Table of contents

ack

L function to maintain the load of a set of container

Purpose

OPL function. A constraint to maintain the load of a set of containers.

context	Model files (.mod) - Not allowed in a CPLEX constraint block.
type	`boolean (1 if the constraint is true, 0 otherwise)`

Syntax

```
pack(int[],int[],int[])
pack(dvar int[ ],dvar int[ ],int[])
pack(int[],int[],int[],int)
pack(dvar int[ ],dvar int[ ],int[],dvar int)
```

Description

The pack constraint maintains the load of a set of containers or bins, given a set of weighted items and an assignment of items to containers. In the last syntax, the fourth parameter represents the number of times where the first variable takes a nonzero value.

In the example that follows, consider that we have n items and m containers. Each item i has an integer weight w[i] and a constrained integer variable p[i] associated with it, indicating in which container item i is to be placed. No item can be split up, and so an item can go in only one container. Associated with each container j is an integer variable l[j] representing the load in that container; that is, the sum of the weights of the items that have been assigned to that container. A capacity can be set for each container, placing an upper bound on this load variable. The constraint also ensures that the total sum of the loads of the containers is equal to the sum of the weights of the items being placed.

This function also works for integer arrays outside constraint blocks both in CP and CPLEX models.

You can use this function within IBM ILOG Script statements by specifying the OPL namespace:

```
( Opl.xxx() )
```

Copy to clipboard

In an MP model, you could write this pack constraint as follows:

```
forall(i in 1..m) sum(j in 1..n) ((p[j]==i)*(w[j]))==l[i];
```

Copy to clipboard

Table of contents

xample

```
sing CP;
nt m = 2;
nt n = 3;

var int l[j in 1..m] in 0..10000;
var int p[i in 1..n] in 1..m;
dvar int nb;

int w[1..n] = [i : 1 | i in 1..n];

subject to {

  pack(l, p, w, nb);

}

assert nb==m-count(l,0);
```

Copy to clipboard

Parent topic:

OPL functions