Práctica 8: Modelo de Urnas

3175

26 de marzo de 2019

1. Introducción

La práctica de urnas trara de simular efectos de coalesencia y fragmentación de cúmulos, parecido a el fenómeno de floculación, usado en tratamientos fisicoquímicos de sustancias.[**?**].

2. Objetivo

En base al modelo de urnas, simular un modelo de coalesencia y fragmentación de partículas y comparar el tiempo entre llevar un proceso paralelizado o uno secuencial.

3. Metodología

Usando de base el código proporcionado[**?**], que usa las funciones de union y separación de partículas n, partiendo de una distribución normal de frecuencias de tamaño de cúmulo k.

```
1   k <- cumu
2   n <- part
3   originales <- rnorm(k)
4   cumulos <- originales - min(originales) + 1
5   cumulos <- round(n * cumulos / sum(cumulos))
6   assert(min(cumulos) > 0)
7   diferencia <- n - sum(cumulos)
8   if (diferencia > 0) {
9     for (i in 1:diferencia) {
10      p <- sample(1:k, 1)
11      cumulos[p] <- cumulos[p] + 1
12    }
13  } else if (diferencia < 0) {
14    for (i in 1:-diferencia) {
15      p <- sample(1:k, 1)
16      if (cumulos[p] > 1) {
17        cumulos[p] <- cumulos[p] - 1
```

Posteriormente se realiza la función de unión o rompimiento dependiendo del tamaño del cumulo, donde se toma como criterio que el tamaño crítico el cual se establece en la mediana de los cúmulos, cumulos mayores al tamaño critico tienden a separarse en 2 partes de tamaños aleatorios diferentes de cero y menores solo pueden unirse

```
1   c <- median(cumulos) # tamanio critico de cumulos
```

```
  d <- sd(cumulos) / 4 # factor arbitrario para suavizar la curva
  rotura <- function(x) {
    return (1 / (1 + exp((c - x) / d)))
  }
  union <- function(x) {
    return (exp(-x / c))
  }
  romperse <- function(tam, cuantos) {
    romper <- round(rotura(tam) * cuantos) # independientes
    resultado <- rep(tam, cuantos - romper) # los demas
    if (romper > 0) {
      for (cumulo in 1:romper) { # agregar las rotas
        t <- 1
        if (tam > 2) { # sample no jala con un solo valor
          t <- sample(1:(tam-1), 1)
        }
        resultado <- c(resultado, t, tam - t)
      }
    }
    assert(sum(resultado) == tam * cuantos) # no hubo perdidas
    return(resultado)
  }
  unirse <- function(tam, cuantos) {
    unir <- round(union(tam) * cuantos) # independientes
    if (unir > 0) {
      division <- c(rep(-tam, unir), rep(tam, cuantos - unir))
      assert(sum(abs(division)) == tam * cuantos)
      return(division)
    } else {
      return(rep(tam, cuantos))
```

Finalmente se determina la duración de éste proceso en 50 pasos, se realizan 5 repeticiones y se varían los valores de k y n para correr el proceso de manera secuencial y paralelizada tomando los tiempos que toma, de ésta manera comparar si se ahorra tiempo en un tratamiento u otro mediante una prueba estadística.

```
replicas <- 5
tiemposSP <- c()
N <- seq(30000,45000,5000)
K <- seq(300,450,50)
for(part in N) {
  for(cumu in K) {
    for(rep in 1:replicas) {
      tsec <- data.matrix(system.time(urnas(cumu,part))[1])[1]
      tiemposSEC <- c(tiemposSEC, tsec)
    }
  }
}
suppressMessages(library(doParallel))
cluster <- makeCluster(detectCores() - 3)
registerDoParallel(cluster)
tiemposCP <- c()
for(part in N) {
  for(cumu in K) {
    tcp <- foreach(r= 1:replicas, .combine = c) %dopar% data.matrix(system.time(urnas(cumu, par
```

```
20        tiemposCP <- c(tiemposP,tp) #se guarda tiempo
21        stopImplicitCluster()
```

4. Resultados

Los resultados mostrados en las figuras 1 y 2 son los resultados de los tiempos a diferentes valores de k y n secuencial y paralelizadamente respectivamente. Se puede observar valores cercanos entre la manera paralelizada y la secuencial, sin embargo es notorio que de manera secuencial los valores presentaron menos variaciones entre repeticiones que los de manera secuencial. La prueba estadística da como resultado que los resultados a éstas repeticiones no son muy diferentes entre uno y otro tratamiento.

```
1          Wilcoxon rank sum test with continuity correction
2
3  data:  tiemposSEC and tiemposP
4  W = 2583, p-value = 0.9825
5  significancia [0.05]
```

