Tarea 10: Algoritmo genético

C. María Montemayor Palos

1. Objetivo

Como objetivo se pretende cambiar la selección de mutación y la selección de los padres para reproducción aplicando la selección de ruleta: cada solución se selecciona como padre con una probabilidad que es linealmente proporcional a su valor de función objetivo y a su factibilidad, combinando los dos a alguna función que parezca conveniente e inversamente proporcional a alguna combinación de factibilidad y objetivo para la mutación.
Se generan instancias con tres distintas reglas:
Regla 1: el peso y el valor de cada objeto se generan independientemente con una distribución normal.
Regla 2: el valor de cada objeto se generan independientemente con una distribución exponencial y su peso es inversamente correlacionado con el valor, con un ruido normalmente distribuido de baja magnitud.
Regla 3: el peso de cada objeto se generan independientemente con una distribución normal y su valor es (positivamente) correlacionado con el cuadrado del peso, con un ruido normalmente distribuido de baja magnitud.
Se determina para cada uno de los tres casos a partir de qué tamaño de instancia el algoritmo genético es competitivo con el algoritmo exacto en términos de valor total obtenido por segundo de ejecución y si la inclusión de la selección de ruleta produce una mejora estadísticamente significativa.

2. Metodología

Para efectos de la tarea [2] se utiliza el programa R versión 4.0.4 [3] para Windows. Esta tarea se basa en el problema de la mochila (o mejor conocido en inglés como: knapsack), es un problema clásico de optimización, particularmente de programación entera, donde la tarea consiste en seleccionar un subconjunto de objetos de tal forma que en primer lugar no se exceda la capacidad de la mochila en términos de la suma de los pesos de los objetos incluidos, y en segundo lugar el valor total de los objetos incluidos sea lo máximo posible. Este problema es pseudo-polinomial ya que existe un algoritmo de tabulación que determina la combinación óptima.

3. Código

Se modifica el código de Schaeffer [1] en la sección de las variables correspondiendo `n` a la capacidad total, añadiendo `p0` como población inicial y `pm` correspondiendo a la población mutada. Esta sección del código es aplicable para las tres reglas.

```
n <- 50
pesos <- generador.pesos(n, 15, 80)
valores <- generador.valores(pesos, 10, 500)
capacidad <- round(sum(pesos) * 0.65)
optimo <- knapsack(capacidad, pesos, valores)
init <- 200
p0 <- poblacion.inicial(n, init)
p<- p0
tam <- dim(p)[1]
assert(tam == init)
pm <- sum(runif(tam) < 0.05)
rep <- 50
tmax <- 50
mejorruleta <- double()
```

Se genera la probabilidad de mutación y las personas mutadas para la regla 1 aplicando la ruleta.

```
  prob.mutacion = NULL
  for (i in 1:tam) {
    prob.mutacion[i] = 1/(obj[i]*(fact[i]+1)*sum(obj*(fact+1)))
  }
  pmutadas<-sample(1:tam, pm, prob = prob.mutacion)
  for (i in pmutadas) {
    p <- rbind(p, mutacion(p[i,], n))
  }
```

Se generan las reproducciones de los padres y se generan el primer y segundo hijo.

```
  prob.reproduccion = NULL
  for (i in 1:tam) {
    prob.reproduccion[i] = obj[i]*(fact[i]+1)/sum(obj*(fact+1))
  }
  for (i in 1:rep) {
    padres <- sample(1:tam, 2, replace=FALSE, prob = prob.reproduccion)
    hijos <- reproduccion(p[padres[1],], p[padres[2],], n)
    p <- rbind(p, hijos[1:n]) # primer hijo
    p <- rbind(p, hijos[(n+1):(2*n)]) # segundo hijo
```

Se generan las iteraciones del algoritmo genético sin ruleta, cada objeto puede mutarse con una probabilidad `pm`.

```
 for (iter in 1:tmax) {
  p$obj <- NULL
  p$fact <- NULL
  for (i in 1:tam) {
    if (runif(1) < pm) {
      p <- rbind(p, mutacion(p[i,], n))
    }
  }
```

Código sin ruleta.

```
for (i in 1:rep) {
    padres <- sample(1:tam, 2, replace=FALSE)
    hijos <- reproduccion(p[padres[1],], p[padres[2],], n)
    p <- rbind(p, hijos[1:n]) # primer hijo
    p <- rbind(p, hijos[(n+1):(2*n)]) # segundo hijo
  }
  tam <- dim(p)[1]
  obj <- double()
  fact <- integer()
  for (i in 1:tam) {
    obj <- c(obj, objetivo(p[i,], valores))
    fact <- c(fact, factible(p[i,], pesos, capacidad))
  }
  p <- cbind(p, obj)
  p <- cbind(p, fact)
  mantener <- order(-p[, (n + 2)], -p[, (n + 1)])[1:init]
  p <- p[mantener,]
  tam <- dim(p)[1]
  assert(tam == init)
  factibles <- p[p$fact == TRUE,]
  mejor <- max(factibles$obj)
  mejorsin <- c(mejorsin, mejor)
}
```

