

Simulación de Sistemas

Práctica 11

Frentes de Pareto
1455175: Ángel Moreno

22 de octubre de 2018

1. Introducción

Se considera un problema multicriterio, en el cual se tiene un conjunto de variables que le asignan valores de tal forma que se optimicen dos o más funciones objetivos. Los problemas típicos es que una mejora en una función objetivo puede corresponder a empeorar otras de las funciones objetivos y además las restricciones que en este caso no consideraremos.

Se tiene un generador de polinomios el cual tomarán el papel de las funciones objetivos. Los polinomios tienen una variable por término y un término por variable.

Se define la dominancia de Pareto para clasificar si una solución es buena: una solución domina a otra si no empeora ninguno de los objetivos y mejora a por lo menos uno. A las soluciones no dominadas se les conoce como un frente de Pareto.

1.1. Tarea

La tarea consiste en paralelizar el cálculo donde convenga y en graficar el porcentaje de soluciones de Pareto como función del número de funciones objetivo como diagramas de violín combinados con diagramas de caja-bigote, verificando que las diferencias observadas sean estadísticamente significativas. Razona en escrito a qué se debe el comportamiento observado.

2. Simulación

Se ejecuto simulaciones con polinomios con 5 términos, de 4 variables y un grado máximo de 3, se considera 300 soluciones iniciales para buscar el porcentaje del frente de Pareto, variando de 2 hasta 10 funciones objetivos con 50 réplicas ya en el código en paralelo.

2.1. Resultados

La figura 1 muestra los resultados.

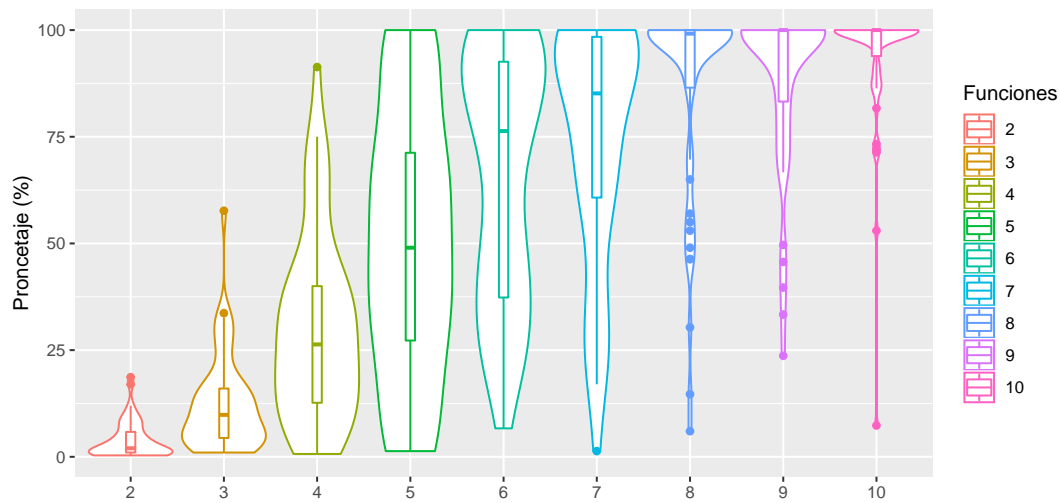


Figura 1: Resultados de cantidad de funciones objetivos contra el porcentaje de 50 réplicas.

Se observa resultados muy variados pero podemos ver una clara tendencia hacia el porcentaje del 100 % al momento de considerar más funciones objetivos. Se hace una prueba estadística Dunn en donde se verifican las diferencias estadísticas a pares.

```
1 > dunn.test(resultados$Porcentaje, resultados$Funciones)
2   Kruskal-Wallis rank sum test
3
4 data: x and group
5 Kruskal-Wallis chi-squared = 300.6936, df = 8, p-value = 0
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
```

