Proyecto Final de Curso: Simmulated annealing para la detección de rutas de contrabando

MA4402-1 Simulación Estocástica, Teoría y Laboratorio

Profesores: Joaquín Fontbona

Auxiliares: Camilo Carvajal Reyes, Arie Wortsman Z., Pablo Zúñiga Rodríguez-Peña.

Integrantes: Andrés Olivares C. Isidora R. Zenteno

2022-12-07

Proyecto Final de Curso:
Simmulated annealing para la detección de rutas de contrabando
MA4402-1 Simulación Estocástica, Teoría y Laboratorio
Profesores: Joaquín Fontbona
Auxiliares: Camilo Carvajal Reyes, Arie Wortsman Z., Pablo Zúñiga Rodríguez-Peña.
Integrantes: Andrés Olivares C. Isidora R. Zenteno

Notas

Introducción:
Motivación
Objetivo
Marco Teórico

Notas El siguiente proyecto tiene como objetivo resolver en el problema de caminos más cortos usando Simmulated Annealing para grafos cualquiera, con la intención de aplicarlo a búsqueda de rutas de contrabando. Una de las inspiraciones es un estudio de las rutas de contrabando de drogas desde México a Estados Unidos, utilizando el algoritmo Dijkstra de caminos más cortos.

Motivación:

Uso práctico: detección de rutas de contrabando.
Anécdota.

Notas

Objetivo:

Resolver el problema de Caminos más cortos usando Simmulated Annealing para grafos no necesariamente regulares (hipótesis del apunte para usar este algoritmo). Comparar los resultados con la solución óptima entregada por Dijkstra.

Objetivo:
Resolver el problema de Caminos más cortos usando Simmulated Annealing para grafos no necesariamente regulares (hipótesis del apunte para usar este algoritmo). Comparar los resultados con la solución óptima entregada por Dijkstra.

Notas

Marco Teórico:

Simmulated Annealing para rutas más cortas

1. Shortest path based simulated annealing algorithm for dynamic facility layout problem under dynamic business environment.
2. Simulated Annealing Approach for the Constrained Shortest Path with Fuzzy Arc and Node Weights.

Dijkstra aplicado a rutas de contrabando

1. Mexico's drug networks: Modeling the smuggling routes towards the United States.

Marco Teórico:
Simmulated Annealing para rutas más cortas
1. Shortest path based simulated annealing algorithm for dynamic facility layout problem under dynamic business environment.
2. Simulated Annealing Approach for the Constrained Shortest Path with Fuzzy Arc and Node Weights.
Dijkstra aplicado a rutas de contrabando
1. Mexico's drug networks: Modeling the smuggling routes towards the United States.

Notas

Implementación: Primera Aproximación

Algoritmo

Resultados

Notas Para la implementación, se definió el grafo como un mapa, utilizando las provincias como nodos, habiendo una arista si es que las dos provincias que representa sus nodos son aledañas. Existía una definición para los pesos que se aplicaría para el problema particular, pero al no estar los datos disponibles, se prefiere simular.

Algoritmo:

Nodos: Provincias. Aristas: Provincias Aledañas

Ruta mínima desde un nodo v_0 hasta un nodo v_n tal que pase por una provincia de cada región que hay entre estos dos. Los vecinos se definen como las rutas que difieren de la original en una provincia.

2022-12-07

Algoritmo:
Nodos: Provincias. Aristas: Provincias Aledañas
Ruta mínima desde un nodo v_0 hasta un nodo v_n tal que pase por una provincia de cada región que hay entre estos dos. Los vecinos se definen como las rutas que difieren de la original en una provincia.

Notas

Problema a Resolver

¿Cómo conectar los dos puntos verdes pasando una vez por región?

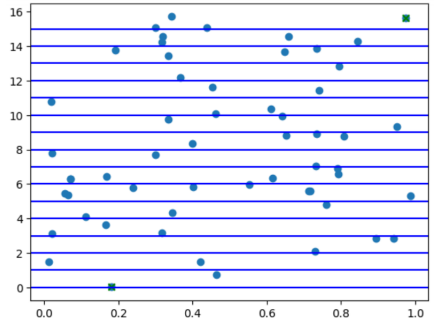


Figure: Mapa aleatorio de Regiones con Provincias

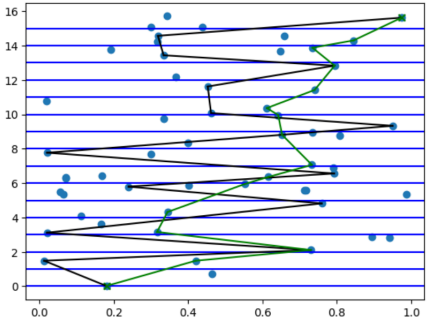
2022-12-07

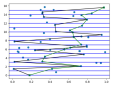
Notas

Problema a Resolver
¿Cómo conectar los dos puntos verdes pasando una vez por región?
Figure: Mapa aleatorio de Regiones con Provincias

