

Reference Manual

Generated by Doxygen 1.7.2

Thu Dec 16 2010 15:53:58

Contents

1	Class Index	1
1.1	Class List	1
2	Class Documentation	3
2.1	MTRand Class Reference	3
2.1.1	Detailed Description	4
2.2	tData Struct Reference	5
2.2.1	Detailed Description	6
2.3	tDemand Struct Reference	6
2.4	tDemandCostumer Struct Reference	6
2.4.1	Detailed Description	6
2.5	tPos Struct Reference	7
2.5.1	Detailed Description	7
2.6	tServiceTimeCostumer Struct Reference	7
2.6.1	Detailed Description	7
2.7	tSolomon Struct Reference	7
2.7.1	Detailed Description	8
2.8	tTimeWindow Struct Reference	8
2.9	tTimeWindowCostumer Struct Reference	8
2.10	tTimeWindowDepot Struct Reference	9
2.10.1	Detailed Description	9
2.11	VRPTWInstanceGenerator Class Reference	9
2.11.1	Constructor & Destructor Documentation	12
2.11.1.1	VRPTWInstanceGenerator	12
2.11.1.2	~VRPTWInstanceGenerator	12
2.11.2	Member Function Documentation	12
2.11.2.1	error	12
2.11.2.2	generateAll	12
2.11.2.3	generateDemands	12
2.11.2.4	generateDistanceMatrix	12
2.11.2.5	generateMatrices	13
2.11.2.6	generateServiceTimes	13
2.11.2.7	generateTimeMatrix	13
2.11.2.8	generateTimeWindows	13
2.11.2.9	getCategoryRandomly	13
2.11.2.10	getData	13
2.11.2.11	getPosition	14
2.11.2.12	print	14
2.11.2.13	printAll	14

2.11.2.14	printDistanceMatrix	14
2.11.2.15	printParameters	14
2.11.2.16	printSpecifications	15
2.11.2.17	printTimeMatrix	15
2.11.2.18	readDemandsFile	15
2.11.2.19	readDistancesFile	15
2.11.2.20	readIds	15
2.11.2.21	readPositions	16
2.11.2.22	readServiceTimesFile	16
2.11.2.23	readTimesFile	16
2.11.2.24	readTimeWindowsFile	17
2.11.2.25	setPrefix	17
2.11.2.26	setSeed	17
2.11.2.27	setSize	17
2.11.2.28	warning	18
2.11.2.29	writeAll	18
2.11.2.30	writeDistanceMatrix	18
2.11.2.31	writeSpecifications	18
2.11.2.32	writeTimeMatrix	18

Chapter 1

Class Index

1.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

MTRand (Mersenne Twister random number generator)	3
tData	5
tDemand	6
tDemandCostumer	6
tPos	7
tServiceTimeCostumer	7
tSolomon	7
tTimeWindow	8
tTimeWindowCostumer	8
tTimeWindowDepot	9
VRPTWInstanceGenerator	9

Chapter 2

Class Documentation

2.1 MTRand Class Reference

Mersenne Twister random number generator.

```
#include <MersenneTwister.h>
```

Public Types

- enum { **N** = 624 }
- enum { **SAVE** = N + 1 }
- typedef unsigned long **uint32**

Public Member Functions

- **MTRand** (const uint32 &oneSeed)
- **MTRand** (uint32 *const bigSeed, uint32 const seedLength=N)
- double **rand** ()
- double **rand** (const double &n)
- double **randExc** ()
- double **randExc** (const double &n)
- double **randDbtExc** ()
- double **randDbtExc** (const double &n)
- uint32 **randInt** ()
- uint32 **randInt** (const uint32 &n)
- double **operator()** ()
- double **rand53** ()
- double **randNorm** (const double &mean=0.0, const double &variance=0.0)
- void **seed** (const uint32 oneSeed)
- void **seed** (uint32 *const bigSeed, const uint32 seedLength=N)
- void **seed** ()

- void **save** (uint32 *saveArray) const
- void **load** (uint32 *const loadArray)

