SPIRAL OPTIMIZATION ALGORITHM UNTUK MENYELESAIKAN INTEGER PROGRAMMING

Tugas Kuliah
SK5001 Analisis Numerik Lanjut

Mohammad Rizka Fadhli
NIM: 20921004

02 December 2021

PENDAHULUAN

Spiral Optimization Algorithm

Pada tugas mata kuliah SK5001 sebelumnya, saya telah menuliskan program **SOA** menggunakan bahasa pemrograman **R** yang dapat digunakan untuk menyelesaikan masalah optimisasi (maksimasi atau minimisasi) dari fungsi *real* **dua peubah**.

Integer Programming

Dua Peubah

Pada tugas ini, saya akan memodifikasi fungsi di atas untuk menyelesaikan permasalahan *integer programming*, yakni dengan melakukan pembulatan terhadap peubah yang memiliki nilai $f(x_1, x_2)$ yang paling minimum.

```
soa_mrf_ip = function(
       N,       # banyak titik
           x1_d,  # batas bawah x1
           x1_u,  # batas atas x1
```

```
          x2_d,  # batas bawah x2
          x2_u,  # batas atas x2
          rot,   # berapa banyak rotasi
          k_max, # iterasi maks
          r){    # berapa rate konstraksi

# N pasang titik random di selang [a,b] di R2
x1 = runif(N,x1_d,x1_u)
x2 = runif(N,x2_d,x2_u)
# hitung theta
theta = 2*pi / rot
# definisi matriks rotasi
A = matrix(c(cos(theta),-sin(theta),
             sin(theta),cos(theta)),
          ncol = 2,byrow = T)

# bikin data frame
temp = data.frame(x1,x2) %>% mutate(f = f(round(x1,0),round(x2,0)))
# proses iterasi
for(i in 1:k_max){
  # mencari titik x* dengan max(f)
  f_min =
      temp %>%
    # memastikan titik ada di D
    filter(x1 >= x1_d & x1 <= x1_u) %>%
    filter(x2 >= x2_d & x2 <= x2_u) %>%
      # mencari titik max fungsi
    filter(f == max(f))
  # definisi pusat rotasi
  pusat = c(f_min$x1[1],f_min$x2[1])
  for(j in 1:N){
      # kita akan ambil titiknya satu persatu
      x0 = c(temp$x1[j],temp$x2[j])
      # proses rotasi dan konstraksi terhadap pusat x*
      xk = A %*% (x0-pusat) # diputar dengan x_bin sebagai pusat
    xk = pusat + (r * xk)
      # proses mengembalikan nilai ke temp
```

```
      temp$x1[j] = xk[1]
      temp$x2[j] = xk[2]
    }
    # hitung kembali nilai f(x1,x2)
    temp = temp %>% mutate(f = f(round(x1,0),round(x2,0)))
    }
# proses output hasil
output =
  temp[N,] %>%
  filter(f == max(f)) %>%
  mutate(x1 = round(x1,0),x2 = round(x2,0),
         g = g(x1,x2),f = f(x1,x2))
return(output)
}
```

Tiga Peubah

Berikut adalah program untuk SOA tiga peubah:

```
soa_mrf_ip_3_var = function(
  N,        # banyak titik
  x1_d,  # batas bawah x1
  x1_u,  # batas atas x1
  x2_d,  # batas bawah x2
  x2_u,  # batas atas x2
  x3_d,  # batas bawah x3
  x3_u,  # batas atas x3
  rot,    # berapa banyak rotasi
  k_max, # iterasi maks
  r){     # berapa rate konstraksi

  # N pasang titik random di selang [a,b] di R3
  x1 = runif(N,x1_d,x1_u)
  x2 = runif(N,x2_d,x2_u)
  x3 = runif(N,x3_d,x3_u)

  # hitung theta
```

```
theta = 2*pi / rot
# definisi matriks rotasi
R12 = matrix(c(cos(theta),-sin(theta),0,
               sin(theta),cos(theta),0,
               0,0,1),
             ncol = 3,byrow = T)
R13 = matrix(c(cos(theta),0,-sin(theta),
               0,1,0,
               sin(theta),0,cos(theta)),
             ncol = 3,byrow = T)
R23 = matrix(c(1,0,0,
               0,cos(theta),-sin(theta),
               0,sin(theta),cos(theta)),
             ncol = 3,byrow = T)


# bikin data frame
temp = data.frame(x1,x2,x3) %>%
  mutate(f = f(round(x1,0),
               round(x2,0),
               round(x3,0)
               )
         )
# proses iterasi
for(i in 1:k_max){
  # mencari titik x* dengan max(f)
  f_min =
    temp %>%
    # memastikan titik ada di D
    filter(x1 >= x1_d & x1 <= x1_u) %>%
    filter(x2 >= x2_d & x2 <= x2_u) %>%
    filter(x3 >= x3_d & x3 <= x3_u) %>%
    # mencari titik max fungsi
    filter(f == max(f))
  # definisi pusat rotasi
  pusat = c(f_min$x1[1],f_min$x2[1],f_min$x3[1])
  for(j in 1:N){
```

```
      # kita akan ambil titiknya satu persatu
      x0 = c(temp$x1[j],temp$x2[j],temp$x3[j])
      # proses rotasi dan konstraksi terhadap pusat x*
      # diputar dengan x_bin sebagai pusat
      xk = (R23 %*% (R13 * R12)) %*% (x0-pusat)
      xk = pusat + (r * xk)
      # proses mengembalikan nilai ke temp
      temp$x1[j] = xk[1]
      temp$x2[j] = xk[2]
      temp$x3[j] = xk[3]
    }
    # hitung kembali nilai f(x1,x2)
    temp = temp %>% mutate(f = f(round(x1,0),round(x2,0),round(x3,0)))
  }
  # proses output hasil
  output =
    temp[N,] %>%
    filter(f == max(f)) %>%
    mutate(x1 = round(x1,0),x2 = round(x2,0),x3 = round(x3,0),
           g = g(x1,x2,x3),f = f(x1,x2,x3))
  return(output)
}
```

