MOrepo result writer

Version 1.0.1

Generated by Doxygen 1.8.13

Contents

Chapter 1

An introduction to the MOrepoResultWriter

Author

Sune Lauth Gadegaard

Version

1.0.1

1.1 License

If you use the software in any academic work, please make a reference to

“An appropriate reference should go here!"

1.2 Description

This software provides a "converter" that given appropritate input generates a result file compatible witht format used on MOrepo. The software basically consists of a single class "MOrepoResultWriter". After constructing an object of the class, the simple "set" methods should be used to set the appropriate entries, and finally, after setting atleast all required entries and the output file path, the `writeFile ( )` function should be called.

1.3 Compiling

The codes were compiled using the Visual Studio 2015 compiler on a Windows 10 machine. The following flags were used: /W3 /Ox /std:[c++14|c++latest]

1.4 Example of usage

This section contains an example showing how the `converter` class can be used to generate a MOrepo compatible json file. In this example we assume the instance is named "instance1", that the contribution is "Foo et al. 2017", and that the class dataReader has the appropriate get methods.

```
  // main.cpp
#include"converter.h"
#include"dataReader.h" // Assume you have written this yourself
#include<vector>
int main(int argc, char** argv){
    try{
        std::string aString;
        MOrepoResultWriter MOwriter = MOrepoResultWriter ( );

        aString = "1.0";
        MOwriter.setVersion ( aString );

        aString = "Gadegaard16_UFLP_Klose_p01_0";
        MOwriter.setInstanceName ( aString );

        aString = "Gadegaard16";
        MOwriter.setContributionName ( aString );

        MOwriter.setObjectives ( 2 );

        std::vector<std::string> objTypes = { "float" , "int" };
        MOwriter.setObjectiveTypes ( objTypes );

        std::vector<std::string> directions = { "min" , "min" };
        MOwriter.setDirections ( directions );

        MOwriter.setOptimal ( true );

        MOwriter.setCardinality ( 6 );

        std::vector<std::vector<double>> points = { { 28463.2 , 92 }, { 28549.6 , 72 },
                                                    { 28550.5 , 69 }, { 28585.8 , 65 },
                                                    { 28606.2 , 60 } , { 28619.7 , 57} };
        std::vector<std::string> types = { "null","null", "null", "null", "null", "null" };
        MOwriter.setPoints ( points, types );

        MOwriter.setValid ( true );

        MOwriter.setExecutionTime ( 13.274200 );

        aString = "Intel Core i7-4785T 2.2 GHz, 16 GB RAM, Linux Ubuntu 64bit";
        MOwriter.setMachineSpecs ( aString );

        aString = "Results from the paper
```

```
                    'A bi-objective approach to discrete cost-bottleneck location problems'
                     by Gadegaard, Klose, Nielsen, Annals of Operations Research, 2016.";
        MOwriter.setComments ( aString );

        MOwriter.setSupportedCardinality ( 3 );

        MOwriter.setExtremeSupportedCardinality ( 3 );


        aString = "This is the miscellaneous entry.";
        MOwriter.setMisc ( aString );

        aString = "./Gadegaard16_UFLP_Klose_p01_0_results.json";
        MOwriter.setOutputFilePath ( aString );

        MOwriter.writeFile ( );
        return 0;
    }
    catch ( std::runtime_error& re )
    {
        std::cerr << re.what () << "\n";
    }
    catch(std::exception &e)
    {
        std::cerr << "Exception: " << e.what() << "\n";
    }
    catch(...)
    {
        std::cerr << "An unexpected exception was thrown. Caught in main.\n";
    }
}
```

1.5 Change log for MOrepoResultWriter.h and MOrepoResultWriter.cpp

FILE:	MOrepoResultWriter.h and MOrepoResultWriter.cpp
Version:	1.0.1

CHANGE LOG:	DATE	VER.-NO.	CHANGES MADE
	2017–07–01	1.0.0	First implementation
	2017–07–19	1.0.1	Added setOutputFilePath function. Added a patch for the std::to_stirng function, so it now compiles using gcc 4.9.2 The patch is in the namespace MOrepo.

