

P10

Ismael Crespo

4 de mayo de 2022

1. Introducción

Este trabajo presenta la aplicación del modelo de mochilas para simular mutaciones genéticas y reproducciones para obtener contenidos óptimos de peso en relación con el valor de cada miembro de la población a partir de la generación de un vector característico para cada miembro. E. Schaeffer [2] describe el modelo de mochilas como un problema que consiste en seleccionar un subconjunto de objetos de tal forma que (i) no se exceda la capacidad de la mochila en términos de la suma de los pesos de los objetos incluidos, y que (ii) el valor total de los objetos incluidos sea lo máximo posible.

2. Objetivo

Generar tres instancias con tres distintas reglas:

- 1.-El peso y el valor de cada objeto se generan independientemente con una distribución uniforme.
- 2.-El valor de cada objeto se generan independientemente con una distribución exponencial y su peso es inversamente correlacionado con el valor.
- 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.

Determinar para cada uno de los tres casos si variar la probabilidad de mutación, la cantidad de cruzamientos y el tamaño de la población tienen un efecto estadísticamente significativo en la calidad de resultado, manteniendo el tiempo de ejecución fijo.

3. Programación en R

La obtención del valor óptimo del peso y valor de cada miembro de la población se realiza utilizando las rutinas elaboradas por E.Schaeffer [2], este valor óptimo se fija como un límite en el que valores mayores a este no serán aceptados dentro de la población y los valores más cercanos a este se considerarán de mejor calidad. Tomando como base el algoritmo genético presentado por E.Schaeffer [2] se agregaron las tres reglas para la generación de los valores y pesos, la generación de los genes se mantuvo con la misma lógica, y se generaron los vectores PM (Probabilidad de Mutación) 'POB (Población) y REP (Reproducción) para estudiar el impacto de estas variables en la simulación.

En el código 1 se presenta como se agregó a la rutina las reglas para la generación de los valores y el peso presentadas como `generacion.peso.1`, `generacion.peso.2` y `generacion.peso.3` y de manera similar para la generación de valores.

Cada regla se estudio variando los valores de PM, POB y REP con dos valores diferentes para cada uno, para realizar un análisis estadístico cada experimento se replicó tres veces. Para consulta a detalle de los códigos consulte el repositorio de I. Hernández [1]

Código 1: Reglas para la generación de los pesos y valores de los posibles componentes del gen.

```

1  generador.pesos_1<- function(cuantos, min, max) {
2      return(sort(round(normalizar(runif(cuantos)) * (max - min) + min)))
3  }
4
5  generador.valores_1 <- function(cuantos, min, max) {
6      return(sort(round(normalizar(runif(cuantos)) * (max - min) + min)))
7  }
8  }
9
10 generador.valores_2 <- function(cuantos, min, max) {
11     return(sort(round(normalizar(rexp(cuantos)) * (max - min) + min)))
12 }
13
14 generador.pesos_2 <- function(pesos, min, max) {
15     n <- length(valores)
16     pesos <- double()
17     for (i in 1:n) {
18         media <- valores[n-i+1]
19         desv <- runif(1)
20         pesos<- c(pesos, rnorm(1, media, desv))
21     }
22     pesos <- normalizar(pesos) * (max - min) + min
23     return(pesos)
24 }
25
26 generador.pesos_3<- function(cuantos, min, max) {
27     return(sort(round(normalizar(rnorm(cuantos)) * (max - min) + min)))
28 }
29
30 generador.valores_3<- function(pesos, min, max) {
31     n <- length(pesos)
32     valores <- double()
33     for (i in 1:n) {
34         media <- (pesos[i])^2
35         desv <- runif(1)
36         valores <- c(valores, rnorm(1, media, desv))
37     }
38     valores <- normalizar(valores) * (max - min) + min
39     return(valores)
40 }

