Cableado global con Simulated annealing

Proyecto final simulación estocástica

Paolo Martiniello R.
Álvaro Márquez S.

Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Matemática

7 de diciembre de 2022

Contenidos

Presentación del problema

Se presenta una placa electrónica que contiene diversas componentes que necesitan ser conectadas entre si mediante el uso de cables, sea de forma recta o con un único doblez del cable, que comparten un mismo plano, y se busca encontrar una distribución de los cables que permitan tener estos repartidos de la forma más uniforme posible dentro de la placa, este problema se conoce como global wiring. Para esto se hara uso de simulated annealing que nos permitirá llegar a una solución óptima bajo los criterios de una función objetivo a definir.

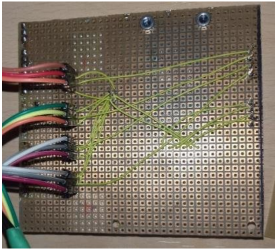

Figura 1: Ejemplo de placa electrónica

Formulación teórica del problema

El modelo utilizado para el problema es una simplificación en una grilla discreta, donde las componentes a conectar se pueden ubicar los vértices y los caminos donde pueden pasar los cables son las aristas.
Se definen los siguientes parámetros para modelar el problema:

- $\varepsilon_i \in \{-1, 0, 1\}$ indica si el cable de la i-ésima conexión dobla y si lo hace en qué sentido, para este problema se definio que 1 representaria una conexión de una componente x a una y con un camino en L con un giro a la derecha, mientras que -1 seria con giro a la izquierda. 0 representa una camino recto entre ambas componentes.

- $m_v \in \mathbb{N}$ cuenta la cantidad de cables que pasa por el camino v, es decir, cuantas conexiones entre componentes pasan por la arista v dentro de la grilla.

Método y función objetivo

La función objetivo a minimizar es

$$F = \sum_v m_v^2$$

Esta función da mayor valor a configuraciones con distribuciones de cables menos saturadas. El método a utilizar es *simulated annealing* con el algoritmo de Metropolis. Este algoritmo utiliza la matriz de transición siguiente:

$$R_{xy} = \begin{cases} \frac{1}{deg_x}, \text{ si y}\sim\text{x} \\ 0, \text{ si no} \end{cases}$$

Y define una transición bajo la comparación con una variable uniforme U de la forma:

$$X_{n+1} = \begin{cases} Y_n, \text{si } U \leq e^{-\beta*(F(Y_n)-F(X_n))} \\ X_n, \text{si } U \geq e^{-\beta*(F(Y_n)-F(X_n))} \end{cases}$$

Donde X_n es la configuración actual y Y_n un vecino de esta.

Espacio de estados de la Cadena de Markov

Primero, se define el espacio de estados como todas las configuraciones posibles, es decir, todas las combinaciones posibles de caminos entre las componentes entregadas, y se dice que dos configuraciones son vecinas si difieren en a los más una orientación de un camino en L, como se puede ver a continuación:

Figura 2: Par de estados que son vecinos entre si

Formulación en código

La información de la grilla sera entregada al código mediante tres componentes:

- Grid_link sera una matriz que contendra el tipo de camino que conecta a dos componentes, el valor en un par [i,j] representa el tipo de conexión que hay de la componente i a la componente j.
- Grid_map sera una matriz que contendra la cantidad de aristas que hay entre dos vertices dentro de la grilla, el valor en un par [i,j] representa la cantidad de conexiones entre componentes que pasan por la arista ij.
- coords sera un listado que contendra las coordenadas de todas las componentes entregadas, un elemento i de la lista sera None si no existe la componente i.

A continuación un ejemplo de una matriz de 3x3 con las siguientes conexiones: (1,4), (1,5), (1,7), (1,9)

Formulación en código

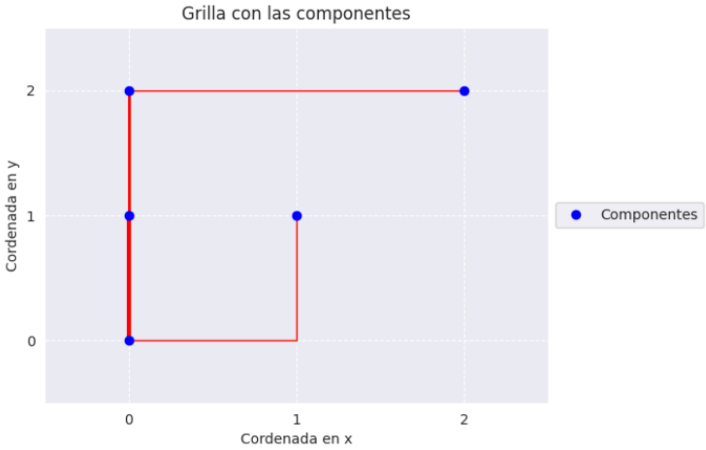

Figura 3: Ejemplo de una configuración en una grilla de 3x3

Formulación en código

$$\begin{bmatrix} N & N & N & 0 & 1 & N & 0 & N & -1 \\ N & N & N & N & N & N & N & N & N \\ N & N & N & N & N & N & N & N & N \\ 0 & N & N & N & N & N & N & N & N \\ -1 & N & N & N & N & N & N & N & N \\ N & N & N & N & N & N & N & N & N \\ 0 & N & N & N & N & N & N & N & N \\ N & N & N & N & N & N & N & N & N \\ 1 & N & N & N & N & N & N & N & N \end{bmatrix}$$

