
schedule$add_act_rel(4, "a4" , 1)
schedule$nr_relations   # 3

## ------------------------------------------------
## Property `Schedule$relations`
## ------------------------------------------------
# Create a schedule
schedule <- Schedule$new()
# Add activities and relations to it.
schedule$add_act_rel(1, "a1" , 0, c(2,3,4))
schedule$add_act_rel(2, "a2" , 4)
schedule$add_act_rel(3, "a3" , 9)
schedule$add_act_rel(4, "a4" , 1)
schedule$relations
```

```
## ------------------------------------------------
## Method `Schedule$new`
## ------------------------------------------------

# An empty schedule.
schedule <- Schedule$new()
schedule$duration
schedule$activities
schedule$relations

# A schedule with activities and relations.
activities <- data.frame(
  id        = 1:17,
  name      = paste("a", as.character(1:17), sep=""),
  duration  = c(1,2,2,4,3,3,3,2,1,1,2,1,1,1,1,2,1)
)

relations <- data.frame(
  from = c(1, 1, 2, 2, 2, 3, 3, 3,  3,  4,  5,  6,
           7,  8,  9, 10, 11, 11, 12, 12, 13, 13, 14, 14, 15, 15),
  to   = c(2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 11, 11,
           12, 13, 14, 15, 16, 17, 16, 17, 16, 17, 16, 17, 16, 17)
)
schedule <- Schedule$new(activities, relations)
schedule$title <- "Project 1: Cost Information System"
schedule$reference <- "VANHOUCKE, Mario.
Integrated project management and control:
  first comes the theory, then the practice.
  Gent: Springer, 2014, p. 6"
schedule$duration
schedule$activities
schedule$relations


## ------------------------------------------------
## Method `Schedule$add_activity`
## ------------------------------------------------

schedule <- Schedule$new()
schedule$add_activity(1, "Task 1", 5)
schedule$add_activity(2, "Task 2", 6)
schedule$add_activity(3, "Task 3", 8)
schedule$add_activity(4, "Task 4", 6)
schedule$add_activity(5, "Task 5", 9)
schedule$add_activity(6, "Task 6", 3)
schedule$add_activity(7, "Task 7", 4)
schedule$duration
schedule$activities

## ------------------------------------------------
## Method `Schedule$add_activities`
```

```
## ------------------------------------------------

activities <- data.frame(
  id        = 1:17,
  name      = paste("a", as.character(1:17), sep=""),
  duration  = c(1,2,2,4,3,3,3,2,1,1,2,1,1,1,1,2,1)
)
schedule <- Schedule$new()
schedule$add_activities(activities)
schedule$duration
schedule$activities

## ------------------------------------------------
## Method `Schedule$get_activity`
## ------------------------------------------------

x <- runif(1)

## ------------------------------------------------
## Method `Schedule$add_relation`
## ------------------------------------------------

# First, create an empty schedule
schedule <- Schedule$new()
schedule$title <- "Project 3: Old Carriage House Renovation"
schedule$reference <-
  "VANHOUCKE, Mario. Integrated project management and control:
  first comes the theory, then the practice. Gent: Springer, 2014, p. 11"

# Second, add activities to it
schedule$add_activity(1, "a1" , 2)
schedule$add_activity(2, "a2" , 2)
schedule$add_activity(3, "a3" , 4)
schedule$add_activity(4, "a4" , 3)
schedule$add_activity(5, "a5" , 4)
schedule$add_activity(6, "a6" , 1)
schedule$add_activity(7, "a7" , 1)
schedule$add_activity(8, "a8" , 1)
schedule$add_activity(9, "a9" , 1)
schedule$add_activity(10, "a10", 1)
schedule$add_activity(11, "a11", 3)
schedule$add_activity(12, "a12", 2)
schedule$add_activity(13, "a13", 1)
schedule$add_activity(14, "a14", 1)
schedule$add_activity(15, "a15", 2)
schedule$add_activity(16, "a16", 1)
schedule$add_activity(17, "a17", 1)

# Finally, add relations to it
schedule$add_relation( 1, 2)
schedule$add_relation( 2, 3)
schedule$add_relation( 3, 4)
schedule$add_relation( 4, 5)
```