Protected Types

- enum { **M** = 397 }

Protected Member Functions

- void **initialize** (const uint32 oneSeed)
- void **reload** ()
- uint32 **hiBit** (const uint32 &u) const
- uint32 **loBit** (const uint32 &u) const
- uint32 **loBits** (const uint32 &u) const
- uint32 **mixBits** (const uint32 &u, const uint32 &v) const
- uint32 **twist** (const uint32 &m, const uint32 &s0, const uint32 &s1) const

Static Protected Member Functions

- static uint32 **hash** (time_t t, clock_t c)

Protected Attributes

- uint32 **state** [N]
- uint32 * **pNext**
- int **left**

Friends

- std::ostream & **operator**<< (std::ostream &os, const [MTRand](#) &mtrand)
- std::istream & **operator**>> (std::istream &is, [MTRand](#) &mtrand)

2.1.1 Detailed Description

Mersenne Twister random number generator. [MersenneTwister.h](#) Mersenne Twister random number generator -- a C++ class [MTRand](#) Based on code by Makoto Matsumoto, Takuji Nishimura, and Shawn Cokus Richard J. Wagner v1.0 15 May 2003 rjwagner@writeme.com

The Mersenne Twister is an algorithm for generating random numbers. It was designed with consideration of the flaws in various other generators. The period, $2^{19937}-1$, and the order of equidistribution, 623 dimensions, are far greater. The generator is also fast; it avoids multiplication and division, and it benefits from caches and pipelines. For more information see the inventors' web page at <http://www.math.keio.ac.jp/~matumoto/emt.html>

Reference M. Matsumoto and T. Nishimura, "Mersenne Twister: A 623-Dimensionally Equidistributed Uniform Pseudo-Random Number Generator", ACM Transactions on Modeling and Computer Simulation, Vol. 8, No. 1, January 1998, pp 3-30.

Copyright (C) 1997 - 2002, Makoto Matsumoto and Takuji Nishimura, Copyright (C) 2000 - 2003, Richard J. Wagner All rights reserved.

Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following conditions are met:

1. Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
2. Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
3. The names of its contributors may not be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

The original code included the following notice:

When you use this, send an email to: matumoto@math.keio.ac.jp with an appropriate reference to your work.

It would be nice to CC: rjwagner@writeme.com and Cokus@math.washington.edu when you write.

The documentation for this class was generated from the following file:

- MersenneTwister.h

2.2 tData Struct Reference

```
#include <dataTypes.h>
```

Public Attributes

- unsigned from

- unsigned **to**
- double **length**

2.2.1 Detailed Description

Cost between nodes: Structure to store the cost between pairs of node.

The documentation for this struct was generated from the following file:

- `dataTypes.h`

2.3 tDemand Struct Reference

Public Attributes

- unsigned **delta**
- `std::vector< tDemandCostumer >` **demandsCostumers**

The documentation for this struct was generated from the following file:

- `dataTypes.h`

2.4 tDemandCostumer Struct Reference

```
#include <dataTypes.h>
```

Public Attributes

- double **type**
- unsigned **probability**

2.4.1 Detailed Description

Demand Specification Structure: "delta" is defined as the percentage change in capacity of the fleet of vehicles. This value will be bounded between 0 and 100. The bigger this value is, the more capacity the vehicles will have. Moreover, we will also specify the demands with their respective probabilities.

The documentation for this struct was generated from the following file:

- `dataTypes.h`

2.5 tPos Struct Reference

```
#include <dataTypes.h>
```

Public Attributes

- unsigned **id**
- double **lat**
- double **lng**

2.5.1 Detailed Description

Position of the costumers: For each costumer, we will hold the the original id and its position (lat, lng).

The documentation for this struct was generated from the following file:

- `dataTypes.h`

2.6 tServiceTimeCostumer Struct Reference

```
#include <dataTypes.h>
```

Public Attributes

- double **type**
- unsigned **probability**

2.6.1 Detailed Description

Service Time Specification Structure For each costumer, a type of time service is defined with its probability

The documentation for this struct was generated from the following file:

- `dataTypes.h`

2.7 tSolomon Struct Reference

```
#include <dataTypes.h>
```

Public Attributes

- unsigned **id**
- double **X**
- double **Y**
- double **demand**
- unsigned long **opens**
- unsigned long **closes**
- unsigned long **serviceTime**

2.7.1 Detailed Description

Solomon-like Specification Structure In order to output the general specifications of the costumers, we will use the structure emplied in the VRP Solomon's instance-set.

