Penelitian Mandiri Sains Komputasi I

Update Progress Week X

Mohammad Rizka Fadhli
Ikang
20921004@mahasiswa.itb.ac.id

29 October 2021

Jenis-Jenis Masalah Optimisasi

Masalah optimisasi bisa dibagi dua menjadi dua kategori berdasarkan tipe *variables* yang terlibat[1], yakni:

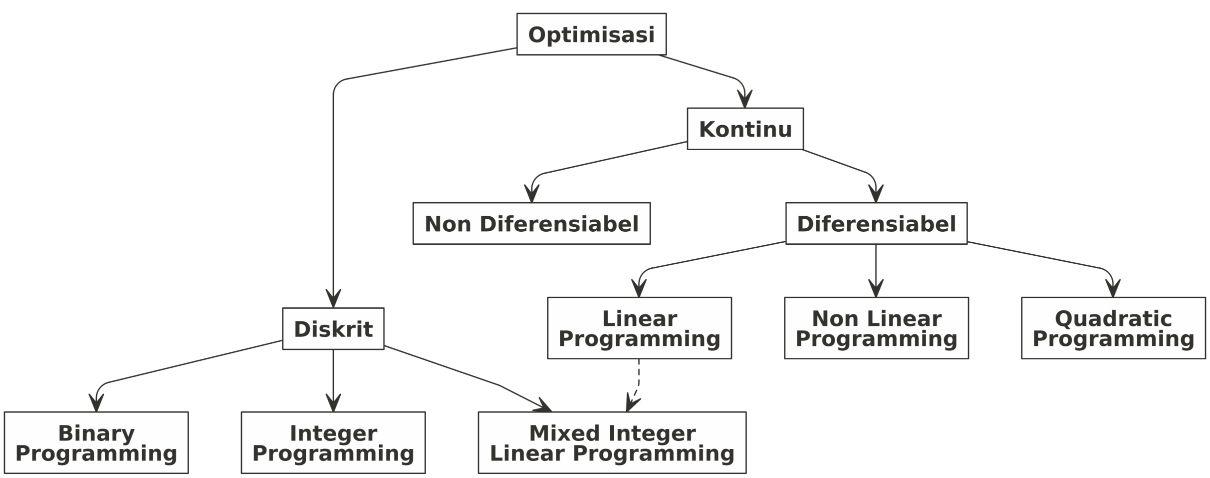

Figure 1: Optimisasi Berdasarkan Jenis Variabel

1. *Discrete Optimization*: merupakan masalah optimisasi di mana variabel yang terkait merupakan variabel diskrit, seperti *binary* atau *integer* (bilangan bulat). Namun pada

[1] Optimization problem. https://en.wikipedia.org/wiki/Optimization_problem

masalah optimisasi berbentuk *mixed integer linear programming*, dimungkinkan suatu masalah optimisasi memiliki berbagai jeni variabel yang terlibat (integer dan kontinu sekaligus).

2. *Continuous Optimization*: merupakan masalah optimisasi di mana variabel yang terkait merupakan variabel kontinu (bilangan *real*). Pada masalah optimisasi jenis ini, fungsi-fungsi yang terlibat bisa diferensiabel atau tidak. Konsekuensinya adalah pada metode penyelesaiannya.

Supplier Selection Problem

Tema penelitian terkait *supplier selection problem* termasuk ke dalam masalah optimisasi deterministik yakni *mixed integer linear programming*, alasannya:

1. Parameter dan variabel yang terlibat merupakan suatu nilai pasti.
2. Variabel yang terlibat meliputi:
    - *Binary* karena melibatkan pengambilan keputusan *raw matt* dari *supplier* mana yang harus dipesan.
    - *Continuous* karena melibatkan angka kuantitas *raw matt* yang harus dipesan.
3. Fungsi *objective* dan *constraints* masih berupa *linear*.

Mixed Integer Linear Programming

Pada bagian sebelumnya, kita telah membahas masalah optimisasi dengan variabel berupa diskrit dan kontinu. Permasalahan *real* yang ada di kehidupan sehari-hari biasanya merupakan memiliki variabel yang *mixed* antara keduanya. Oleh karena itu, ada metode yang disebut dengan *mixed integer linear programming*. Pada masalah optimisasi tipe ini, *decision variables* yang terlibat bisa saja berupa *binary*, *integer*, dan *continuous* sekaligus.

Menyelesaikan *MILP*

MILP secara eksak bisa diselesaikan dengan metode *simplex*.

Contoh *MILP*

Pemilihan dan Penentuan Item Produksi Suatu pabrik makanan dan minuman berencana untuk membuat tiga produk baru yang bisa diproduksi di dua *plants* yang berbeda.

Table 1: Tabel Runtime Item Produk per Plant (harian - dalam jam)

Produk	Runtime Plant 1	Runtime Plant 2
Item 1	3	4
Item 2	4	6
Item 3	2	2

Plant 1 memiliki maksimum *working hours* sebesar 30 jam perhari.

Plant 2 memiliki maksimum *working hours* sebesar 40 jam perhari.

Table 2: Tabel Profit dan Potensi Sales Item Produk

Produk	Profit per ton	Sales potential per ton
Item 1	5	7
Item 2	7	5
Item 3	3	9

Masalah timbul saat mereka harus memilih **dua dari tiga** produk baru tersebut yang harus di produksi. Selain itu, mereka juga harus memilih **satu dari dua** *plants* yang memproduksi *items* tersebut.

