Práctica 11: Frentes de Pareto

Alumno: José Adrian Garcia Fuentes
Profesor: Satu Elisa Schaeffer

Universidad Autónoma de Nuevo León. Facultad de Ingeniería Mecánica y Eléctrica.
09/mayo/2021

Resumen

Se realizara una optimización de multicriterio implementando un frente de pareto con la finalidad de definir un conjunto de soluciones que son buenas al considerar todos los objetivos, determinando los puntos con mejor compromiso en cada función.

Palabras Claves: *pareto, multicriterio, polinomios.*

1. Introducción

En muchas ocasiones es muy difícil tomar decisiones, sobre todo cuando tenemos varios objetivos para alcanzar y no nos decidimos cuál es la mejor ruta para obtener el óptimo desempeño, por lo cual existe una herramienta llamada frentes de pareto que nos permite contribuir en la mejor optimización de un objetivo. Muchos problemas reales deben ser predecibles al mismo tiempo y en ocasiones de forma conjunta, sin embargo, es muy común que los objetivos se encuentren en conflictos entre sí, para llegar al objetivo hay que considerar ciertos conceptos que nos permiten elegir las condiciones necesarias para obtener un mejor resultado, tales como la simplicidad, en muchas ocasiones los problemas tienden a modelarse para cumplir con un solo. Cuando se intentan optimizar simultáneamente múltiples objetivos, estos se contraponen entre sí, pues mientras uno mejora los otros empeoran y viceversa. Si analizamos las soluciones de estos problemas y sus evaluaciones es fácil notar la dificultad de comparar en vía de encontrar la mejor. Para estudiar este problema, vamos a primero implementar un generador de polinomios aleatorios [1]. Estos polinomios los utilizaremos como funciones objetivo. Vamos a permitir solamente una variable por término y un término por grado por variable. Para simplicidad de visualización, vamos a concentrarnos en el caso de dos funciones objetivo [1]. El análisis busca mostrar hasta cuantas funciones objetivo tiene sentido considerar, cómo encontrar un subconjunto de soluciones no dominadas que sea diverso a partir de un frente dado y como encontrar el frente de pareto. Pues bien, existen soluciones que en sí misma cumplen la definición de optimas al poder asegurar que no existe solución que sea mejor que ella misma; sin embargo, también existen soluciones que no se pueden comparar entre sí. Este conjunto de soluciones es mejor conocido como soluciones eficientes y su evaluación como frente de pareto o conjunto de soluciones no dominadas.

2. Objetivo

- Grafica el porcentaje de soluciones de Pareto como función del número de funciones objetivo para $k\epsilon[2, 8]$ en pasos de dos con diagramas de violín combinados con diagramas de caja-bigote [1].

- Razona en escrito a qué se debe el comportamiento observado [1].

3. Metodologia

La metodología empleada se realizó a través de Rstudio [2] llevando a cabo los pasos señalados en la *Práctica 11: Frentes de Pareto* [1], a partir del código en el repositorio de Schaeffer [3], se realizaron modificaciones. El código completo de la metodología empleada se encuentra en el repositorio de GitHub [4].

4. Resultados

Con la finalidad de poner un numero mínimo de réplicas se añade el valor de 30 a r posteriormente se agregan las funciones objetivo, se crean los polinomios y las soluciones son evaluadas para todos los objetivos.

```
r <- 30
vectorfunciones <- c(2,4,6,8)
poncentajefrente<-matrix(0, nrow=r, ncol=
    length(vectorfunciones))
for(f in vectorfunciones) {
k <- f
for (q in 1:r) {
for (i in 1:k) {
  obj[[i]] <- poli(md, vc, tc)
}
minim <- (runif(k) > 0.5)
sign <- (1 + -2 * minim)
n <- 200
sol <- matrix(runif(vc * n), nrow=n, ncol=vc)
val <- matrix(rep(NA, k * n), nrow=n, ncol=k)
for (i in 1:n) {
  for (j in 1:k) {
    val[i, j] <- eval(obj[[j]], sol[i,], tc)
  }
}
```