The documentation for this struct was generated from the following file:

- `dataTypes.h`

2.8 tTimeWindow Struct Reference

Public Attributes

- [tTimeWindowDepot](#) `timeWindowDepot`
- `std::vector< tTimeWindowCostumer >` `timeWindowsCostumers`

The documentation for this struct was generated from the following file:

- `dataTypes.h`

2.9 tTimeWindowCostumer Struct Reference

Public Attributes

- double **opens**
- double **closes**
- unsigned **probability**

The documentation for this struct was generated from the following file:

- `dataTypes.h`

2.10 tTimeWindowDepot Struct Reference

```
#include <dataTypes.h>
```

Public Attributes

- double **opens**
- double **closes**

2.10.1 Detailed Description

Time Window Scepecifications Structure: In order to avoid the use of inheritance and reduce the possible overhead to the minimum, we will use two structs, one for the depot and another for the costumers.

The documentation for this struct was generated from the following file:

- `dataTypes.h`

2.11 VRPTWInstanceGenerator Class Reference

Public Member Functions

- [VRPTWInstanceGenerator](#) ()
Default Ctor.
- [~VRPTWInstanceGenerator](#) ()
Default Destructor.
- void [readDistancesFile](#) (const char *distanceFileName)
Method that reads the distance between each par of costumers.
- void [readTimesFile](#) (const char *timeFilename)
Method that reads the time between each par of costumers.
- void [readIds](#) (const char *idsFileName)
Method that reads the real ids of the costumers.
- void [readPositions](#) (const char *idslatlngFileName)
Method that reads the position of the costumers.
- void [readTimeWindowsFile](#) (const char *fileName)
Method that parsers the time windows specifications.

- void [readDemandsFile](#) (const char *fileName)
Method that parsers the demands specifications.
- void [readServiceTimesFile](#) (const char *fileName)
Method that parsers the service times specifications.
- void [setSize](#) (unsigned)
Sets the size of the instance.
- void [setSeed](#) (const std::string &, unsigned)
Sets the seeds to generate the data for this instance.
- void [setPrefix](#) (const std::string &)
Sets the prefix for the output files.
- void [generateMatrices](#) ()
Method that generates the distance and time windows matrices with random costumers.
- void [generateTimeWindows](#) ()
Method that associates a type of time window to each costumer.
- void [generateDemands](#) ()
Method that associates a type of demand to each costumer.
- void [generateServiceTimes](#) ()
Method that associates a type of service time to each costumer.
- void [generateAll](#) ()
Method that call all generates.
- void [printParameters](#) ()
Method to print the parameters.
- void [printTimeMatrix](#) ()
Method to print the time matrix.
- void [printDistanceMatrix](#) ()
Method to print the distance matrix.
- void [printSpecifications](#) ()
Method to print the specifications.
- void [print](#) ()
Method that prints all.

- void [printAll](#) ()
Method that prints all.
- void [writeDistanceMatrix](#) ()
Method that writes the distance matrix to a file.
- void [writeTimeMatrix](#) ()
Method that writes the time matrix to a file.
- void [writeSpecifications](#) ()
Method that writes the specification to a file.
- void [writeAll](#) ()
Method that call all writes.

Protected Member Functions

- double [getData](#) (const rawInfoVector &, unsigned, unsigned)
Method that, given a table and 'from' 'to' information, returns the length in time or distance.
- unsigned [getCategoryRandomly](#) (double, const accProbabilityType &)
This method will receive a random number generator and a vector of accumulative probability and will return a category.
- void [error](#) (const std::string &)
Method to output errors.
- void [warning](#) (const std::string &)
Method to output warnings.
- unsigned [getPosition](#) (unsigned)
Method to get the position of a costumer within vector.
- void [generateDistanceMatrix](#) ()
Method to generate the distance matrix for this instance.
- void [generateTimeMatrix](#) ()
Method to generate the time matrix for this instance.

2.11.1 Constructor & Destructor Documentation

2.11.1.1 VRPTWInstanceGenerator::VRPTWInstanceGenerator ()

Default Ctor.