Implementación

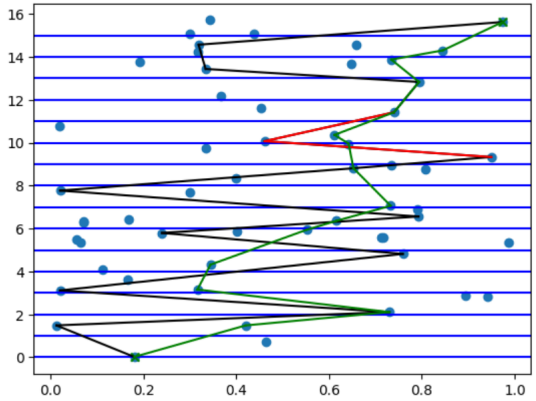
Comenzamos con una ruta inicial.

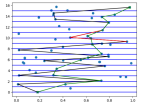


Implementación
Comenzamos con una ruta inicial.


Implementación

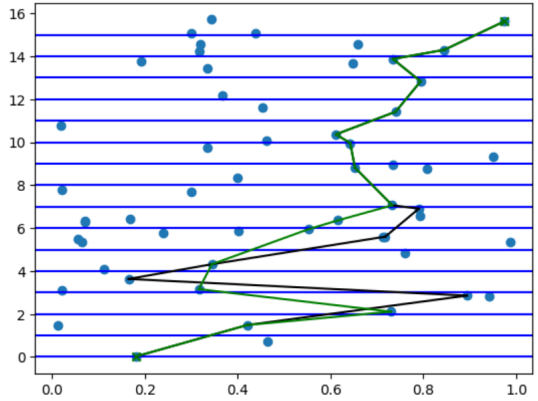
Cambiamos una provincia al azar de una región al azar.



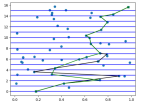
Implementación
Cambiamos una provincia al azar de una región al azar.


Implementación

Y así sucesivamente:



Implementación
Y así sucesivamente:



Resultados

Y llegamos a la solución entregado por Dijkstra

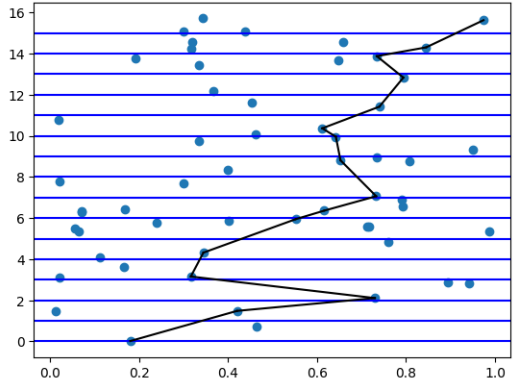


Figure: Ruta solución Entregada por Dijkstra

Resultados
Y llegamos a la solución entregado por Dijkstra
Figure: Ruta solución Entregada por Dijkstra

Resultados

A partir de 100 mapas distintos, resolvemos el mismo problema para distintos parámetros:

β	C=1	C=5	C=10	C=100
Lineal	0.38	0.4	0.39	0.029
Cuadrático	0.32	0.33	0.31	0.36
Exponencial	0.33	0.32	0.34	0.34

Tabla de comparación de precisión por los modelos de Simulated Annealing

Resultados

A partir de 100 mapas distintos, resolvemos el mismo problema para distintos parámetros:

β	C=1	C=5	C=10	C=100
Lineal	0.38	0.4	0.39	0.029
Cuadrático	0.32	0.33	0.31	0.36
Exponencial	0.33	0.32	0.34	0.34

Tabla de comparación de precisión por los modelos de Simulated Annealing

Resultados

A partir de 100 mapas distintos, resolvemos el mismo problema para distintos parámetros. Considerar que Dijkstra tarda en promedio 0.00037 segundos:

β	C=1	C=5	C=10	C=100
Lineal	0.0277	0.0278	0.0276	0.0259
Cuadrático	0.0283	0.0274	0.0278	0.0275
Exponencial	0.0299	0.0315	0.0296	0.030

