

optimr: An An Integrated Framework for Optimization

Emilio L. Cano

August 23, 2012

Contents

| | | |
|----------|-----------------------------------|----------|
| 1 | Introduction | 1 |
| 2 | Definitions | 1 |
| 3 | Optimization Workflow | 2 |
| 4 | Equations | 4 |
| 4.1 | Modified Indices | 4 |
| 4.2 | Conditional sums | 4 |
| 5 | Problem Instance | 4 |
| 5.1 | Sets | 4 |
| 5.2 | Parameter Values | 4 |
| A | Optimization Services (OS) | 5 |

1 Introduction

2 Definitions

entity An entity is a basic type items in an optimisation problem. These main entities are: sets, variables, parameters and equations. Terms are a special entity composed by a variable or parameter and further information to compound an equation. Constants and Scalars are auxiliary entities for modeling.

item An item is an specific instance of an entity. It is represented by an id and a symbol, and can contain more information as descriptions, indices, values or others.

Set

Variable

Parameter

Equation

Term

Constant

Scalar

3 Optimization Workflow

First, an `optimSMS` object has to be created. At least a name (`name`) must be provided. Other available arguments are short description (`sDes`) and long description (`lDes`):

```
> mySMS <- newSMS("MyFirstSMS")
```

After its creation, besides the name and descriptions, it contains the data structures for each entity of the model: sets, variables, parameters, equations and terms (of equations).

```
> getClass("optimSMS")
```

```
Class "optimSMS" [package "optimr"]
```

Slots:

```
Name:      consts      sets      vars
Class: data.frame data.frame data.frame
```

```
Name:      pars      eqs      terms
Class: data.frame data.frame data.frame
```

```
Name:      name      sDes      lDes
Class: character character character
```

```
> str(mySMS)
```

```
Formal class 'optimSMS' [package "optimr"] with 9 slots
 ..@ consts:'data.frame':      0 obs. of  6 variables:
 .. ..$ id      : logi(0)
 .. ..$ symbol: logi(0)
 .. ..$ tag     : logi(0)
 .. ..$ sDes    : logi(0)
 .. ..$ lDes    : logi(0)
 .. ..$ value   : logi(0)
 ..@ sets  :'data.frame':      0 obs. of  8 variables:
 .. ..$ id      : logi(0)
 .. ..$ symbol: logi(0)
 .. ..$ tag     : logi(0)
 .. ..$ sDes    : logi(0)
 .. ..$ lDes    : logi(0)
```

```

.. ..$ loc      : logi(0)
.. ..$ inSet    : logi(0)
.. ..$ part     : logi(0)
..@ vars  : 'data.frame':      0 obs. of  11 variables:
.. ..$ id       : logi(0)
.. ..$ symbol   : logi(0)
.. ..$ tag      : logi(0)
.. ..$ sDes     : logi(0)
.. ..$ lDes     : logi(0)
.. ..$ nature   : logi(0)
.. ..$ units    : logi(0)
.. ..$ dataType: logi(0)
.. ..$ ind      : logi(0)
.. ..$ integer  : logi(0)
.. ..$ positive: logi(0)
..@ pars  : 'data.frame':      0 obs. of  10 variables:
.. ..$ id       : logi(0)
.. ..$ symbol   : logi(0)
.. ..$ tag      : logi(0)
.. ..$ sDes     : logi(0)
.. ..$ lDes     : logi(0)
.. ..$ nature   : logi(0)
.. ..$ units    : logi(0)
.. ..$ dataType: logi(0)
.. ..$ group    : logi(0)
.. ..$ ind      : logi(0)
..@ eqs   : 'data.frame':      0 obs. of  8 variables:
.. ..$ id       : logi(0)
.. ..$ symbol   : logi(0)
.. ..$ tag      : logi(0)
.. ..$ sDes     : logi(0)
.. ..$ lDes     : logi(0)
.. ..$ nature   : logi(0)
.. ..$ relation: logi(0)
.. ..$ domain   : logi(0)
..@ terms : 'data.frame':      0 obs. of  14 variables:
.. ..$ id       : logi(0)
.. ..$ symbol   : logi(0)
.. ..$ tag      : logi(0)
.. ..$ sDes     : logi(0)
.. ..$ lDes     : logi(0)
.. ..$ eq       : logi(0)
.. ..$ side     : logi(0)
.. ..$ parent   : logi(0)
.. ..$ nature   : logi(0)
.. ..$ item     : logi(0)
.. ..$ setSums  : logi(0)
.. ..$ power    : logi(0)
.. ..$ sign     : logi(0)
.. ..$ setSubEq: logi(0)

```

```

..@ name : chr "MyFirstSMS"
..@ sDes : chr ""
..@ lDes : chr ""

```

Now we can add items to each entity of the model. Before adding an indexed item, the corresponding set must have been created. Thus, it is better to add first the sets, then the variables and parameters, and finally the equations and terms.

Let us add some sets:

4 Equations

4.1 Modified Indices

A term index can vary dynamically as a function of other indices or parameters. In this case, the term in the equation must include which (auxiliary) equation should substitute a given index.

NA

This equations are composed by the set to be substituted on the left side, and the substituting expression on the right side.

When an index in a term of an equation must be substituted, we add a value for the variable `setSubEq` to the term `data.frame`.

4.2 Conditional sums

A sum over a term can be made over a whole set (or defined subset), or over a conditional statement over sets. Thus, the property `setSums` of a term indicates which sets are in the sum, while the property `condSums` of a term indicates which conditional equation should be added to the sum. This is usually used for equations of the type:

$$\sum_{i' < i} x_{i'} \quad \forall i \in \mathcal{I}.$$

5 Problem Instance

5.1 Sets

The sets must be added in the same order than they are in the `data.frame` of the `sms`.

5.2 Parameter Values

The list must be in the same order than the indices were saved to the parameter in the `SMS`.

A Optimization Services (OS)

Implemented the Optimization Services from COIN-OR:

<http://www.coin-or.org>

<https://projects.coin-or.org/OS>

<http://www.optimizationservices.org/>