Se obtiene el porcentaje de cada frente entre soluciones y es graficado, se añaden pruebas estadísticas para comparar las medias obteniendo resultados estadísticamente diferentes entre las diferentes cantidades de funciones objetivo.

```
poncentajefrente[q,(f/2)]  <- (length(frente
    [,1])/n)*100
}
}
frentes<-c(poncentajefrente[,1],
    poncentajefrente[,2], poncentajefrente
    [,3], poncentajefrente[,4])
funciones<-c(rep(2, r), rep(4, r), rep(6, r),
    rep(8, r))
funciones<-as.factor(funciones)
data2 <- data.frame(pos=funciones, dom=frentes
    )
g2 <- ggplot(data2, aes(x=pos, y=dom)) +
    geom_violin(fill="orange", color="red")
g2 + geom_boxplot(width=0.2, fill="blue",
    color="green", lwd=2) +
  xlab("") +
  ylab("Frecuencia") +
  ggtitle("Porcentaje de Frentes de Pareto
    entre las Soluciones Totales")
t.test(poncentajefrente[,1], poncentajefrente
    [,2])$p.value
t.test(poncentajefrente[,1], poncentajefrente
    [,3])$p.value
t.test(poncentajefrente[,1], poncentajefrente
    [,4])$p.value
t.test(poncentajefrente[,2], poncentajefrente
    [,3])$p.value
t.test(poncentajefrente[,2], poncentajefrente
    [,4])$p.value
t.test(poncentajefrente[,3], poncentajefrente
    [,4])$p.value
```

Basándonos en los resultados del "violín plot"(figura 1) podemos observar que mientras mayor sea el numero de funciones objetivo, mayor sera el porcentaje de frentes de pareto entre la cantidad de soluciones existentes. Se pudiera inferir que este resultado tiene como causa que al haber muchos funciones que se tienen que optimizar, las soluciones existentes si bien no cumplen con algunas funciones en específico, pudieran estar optimizando en gran manera alguna otra, en cierta manera lo veo de esta forma, cada solución al menos le tiene que gustar a alguna función,

