

---

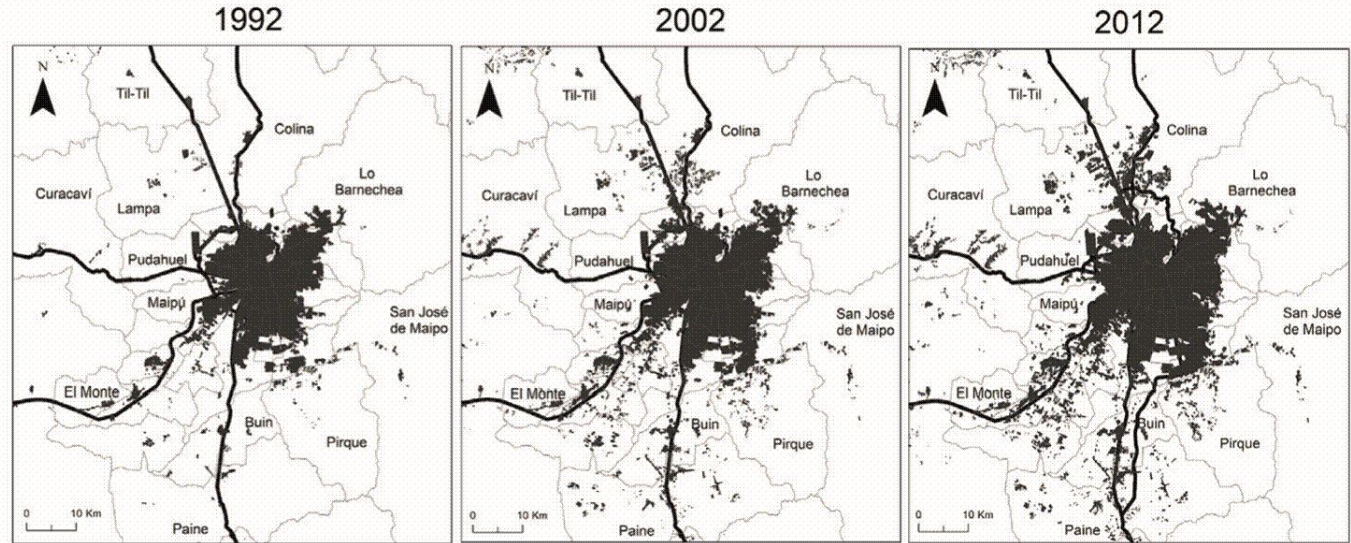
# CC6909 - Ayatori

Implementación eficiente de Connection Scan en  
GTFS con caso de estudio de movilidad en Santiago

---

# S/C/P: Situación, Complicación, Propuesta

Situación:



## Leyenda

División comunal    Mancha Urbana    Autopistas

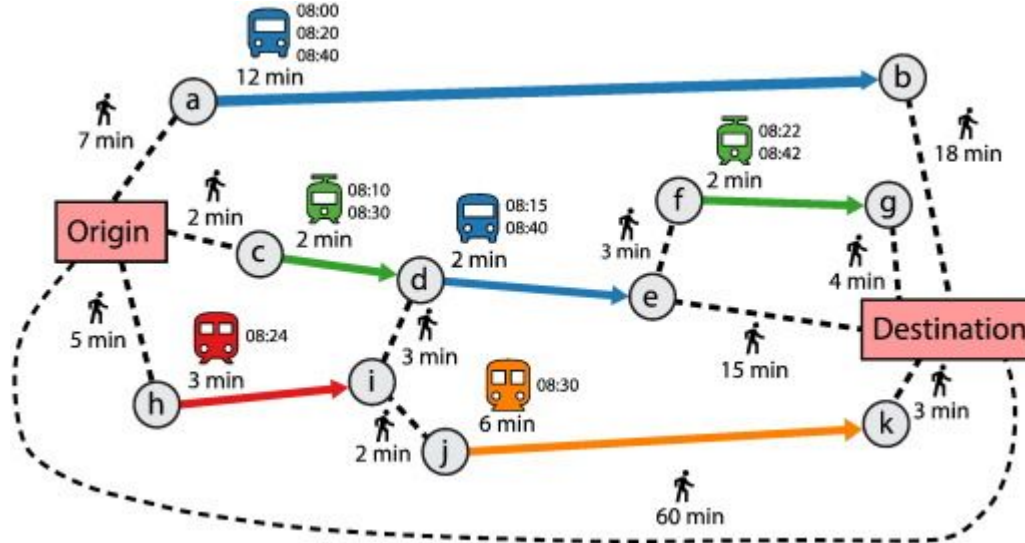
# S/C/P: Situación, Complicación, Propuesta

Complicación:



# S/C/P: Situación, Complicación, Propuesta

Propuesta:



**Red**  
Metropolitana de Movilidad

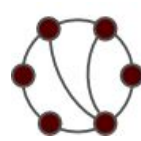
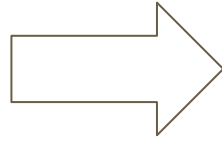
Orden depende del grafo.  $O(V+E)$

# Avances

# Cambio de librerías



NetworkX  
Network Analysis in Python



graph-tool

# Mapeo de Santiago

Santiago - Rutas a pie



# Mapeos por comuna

Maipú - Rutas a pie



Maipú - Mapa completo





# Avances de programación

```
def connection_scan(graph, source, destination, departure_time, date, hour):
    """
    Aplica el algoritmo Connection Scan para buscar rutas de viaje desde la fuente (source) hasta el destino (destination)
    dado un tiempo de partida (departure_time). Se solicita también la fecha del viaje (date) y la hora de inicio (hour).
    Por defecto, se utilizan la fecha y hora actuales del sistema. (Por implementar).
    """

    connections = []
    visited = set()

    def recursive_dfs(node, current_time):
        """
        Realiza una búsqueda en profundidad (DFS) recursiva desde el nodo dado (node) con un tiempo actual (current_time).
        """
        if current_time > departure_time:
            return

        visited.add(node)

        if node == destination:
            connections.append([])
            return

        for neighbor in node.out_neighbors():
            if neighbor not in visited:
                travel_time = graph.ep['time'][graph.edge(node, neighbor)]
                arrival_time = current_time + travel_time
                recursive_dfs(neighbor, arrival_time)

        visited.remove(node)

    recursive_dfs(source, departure_time)
    return connections
```

# Plan actual (próximas semanas)



**Continuar la programación del algoritmo + Testeos**

# Estado final esperado



---

# CC6909 - Ayatori

— ¡Gracias por su atención! —

---