Chapter 2

Class Index

2.1 Class List

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

Chapter 3

Class Documentation

3.1 MOrepoResultWriter Class Reference

Public Member Functions

- void **writeFile** ()
- void setVersion (std::string &version)

  Sets the version of the resultfile.
- void setInstanceName (std::string &instanceName)
- void setContributionName (std::string &contributionName)

  Sets the name of the contribution.
- void setObjectives (int objectives)
- void setObjectiveTypes (std::vector< std::string > &types)

  Sets the objective types.
- void setDirections (std::vector< std::string > &directions)

  Sets the directions of the objective functions.
- void setOptimal (bool isOptimal)
- void setCardinality (int cardinality)
- void setPoints (std::vector< std::vector< double >> &points, std::vector< std::string > &types)
- void setValid (bool valid)

  Sets the validity of the solution to either true or false.
- void setExecutionTime (double executionTime)

  Sets the execution time.
- void setMachineSpecs (std::string &machineSpecs)

  Sets the machine specifics.
- void setComments (std::string &comments)
- void setSupportedCardinality (int suppCard)
- void setExtremeSupportedCardinality (int extCard)
- void setMisc (std::string &misc)
- void setOutputFilePath (std::string &outputPath)

3.1.1 Member Function Documentation

3.1.1.1 setCardinality()

```
void MOrepoResultWriter::setCardinality (
            int cardinality ) [inline]
```

Sets the cardinality of the non–dominated frontier

Parameters

cardinality	integer specifying the number of points on the efficient frontier.

3.1.1.2 setComments()

```
void MOrepoResultWriter::setComments (
            std::string & comments )  [inline]
```

Sets the comments entry

Parameters

comments	reference to a string. Contains the comment that should be attached to the result file.

3.1.1.3 setContributionName()

```
void MOrepoResultWriter::setContributionName (
            std::string & contributionName )  [inline]
```

Sets the name of the contribution.

Sets the contribution name. It should be a string with the name of the contribution in which the instances and results have been published.

Parameters

contributionName	reference to a string. Contains the name of the constribution, e.g. "Pedersen08".

3.1.1.4 setDirections()

```
void MOrepoResultWriter::setDirections (
            std::vector< std::string > & directions )  [inline]
```

Sets the directions of the objective functions.

Sets the directions of the objective functions. The directions can either be "min" or "max".

Parameters

directions	reference to a vector of strings. If for example there are three objective functions where the two first are of the minimization-kind and the last is a maximization, we should specify a vector { "min" , "min" , "max" } as the function argument.

3.1.1.5 setExecutionTime()

```
void MOrepoResultWriter::setExecutionTime (
            double executionTime )  [inline]
```

Sets the execution time.

Sets the execution time, that is, the time in seconds it took to generate the solution contained in the results file.

Parameters

executionTime	double containing the execution time in seconds.

3.1.1.6 setExtremeSupportedCardinality()

```
void MOrepoResultWriter::setExtremeSupportedCardinality (
            int extCard )  [inline]
```

Sets the cardinality of the set of extreme supported non–dominated solutions

Parameters

extCard	integer containing the number of extreme supported non–dominated solutions.

3.1.1.7 setInstanceName()

```
void MOrepoResultWriter::setInstanceName (
            std::string & instanceName )  [inline]
```

Sets the name of the instance for which the result file contains information

Parameters

instanceName	reference to a string containing the name of the instance for which the results are for.

3.1.1.8 setMachineSpecs()

```
void MOrepoResultWriter::setMachineSpecs (
            std::string & machineSpecs )  [inline]
```

Sets the machine specifics.

