Manual Técnico

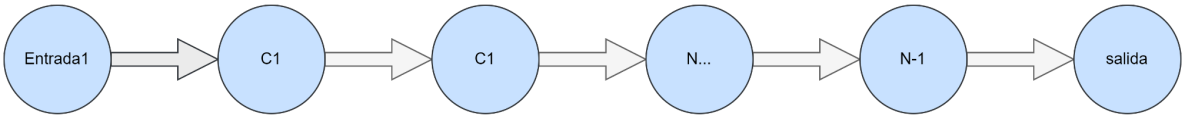
Yefer Alvarado 201731163

Requerimientos

- Python 3.9
- Conocimientos de Algoritmo Genético
- Estructura de Datos
- DearpyGui

Algoritmo Genético (Planteamiento de la función de Aptitud)

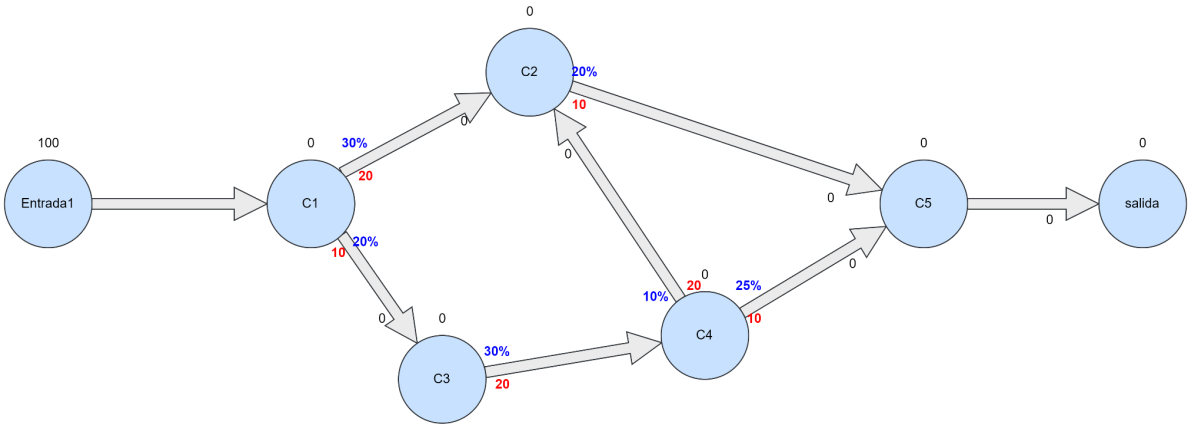
Tomando en cuenta que en el peor de los casos, viendo el grafo como una lista, es decir que cada nodo se conecta de manera lineal, o secuencial:



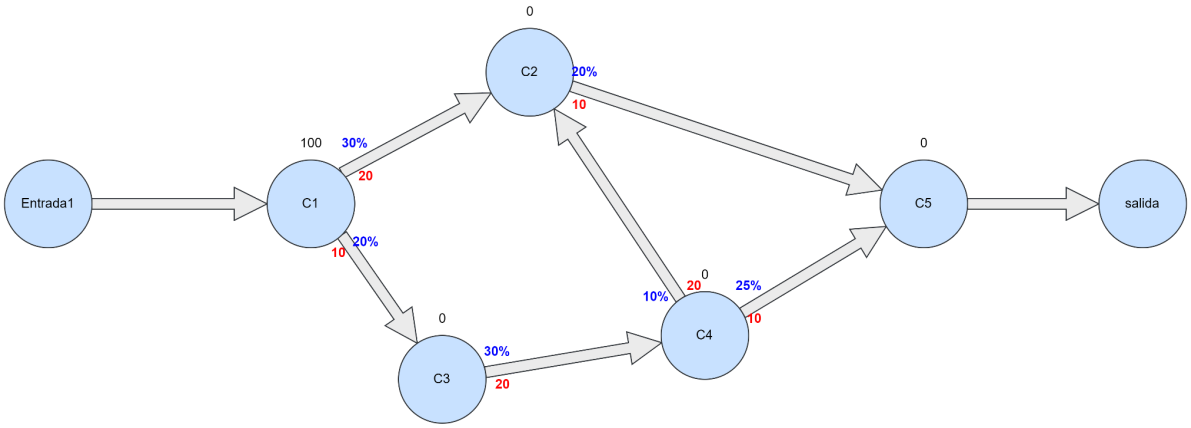
La cantidad de interacciones para que los vehículos puedan tener un gráfico fluido es de N nodos, esto quiere decir que para que puedan salir los vehículos por lo menos tiene que realizarse N iteraciones.