Mengubah Optimisasi Menjadi Pencarian Akar

Spiral optimization algorithm adalah suatu metode untuk mencari solusi minimum global. Jika kita hendak memakainya untuk mencari suatu akar persamaan (atau sistem persamaan), kita bisa melakukan modifikasi pada fungsi-fungsi yang terlibat (membuat fungsi *merit*).

Misalkan suatu sistem persamaan non linear:

$$g_1(x_1, x_2, .., x_n) = 0$$

$$g_2(x_1, x_2, .., x_n) = 0$$

$$g_n(x_1, x_2, .., x_n) = 0$$

dengan $(x_1, x_2, .., x_n)^T \in D$

$$D = a_1, b_1 \times a_2, b_2 \times .. \times a_n, b_n \subset \mathbb{R}^n$$

Pencarian Akar

Sistem di atas memiliki solusi $x = (x_1, x_2, .., x_n)^T$ jika $F(x)$ yang kita definisikan sebagai:

$$F(x) = \frac{1}{1 + \sum_{i=1}^{n} |g_i(x)|}$$

memiliki nilai maksimum sama dengan 1. **Akibatnya algoritma yang sebelumnya adalah mencari** $\min F(x)$ **diubah menjadi** $\max F(x)$. Kenapa demikian?

> Karena jika $F(x) = 1$ artinya $\sum_{i=1}^{n} |g_i(x)| = 0$ yang merupakan akar dari $g_i, i = 1, 2, .., n$.

Kelak $F(x)$ akan digunakan untuk menjawab soal-soal yang ada dalam tugas ini.

SOAL

Tentukanlah solusi-solusi persamaan *diophantine* berikut dengan algoritma optimisasi spiral:

1. $x^2 + y^2 = 625, 0 \leq x, y \leq 25$.
2. $x^3 + y^3 = 1008, 0 \leq x, y \leq 50$.
3. $x^3 - 3xy^2 - y^3 = 1, -10 \leq x, y \leq 10$.
4. $x^2 + y^2 + z^2 = 2445, 0 \leq x, y, z \leq 50$.

JAWAB

Persamaan *Diophantine* 1

Persamaan pada soal bisa dituliskan sebagai berikut:

$$g(x, y) = x^2 + y^2 - 625 = 0, 0 \leq x, y \leq 25$$

Untuk menyelesaikannya dengan **SOA**, saya akan ubah $g(x, y)$ menjadi fungsi merit sebagai berikut:

$$F(x, y) = \frac{1}{1 + |g(x, y)|}, 0 \leq x, y \leq 25$$

Oleh karena itu, tugas saya sekarang adalah mencari $(x, y) \in D$ yang membuat $\max F(x, y)$. Saya akan coba *run* programnya berulang kali agar mendapatkan sebanyak-banyaknya solusi yang tepat.

Berikut adalah penyelesaiannya di **R**:

```
# solving
N = 100
a = 0  # x dan y punya batas yang sama
b = 25 # x dan y punya batas yang sama
rot = 30
k_max = 50
r = .65
# membuat fungsi g dan f
g = function(x,y){x^2 + y^2 - 625}
f = function(x,y){1 / (1 + g(x,y))}
# iterasi berulang kali agar mendapatkan hasil yang tepat
solusi = data.frame()
for(num in 1:70){
  temporary = soa_mrf_ip(N,a,b,a,b,rot,k_max,r)
  solusi = rbind(solusi,temporary)
}
```

Berikut adalah tabel solusi yang didapatkan:

Table 1: Solusi Soal I

x	y	F(x,y)	g(x,y)
0	25	1	0
7	24	1	0
15	20	1	0
20	15	1	0
24	7	1	0
25	0	1	0

Persamaan *Diophantine* 2

Persamaan pada soal bisa dituliskan sebagai berikut:

$$g(x,y) = x^3 + y^3 - 1008 = 0, 0 \leq x, y \leq 50$$

Untuk menyelesaikannya dengan **SOA**, saya akan ubah $g(x,y)$ menjadi fungsi merit sebagai berikut:

$$F(x,y) = \frac{1}{1 + |g(x,y)|}, 0 \leq x, y \leq 50$$

Oleh karena itu, tugas saya sekarang adalah mencari $(x,y) \in D$ yang membuat $\max F(x,y)$. Saya akan coba *run* programnya berulang kali agar mendapatkan sebanyak-banyaknya solusi yang tepat.