Matriz con los valores de ε_i

$$\begin{bmatrix} 0 & 1 & 0 & 3 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 2 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

Matriz con los valores de m_v

Resultados

Figura 4: Gif solución en grilla 4x4 con conexión entre todas las componentes

Resultados

Valores objetivos con $n = 10000$ pasos y grilla de 11 x 11 usando np.seed(5), donde se buscaba encontrar la forma más uniforme de conectar todas las componentes entre si colocando una de estas en cada coordenada de la grilla, con $F_{inicial} = 14321902.0$:

Tabla 1: Comparación distintos valores de β.

$\beta(n)$	F_{final}	Tiempo
n^2	14289740.0	84.93386745452881
n^3	14289740.0	85.24781107902527
$\log(n)$	14289728.0	82.28722190856934
$n\log(n)$	14289728.0	82.68437933921814
$\exp(n)$	14289774.0	82.40791392326355
$\sqrt{n}$	14289740.0	89.11154079437256

Otros métodos utilizados: Programación Lineal

- Este método usa una función objetivo que va en la misma linea que la que se definió para simulated annealing, dado que busca minimizar la capacidad E de cada arista, sujeto a que cada arista menos una cantidad de canales que se define en este método sea menor que la capacidad E. En términos de costo, este método funciona más rápido que simulated annealing, llegando a solución de grillas más complejas que la que se utilizo en este proyecto en tiempos bajo un minuto.
- Ahora, este método presenta un problema no menor que afecta en su rendimiento, que es que bajo sus condiciones no puede definir a que conexión entre dos componentes pertenece cada arista entre unas coordenadas [i,j], lo que lleva que se consideren dos veces el mismo cable para la capacidad (por ejemplo bajo el caso en que se conecta una componente a y una b, y a también se conecta con c pasando previamente por b), problema que no se presenta en simulated annealing, al poder redefinir las conexiones entre componentes dentro de las matrices utilizadas.
- En términos de costo computacional, programación lineal supera con creces a simulated annealing, pero esto a cambio de perder generalización sobre el problema de global wiring, generalización que si presenta simulated annealing.

Generalizaciones del problema

Inclusión de caminos con dos curvas

Se puede utilizar la configuración encontrada en el caso simplificado como configuración inicial para el annealing, recortando el proceso.
El código en esencia sería el mismo pero cada configuración tendría muchos más vecinos y por lo tanto la extensión sería mucho mayor y más costoso de simular.

Componentes más grandes

Se podría modelar componentes que utilicen más de un nodo de espacio. En la teoría esto se hace eliminando el enlace que existe entre los nodos de la componente y tomando la distancia como la menor a todos nodos de la componente.
Quitando las componentes grandes se puede utilizar el mismo código para encontrar una configuración inicial más óptima.
El código debería ser más sencillo de parchar, simplemente eliminando conexiones internas a la componente y considerando caminos desde un vértice a un conjunto de estos en lugar caminos de vértice a vértice.

Conclusiones

- El algoritmo de simulated annealing da buenos resultado para encontrar mejores distribuciones de cables con las restricciones dadas y es lo suficientemente simple como para que ampliar el problema sea relativamente sencillo en términos de codificación.
- Aunque los resultados con distintos valores de β no varíen mucho esto puede ser dado a las conexiones utilizadas, una configuración con menos conexiones podría mostrar una mejor relación entre los valores de β.
- Una normalización de la función objetivo también puede ser útil para mostrar de forma más notoria la mejoría antes y después de ser aplicado el algoritmo.

Referencias

- Vecchi, M. P., Kirkpatrick, S. (1983) Global wi- ring by simulated annealing., IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 2(4), 215-222.
- Ng, A. P.-C., Thompson"', C. D., Hi', -K, Kabah, K., Tomncoh, :. (s/f). Experimental results for a linear program global router. Auckland.ac.nz. Recuperado el 7 de diciembre de 2022, de https://www.cs.auckland.ac.nz/ cthombor/Pubs/Old/exptpgrouter.pdf

Cableado global con Simulated annealing

Proyecto final simulación estocástica

Paolo Martiniello R.
Álvaro Márquez S.

Ingeniería Matemática
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS UNIVERSIDAD DE CHILE

Universidad de Chile
Facultad de Ciencias Físicas y Matemáticas
Departamento de Ingeniería Matemática

7 de diciembre de 2022