```

4. Resultados

Para medir que tan cerca se esta del valor óptimo se utilizó la ecuación 1

$$Desviación = \frac{\text{óptimo} - \text{mejor}}{\text{óptimo}} \quad (1)$$

Donde mejor es el valor mas grande encontrado al tiempo final sin que sea mayor al óptimo. La figura 1 presenta los resultados para las tres reglas variando la probabilidad de mutación, la figura 2 presenta los resultados para las tres reglas variando las reproducciones y en la figura 3 se presentan los resultados de las tres reglas variando la población. En la figura 4 se presenta los resultados del algoritmo utilizando la primer regla y utilizando el

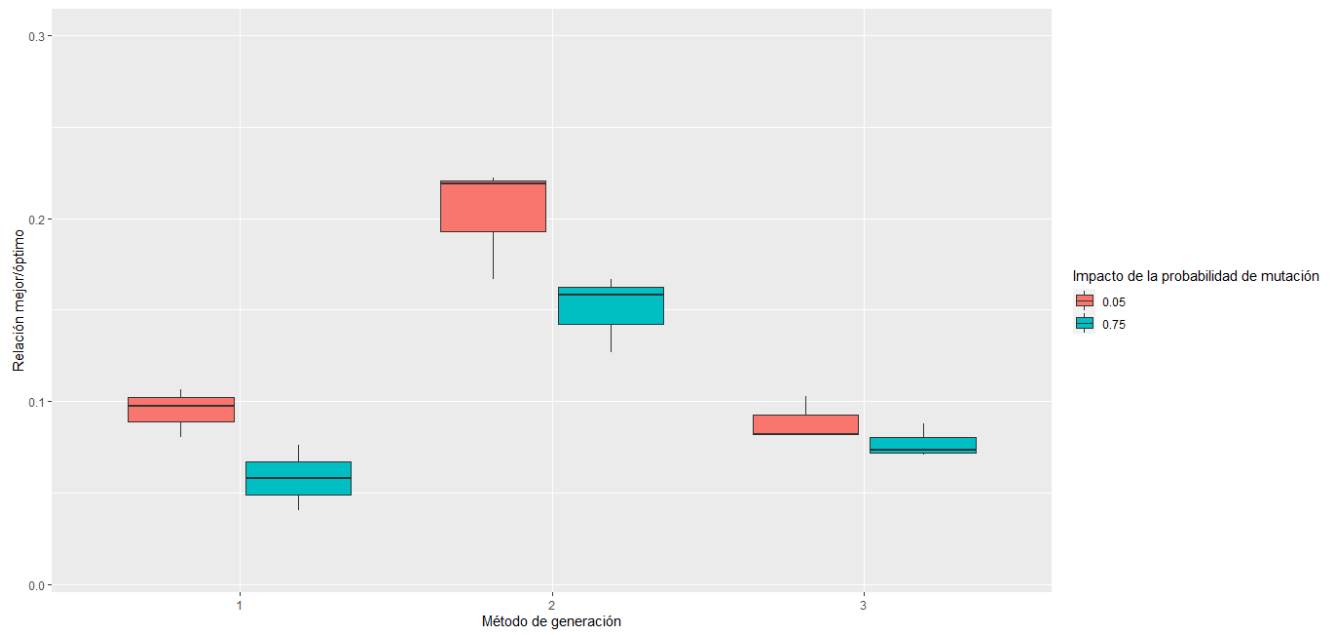


Figura 1: Resultados de la aplicación de un algoritmo genético aplicando las reglas descritas y variando la probabilidad de mutación.