Comparison of x by group (No adjustment)						
Col	Mean	10	2	3	4	5
Row	Mean					
2	12.04222	0.0000*				
3	10.21771	-1.824509	0.0000*	0.0340		
4	8.028922	-4.013302	-2.188793	0.0143*		
5	5.953316	-6.088908	-4.264399	-2.075606	0.0190*	
6	4.073179	-7.969045	-6.144536	-3.955743	-1.880136	0.0300
7	3.051021	-8.991203	-7.166694	-4.977901	-2.902294	-1.022157
8	1.191358	-10.85086	-9.026357	-6.837564	-4.761957	-2.881820
9	0.608427	-11.43379	-9.609288	-7.420495	-5.344889	-3.464752
Col	Mean					
Row	Mean					
8	-1.859662	0.0315				
9	-2.442594	-0.582931	0.0073*	0.2800		

Se concluye que considerar más de 8 funciones objetivos no existen diferencias significativas entre los porcentajes de la frente de Pareto y mientras para menores que 8 funciones objetivos solo no existen diferencias significativas cuando se toman consecutivas las funciones, es decir, 2 funciones contra 3 funciones, 3 funciones contra 4 funciones, etc.

2.2. Reto 1

El primer reto es seleccionar un subconjunto del frente de Pareto de tal forma que la selección esté diversificada, es decir, que no estén agrupados juntos en una sola zona del frente las soluciones seleccionadas. Graficar los resultados de la selección, indicando con un color cuáles se incluyen en el subconjunto diverso.

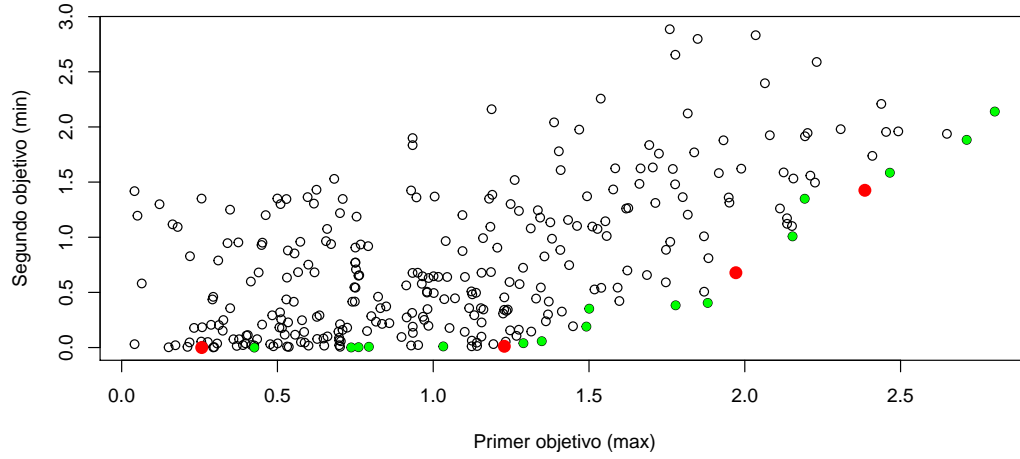


Figura 2: Gráfica que muestra el 20 % de la frente de Pareto (rojo) del total (verde).

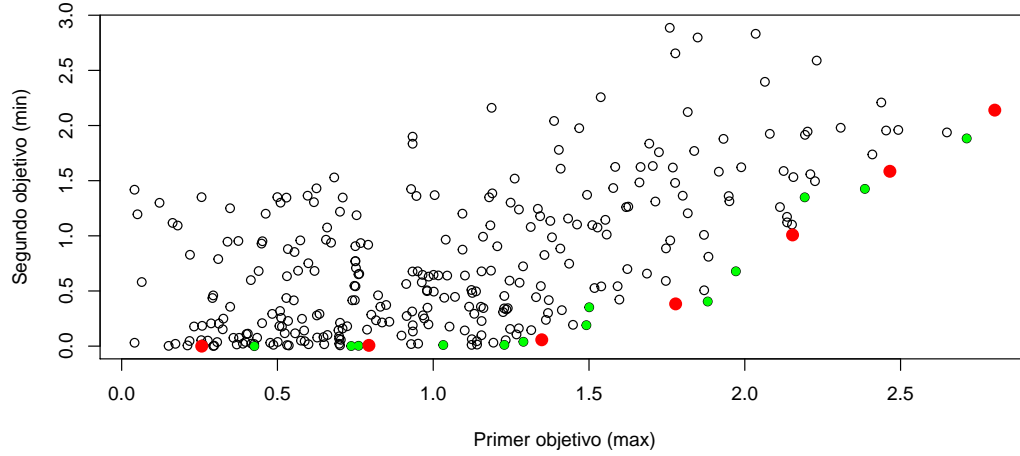


Figura 3: Gráfica que muestra el 35 % de la frente de Pareto (rojo) del total (verde).