Berikut adalah penyelesaiannya di **R**:

```
# solving
N = 200
a = 0  # x dan y punya batas yang sama
b = 50 # x dan y punya batas yang sama
rot = 30
k_max = 50
r = .65
# membuat fungsi g dan f
```

```r
g = function(x,y){x^3 + y^3 - 1008}
f = function(x,y){1 / (1 + g(x,y))}
# iterasi berulang kali agar mendapatkan hasil yang tepat
solusi = data.frame()
for(num in 1:80){
  temporary = soa_mrf_ip(N,a,b,a,b,rot,k_max,r)
  solusi = rbind(solusi,temporary)
}
```

Berikut adalah tabel solusi yang didapatkan:

Table 2: Solusi Soal II

x	y	F(x,y)	g(x,y)
2	10	1	0
10	2	1	0

Persamaan *Diophantine* 3

Persamaan pada soal bisa dituliskan sebagai berikut:

$$g(x,y) = x^3 - 3xy^2 - y^3 - 1 = 0, -10 \leq x, y \leq 10$$

Untuk menyelesaikannya dengan **SOA**, saya akan ubah $g(x,y)$ menjadi fungsi merit sebagai berikut:

$$F(x,y) = \frac{1}{1 + |g(x,y)|}, -10 \leq x, y \leq 10$$

Oleh karena itu, tugas saya sekarang adalah mencari $(x,y) \in D$ yang membuat $\max F(x,y)$. Saya akan coba *run* programnya berulang kali agar mendapatkan sebanyak-banyaknya solusi yang tepat.

Berikut adalah penyelesaiannya di **R**:

```r
# solving
N = 100
a = -10  # x dan y punya batas yang sama
```

```
b = 10   # x dan y punya batas yang sama
rot = 50
k_max = 80
r = .75
# membuat fungsi g dan f
g = function(x,y){x^3 - 3*x*y^2 - y^3 - 1}
f = function(x,y){1 / (1 + abs(g(x,y)))}
# iterasi berulang kali agar mendapatkan hasil yang tepat
solusi = data.frame()
for(num in 1:80){
  temporary = soa_mrf_ip(N,a,b,a,b,rot,k_max,r)
  solusi = rbind(solusi,temporary)
}
```

Berikut adalah tabel solusi yang didapatkan:

Table 3: Solusi Soal III

x	y	F(x,y)	g(x,y)
-3	2	1	0
-1	1	1	0
0	-1	1	0
1	-3	1	0
1	0	1	0
2	1	1	0

Persamaan *Diophantine* 4

Persamaan pada soal bisa dituliskan sebagai berikut:

$$g(x,y,z) = x^2 + y^2 + z^2 - 2445 = 0, 0 \leq x,y,z \leq 50$$

Untuk menyelesaikannya dengan **SOA**, saya akan ubah $g(x,y,z)$ menjadi fungsi merit sebagai berikut:

$$F(x,y,z) = \frac{1}{1 + |g(x,y,z)|}, 0 \leq x,y,z \leq 50$$

Berikut adalah penyelesaiannya di **R**:

```
# solving
N = 200
a = 0  # x dan y punya batas yang sama
b = 50   # x dan y punya batas yang sama
rot = 70
k_max = 90
r = .85
# membuat fungsi g dan f
g = function(x,y,z){x^2 + y^2 + z^2 - 2445}
f = function(x,y,z){1 / (1 + abs(g(x,y,z)))}
# iterasi berulang kali agar mendapatkan hasil yang tepat
solusi = data.frame()
for(num in 1:70){
  temporary = soa_mrf_ip_3_var(N,a,b,a,b,a,b,rot,k_max,r)
  solusi = rbind(solusi,temporary)
}
```

Berikut adalah tabel solusi yang didapatkan:

Table 4: Solusi Soal IV

x	y	z	F(x,y,z)	g(x,y,z)
13	40	26	1	0
13	26	40	1	0
14	43	20	1	0
14	32	35	1	0
19	40	22	1	0
19	22	40	1	0
20	43	14	1	0
20	26	37	1	0
20	14	43	1	0
22	40	19	1	0
22	19	40	1	0
26	37	20	1	0
26	13	40	1	0

x	y	z	F(x,y,z)	g(x,y,z)
26	20	37	1	0
26	40	13	1	0
32	14	35	1	0
32	35	14	1	0
34	8	35	1	0
35	14	32	1	0
35	34	8	1	0
37	26	20	1	0
40	13	26	1	0
43	14	20	1	0

END

Dibuat oleh: 20921004 Mohammad Rizka Fadhli