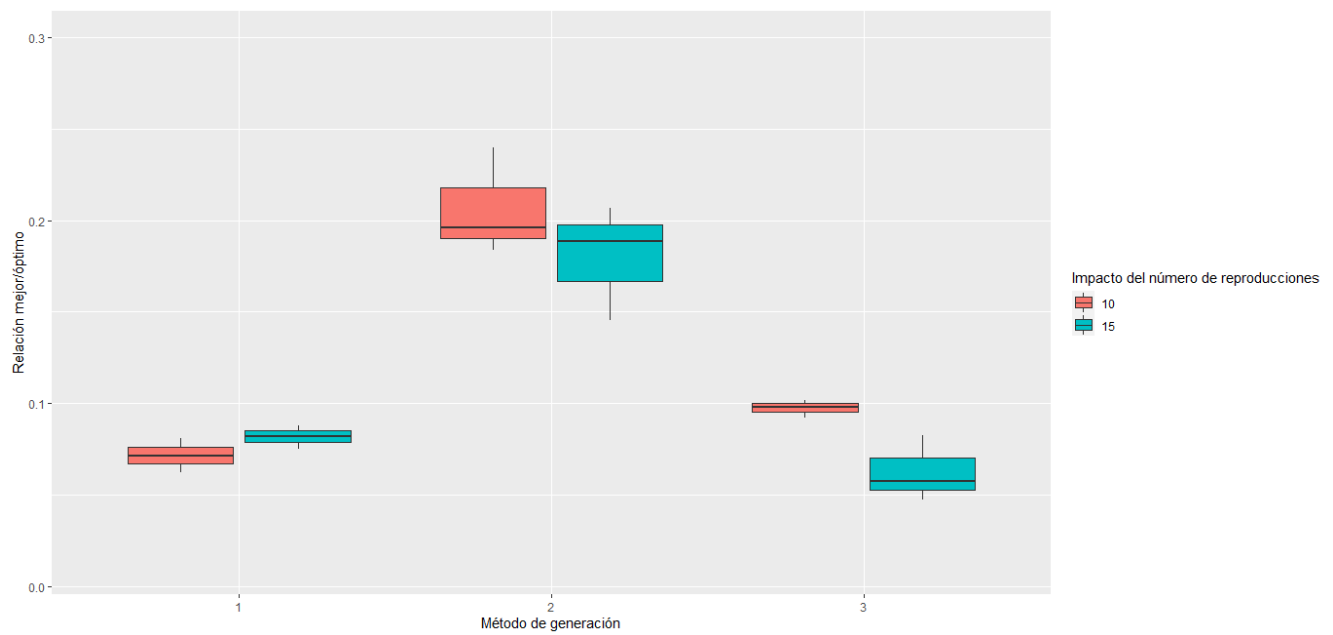


Figura 2: Resultados de la aplicación de un algoritmo genético aplicando las reglas descritas y variando las reproducciones para generar hijos e hijas.