Tabla de comparación de tiempos de ejecución (en segundos) por los modelos de Simulated Annealing

Resultados

A partir de 100 mapas distintos, resolvemos el mismo problema para distintos parámetros. Considerar que Dijkstra tarda en promedio 0.00037 segundos:

β	C=1	C=5	C=10	C=100
Lineal	0.0277	0.0278	0.0276	0.0259
Cuadrático	0.0283	0.0274	0.0278	0.0275
Exponencial	0.0299	0.0315	0.0296	0.030

Tabla de comparación de tiempos de ejecución (en segundos) por los modelos de Simulated Annealing

Implementación: Segunda Aproximación

Algoritmo

Resultados

Notas La primera aproximación, se hizo una versión de Chile como un grafo de 16 regiones (niveles) con 4 provincias cada una, en el que todas las provincias son aledañas con aquellas de las regiones inmediatamente al norte e inmediatamente al sur, pero no con el resto. Para este caso, las posiciones de las provincias son aleatorias en rango de [0,1] de este a oeste y de [i,i+1] para $i = 0 \ldots 15$ de norte a sur. Los pesos de las aristas se definen como la distancia euclideana. Luego, un vecino de una ruta se define cuando estas difieren en una ciudad. Con esto, se aplica Simmulated Annealing para distintos β y C y se compara con la ruta obtenida por Dijkstra. El peso total de la ruta para β_n cuadrático, con C=5 y 100 fue el mismo que el algoritmo determinista, pero el tiempo de ejecución es mayor.

Algoritmo:

Nodos: Provincias.
Aristas: Provincias Aledañas.
Ruta mínima desde un nodo v_0 hasta un nodo v_n. Los vecinos se definen como las rutas que difieren de la original en una sección de a lo más 2 nodos y que luego continúan con la ruta hasta llegar al nodo final.

Algoritmo:
Nodos: Provincias.
Aristas: Provincias Aledañas.
Ruta mínima desde un nodo v_0 hasta un nodo v_n. Los vecinos se definen como las rutas que difieren de la original en una sección de a lo más 2 nodos y que luego continúan con la ruta hasta llegar al nodo final.

Notas

Implementación

Comenzamos con grafos simples:

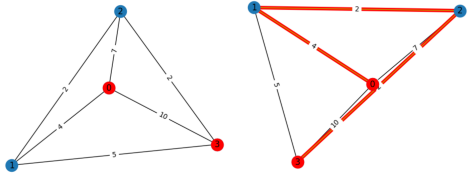


Figure: Ejemplo grafo con 4 vértices

Implementación
Comenzamos con grafos simples:

Figure: Ejemplo grafo con 4 vértices

Notas

Implementación

Comenzamos con grafos simples:

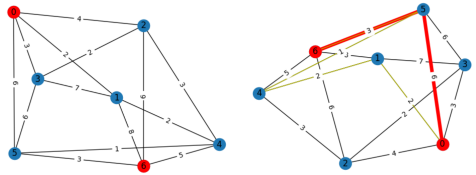


Figure: Ejemplo grafo de 7 vértices

Implementación
Comenzamos con grafos simples:
Figure: Ejemplo grafo de 7 vértices

Notas

Resultados

A partir de un mismo grafo regular de 50 nodos, y 50 caminos distintos por buscar, resolvemos el mismo problema para distintos parámetros:

β	C=1	C=5	C=10	C=100
Lineal	0.433	0.433	0.466	0.433
Cuadrático	0.466	0.433	0.466	0.433
Exponencial	0.433	0.433	0.533	0.433

Tabla de comparación de precisión por los modelos de Simulated Annealing

Resultados

A partir de un mismo grafo regular de 50 nodos, y 50 caminos distintos por buscar, resolvemos el mismo problema para distintos parámetros:

β	C=1	C=5	C=10	C=100
Lineal	0.433	0.433	0.466	0.433
Cuadrático	0.466	0.433	0.466	0.433
Exponencial	0.433	0.433	0.533	0.433

Tabla de comparación de precisión por los modelos de Simulated Annealing

Resultados

A partir de un mismo grafo regular de 50 nodos, y 50 caminos distintos por buscar, resolvemos el mismo problema para distintos parámetros. Considerar que Dijkstra tarda en promedio 0.0003 segundos:

β	C=1	C=5	C=10	C=100
Lineal	0.858	0.842	0.869	0.916
Cuadrático	0.887	0.881	0.881	0.844
Exponencial	0.888	0.899	1.019	0.909

Tabla de comparación de tiempos de ejecución (en segundos) por los modelos de Simulated Annealing

