Gurobify

Gurobify provides an interface to Gurobi from GAP.

0.1

23/02/2017

Jesse Lansdown

Jesse Lansdown
Email: jesse.lansdown@research.uwa.edu.au
Homepage: www.jesselansdown.com
Address: Jesse Lansdown
Lehrstuhl B für Mathematik
RWTH Aachen
Pontdriesch 10 - 16
52062 Aachen
Germany

Abstract

TODO

Copyright

Acknowledgements

TODO

Contents

Chapter 1

Introduction

1.1 Introducing Gurobify

Gurobify is a GAP package which provides an interface to the optimisation software Gurobi.

Chapter 2

Installation

2.1 Installing Gurobify

TODO

Chapter 3

Getting Started

3.1 Getting Started

TODO

3.2 Minimal working example?

TODO

Chapter 4

Gurobify Functions

4.1 Functions

In addition to the methods provided by Gurobify for altering specific attibute, Gurobify also offers direct manipulation of the Gurobi model using the following functions. Note that these require a greater familiarity with Gurobi, since they require the exact names of a gurobi attribute or parameter, and they are strict about the value types, and will only take integer or double values depending on the given attribute or parameter.

4.1.1 GurobiReadModel

▷ `GurobiReadModel(`*`ModelFile`*`)` (function)

Returns: a Gurobi model.

Takes a model file, reads it and creates a Gurobi model from it. ModelFile is the name of the file as a string, with the appropriate extension, and including the path if the file is not located in the current GAP working directory. Gurobi accepts files of type .mps, .rew, .lp, .rlp, .ilp, or .opb. Refer to the gurobi documentation for more infomation on which file types can be read.

4.1.2 GurobiNewModel

▷ `GurobiNewModel(`*`VariableTypes, ObjectiveFunction`*`)` (function)

Returns: a Gurobi model.

Creates a gurobi model with variables defined by VariableTypes and an objective function given by ObjectiveFunction. VariableTypes must be a list, with entries indexed by the set of variables, and entries corresponding to the type of variable, as a string. Accepted variable types are "CONTINUOUS", "BINARY", "INTEGER", "SEMICONT", or "SEMIINT". Refer to the Gurobi documentation for more information on the variable types. ObjectiveFunction is a list, with entries indexed by the set of variables, where each entry corresponds to the coefficient of the variable in the objective function. ObjectiveFunction takes only double values.

4.1.3 GurobiOptimizeModel

▷ `GurobiOptimizeModel(`*`Model`*`)` (function)

Returns: Optimisation status.

Takes a Gurobi model and optimises it. Returns the optimisation status code which indicates the outcome of the optimisation. A status code of 2 indicates that a feasible solution was found, a status code of 3 indicates the model is infeasible. There a number of other status codes. Refer to the Gurobi documentation for more information about status codes. The model itself is altered to reflect the optimisation, and more information about can be obatained using other functions, in particular the GurobiGetAttribute and GurobiGetAttributeArray functions.

4.1.4 GurobiReset

▷ `GurobiReset(`*`Model`*`)` (function)

Returns:

Reset all information associated with a solution for the model.

4.1.5 GurobiSetIntegerParameter

▷ `GurobiSetIntegerParameter(`*`Model, ParameterName, ParameterValue`*`)` (function)

Returns:

Takes a Gurobi model and assigns a value to a given integer-valued parameter. ParameterValue must be a integer value. Refer to the Gurobi documentation for a list of parameters and their types.

4.1.6 GurobiSetDoubleParameter

▷ `GurobiSetDoubleParameter(`*`Model, ParameterName, ParameterValue`*`)` (function)

Returns:

Takes a Gurobi model and assigns a value to a given double-valued parameter. ParameterValue must be a double value. Refer to the Gurobi documentation for a list of parameters and their types.

4.1.7 GurobiGetIntegerParameter

▷ `GurobiGetIntegerParameter(`*`Model, ParameterName`*`)` (function)

Returns: parameter value

Takes a Gurobi model and retrieve the value of a integer-valued parameter. Refer to the Gurobi documentation for a list of parameters and their types.

4.1.8 GurobiGetDoubleParameter

▷ `GurobiGetDoubleParameter(`*`Model, ParameterName`*`)` (function)

Returns: parameter value

Takes a Gurobi model and retrieve the value of a double-valued parameter. Refer to the Gurobi documentation for a list of parameters and their types.

