

UNIVERSIDAD PANAMERICANA

CAMPUS BONATERRRA

OPTIMIZACIÓN Y METAHEURÍSTICAS

PROYECTO FINAL

DAVID GAMALIEL ARCOS BRAVO - 0223826

JESUS ERNESTO GALVAN VALDEZ - 0222460

LUIS FERNANDO CARO REYNA - 0227782

CARLOS DANIEL CORRAL CORNEJO - 0226934

Problema 1: Asignación de terminales

Se tienen n estaciones de trabajo y m hubs (terminales). El costo de asignar una estación de trabajo ( i ) a un hub ( j ) es . Cada estación de trabajo consume o demanda unidades de la capacidad de un hub. La capacidad de un hub j es . El objetivo es encontrar la asignación de mínimo costo para formar una red de conexiones entre estaciones de trabajo y terminales. Cada estación de trabajo debe ser asignada exactamente a una terminal (hub). El costo , se calcula usando las coordenadas de las terminales (x, y) y las estaciones de trabajo (x, y,) de la siguiente forma:

**Primer Recocido simulado**: Para obtener una solución inicial, llamamos la función “solución inicial”, esta nos da el primer estado viable o primera solución que cumple las condiciones dadas, hacemos el recocido simulado, si es mejor, se toma la solución, en caso contrario, modificamos la temperatura para obtener una mejor solución.

Modificamos el vector que tiene nuestras estaciones de trabajos, checamos cuanto nos da la suma de cada estación y separan en dos grupos, las que están en el rango y las que no, las que se pasan del rango, de forma aleatoria se les quita elementos, estos elementos removidos son distribuidos en las estaciones de trabajo que tienen cupo hasta llegar a una solución dentro del rango

**Segundo Recocido simulado** para optimizar la solución anterior, llamamos la función “optimize”. La selección de posibles soluciones hace uso de la función generadora de vecinos usada en la búsqueda tabú, con una probabilidad menor de generación para reducir la complejidad, y de esta se toma un vector al que le realizamos muchas operaciones creando así vectores vecinos a este donde se comparan y se toma el mejor valor, si es mejor o la temperatura lo permite.

**Búsqueda Tabú** es aplicado al resultado anterior para llegar a nuestra solución

A diferencia de la tabla finita expuesta en clase, nosotros decidimos hacer una tabla dinámica que fuimos ajustando dependiendo el número de elementos que se tiene, la tabla crece de forma logarítmica, ya que, si fuera de manera lineal, perderíamos mucha memoria, por esto lo hacemos de forma logarítmica, cada vez que llegamos a una potencia de 2, incrementamos el tamaño de la tabla tabu, esto nos ayuda ya que entre más grande sea la tabla tabu, tendremos resultados mejores.

Si usáramos el tabú directo, nos arrogaría un mínimo local y no llegaríamos a la mejor solución.

Generamos 5 instancias con 30 datos cada una, estos fueron los resultados que arrojo cada instancia.

Instancia 1

|  |  |
| --- | --- |
| Numero de datos: | 30 |
| Desviación estándar: | 27.70497348050627 |
| Promedio: | 702.3666666666667 |
| Punto máximo f(x): | 770.0 |
| Punto mínimo f(x): | 671.0 |
| índice de mejor resultado: | 26 |

Instancia 2

|  |  |
| --- | --- |
| Numero de datos: | 30 |
| Desviación estándar: | 38.224497671286954 |
| Promedio: | 1037.2333333333333 |
| Punto máximo f(x): | 1148.0 |
| Punto mínimo f(x): | 987.0 |
| índice de mejor resultado: | 15 |

Instancia 3

|  |  |
| --- | --- |
| Numero de datos: | 30 |
| Desviación estándar: | 38.31964973163971 |
| Promedio: | 809.2666666666667 |
| Punto máximo f(x): | 912.0 |
| Punto mínimo f(x): | 754.0 |
| índice de mejor resultado: | 18 |

Instancia 4

|  |  |
| --- | --- |
| Numero de datos: | 30 |
| Desviación estándar: | 18.53707995703027 |
| Promedio: | 861.1 |
| Punto máximo f(x): | 908.0 |
| Punto mínimo f(x): | 830.0 |
| índice de mejor resultado: | 0 |

Instancia 5

|  |  |
| --- | --- |
| Numero de datos: | 30 |
| Desviación estándar: | 29.11631844859511 |
| Promedio: | 800.8 |
| Punto máximo f(x): | 877.0 |
| Punto mínimo f(x): | 756.0 |
| índice de mejor resultado: | 26 |

Al ver los datos podemos concluir que de las mejores instancias fueron tanto la primera como la última, ya que esas dos nos dieron los mejores índices de resultados. Con la única diferencia que la última instancia tuvimos un mejor promedio.

Resultados graficados por instancias

Gráfico

Descripción generada automáticamenteGráfico

Descripción generada automáticamenteINSTANCIA 1

INSTANCIA 2

Gráfico, Gráfico de líneas

Descripción generada automáticamenteDiagrama, Gráfico radial

Descripción generada automáticamente

Gráfico, Gráfico de líneas

Descripción generada automáticamenteINSTANCIA 3

Diagrama

Descripción generada automáticamente

INSTANCIA 4

Gráfico

Descripción generada automáticamenteDiagrama

Descripción generada automáticamente

Gráfico

Descripción generada automáticamenteDiagrama

Descripción generada automáticamenteINSTANCIA 5

ÁNALISIS DEL ALGORITMO

Gráfico, Gráfico de líneas

Descripción generada automáticamente

Gráfico, Histograma

Descripción generada automáticamente

CONCLUSIONES

Podemos concluir que los algoritmos vistos tienen sus mejores áreas de desempeño tanto como sus puntos débiles, el punto es encontrar el punto medio de eso y lo que me piden. Podemos apoyarnos de diferentes códigos para cubrir esos puntos débiles y así poder lograr una mejor solución, tal como lo hicimos nosotros, primero usamos un recocido simulado para obtener un primer estado viable que cumpliera las condiciones dadas, para así después aplicarle de nuevo un recocido simulado para optimizar ese estado y al final aplicamos la búsqueda tabú para crear una tabla de puntos ya visitados y evitarlos para buscar la mejor solución.

Fue a través de pruebas de comparar los resultados generados por las otras versiones del algoritmo, como poner primero la búsqueda tabu y después el recocido, o manejando el numero de iteraciones, fue como llegamos a una distribución optima del algoritmo, con un numero de iteraciones que nos ayuda a reducir de manera notable la desviación estándar de nuestras soluciones, reduciéndola de aproximadamente 100 a una desviación de 25 – 30.

Es por ello que utilizar un recocido simulado para una solución inicial nos ayuda a escapar de mínimos locales y llegar a mejores soluciones.