```
schedule$add_relation( 5, 6)
schedule$add_relation( 6, 7)
schedule$add_relation( 6, 8)
schedule$add_relation( 6, 9)
schedule$add_relation( 7, 10)
schedule$add_relation( 8, 10)
schedule$add_relation( 9, 10)
schedule$add_relation( 10, 11)
schedule$add_relation( 10, 13)
schedule$add_relation( 11, 12)
schedule$add_relation( 12, 15)
schedule$add_relation( 13, 14)
schedule$add_relation( 14, 15)
schedule$add_relation( 15, 16)
schedule$add_relation( 16, 17)
schedule$duration
schedule$activities
schedule$relations

## ------------------------------------------------
## Method `Schedule$add_relations`
## ------------------------------------------------

# A schedule with activities and relations.
activities <- data.frame(
  id        = 1:17,
  name      = paste("a", as.character(1:17), sep=""),
  duration  = c(1,2,2,4,3,3,3,2,1,1,2,1,1,1,1,2,1)
)

relations <- data.frame(
  from = c(1, 1, 2, 2, 2, 3, 3, 3,  3,  4,  5,  6,
           7,  8,  9, 10, 11, 11, 12, 12, 13, 13, 14, 14, 15, 15),
  to   = c(2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 11, 11,
           12, 13, 14, 15, 16, 17, 16, 17, 16, 17, 16, 17, 16, 17)
)
schedule <- Schedule$new(activities)
schedule$title <- "Project 1: Cost Information System"
schedule$reference <- "VANHOUCKE, Mario.
Integrated project management and control:
  first comes the theory, then the practice.
  Gent: Springer, 2014, p. 6"
schedule$relations # Empty

schedule$add_relations(relations)
schedule$relations # Not empty


## ------------------------------------------------
## Method `Schedule$add_act_rel`
## ------------------------------------------------

# Create a schedule
```

```
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"

# Add activities and relations to it.
schedule$add_act_rel(  1, "a1" , 0, c(2,3,4))
schedule$add_act_rel(  2, "a2" , 4, c(5))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)
schedule$duration
schedule$activities
schedule$relations

## ------------------------------------------------
## Method `Schedule$print`
## ------------------------------------------------

schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"
schedule

## ------------------------------------------------
## Method `Schedule$all_successors`
## ------------------------------------------------


# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"

# Add activities and relations to it.
schedule$add_act_rel(  2, "a2" , 4, c(5, 12))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
```

```
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)

schedule$all_successors(2) # 5, 9, 12
schedule$all_successors(7) # 8, 11, 12
schedule$all_successors(10) # 12


## ------------------------------------------------
## Method `Schedule$all_predecessors`
## ------------------------------------------------


# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"

# Add activities and relations to it.
schedule$add_act_rel(  2, "a2" , 4, c(5, 12))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)

schedule$all_predecessors(2) # nothing
schedule$all_predecessors(7) # 6, 4
schedule$all_predecessors(10) # 3


## ------------------------------------------------
## Method `Schedule$is_redundant`
## ------------------------------------------------


# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"
```

```
# Add activities and relations to it.
schedule$add_act_rel(  2, "a2" , 4, c(5, 12))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)

schedule$is_redundant(2, 5)  #FALSE
schedule$is_redundant(2, 12) #TRUE


## ------------------------------------------------
## Method `Schedule$change_durations`
## ------------------------------------------------

activities <- data.frame(
  id        = 1:17,
  name      = paste("a", as.character(1:17), sep=""),
  duration  = c(1,1,3,2, 2,2,2,1, 4,5,3,3, 4,5,1,5,2)
)

relations <- data.frame(
  from = c(1, 2, 3, 3, 4, 5, 6, 7, 8,  8,  8,
    8,  8,  9, 10, 11, 12, 13, 13, 14, 14, 15, 15),
  to   = c(2, 3, 4, 6, 5, 8, 7, 8, 9, 10, 11,
   12, 13, 14, 14, 14, 14, 14, 15, 16, 17, 16, 17)
)

schedule <- Schedule$new(activities, relations)
schedule$title <- "Project 2: Patient Transport System"
schedule$reference <-
  "VANHOUCKE, Mario. Integrated project management and control:
  first comes the theory, then the practice. Gent: Springer, 2014, p. 9"
#Project duration
schedule$duration # 25
#Activities duration
schedule$activities$duration