4.1.9 GurobiAddConstraint

▷ `GurobiAddConstraint(`*`Model, ConstraintEquation, ConstraintSense, ConstraintRHSValue, ConstraintName`*`)` (function)

Returns:

Adds a constraint to a gurobi model. ConstraintEquation must be a list, with entries indexed by the variable set, such that each entry is the coefficient of the corresponding variable in the constraint

equation. The ConstraintSense must be one of "<", ">" or "=", where Gurobi interprets < as <= and > as >=. The ConstraintRHSValue is the value on the right hand side of the constraint. A constraint may also be given a name, which helps to identify the constraint if it is to be deleted at some point. May also take an empty string "" if no name is needed. Note that a model must be updated or optimised before any additional constraints become effective.

4.1.10 GurobiAddConstraint

▷ `GurobiAddConstraint(`*`Model, ConstraintName`*`)` (function)

Returns:

Deletes all constraints from a model with the name ConstraintName. Returns the updated model.

4.1.11 GurobiSetIntegerAttribute

▷ `GurobiSetIntegerAttribute(`*`Model, AttributeName, AttributeValue`*`)` (function)

Returns:

Takes a Gurobi model and assigns a value to a given integer-valued attribute. AttributeValue must be a double value Refer to the Gurobi documentation for a list of attributes and their types.

4.1.12 GurobiSetDoubleAttribute

▷ `GurobiSetDoubleAttribute(`*`Model, AttributeName, AttributeValue`*`)` (function)

Returns:

Takes a Gurobi model and assigns a value to a given double-valued attribute. AttributeValue must be a double value Refer to the Gurobi documentation for a list of attributes and their types.

4.1.13 GurobiGetIntegerAttribute

▷ `GurobiGetIntegerAttribute(`*`Model, AttributeName`*`)` (function)

Returns: attibute value

Takes a Gurobi model and retrieve the value of an integer-valued attribute. Refer to the Gurobi documentation for a list of attributes and their types.

4.1.14 GurobiGetDoubleAttribute

▷ `GurobiGetDoubleAttribute(`*`Model, AttributeName`*`)` (function)

Returns: attibute value

Takes a Gurobi model and retrieve the value of a double-valued attribute. Refer to the Gurobi documentation for a list of attributes and their types.

4.1.15 GurobiGetAttributeArray

▷ `GurobiGetAttributeArray(`*`Model, AttributeName`*`)` (function)

Returns: attibute array

Takes a Gurobi model and retrieve an attribute array. Can only get value of attributes arrays which take integer or double values, Refer to the Gurobi documentation for a list of attributes and their types.

4.1.16 GurobiWriteToFile

▷ `GurobiWriteToFile(`*`Model, FileName`*`)` (function)

Returns:

Takes a model and writes it to a file. File type written is determined by the FileName suffix. File types include .mps, .rew, .lp, .rlp, .ilp, .sol, or .prm Refer to the gurobi documentation for more infomation on which file types can be read.

4.1.17 GurobiUpdateModel

▷ `GurobiUpdateModel(`*`Model`*`)` (function)

Returns:

Takes a model and updates it. Changes to parameters or constraints are not processed until the model is either updated or optimised.

Chapter 5

Examples

5.1 Examples

TODO

Chapter 6

Apendix

6.1 Links to some Gurobi documentation

TODO

Chapter 7

Gurobify automatic generated documentation

7.1 Gurobify automatic generated documentation of methods

7.1.1 SetTimeLimit (for IsGurobiModel, IsFloat)

▷ `SetTimeLimit(`*`Model, TimeLimit`*`)` (operation)

Returns: true

Set a time limit for a Gurobi model.

7.1.2 SetBestObjectiveBoundStop (for IsGurobiModel, IsFloat)

▷ `SetBestObjectiveBoundStop(`*`Model, BestObjectiveBoundStop`*`)` (operation)

Returns: true

Optimisation will terminate if a feasible solution is found with objective value at least as good as BestObjectiveBoundStop.

7.1.3 SetCutOff (for IsGurobiModel, IsFloat)

▷ `SetCutOff(`*`Model, CutOff`*`)` (operation)

Returns: true

Optimisation will terminate if the objective value is worse than CutOff

Index