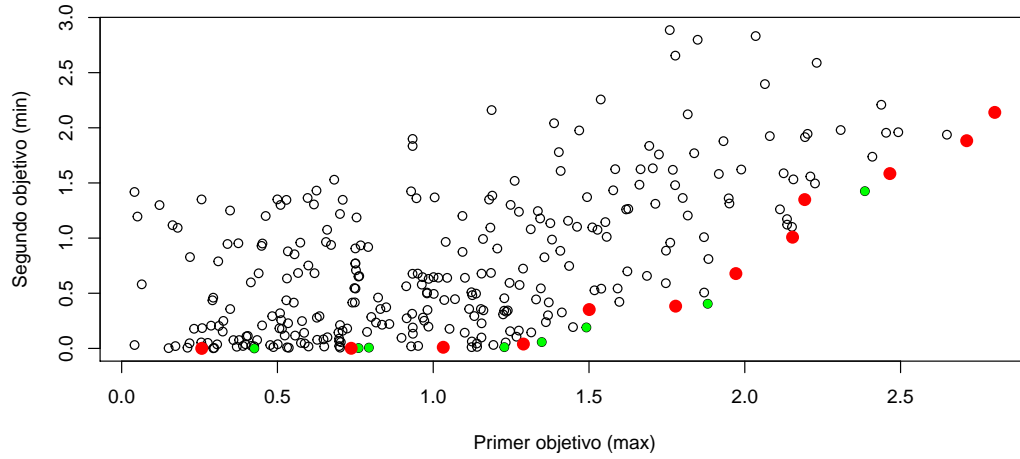


Figura 4: Gráfica que muestra el 60 % de la frente de Pareto (rojo) del total (verde).

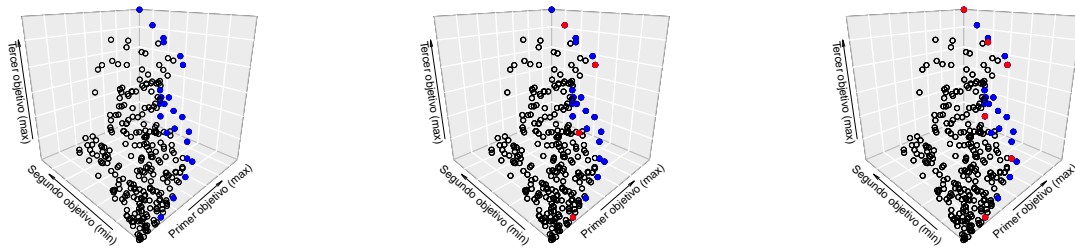


Figura 5: Gráfica en 3D de la frente de Pareto.

2.3. Reto 2

El segundo reto es adaptar el algoritmo genético de la tarea anterior para que vaya buscando mejora a un frente dado como entrada. Visualiza con un GIF animado cómo avanza la frente de una generación a otra.

Se toma como parámetros una población inicial de 300 soluciones, un probabilidad de mutación del 5 % y se hacen 50 reproducciones. La mutación consiste en alterar la solución escogiendo la variable aleatoriamente y se modifica aleatoriamente y la reproducción con la práctica escoger dos soluciones al azar y cambiar variables.

```

1 pm <- 0.05
2 rep <- 50
3
4 probabilidades <- runif(n) < pm
5 mutados <- which(probabilidades %in% TRUE)
6
7 mutaciones <- t(parSapply(cl, mutados, function(i) {
8   pos <- sample(1:vc, 1)
9   mut <- sol[i,]
10  mut[pos] <- runif(1)
11  return(as.numeric(mut))
12  }))
13
14 hijos <- matrix(parSapply(cl, 1:rep, function(i) {
15   padres <- sample(1:n, 2, replace = FALSE)
16   pos <- sample(2:(vc-1), 1)
17   x <- sol[padres[1],]
18   y <- sol[padres[2],]
19   xy <- c(x[1:pos], y[(pos+1):vc])
20   yx <- c(y[1:pos], x[(pos+1):vc])
21   return(as.numeric(c(xy, yx)))
22  }), ncol = vc, byrow = TRUE)

```

Luego ordenamos con respecto a la dominancia de cada solución, es decir, desde las no dominadas hasta las mas dominadas y se quedan las 200 primeras soluciones. La figura 6 muestrs algunas imágenes del GIF del genético.