Modificación del código para la regla 1 en el cual el peso y el valor de cada objeto se genera independientemente con una distribución normal.

```
generador.pesos <- function(cuantos, min, max) {
  return(sort(round(normalizar(rnorm(cuantos)) * (max - min) + min)))
}
```

Modificación del código para la regla 2 en la cual el valor de cada objeto se genera independientemente con una distribución exponencial y su peso es inversamente correlacionado con el valor, con un ruido normalmente distribuido de baja magnitud.

```
generador.pesos <- function(valores, min, max) {
  n <- length(valores)
  pesos <- double()
  for (i in 1:n) {
    media <- valores[i]
    desv <- runif(1, max=.1)
    ruido <- rnorm(1, sd=.1)
    pesos <- c(pesos, rnorm(1, (1/media), desv) + ruido)
  }
  pesos <- normalizar(pesos) * (max - min) + min
  return(pesos)
}
```

Modificación del código para la regla 3 en la cual el peso de cada objeto se generan independientemente con una distribución normal y su valor correlacionado con el cuadrado del peso, con un ruido normalmente distribuido de baja magnitud.

```
generador.valores <- function(pesos, min, max) {
  n <- length(pesos)
  valores <- double()
  for (i in 1:n) {
    media <- pesos[i]
    desv <- runif(1)
    ruido <- rnorm(1, sd=.1)
    valores <- c(valores, rnorm(1, media^2, desv) + ruido)
  }
  valores <- normalizar(valores) * (max - min) + min
  return(valores)
}
```

4. Resultados y discusión

En las siguientes figuras se presentan los resultados obtenidos de las reglas 1, 2 y 3 respectivamente, correspondiendo la línea verde al valor óptimo, la línea morada a la ejecución del código sin ruleta y a la línea rosa pastel a la ejecución del código con la ruleta.

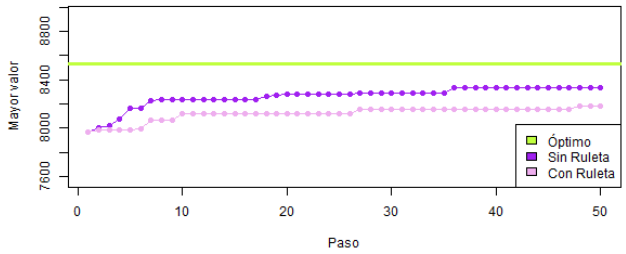

Figura 1. Diagrama de ejecución de la regla 1 con ruleta y sin ruleta.

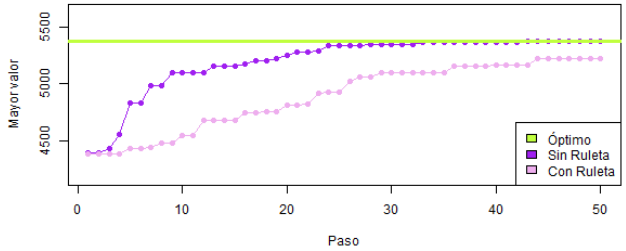

Figura 2. Diagrama de ejecución de la regla 2 con ruleta y sin ruleta.

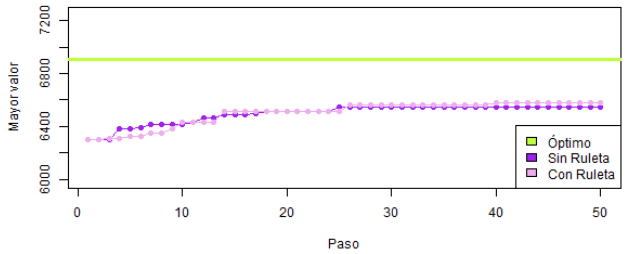

Figura 3. Diagrama de ejecución de la regla 3 con ruleta y sin ruleta.

De acuerdo a los datos arrojados para cada regla, se comparan los valores obtenidos en la siguiente tabla.

Cuadro 1. Fragmento de los datos recopilados del experimento.

Regla	Mejor resultado	Valor óptimo	Porcentaje
1	8331	8530	0.0233
2	5373	5376	0.0005
3	6549	6907	0.0518

5. Conclusión

A simple vista se puede deducir por las figuras presentadas que la figura 2, en cuanto a la ejecución del código sin ruleta, fue la más cercana al valor óptimo. Se puede comprobar mediante los datos arrojados observados en la tabla 1 que su mejor resultado fue de 5,373 siendo que el valor óptimo fue de 5,376 obtienendo un porcentaje de 0.0005. Comparando los demás valores le sigue la regla 1 (figura 1) resultando como último la regla 3 (figura 3).