Planteamiento

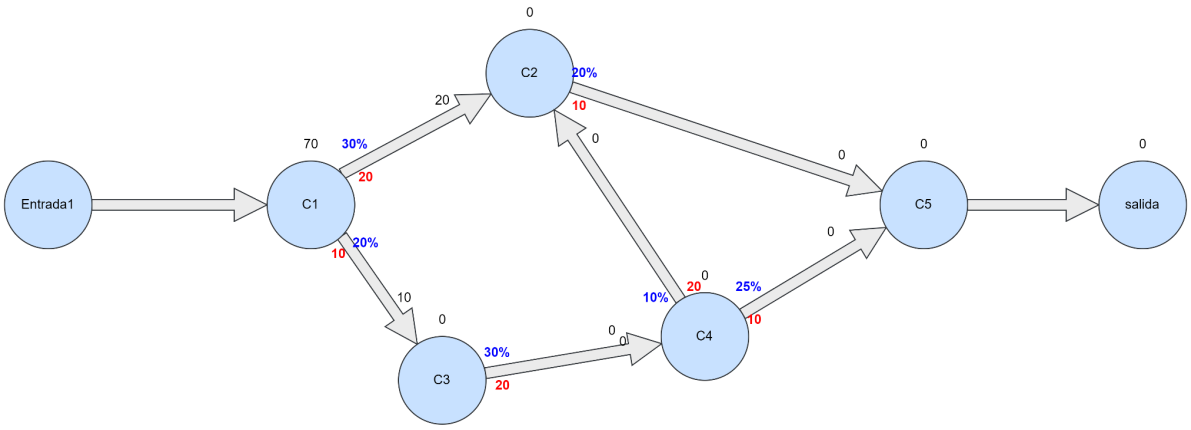
Inicialmente solo se tienen 100 vehicuilos en la entrada, por cada iteracion cada nodo tiene un estado de los vehiculos que estan en cola, y su respectivos datos por cada calle.



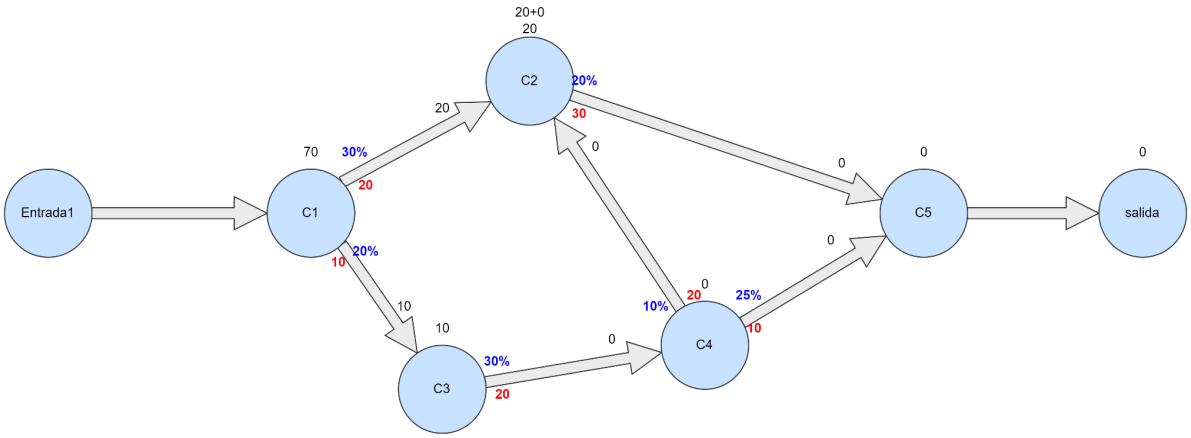
En la siguiente iteración la calles de tipo entrada envian los vehículos a las colas