Misalkan saya definisikan:

- $x_i \geq 0, i = 1, 2, 3$ sebagai `berapa ton` yang harus diproduksi dari item i.
- $y_i \in [0, 1], i = 1, 2, 3$ sebagai *binary*.

- Jika bernilai 0, maka produk i tidak dipilih.
- Jika bernilai 1, maka produk i dipilih.

- $z \in [0, 1]$ sebagai *binary*.
  - Jika bernilai 0, maka *plant* pertama dipilih.
  - Jika bernilai 1, maka *plant* kedua dipilih.

Saya akan mendefinisikan suatu variabel *dummy* $M = 99999$ berisi suatu nilai yang besar. Kelak variabel ini akan berguna untuk *reinforce model* (metode pemberian *penalty*) agar bisa memilih *items* dan *plants* secara bersamaan.

Objective function dari masalah ini adalah memaksimalkan *profit*.

$$\max \sum_{i=1}^{3} x_i \times \text{profit}_i$$

Constraints dari masalah ini adalah:

Tonase produksi tidak boleh melebihi angka *sales potential* per items.

$$x_i \leq \text{sales potential}_i, i = 1, 2, 3$$

Kita akan memilih dua produk sekaligus menghitung tonase. Jika produk tersebut **dipilih**, maka akan ada angka tonase produksinya. Sebaliknya, jika produk tersebut **tidak dipilih**, maka tidak ada angka tonase produksinya.

$$x_i - y_i \times M \leq 0, i = 1, 2, 3$$

$$\sum_{i=1}^{3} y_i \leq 2$$

Kita akan memilih *plant* dari waktu produksinya.

$$3x_1 + 4x_2 + 2x_3 - M \times z \leq 30$$

$$4x_1 + 6x_2 + 2x_3 + M \times z \leq 40 + M$$

Penyelesaian Contoh Soal

Dengan menggunakan `library(ompr)`

```
rm(list=ls())

library(dplyr)
library(ompr)
library(ompr.roi)
library(ROI.plugin.glpk)

# data yang dibutuhkan
profit = c(5,7,3)
sales = c(7,5,9)
M = 99999

# membuat model
mil_prog =
  MIPModel() %>%
  # menambah variabel
  # xi
  add_variable(x[i],
           i = 1:3,
           type = "continuous",
           lb = 0) %>%
  # yi
  add_variable(y[i],
           i = 1:3,
```

```
           type = "binary",
           lb = 0) %>%
  # z
  add_variable(z,type = "binary",lb = 0) %>%
  # membuat objective function
  set_objective(sum_expr(x[i] * profit[i],
             i = 1:3),
        "max") %>%
  # menambah constraints
  # max tonase
  add_constraint(x[i] <= sales[i],
         i = 1:3) %>%
  # memilih 2 produk
  add_constraint(x[i] - y[i] * M <= 0,
         i = 1:3) %>%
  add_constraint(sum_expr(y[i],
         i = 1:3) <= 2) %>%
  # memilih 1 plant
  add_constraint(3*x[1] + 4*x[2] + 2*x[3] - M * z <= 30) %>%
  add_constraint(4*x[1] + 6*x[2] + 2*x[3] + M * z <= 40 + M)

mil_prog
```

```
## Mixed integer linear optimization problem
## Variables:
##   Continuous: 3
##   Integer: 0
##   Binary: 4
## Model sense: maximize
## Constraints: 9
```

```r
hasil =
  mil_prog %>%
  solve_model(with_ROI(solver = "glpk",
                       verbose = T))
```

```
## <SOLVER MSG>  ----
## GLPK Simplex Optimizer, v4.65
## 9 rows, 7 columns, 20 non-zeros
## *     0: obj =  -0.000000000e+00 inf =   0.000e+00 (3)
## *     7: obj =   9.700000000e+01 inf =   0.000e+00 (0)
## OPTIMAL LP SOLUTION FOUND
## GLPK Integer Optimizer, v4.65
## 9 rows, 7 columns, 20 non-zeros
## 4 integer variables, all of which are binary
## Integer optimization begins...
## Long-step dual simplex will be used
## +     7: mip =     not found yet <=              +inf        (1; 0)
## +    12: >>>>>   5.450000000e+01 <=   5.450000000e+01   0.0% (4; 0)
## +    12: mip =   5.450000000e+01 <=     tree is empty   0.0% (0; 7)
## INTEGER OPTIMAL SOLUTION FOUND
## <!SOLVER MSG> ----
```

```r
xi =
  hasil %>%
  get_solution(x[i])

yi =
  hasil %>%
  get_solution(y[i])
```

```r
zi =
  hasil %>%
  get_solution(z)
```

Berikut adalah hasilnya:

```
##   variable i value
## 1        x 1   5.5
## 2        x 2   0.0
## 3        x 3   9.0
```

```
##   variable i value
## 1        y 1     1
## 2        y 2     0
## 3        y 3     1
```

```
## z
## 1
```

Dari ketiga produk baru, perusahaan bisa memilih produk **1 dan 3** sebanyak **5.5 dan 9 ton** di *plant* **2** sehingga *profit* yang bisa diraih adalah sebesar **54.5**.