6. Reto 1

El primer reto consiste en extender la selección de ruleta a la fase de supervivencia: en vez de quedarse con las mejores soluciones, cada solución tiene una probablidad de entrar a la siguiente generación que es proporcional a su valor de la función objetivo, incorporando el sí o no es factible la solución en cuestión, permitiendo que los k mejores entre las factibles entren siempre (donde k es un parámetro). Se estudia nuevamente el efecto de este cambio en la calidad de la solución en los tres casos.
Modificando el primer código de la regla 1,2 y 3, en la creación de variables se agrega la probabilidad de supervivencia.

```
n <- 50
pesos <- generador.pesos(n, 15, 80)
valores <- generador.valores(pesos, 10, 500)
capacidad <- round(sum(pesos) * 0.65)
optimo <- knapsack(capacidad, pesos, valores)
init <- 200
p1 <- poblacion.inicial(n, init)
p<- p1
tam <- dim(p)[1]
assert(tam == init)
pm <- sum(runif(tam) < 0.05)
rep <- 50
tmax <- 50
mejorescon <- double()
prob.supervivencia <- NULL
```

En los vectores de factibilidad y objetivos se modifica añadiendo los `sobrevivientes` para obtener la mejor combinación.

```
  p <- cbind(p, obj)
  p <- cbind(p, fact)
  for (i in 1:tam) {
    prob.supervivencia[i] = obj[i]*(fact[i]+1)/sum(obj*(fact+1))
  }
  sobrevivientes<-sample(1:tam,init,prob = prob.supervivencia)
  p <- p[sobrevivientes,]
  tam <- dim(p)[1]
  assert(tam == init)
  factibles <- p[p$fact == TRUE,]
  mejor <- max(factibles$obj)
  mejorruleta <- c(mejorruleta, mejor)
}
```

En las siguientes figuras se presentan los resultados obtenidos del reto 1 para las reglas 1, 2 y 3 respectivamente, correspondiendo la línea verde al valor óptimo, la línea morada a la ejecución del código sin ruleta y a la línea rosa pastel a la ejecución del código con la ruleta.

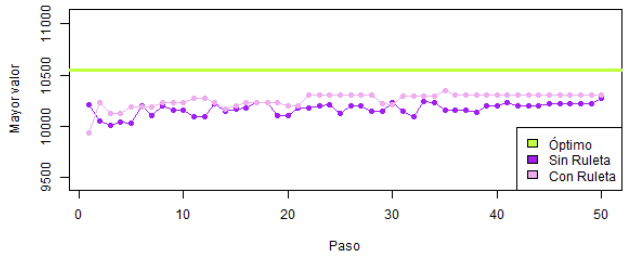

FIGURA 4. Diagrama de ejecución de la regla 1 con ruleta y sin ruleta.

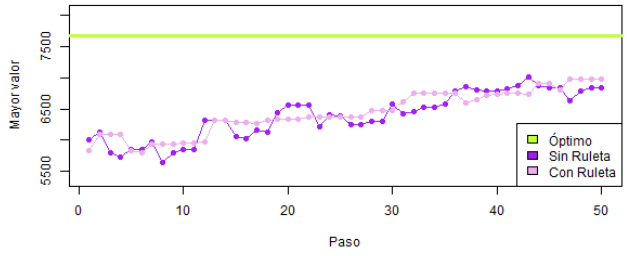

FIGURA 5. Diagrama de ejecución de la regla 2 con ruleta y sin ruleta.

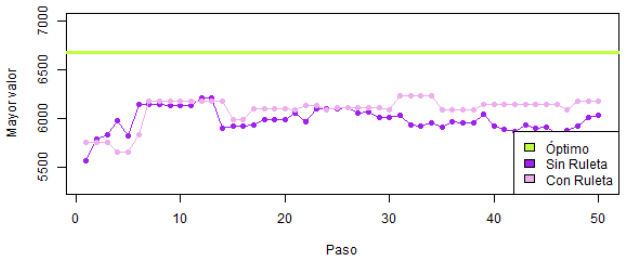

FIGURA 6. Diagrama de ejecución de la regla 3 con ruleta y sin ruleta.

De acuerdo a los datos arrojados para cada regla del reto 1, se comparan los valores obtenidos en la siguiente tabla.

CUADRO 2. Comparación de los datos recopilados del experimento.

Regla	Mejor resultado	Valor óptimo	Porcentaje
1	10279	10553	0.0259
2	6837	7676	0.1093
3	6031	6682	0.0973

Como conclusión observando la figura 4 tiene valores más cercanos al valor óptimo a comparación de las figuras 5 y 6. Comparando los valores del experimento en la tabla 2 se aprecia que el mejor resultado es para la regla 1 obteniendo un porcentaje de 0.0259, seguido de la regla 3 con un porcentaje de 0.0973, y por último la regla 2 con un porcentaje de 0.1093.

Referencias

[1] Schaeffer E. Algoritmo genético., 2021. URL https://github.com/satuelisa/Simulation/blob/master/GeneticAlgorithm/perfGA.R.

[2] Schaeffer E. Práctica 10: Algoritmo genético., 2021. URL https://elisa.dyndns-web.com/teaching/comp/par/p10.html.

[3] The R Foundation. The R Project for Statistical Computing, 2021. URL https://www.r-project.org/.