Mathematical Optimization
Using R to model optimization problem

A Gentle Introduction of Mathematical Optimization for R users!

library(rcbc) & minizinc

Shuai Wang Eugene Pyatigorsky

84.51 Operations Research

CinDay R User Meetup, May 22 2019

Table of Contents

Mathematical Optimization

Using R to model optimization problem

1 Mathematical Optimization

2 Using R to model optimization problem

Table of Contents

1 Mathematical Optimization

2 Using R to model optimization problem

What's mathematical optimization anyway?

Mathematical Optimization
Using R to model optimization problem

"Optimization" comes from the same root as "optimal", which means best. When you optimize something, you are "making it best".
But "best" can vary. If you're a football player, you might want to maximize your running yards, and also minimize your fumbles. Both maximizing and minimizing are types of optimization problems.

Mathematical Optimization in the "Real World"

Mathematical Optimization
Using R to model optimization problem

Mathematical Optimization is a branch of applied mathematics which is useful in many different fields. Here are a few examples:

- Manufacturing
- Production
- Inventory control
- Transportation
- Scheduling
- Networks
- Finance
- Economics
- Control engineering
- Marketing
- Policy Modeling
- Mechanics

Optimization Model Components

Mathematical Optimization
Using R to model optimization problem

Your basic optimization problem consists of:

1. The objective function, f(x), which is the output you're trying to maximize or minimize. e.g. maximize the gross profit margin; minimize travel distance of a pizza delivery car.
2. Variables, x_1, x_2, x_3 and so on, which are the inputs – things you can control.
3. Constraints, which are equations that place limits on how big or small some variables can get. e.g. The pizza delivery should be on time.

Optimization Example

Mathematical Optimization
Using R to model optimization problem

A football coach is planning practices for his running backs.

- His main goal is to maximize running yards – this will become his objective function.
- He can make his athletes spend practice time in the weight room; running sprints; or practicing ball protection. The amount of time spent on each is a variable.
- However, there are limits to the total amount of time he has. Also, if he completely sacrifices ball protection he may see running yards go up, but also fumbles, so he may place an upper limit on the amount of fumbles he considers acceptable. These are constraints.

Note that the variables influence the objective function and the constraints place limits on the domain of the variables.

Table of Contents

1 Mathematical Optimization

2 Using R to model optimization problem

Knapsack problem

- You only bring one knapsack with a capacity limit to rob a bank.
- Different gold has various amount of value and weight.
- Try to get as much value as possible.
- So, which ones to choose with the capacity limit of the knapsack.

Knapsack problem math modeling

KP has the following Integer Linear Programming (ILP) formulation:

$$\text{maximize} \quad \sum_{j \in N} p_j x_j \tag{1}$$

$$\text{subject to} \quad \sum_{j \in N} w_j x_j \leq c \tag{2}$$

$$x_j \in \{0, 1\}, \quad j \in N, \tag{3}$$

where each binary variable x_j, $j \in N$, is equal to 1 if and only if item j is selected.
p_j: price/value of each item; w_j: weight of each item.
We cannot take all items because the total weight of the chosen items cannot exceed the knapsack capacity c.

Using library(rcbc)

https://github.com/dirkschumacher/rcbc

```r
max_capacity <- 1000
n <- 100
weights <- round(runif(n, max = max_capacity))
price <- round(runif(n) * 100)

A <- matrix(weights, ncol = n, nrow = 1) # matrix for constrains

result <- cbc_solve(
 obj = price,  # define objective sum(price_i)
 mat = A,      # weight_i * n
 # variable x(j) is binary
 is_integer = rep.int(TRUE, n),
# row bound for contraints
 row_lb = 0, row_ub = max_capacity, max = TRUE,
#column bound for vairable
 col_lb = rep.int(0, n), col_ub = rep.int(1, n))
```

Minizinc: The Right Tool for the Right Job.

MiniZinc is a free and open-source constraint modeling language.
You can use MiniZinc to model constraint satisfaction and optimization problems in a high-level, solver-independent way.

```
int: n; % number of objects
int: capacity;
array[1..n] of int: profit;
array[1..n] of int: size;

array[1..n] of var 0..1: x;

constraint sum(i in 1..n)(size[i] * x[i]) <= capacity;
solve maximize sum(i in 1..n)(profit[i] * x[i]);
```

Linking Minizinc with R

Minizinc requires a unique data format to feed into the model as input. In order to make the best use of the general purpose language's data processing, visiulization function, the api is needed.
Only python's pymzn exists.
Eugene and I wrote an r package to