Resultados

A partir de un mismo grafo regular de 50 nodos, y 50 caminos distintos por buscar, resolvemos el mismo problema para distintos parámetros. Considerar que Dijkstra tarda en promedio 0.0003 segundos:

β	C=1	C=5	C=10	C=100
Lineal	0.858	0.842	0.869	0.916
Cuadrático	0.887	0.881	0.881	0.844
Exponencial	0.888	0.899	1.019	0.909

Tabla de comparación de tiempos de ejecución (en segundos) por los modelos de Simulated Annealing

Conclusiones

Notas La segunda aproximación (que luego serviría para aplicarlo como mapa) sería usando un grafo cualquiera. Los primeros ejemplos versiones simples, como un grafo completo y grafos de pocos vértices y aristas. Luego se implimenta en un grafo proveniente de una base de datos (también debería ser parte de los anexos).

Resultados

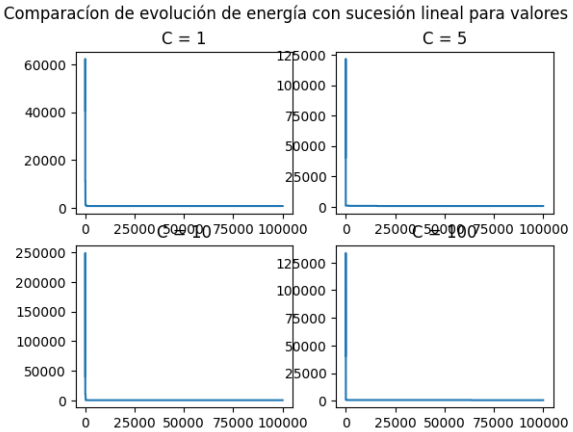
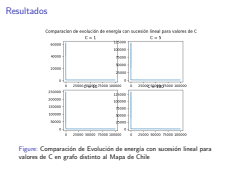


Figure: Comparación de Evolución de energía con sucesión lineal para valores de C en grafo distinto al Mapa de Chile



Notas

Resultados

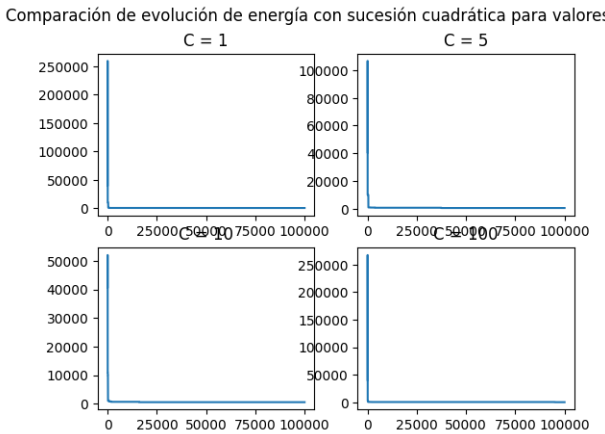
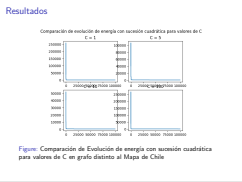


Figure: Comparación de Evolución de energía con sucesión cuadrática para valores de C en grafo distinto al Mapa de Chile



Notas

Resultados

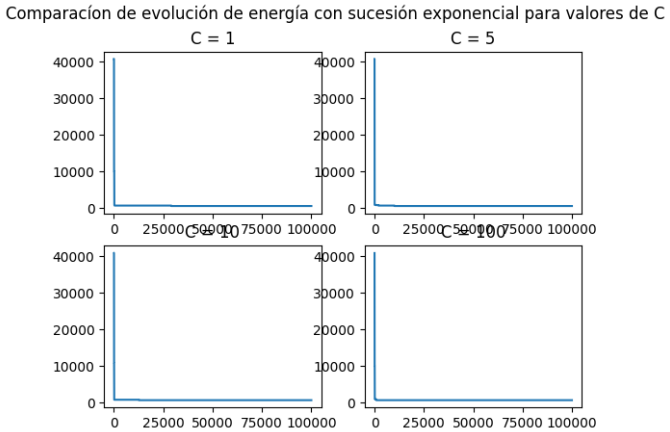
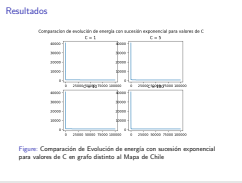


Figure: Comparación de Evolución de energía con sucesión exponencial para valores de C en grafo distinto al Mapa de Chile



Notas