

两辆铁路平板车的装货问题

1988: The Railroad Flatcar Problem

有七种规格的包装箱要装到两辆平板车上去。包装箱的宽和高是一样的，但是厚度（ t_i ，以厘米计）及其重量（ w_i 以公斤计）是不同的。下表给出了每种包装箱的厚度、重量和数量。

	C1	C2	C3	C4	C5	C6	C7
t_i 厘米	48.7	52.0	61.3	72.0	48.7	52.0	64.0
w_i 公斤	2000	3000	1000	500	4000	2000	1000
n_i 件数	8	7	9	6	6	4	8

每辆平板车有 10.2 米长的地方可用来装包装箱（像面包片一样），载重为 40 吨。

由于当地货运的限制，对 C5, C6, C7 类的包装箱的总数有一定的限制：这类箱子所占的空间（厚度）不能超过 302.7 厘米。试把包装箱装到平板车上去使得浪费的空间最小。

其解是 $x = [5, 7, 0, 5, 0, 1, 0, 3, 0, 9, 1, 3, 2, 0]$ ，最优值是 0.600000000000014。

```
local print = require('package.print').print
local mol = libminoptlab

local t = {48.7, 52.0, 61.3, 72.0, 48.7, 52.0, 64.0}
local w = {2000, 3000, 1000, 500, 4000, 2000, 1000}
local n = {8, 7, 9, 6, 6, 4, 8}
local len = #t
```

```

local f = {table.unpack(t)}
for i = 1, len do
    table.insert(f, t[i])
end

local c = mol.matrix.new(f):prod(-1):tovector()

local Aeq = mol.matrix.new()
local beq = {}

local A = mol.matrix.new(len + 5, 2 * len)
for i = 1, len do
    A:set(i, i, 1)
    A:set(i, i + len, 1)
    A:set(1 + len, i, w[i])
    A:set(2 + len, i + len, w[i])
    A:set(3 + len, i, t[i])
    A:set(4 + len, i + len, t[i])
    if i > 4 then
        A:set(5 + len, i, t[i])
        A:set(5 + len, i + len, t[i])
    end
end

local bcon = {40000, 40000, 1020, 1020, 302.7}
local b = {table.unpack(n)}
for i = 1, #bcon do
    table.insert(b, bcon[i])
end

local lb = {}
local ub = {}
local intindex = {}

for i = 1, 2 * len do
    table.insert(lb, 0)

```

```
    table.insert(ub, 9)
    table.insert(intindex, i)
end

local best, fbest = mol.lpsolve(c, Aeq, beq, A, b, lb, ub, intindex)
fbest = 2040 + fbest

print({
    best = best,
    fbest = fbest,
    ineqcon = A:mul(best):sub(b):tovector()
})
```

长沙市先验真理数据科技有限公司 2023-2026

 湘公网安备43019002002046  湘ICP备2023033902号-1

所有权利归长沙市先验真理数据科技有限公司所有,侵权必究.