Ctor. It sets default values for data members

2.11.1.2 VRPTWInstanceGenerator::~~VRPTWInstanceGenerator ()

Default Destructor.

Dtor. It does nothing

2.11.2 Member Function Documentation

2.11.2.1 void VRPTWInstanceGenerator::error (const std::string & *errorMessage*) [protected]

Method to output errors.

Method that outputs message and force the exit of the app.

Parameters

<i>errorMes- sage</i>	is the message to be output.
---------------------------	------------------------------

2.11.2.2 void VRPTWInstanceGenerator::generateAll ()

Method that call all generates.

Method that invokes serveral other methods to generate the instance.

2.11.2.3 void VRPTWInstanceGenerator::generateDemands ()

Method that associates a type of demand to each costumer.

Method that generates a vector of size (size) with all the demands.

2.11.2.4 void VRPTWInstanceGenerator::generateDistanceMatrix () [protected]

Method to generate the distance matrix for this instance.

Method that generates the distance matrix using random ids.

2.11.2.5 void VRPTWInstanceGenerator::generateMatrices ()

Method that generates the distance and time windows matrices with random costumers.

Method that generates a vector of random ids (size value random ids) and invokes generateDistanceMatrix and [generateTimeMatrix\(\)](#).

2.11.2.6 void VRPTWInstanceGenerator::generateServiceTimes ()

Method that associates a type of service time to each costumer.

Method that generates a vector of size (size) with all the time services.

2.11.2.7 void VRPTWInstanceGenerator::generateTimeMatrix () [protected]

Method to generate the time matrix for this instance.

Method that generates the travel time matrix using random ids.

2.11.2.8 void VRPTWInstanceGenerator::generateTimeWindows ()

Method that associates a type of time window to each costumer.

Method that generates a vector of size (size) with all the time windows.

2.11.2.9 unsigned VRPTWInstanceGenerator::getCategoryRandomly (double *randNumber*, const accProbabilityType & *accProbability*) [protected]

This method will receive a random number generator and a vector of accumulative probability and will return a category.

Method that given a vector with accumulate probabilities returns the section in which a random number falls.

Parameters

<i>randNumber</i>	is the random number we want to know the place it falls in
<i>accProbability</i>	is the vector containing the accumulate probabilities.

Returns

the section in which the random number falls.

2.11.2.10 double VRPTWInstanceGenerator::getData (const rawInfoVector & *table*, unsigned *from*, unsigned *to*) [protected]

Method that, given a table and 'from' 'to' information, returns the length in time or distance.

Method that given the two first two elements of a table of nx3 returns the third element.

Parameters

<i>table</i>	is the matrix in which we want to look for the third element.
<i>from</i>	is the first element or index.
<i>to</i>	is the second element or index.

Returns

the value of the third element.

2.11.2.11 unsigned VRPTWInstanceGenerator::getPosition (unsigned *id*) [protected]

Method to get the position of a costumer within vector.

Method that returns the index of a given element (*id*) within the positions vector

Parameters

<i>id</i>	is the id of the element we are looking for
-----------	---

Returns

the position (index) within the vector

2.11.2.12 void VRPTWInstanceGenerator::print ()

Method that prints all.

Method that invokes all printing methods.

2.11.2.13 void VRPTWInstanceGenerator::printAll ()

Method that prints all.

Method that invokes all printing methods.

2.11.2.14 void VRPTWInstanceGenerator::printDistanceMatrix ()

Method to print the distance matrix.

Method that prints the distance matrix.

2.11.2.15 void VRPTWInstanceGenerator::printParameters ()

Method to print the parameters.

Method that prints the parameters given by the user (debug purposes).

2.11.2.16 void VRPTWInstanceGenerator::printSpecifications ()

Method to print the specifications.

Method that prints the specifications.

2.11.2.17 void VRPTWInstanceGenerator::printTimeMatrix ()

Method to print the time matrix.

Method that prints the time matrix.

2.11.2.18 void VRPTWInstanceGenerator::readDemandsFile (const char * *fileName*)

Method that parsers the demands specifications.