Sets the specs of the machine used to carry out the experiments, e.g. "Intel Core i7-4785T 2.2 GHz, 16 GB RAM, Linux Ubuntu 64bit"

Parameters

machineSpecs	reference to a string. Contains the specs of the machine used to perform the experiments.

3.1.1.9 setMisc()

```
void MOrepoResultWriter::setMisc (
            std::string & misc )  [inline]
```

Sets the misc entry

Parameters

misc	reference to a string containing the misc that should be attached to the result file

Note

This entry may be used as you like. It could e.g. contain an object with more detailed entries about the experiment.

3.1.1.10 setObjectives()

```
void MOrepoResultWriter::setObjectives (
            int objectives )  [inline]
```

Sets the number of objectives

Parameters

objectives	integer specififying the number of objective functions of the multiobjective optimization problem.

3.1.1.11 setObjectiveTypes()

```
void MOrepoResultWriter::setObjectiveTypes (
            std::vector< std::string > & types )
```

Sets the objective types.

Sets the objective types to either int, float, or null (if unknown).

Parameters

types	reference to a vector of strings containing the type of each objective. That is if the i'th objective is integral, then objType[i] = "int"

Note

if the function "setObjectives()" has been called prior to calling setObjectiveTypes, and the number of objectives does not equal the lenght of the vector "types", a runtime error is thrown.

3.1.1.12 setOptimal()

```
void MOrepoResultWriter::setOptimal (
            bool isOptimal )  [inline]
```

Specifies whether the solution is an optimal solution to the specific instance or not.

Parameters

isOptimal	boolean. If isOptimal = true, it is assumed that the solutions is optimal solution, and if itsOptimal = false, it has not been verified optimal, or it is known to be suboptimal

Note

If setOptimal has not been called when creating the results file, the entry "optimal" is set to null.

3.1.1.13 setOutputFilePath()

```
void MOrepoResultWriter::setOutputFilePath (
            std::string & outputPath )  [inline]
```

Sets the output file path

Sets the output file path. If this function is not called before calling writeFile(), then the output file will be "./results.↩ json".

Parameters

outputPath	reference to a string containing the path (relative or absolute) to the output file.

Note

If the file specified by the outputPath does not exist, then the file will be created. If it does exist, then the file content will be overwritten.

3.1.1.14 setPoints()

```
void MOrepoResultWriter::setPoints (
            std::vector< std::vector< double >> & points,
            std::vector< std::string > & types )
```

Sets the points and the point types

Parameters

points	reference to a vector of vectors of doubles. points[i] contains the i'th point on the frontier and points[i][j] contains the j'th entry of the i'th non–dominated point.
types	reference to a vector of strings. Contains a specification of the type of each point. type can be either extreme supported ("se"), non-extreme supported ("sne"), supported (may be extreme or non–extreme) ("s"), unsuported ("un") or if this information is unknown ("null").

3.1.1.15 setSupportedCardinality()

```
void MOrepoResultWriter::setSupportedCardinality (
            int suppCard )  [inline]
```

Sets the cardinality of the set of supported non–dominated solutions

Parameters

suppCard	integer containing the number of supported non–dominated solutions.

3.1.1.16 setValid()

```
void MOrepoResultWriter::setValid (
            bool valid )  [inline]
```

Sets the validity of the solution to either true or false.

Sets the validity of the solution to either true or false. If valid is false, the solution might be in conflict with another solution on MOrepo. This will be sorted out eventually.

Parameters

valid	boolean. If true, the solution is not in conflict with other known solutions. If false, it may be in conflict with a known solution.

3.1.1.17 setVersion()

```
void MOrepoResultWriter::setVersion (
            std::string & version )  [inline]
```

Sets the version of the resultfile.

Sets the version of the results file using the provided string

Parameters

version	reference to a string. If the version is 5.4 the input should be a string "5.4"

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

- MOrepoResultWriter.h
- MOrepoResultWriter.cpp