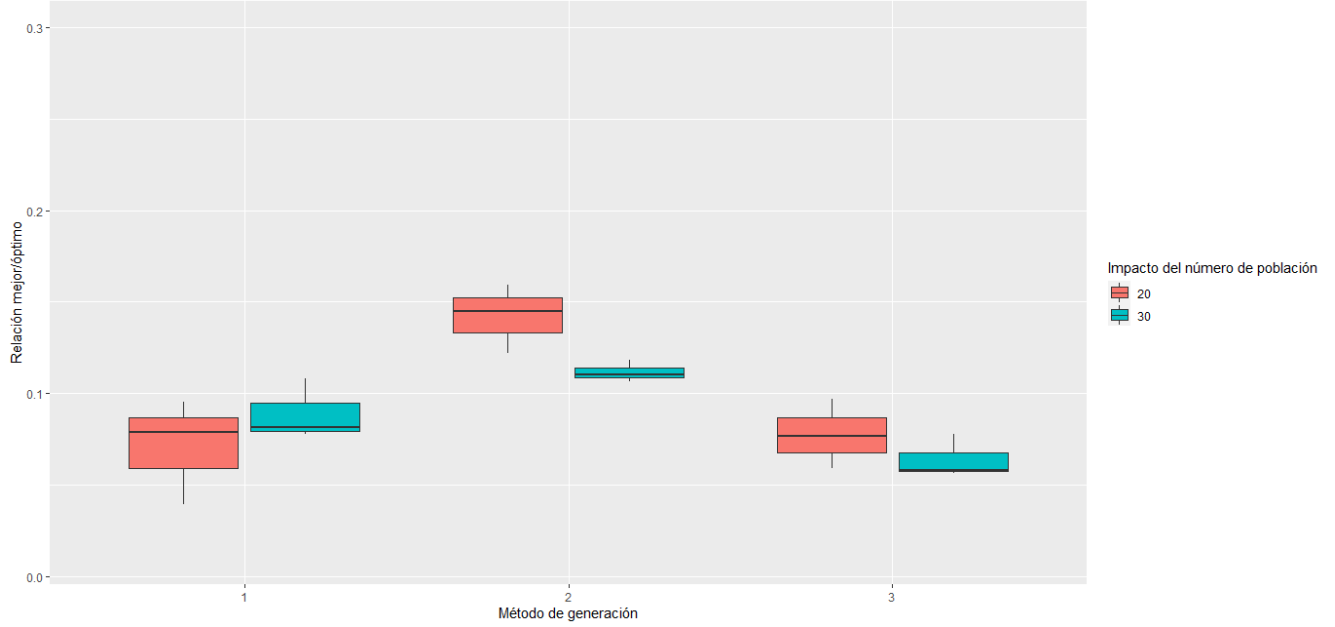


Figura 3: Resultados de la aplicación de un algoritmo genético aplicando las reglas descritas y variando la población permisible.

total de las variables para estudiar su impacto en los resultados. Cada simulación se replicó un total de 3 veces, el cuadro 1 presenta un análisis estadístico de cada experimento, por medio del `wilcox.test` se obtuvo el $p.value$ de la relación entre métodos, los resultados no presentan significancia ($p.value > 0.05$) cuando se varía PM, POB y REP entre las reglas 1 Y 3, solo tiene significancia utilizando la regla 2. Al comprar las reglas 3 y 1 en todos los casos tampoco se obtuvo significancia ($p.value > 0.05$), la regla 2 si presentó significancia al compararse con la regla 1 y 3 ($p.value < 0.05$). La figura 4 presenta la combinación de todos los valores de las variables utilizando la regla 1.

Cuadro 1: Estadística de cada simulación.

	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
REGLA 1 PM=0.50	3	0.095	0.013	0.080	0.089	0.102	0.107
REGLA 1 PM=0.75	3	0.058	0.018	0.040	0.049	0.067	0.076
REGLA 2 PM=0.50	3	0.203	0.031	0.167	0.193	0.221	0.222
REGLA 2 PM=0.75	3	0.150	0.021	0.127	0.142	0.162	0.167
REGLA 3 PM=0.50	3	0.089	0.012	0.082	0.082	0.092	0.103
REGLA 3 PM=0.75	3	0.077	0.009	0.071	0.072	0.080	0.088
REGLA 1 POB=20	3	0.071	0.029	0.039	0.059	0.087	0.095
REGLA 1 POB=30	3	0.089	0.017	0.077	0.079	0.095	0.108
REGLA 2 POB=20	3	0.142	0.019	0.122	0.133	0.152	0.159
REGLA 2 POB=30	3	0.112	0.006	0.107	0.108	0.114	0.118
REGLA 3 POB=20	3	0.077	0.019	0.059	0.068	0.087	0.097
REGLA 3 POB=30	3	0.064	0.012	0.056	0.057	0.068	0.078
REGLA 1 REP=10	3	0.071	0.009	0.062	0.067	0.076	0.081
REGLA 1 REP=15	3	0.082	0.006	0.075	0.079	0.085	0.088
REGLA 2 REP=10	3	0.207	0.029	0.184	0.190	0.218	0.240
REGLA 2 REP=15	3	0.180	0.032	0.145	0.167	0.198	0.207
REGLA 3 REP=10	3	0.097	0.005	0.092	0.095	0.100	0.101
REGLA 3 REP=15	3	0.062	0.018	0.047	0.052	0.070	0.083