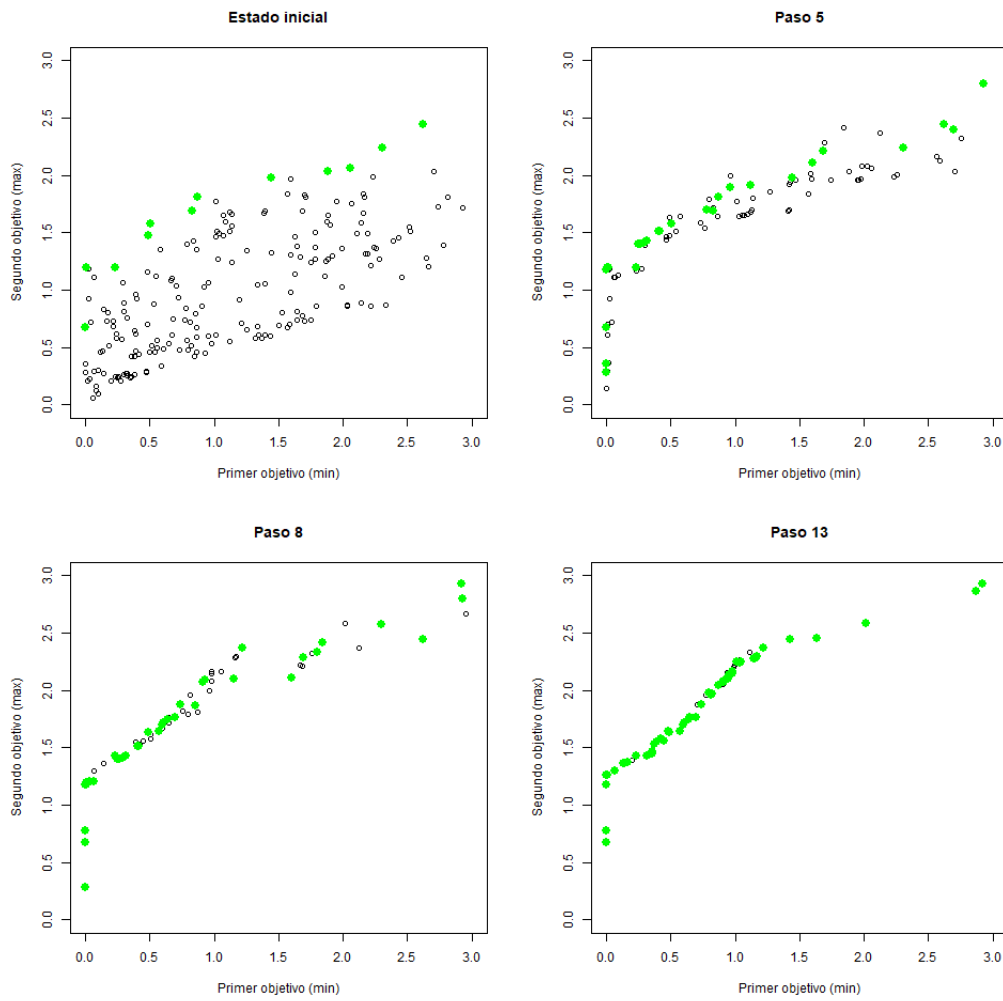


Figura 6: Imágenes de como avanza el genético.

Referencias

- [1] SCHAEFFER E. *R paralelo: simulación y análisis de datos*, 2018.
<https://elisa.dyndns-web.com/teaching/comp/par/>
- [2] VALDES E. *Repository of Github*, 2017.
<https://github.com/eduardovaldesga/SimulacionSistemas>
- [3] SAUS L. *Repository of Github*, 2018.
<https://github.com/pejli/simulacion>