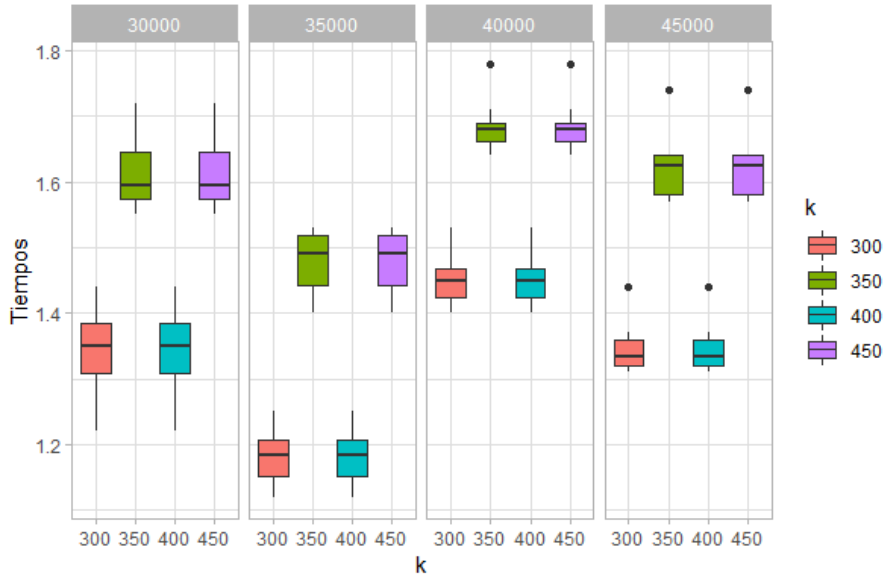

Figura 1: Tiempos en el proceso de manera secuencial

5. Conclusiones

Se concluye en base a los resultados que la diferencia de tratamientos no influye de manera significativa en los tiempos de procesamiento a éste número de repeticiones. Sería importante verificar posteriormente si esto cambia a diferente número de repeticiones.

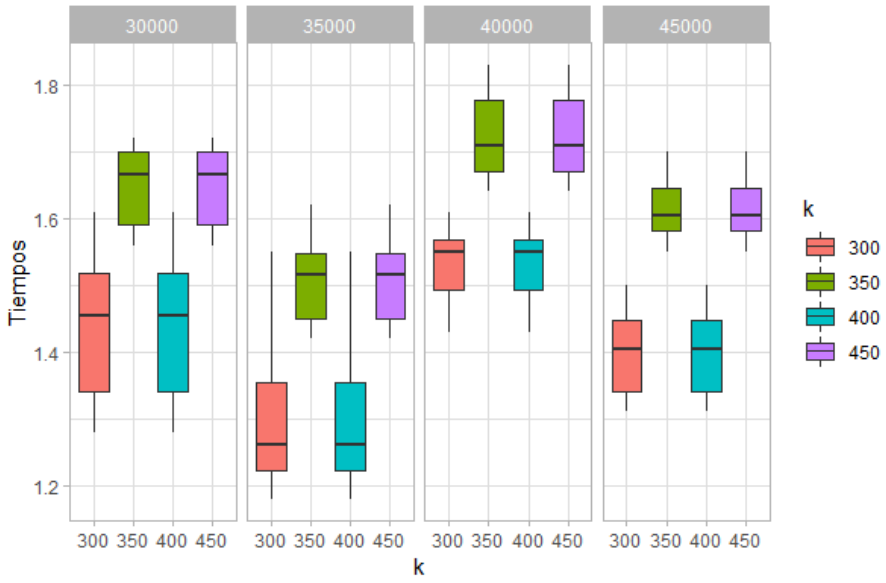

Figura 2: Tiempos en el proceso de manera paralelizada

Referencias

[] María Gonzales Gonzalo. Aplicación de procesos de coagulación floculación en la regeneración de aguas depuradas. Universidad de Zaragoza, 2010. URL `https://zaguan.unizar.es/record/4935?ln=es`.

[] Elisa Schaeffer. Práctica 8: Modelo de urnas. Página Web, 2019. URL `https://elisa.dyndns-web.com/teaching/comp/par/p8.html`.