# Now, change activities duration
new_durations <- c(1,2,5, 4,3, 2,1, 5, 3,5,5,3,4, 2,1, 2,4)
schedule$change_durations(new_durations)

#Project duration
schedule$duration # 31
#Activities duration
schedule$activities$duration
```

```
## ------------------------------------------------
## Method `Schedule$gantt_matrix`
## ------------------------------------------------

activities <- data.frame(
  id        = c( 1,   2,   3,   4 ),
  name      = c("A", "B", "C", "D"),
  duration  = c( 2,   3,   1,   2 )
)
relations <- data.frame(
  from = c(1, 2, 4, 4),
  to   = c(3, 3, 1, 2)
)
schedule <- Schedule$new(activities, relations)
gantt <- schedule$gantt_matrix()
gantt
# What is the effort by time period?
colSums(gantt) # 1 1 2 2 1 1
# What is the duration by activities?
rowSums(gantt) # 2 3 1 2
# what is the S curve
cumsum(colSums(gantt))
plot(cumsum(colSums(gantt)), type="l", lwd=3)


## ------------------------------------------------
## Method `Schedule$xy_gantt_matrix`
## ------------------------------------------------

activities <- data.frame(
  id        = c( 1,   2,   3,   4 ),
  name      = c("A", "B", "C", "D"),
  duration  = c( 2,   3,   1,   2 )
)
relations <- data.frame(
  from = c(1, 2, 4, 4),
  to   = c(3, 3, 1, 2)
)
schedule <- Schedule$new(activities, relations)
gantt <- schedule$gantt_matrix()
xyw <- schedule$xy_gantt_matrix()
xyw
plot(xyw[, 1:2])


## ------------------------------------------------
## Method `Schedule$topoi_sp`
## ------------------------------------------------

# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
```

```
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"

# Add activities and relations to it.
schedule$add_act_rel(  1, "a1" , 0, c(2,3,4))
schedule$add_act_rel(  2, "a2" , 4, c(5))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)

schedule$topoi_sp()


## ------------------------------------------------
## Method `Schedule$topoi_ad`
## ------------------------------------------------

# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"

# Add activities and relations to it.
schedule$add_act_rel(  1, "a1" , 0, c(2,3,4))
schedule$add_act_rel(  2, "a2" , 4, c(5))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)

schedule$topoi_ad()


## ------------------------------------------------
## Method `Schedule$topoi_la`
## ------------------------------------------------
```

```
# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"

# Add activities and relations to it.
schedule$add_act_rel(  1, "a1" , 0, c(2,3,4))
schedule$add_act_rel(  2, "a2" , 4, c(5))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)

schedule$topoi_la()


## ------------------------------------------------
## Method `Schedule$topoi_tf`
## ------------------------------------------------

# Create a schedule
schedule <- Schedule$new()
schedule$title <- "Fictitious Project Example"
schedule$reference <- "VANHOUCKE, Mario. Measuring time:
  improving project performance using earned value management.
  Gent: Springer, 2009, p. 18"

# Add activities and relations to it.
schedule$add_act_rel(  1, "a1" , 0, c(2,3,4))
schedule$add_act_rel(  2, "a2" , 4, c(5))
schedule$add_act_rel(  3, "a3" , 9, c(10))
schedule$add_act_rel(  4, "a4" , 1, c(6))
schedule$add_act_rel(  5, "a5" , 4, c(9))
schedule$add_act_rel(  6, "a6" , 5, c(7))
schedule$add_act_rel(  7, "a7" , 1, c(8,11))
schedule$add_act_rel(  8, "a8" , 7, c(12))
schedule$add_act_rel(  9, "a9" , 8, c(12))
schedule$add_act_rel( 10, "a10", 3, c(12))
schedule$add_act_rel( 11, "a11", 3, c(12))
schedule$add_act_rel( 12, "a12", 0)

schedule$topoi_tf()
```

Index

criticalpath, 2, *19*

Schedule, *3*, 4
schedule *(*Schedule*)*, 4