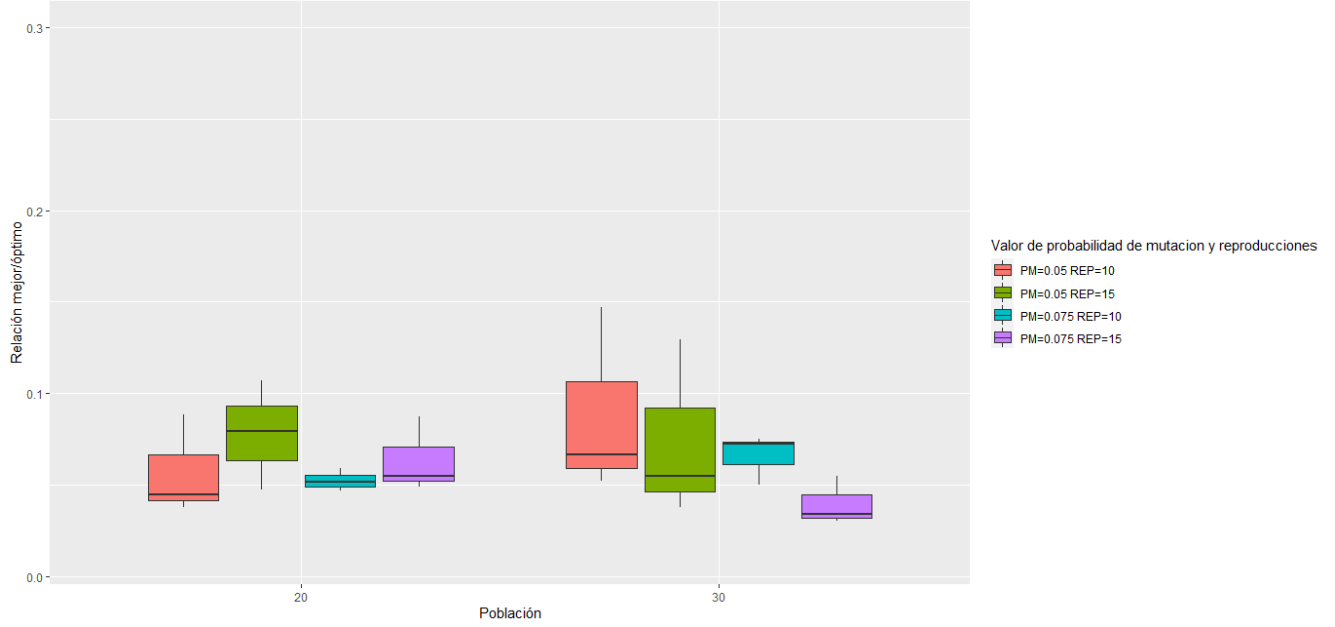


Figura 4: Resultados de la aplicación de un algoritmo genético aplicando la primera regla e intercambiando todas las variables.

El cuadro 2 presenta el análisis estadístico para el algoritmo utilizando la regla 1 y combinando todos los valores de las variables. Los resultados obtenidos al combinar los valores de las variables no representan significancia ($p.value > 0.05$).

Cuadro 2: Estadística de la aplicación del algoritmo con la regla de generación de valores y peso 1 y la combinación de todas las variables.

	N	Mean	St. Dev.	Min	Pctl(25)	Pctl(75)	Max
PM=0.05 REP=10 POB=20	3	0.057	0.027	0.038	0.041	0.066	0.088
PM=0.075 REP=10 POB=20	3	0.052	0.006	0.047	0.049	0.055	0.059
PM=0.05 REP=15 POB=20	3	0.078	0.030	0.047	0.063	0.093	0.107
PM=0.075 REP=15 POB=20	3	0.064	0.021	0.049	0.052	0.071	0.087
PM=0.05 REP=10 POB=30	3	0.088	0.051	0.052	0.059	0.107	0.147
PM=0.075 REP=10 POB=30	3	0.066	0.014	0.050	0.061	0.074	0.075
PM=0.05 REP=15 POB=30	3	0.074	0.049	0.038	0.046	0.092	0.129
PM=0.075 REP=15 POB=30	3	0.040	0.013	0.030	0.032	0.044	0.055

5. Conclusiones

La regla 2 para la generación de valores y pesos presenta los peores resultados en relación con el óptimo. El cambio de las reglas tiene significancia, sin embargo el uso de los diferentes valores para cada variable con la misma regla no presenta cambios importantes en el desarrollo del algoritmo.

Referencias

- [1] I. Crespo. P10: algoritmo genético. 2022. URL <https://github.com/IsmaelHC/Simulacion-NANO-2022/tree/main/P10>.
- [2] E. Schaeffer. Simulación. 2022. URL <https://satuelisa.github.io/simulation>.