Parameters

<i>fileName</i>	is the name of the file containing the demands specifications.
-----------------	--

Method that reads the demand specifications

Parameters

<i>fileName</i>	is the path to the file containing the demands specifications.
-----------------	--

2.11.2.19 void VRPTWInstanceGenerator::readDistancesFile (const char * *distanceFileName*)

Method that reads the distance between each par of costumers.

Parameters

<i>distanceFile-Name</i>	is the name of the file containing the distances.
--------------------------	---

Method that reads the distances from a file between pairs of costumers.

Parameters

<i>distanceFilename</i>	is the path to the file containing the distances information.
-------------------------	---

2.11.2.20 void VRPTWInstanceGenerator::readIds (const char * *idsFileName*)

Method that reads the real ids of the costumers.

Parameters

<i>distanceFileName</i>	is the name of the file containing the distances.
-------------------------	---

Method that reads the real ids of the costumers.

Parameters

<i>idsFileName</i>	is the path to the file containing the real ids.
--------------------	--

2.11.2.21 void VRPTWInstanceGenerator::readPositions (const char * *idslatlngFileName*)

Method that reads the position of the costumers.

Parameters

<i>idslatlngFileName</i>	is the name of the file containing the distances.
--------------------------	---

Method that reads the real position of the costumers (lat, lng).

Parameters

<i>idslatlngFileName</i>	is the path to the file containing the information.
--------------------------	---

2.11.2.22 void VRPTWInstanceGenerator::readServiceTimesFile (const char * *fileName*)

Method that parsers the service times specifications.

Parameters

<i>fileName</i>	is the name of the file containing the service times specifications.
-----------------	--

Method that reads the service times specifications.

Parameters

<i>fileName</i>	is the path to the file containing the specifications.
-----------------	--

2.11.2.23 void VRPTWInstanceGenerator::readTimesFile (const char * *timeFileName*)

Method that reads the time between each par of costumers.

Parameters

<i>timeFileName</i>	is the name of the file containing the times.
---------------------	---

Method that reads the travel times from a file between pairs of costumers.

Parameters

<i>timeFile-Name</i>	is the path to the file containing the travel time information.
----------------------	---

2.11.2.24 void VRPTWInstanceGenerator::readTimeWindowsFile (const char * *fileName*)

Method that parsers the time windows specifications.

Parameters

<i>fileName</i>	is the name of the file containing the time windows specifications.
-----------------	---

Method that reads the time windows specification *fileName* is the path to the file containing the time windows specification

2.11.2.25 void VRPTWInstanceGenerator::setPrefix (const std::string & *prefix*)

Sets the prefix for the output files.

Method to set the prefix for the output files

Parameters

<i>prefix</i>	is the prefix of the output files.
---------------	------------------------------------

2.11.2.26 void VRPTWInstanceGenerator::setSeed (const std::string & *object*, unsigned *value*)

Sets the seeds to generate the data for this instance.

Method to set the value of a seed.

Parameters

<i>object</i>	is the name of seed to be set up.
<i>value</i>	is the value to be set.

2.11.2.27 void VRPTWInstanceGenerator::setSize (unsigned *size*)

Sets the size of the instance.

Method to set the number of costumers of the instance.

Parameters

<i>size</i>	if the number of costumers.
-------------	-----------------------------

2.11.2.28 void VRPTWInstanceGenerator::warning (const std::string & *warningMessage*) [protected]

Method to output warnings.

Method that outputs a warning message.

Parameters

<i>warningMessage</i>	is the message to be output.
-----------------------	------------------------------

2.11.2.29 void VRPTWInstanceGenerator::writeAll ()

Method that call all writes.

Method that invokes all writing methods.

2.11.2.30 void VRPTWInstanceGenerator::writeDistanceMatrix ()

Method that writes the distance matrix to a file.

Method that creates the file containing the distance matrix.

2.11.2.31 void VRPTWInstanceGenerator::writeSpecifications ()

Method that writes the specification to a file.

Method that creates the file containing the specifications.

2.11.2.32 void VRPTWInstanceGenerator::writeTimeMatrix ()

Method that writes the time matrix to a file.

Method that creates the file containing the time matrix.

