

Optimización de la red eléctrica de granja eólicas

Manuel Valenzuela-Rendón
valenzuela@itesm.mx

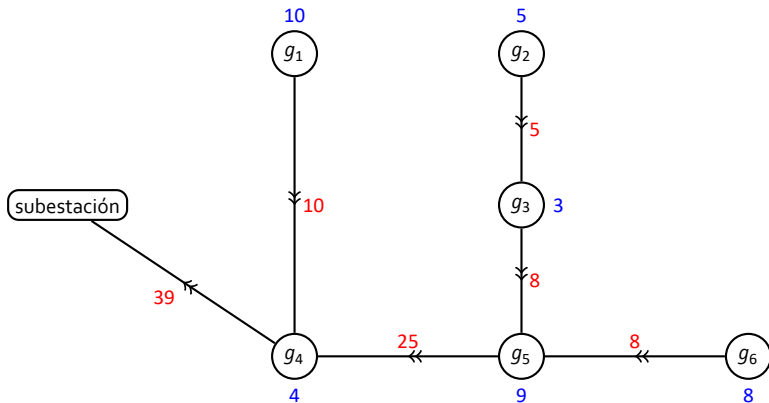
Tecnológico de Monterrey

27 de marzo de 2017

- En una granja eólica se tienen aerogeneradores que transforma energía del viento en energía eléctrica.
- Esta energía debe concentrarse en una subestación para su distribución y consumo fuera de la granja.
- Cada generador es capaz de producir una cierta cantidad de energía que puede especificarse en términos de la potencia eléctrica que se produce.
- Los generadores se conectan mediante una red eléctrica que debe tener la capacidad de transmitir la potencia producida por los generadores.

Caso 1

Una posible solución

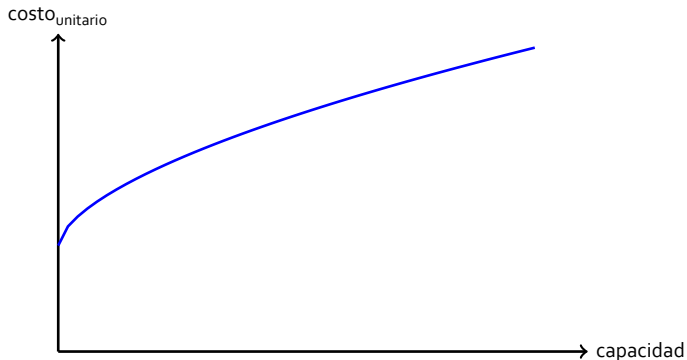


La cantidad en azul cerca de un generador es su producción. Las cantidades en rojo son las capacidades necesarias en los conductores.

El costo de la red eléctrica depende del costo de los conductores. El costo C de un tramo de conductor es su longitud d multiplicada por el costo unitario en función de su capacidad:

$$C = d \times \text{costo}_{\text{unitario}}(\text{capacidad}) \quad (1)$$

donde $\text{costo}_{\text{unitario}}(\text{capacidad})$ es una función cóncava como la mostrada en la figura.

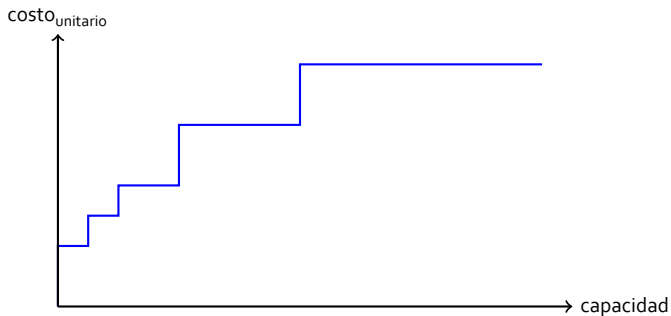


Definición del problema

Dado un conjunto de generadores con capacidades de producción conocidas, y dadas las coordenadas de estos generadores y de la subestación a la que deben conectarse, se desea diseñar una red eléctrica que conecte con tramos rectos entre generadores y la subestación, y que tenga mínimo costo. La función de costo unitario de los conductores eléctricos está dada.