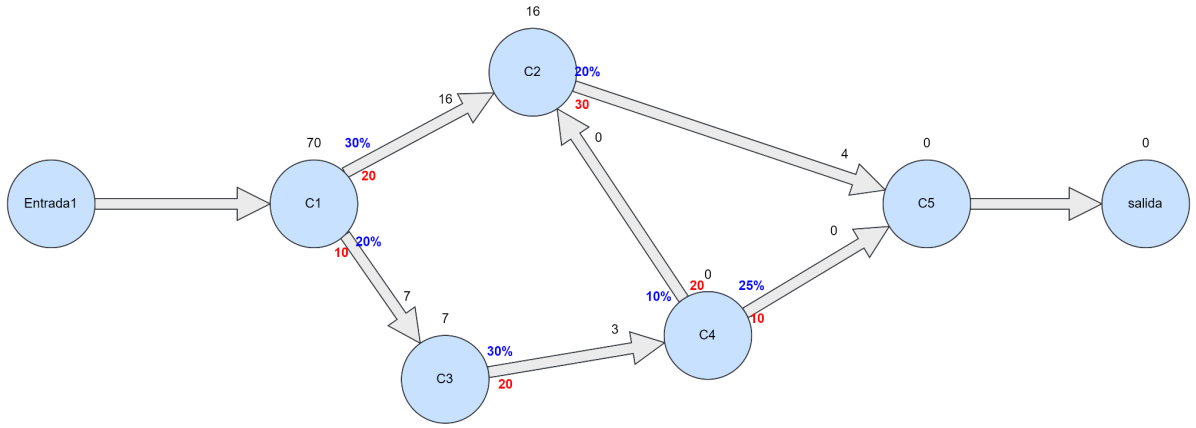
Cada nodo tiene un estado que registra cuántos hay en cola, y cuanto va a enviar a su dato, entonces se toma como un screenshot del estado actual del sistema y se envian los vehiculo con es estado actual mas no el nuevo estado los nuevo envios no se toman en cuenta.



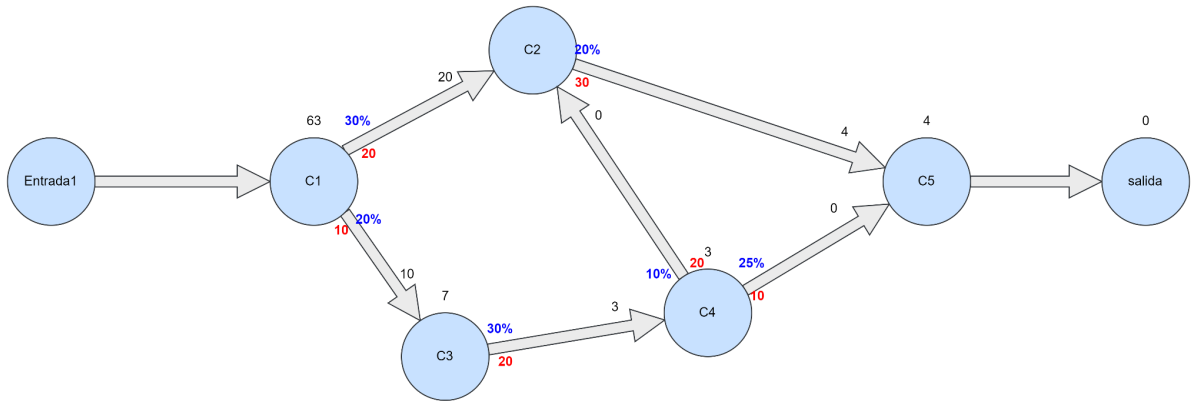
Se actualizan los estado de los nodos



Se iteran los nodos una vez mas, un screenshot y envia los vehiculos de su estado actual



Se vuelve a iterar



Entonces aparecen los primeros vehículos que saldrán del sistema, pero vemos que del nodo C4 a C2 aun no existe la fluidez de los vehículos por lo que es necesario volver a iterar hasta que se llegue al número de nodos por iteración

Algoritmo de la Función de Aptitud

```
Para - Recorrer nodos del individuo
     si el nodo ==  Entrada entonces
          Para - recorrer sus calles o arista del nodo
               enviar los vehículos a los nodos siguientes de
cada entrada
               actualizar los datos del nodo o cruce
```

```
Para -  N nodos = n iteraciones
      Para - Recorrer nodos del individuo
            Si las entrada del nodo > 0
                  Para - recorrear su aristas o calles
                        Si el nodo == CRUCE
                              num_vh = obtener entradas nodo
                              actualizar cola
                              si vehiclos_entrada > en_cola
                                    num_vh = en cola
                              arista.en_cola += num_vh
            Si no Si nodo == SALIDA
                  num_vh = obtener entradas nodo
                  arista.en_cola += num_vh
            Uptadate Arista o Calles que ingresan al nodo colas
      update_entradas_nodo actual

Para - Recorrer nodos del individuo
      Si no Si nodo == SALIDA
            Obtener total vehiculos salieron
            total_fitness = total_enter_nodo
return total_fitness
```

Objetos y Función

GraphIndividual: Cada grafo se toma como un individuo

NodeChromosome: Los nodos del grafo se toman como cromosomas

EdgeStreetGene: cada calle o arista es un gen

Tipo de algoritmo que se utilizaron

- Selection: Selección por ruleta
- Cruce: por punto, se divide el cromosoma en dos partes en función del punto de separación
- mutación: intercambio de dos genes