The documentation for this class was generated from the following files:

- vrptwinstancegenerator.h
- vrptwinstancegenerator.cpp

Index

- ~VRPTWInstanceGenerator
 - VRPTWInstanceGenerator, [12](#)
- error
 - VRPTWInstanceGenerator, [12](#)
- generateAll
 - VRPTWInstanceGenerator, [12](#)
- generateDemands
 - VRPTWInstanceGenerator, [12](#)
- generateDistanceMatrix
 - VRPTWInstanceGenerator, [12](#)
- generateMatrices
 - VRPTWInstanceGenerator, [12](#)
- generateServiceTimes
 - VRPTWInstanceGenerator, [13](#)
- generateTimeMatrix
 - VRPTWInstanceGenerator, [13](#)
- generateTimeWindows
 - VRPTWInstanceGenerator, [13](#)
- getCategoryRandomly
 - VRPTWInstanceGenerator, [13](#)
- getData
 - VRPTWInstanceGenerator, [13](#)
- getPosition
 - VRPTWInstanceGenerator, [14](#)
- MTRand, [3](#)
- print
 - VRPTWInstanceGenerator, [14](#)
- printAll
 - VRPTWInstanceGenerator, [14](#)
- printDistanceMatrix
 - VRPTWInstanceGenerator, [14](#)
- printParameters
 - VRPTWInstanceGenerator, [14](#)
- printSpecifications
 - VRPTWInstanceGenerator, [14](#)
- printTimeMatrix
 - VRPTWInstanceGenerator, [15](#)
- readDemandsFile
 - VRPTWInstanceGenerator, [15](#)
- readDistancesFile
 - VRPTWInstanceGenerator, [15](#)
- readIds
 - VRPTWInstanceGenerator, [15](#)
- readPositions
 - VRPTWInstanceGenerator, [16](#)
- readServiceTimesFile
 - VRPTWInstanceGenerator, [16](#)
- readTimesFile
 - VRPTWInstanceGenerator, [16](#)
- readTimeWindowsFile
 - VRPTWInstanceGenerator, [17](#)
- setPrefix
 - VRPTWInstanceGenerator, [17](#)
- setSeed
 - VRPTWInstanceGenerator, [17](#)
- setSize
 - VRPTWInstanceGenerator, [17](#)
- tData, [5](#)
- tDemand, [6](#)
- tDemandCostumer, [6](#)
- tPos, [7](#)
- tServiceTimeCostumer, [7](#)
- tSolomon, [7](#)
- tTimeWindow, [8](#)
- tTimeWindowCostumer, [8](#)
- tTimeWindowDepot, [9](#)
- VRPTWInstanceGenerator, [9](#)
 - ~VRPTWInstanceGenerator, [12](#)
 - error, [12](#)
 - generateAll, [12](#)
 - generateDemands, [12](#)
 - generateDistanceMatrix, [12](#)
 - generateMatrices, [12](#)
 - generateServiceTimes, [13](#)
 - generateTimeMatrix, [13](#)

- generateTimeWindows, [13](#)
- getCategoryRandomly, [13](#)
- getData, [13](#)
- getPosition, [14](#)
- print, [14](#)
- printAll, [14](#)
- printDistanceMatrix, [14](#)
- printParameters, [14](#)
- printSpecifications, [14](#)
- printTimeMatrix, [15](#)
- readDemandsFile, [15](#)
- readDistancesFile, [15](#)
- readIds, [15](#)
- readPositions, [16](#)
- readServiceTimesFile, [16](#)
- readTimesFile, [16](#)
- readTimeWindowsFile, [17](#)
- setPrefix, [17](#)
- setSeed, [17](#)
- setSize, [17](#)
- VRPTWInstanceGenerator, [12](#)
- warning, [18](#)
- writeAll, [18](#)
- writeDistanceMatrix, [18](#)
- writeSpecifications, [18](#)
- writeTimeMatrix, [18](#)

warning

- VRPTWInstanceGenerator, [18](#)

writeAll

- VRPTWInstanceGenerator, [18](#)

writeDistanceMatrix

- VRPTWInstanceGenerator, [18](#)

writeSpecifications

- VRPTWInstanceGenerator, [18](#)

writeTimeMatrix

- VRPTWInstanceGenerator, [18](#)