Ejercicio 5: Desde un cierto conjunto grande de ciudades del interior de una provincia, se desean transportar cereales hasta alguno de los 3 puertos pertenecientes al litoral de la provincia.

Se pretende efectuar el transporte total con mínimo costo sabiendo que el flete es más caro cuanto más distancia tiene que recorrer. Dé un algoritmo que resuelva este problema, devolviendo para cada ciudad el camino que debería recorrer hacia el puerto de menor costo.

Solo en esta ocasión pedimos un pseudocódigo del algoritmo que resuelve el problema y un seguimiento completo del mismo con un grafo de ejemplo.

Pseudocodigo: Metodo principal y funciones: Solucion, Factible y Seleccionar. Se pueden incluir otras funciones y métodos que ayuden a resolver el problema.

Estructura general de greedy

Funcion voraz(conjunto<candidatos> C)

{ // Inicialmente, C contiene todos los candidatos

conjunto<candidatos> S; // Solución inicial Vacía

while (!C.vacio() && !solucion(S)) {

x = seleccionar(C); // determina el mejor candidatos del conjunto a seleccionar

C.borrar(x);

if (factible(S,x))

S.agregar(x);

}

if solucion(S)

return S

else

return “No\_hay\_solucion”;

}

Elementos:

• Conjunto de candidatos a seleccionar C.

• Conjunto de candidatos seleccionados S.

• Función Solución: determina si los candidatos seleccionados

han alcanzado una solución.

• Función de Factible: determina si es posible llegar a una

solución con el candidato seleccionado.

• Función Seleccionar: determina el mejor candidato del

conjunto a seleccionar en ese momento.

• Siempre está presente una Función Objetivo: da el valor de la

solución alcanzada que queremos sea óptima (maximice o

minimice algo).

Dijkstra

Procedura Dijkstra;

Begin

S := { 1 };

For i := 2 to n do

D[i] := C[1,i];

For i := 1 to n-1 do begin

Elige un vertice w en V-S tal que D[w] sea un minimo;

agrega w a S;

For cada vertice v en V-S do

D[v] := min(D[v],D[w] + C[w,v]) -----🡪 P[v] := w

End

End;