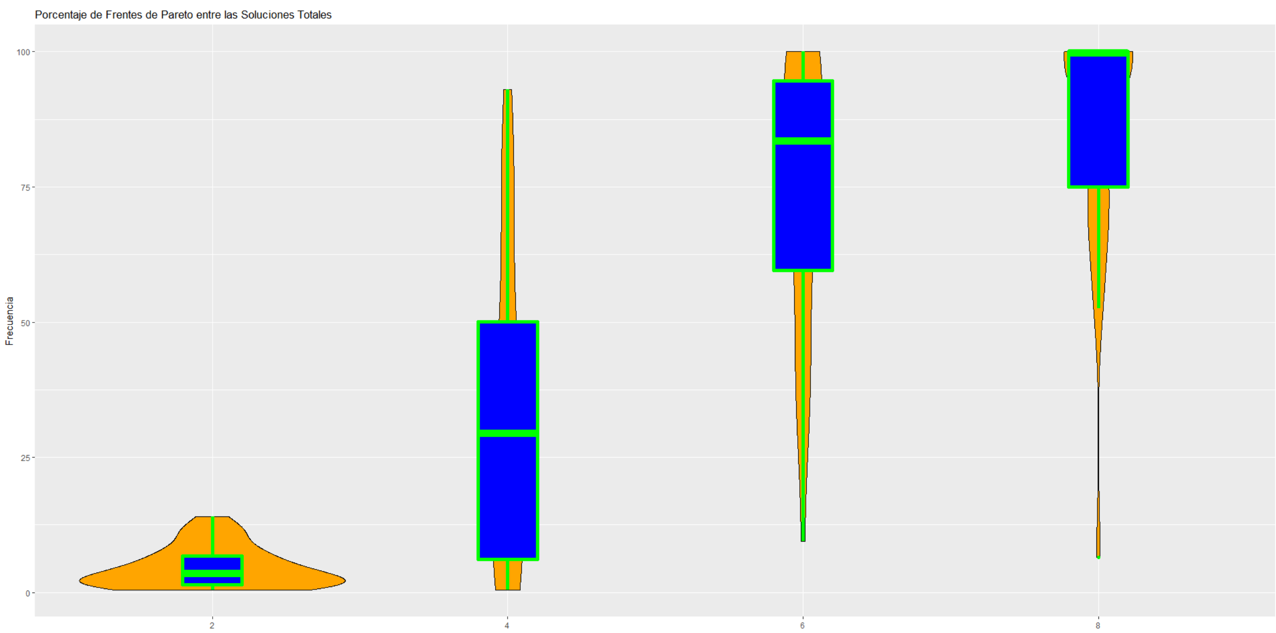

Figura 1: Porcentaje de frentes de pareto entre las soluciones totales.

3,959976$e-06$
3,450318$e-14$
1,729815$e-26$
2,321364$e-05$
1,711739$e-12$
2,845312$e-05$

Cuadro 1: % Porcentaje estadístico de pareto

En el cuadro 1 se muestran los valores estadisticos en cuanto a comparacion de las medias del porcentaje de frente de pareto

5. Reto 1

Seleccionar un subconjunto (cuyo tamaño como un porcentaje del frente original se proporciona como un parámetro) 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.

```
distancia <- function(frente) {
  m <- length(frente[,1])
  k <- length(frente[1,])
  distgrandes <- matrix(0, nrow=m, ncol=m)
  for (p1 in 1:m) {
    coordenadas1<-frente[p1,]
    dist<-c()
    t=1
    for (punto2 in 1:m) {
      if (p1 != punto2) {
      coordenadas2<-frente[punto2,]
      dist[t] <-sqrt(sum((coordenadas2-coordenadas1)^2))
      t=t+1
      }
    }
  if (p1==1) { dist <- c(max(dist), dist)
  } else if (p1 == m) { dist <- c(dist, max(dist))
  } else { dist<-c(dist[1:(p1-1)],max(dist), dist[p1:length(dist)])
  }
  distgrandes[p1,] <-dist > max(dist)*1
  }
  for (i in 1:m) {
    tope<-length(distgrandes[,1])
    if (tope>m) {
    distgrandes<-subset(distgrandes, distgrandes[i,] > 0)
    }
  }
  return(distgrandes)
}
length(distancia(frente)[,1])
```

