

# Proyecto intermedio - Herramientas computacionales

William Oquendo, woquendo@gmail.com

## 1 Percolación de sitio y probabilidad crítica

La percolación está relacionada con el paso de una sustancia o elemento o información de una frontera a otra, generalmente a través de un filtro. En este proyecto se simulará el llenado aleatorio de una malla bidimensional.

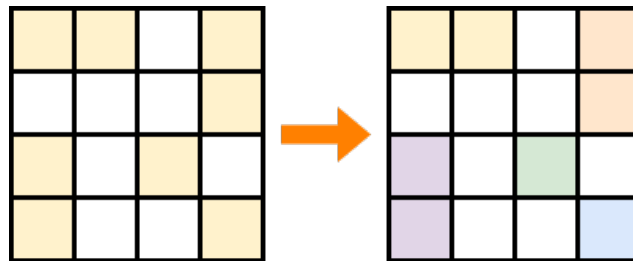
Imagine que usted escoge una probabilidad de llenado  $p$ , y va recorriendo uno a uno los sitios de una matriz de tamaño  $L \times L$ , y ocupa el sitio con esa probabilidad  $p$ . Teniendo en cuenta solamente vecinos inmediatos (arriba-abajo-izquierda-derecha), el anterior llenado hará que aparezcan clusters de vecinos con tamaños específicos que dependerán de esa probabilidad  $p$ . Luego de cierta probabilidad crítica,  $p_c$ , siempre aparecerá un cluster percolante (que une de arriba a abajo o de izquierda a derecha), cuyo tamaño será cada vez mayor a medida que aumenta  $p > p_c$ .

Las preguntas que debe explorar y responder son las siguientes:

1. Cuál es la probabilidad  $P(p, L)$  de que exista un cluster percolante, en función de  $p$  y de  $L$ ? Esto se resumirá en una figura de  $P$  en función de  $p$ , para diferentes  $L$ .
2. Cuál es el tamaño promedio del clúster percolante más grande,  $s(p, L)$  en función de  $p$  y  $L$ ? El tamaño del cluster deberá estar normalizado al tamaño total del sistema,  $L \times L$ .

Para lo anterior, explore al menos 30 valores de  $p \in [0, 1]$ , con al menos 10 de esos valores en el intervalo  $[0.55, 0.65]$  y al menos cinco valores para  $L = \{32, 64, 128, 256, 512\}$ . Cada simulación deberá repetirse al menos 20 veces para al final reportar promedios y desviaciones estándar.

Como ejemplo, la siguiente figura muestra una realización de una red de  $4 \times 4$  sitios, llenada con probabilidad  $p = 0.4$ , en donde primero se realiza el llenado y luego se realiza la clasificación de los clusters (información codificada en colores). En esta realización particular no existe cluster percolante.



Si después de hacer las 20 realizaciones con esta misma  $p = 0.4$  y  $L = 3$  obtiene que solamente 4 de ellas crearon cluster percolante, entonces  $P(0.4, 3) = 4/20 = 0.2$  (deberá reportar también la desviación estándar) y además deberá reportar el tamaño promedio del cluster percolante (y su desviación estándar).

### 1.1 Entregables

El siguiente link es en donde debe entregar su proyecto (grupos mínimo de 2, máximo de 3): <https://classroom.github.com/g/o7ZRKAxB>. El deadline es [2020-05-15 Fri 11:00] .

En este repositorio asignado a su equipo, deberá subir:

- Todo el código necesario para realizar las simulaciones.
- Makefile que compile, ejecute, cree datos y figuras, y realice tests. También debe compilar con sanitizers.
- Al menos tres (3) test, que se ejecuten cuando se escriba `make test` (es decir, el makefile debe tener ese target creado), que prueben, por ejemplo, que su función para detectar clusters da probabilidad y tamaño de clúster cero cuando  $p = 0$ , para varios tamaños  $L$ , y que da 1, también para varios  $L$ , cuando  $p = 1.0$ . Pueden pensar en otros tests, que se ejecuten rápido. Pueden usar `catch2` o `google test`, como prefieran. El código debe estar modularizado para separar el main del estudio de la percolación de los test.

- Un informe en pdf, tipo informe de laboratorio, que presenta los resultados anteriores e incluya una figura del tiempo de cómputo en función del tamaño del sistema. Esta figura debe hacerse para dos niveles diferentes de optimización del compilador. Indique cualquier otra optimización que haya aplicado. Es importante indicar también las dificultades encontradas y las soluciones aplicadas.
- Un reporte de profiling en un archivo de texto llamado `profiling-report.txt` en donde se observe el flat profile de su código para la probabilidad crítica (a partir de la cuál , en promedio, aparece un cluster percolante,  $p_c \simeq 0.59271$ ) , y  $L = 128$ . Los resultados de este profiling deben ser analizados en el informe para reportar cualquier tipo de optimización que se haya realizado.