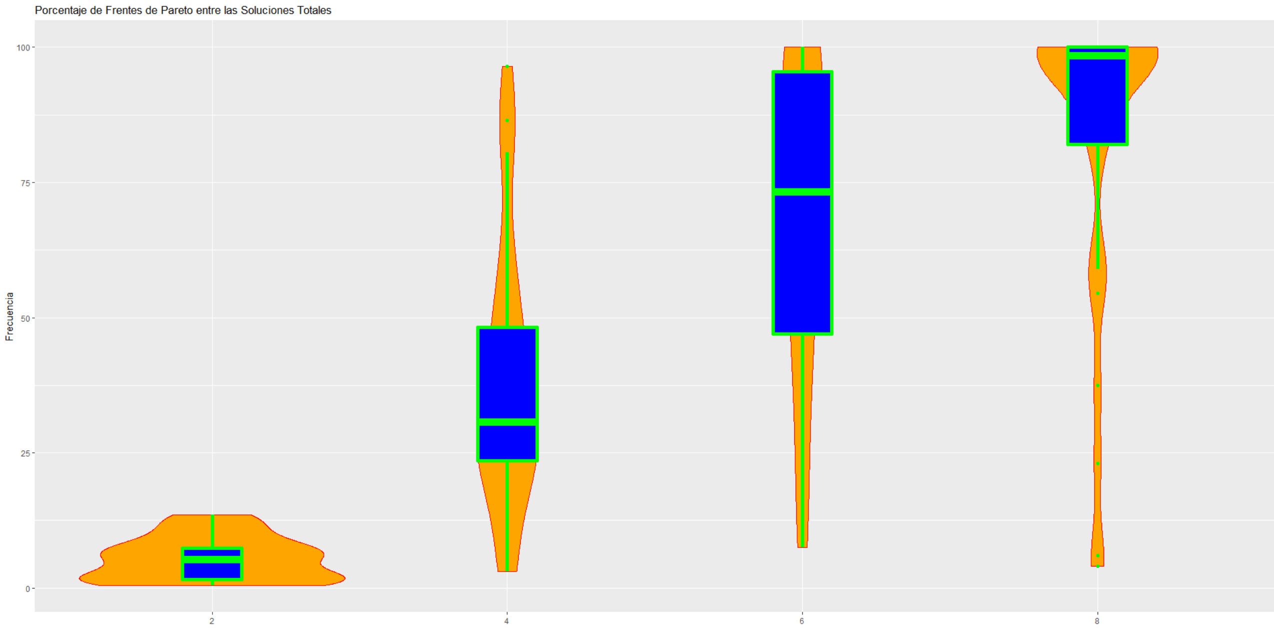

Figura 2: Porcentaje de frentes de pareto (subconjunto).

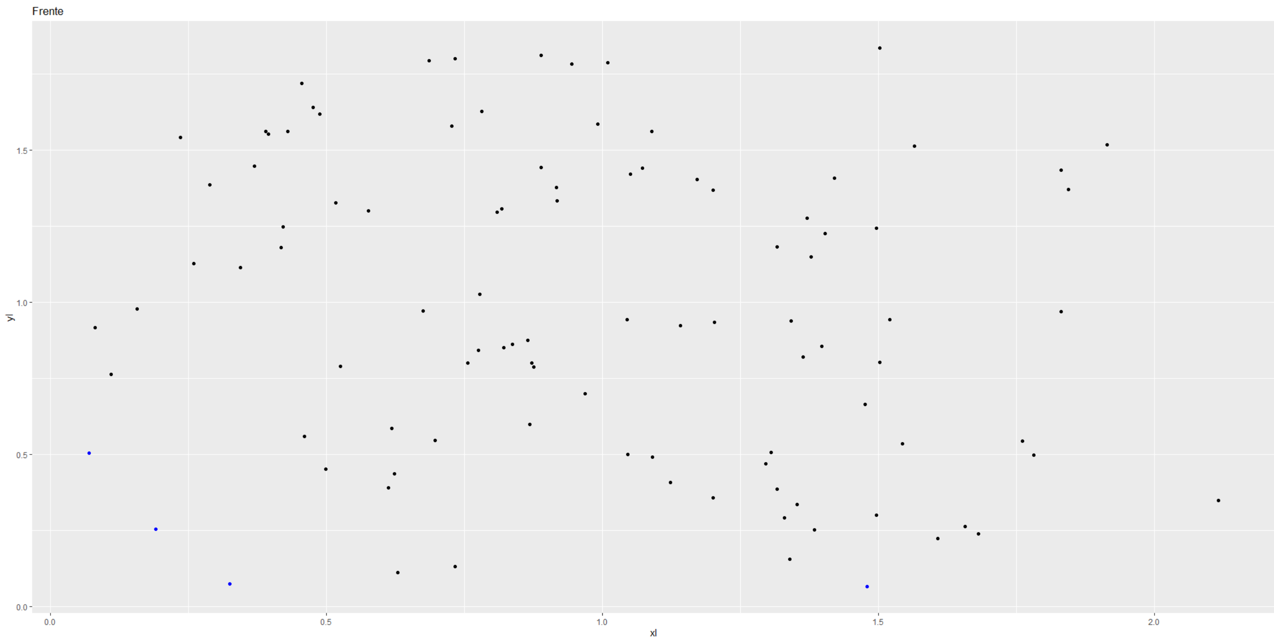

Figura 3: Frentes de pareto .

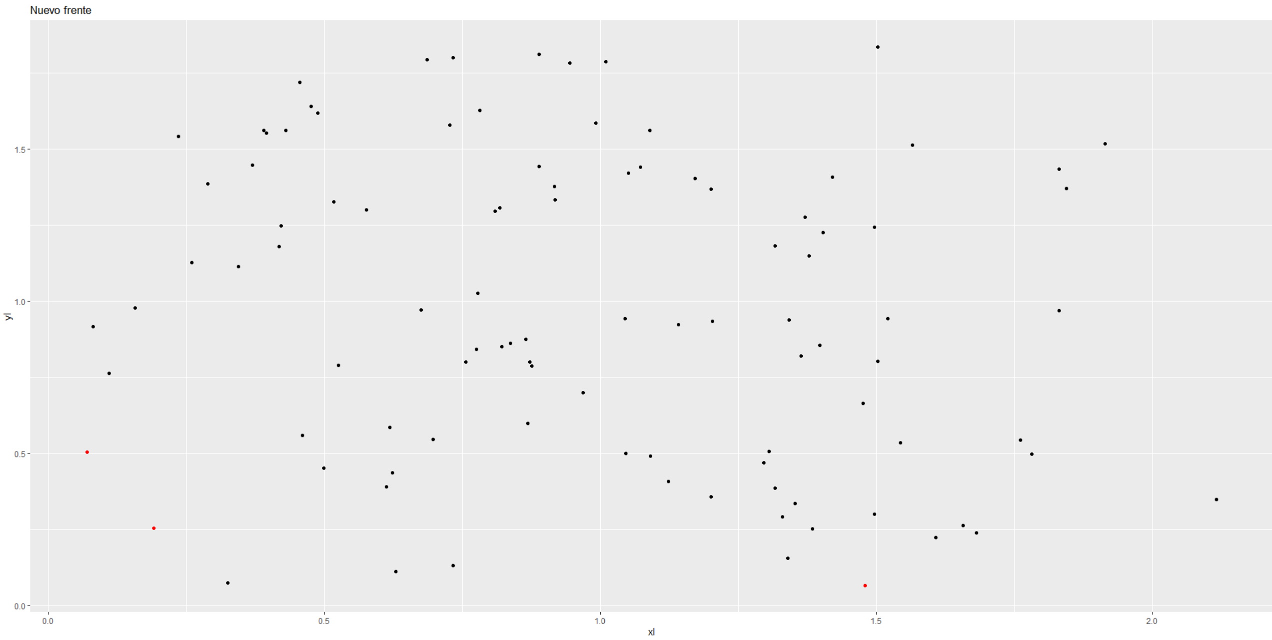

Figura 4: Frentes de pareto (puntos rojos).

En la figura 3 se muestra un frente de pareto sin embargo lo que se busca es de todas las mejores opciones mostrar solo algunas con la finalidad de que no estén agrupadas tal como se muestra en la figura 4 en comparación con la tarea base la figura 2 nos indica un porcentaje similar al de la figura 1

6. Reto 2

Adaptar el algoritmo genético de la tarea anterior para que vaya buscando mejora a un frente; la población inicial es el frente generado en la tarea y se aplica la diversificación del primer reto a cada generación después de los cruzamientos y las mutaciones. Visualiza con un GIF animado cómo avanza la frente de una generación a otra.

7. Conclusión

La mayoría de las pruebas realizadas muestran que las medias son estadísticamentes diferentes entre las diferentes cantidades de funciones objetivo basándonos en los resultados se observa que mientras mayor sea el numero de funciones objetivo, mayor sera el porcentaje de frentes de pareto entre la cantidad de soluciones existentes.

Referencias

[1] E. Schaeffer, "Práctica 11: frentes de pareto," mayo 2021. https://https://elisa.dyndns-web.com/teaching/comp/par/p11.html.

[2] J. J. Allaire, "Rstudio," mayo 2021. https://rstudio.com.

[3] E. Schaeffer, "Práctica 11: frentes de pareto," mayo 2021. https://github.com/satuelisa/Simulation/tree/master/ParetoFronts.

[4] J. A. Garcia mayo 2021. https://github.com/fuentesadrian/SIMULACION-DE-NANOMATERIALES/tree/main/Tarea%2011.