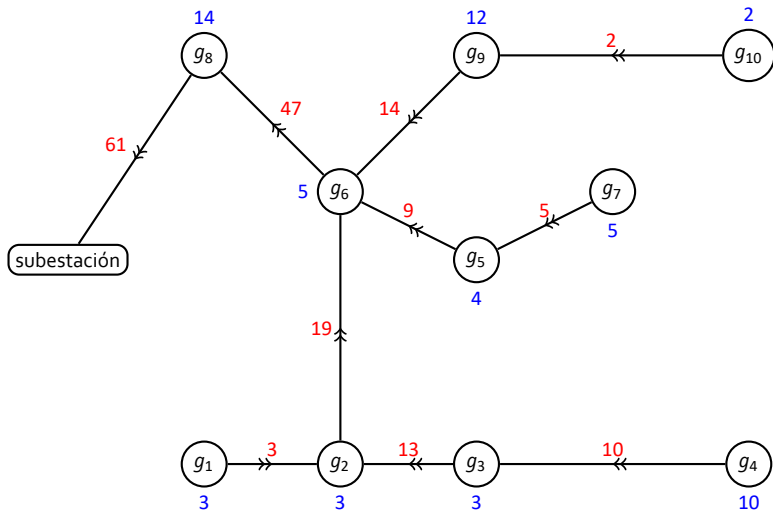
La red eléctrica estará definida por los arcos dirigidos entre generadores y subestación, y por las capacidades de todos los conductores. La red no puede tener ciclos, es decir, cada generador puede tener solamente una salida, y la subestación no puede tener salidas.

$$\text{costo}_{\text{unitario}} = \begin{cases} 10 & \text{si } 0 < \text{capacidad} \leq 5 \\ 15 & \text{si } 5 < \text{capacidad} \leq 10 \\ 20 & \text{si } 10 < \text{capacidad} \leq 20 \\ 30 & \text{si } 20 < \text{capacidad} \leq 40 \\ 40 & \text{si } 40 < \text{capacidad} \leq 80 \end{cases} \quad (2)$$



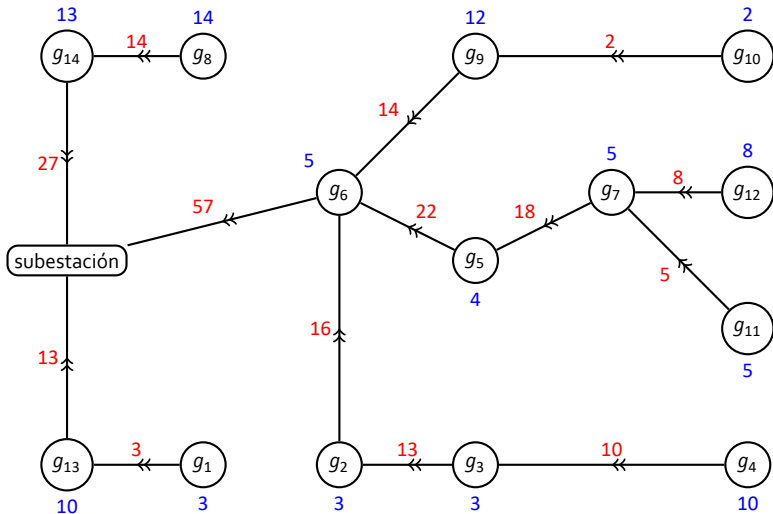
Caso 2

Una posible solución



Caso 3

Una posible solución



- Los nodos se numeran a partir de 0, donde 0 es la subestación.
- El arreglo **hijo** contiene el hijo del nodo $i - 1$. En este arreglo el hijo se numera a partir de 0, donde 0 indica que no tiene hijos (todos los nodos, excepto la subestación, deben tener un hijo) 1 se refiere a la subestación (nodo 0). Por ejemplo, en el caso 1, **hijo**(1) = 0 es el hijo de la subestación (nodo 0) que no tiene hijo, **hijo**(2) = 5 indica que el hijo del nodo 1 es 4, etc.
- El arreglo **prod** contiene la producción del nodo $i - 1$.
- Cada renglón del arreglo **C** contiene las coordenadas del nodo $i - 1$.
- **D** es la matriz de distancias, donde **D**(i, j) es la distancia entre los nodos $i - 1$ y $j - 1$.
- **A** es la matriz de incidencias, donde **A**(i, j) = 1 si el nodo $i - 1$ es el padre del nodo $j - 1$.
- **N** es el número de nodos contando a la subestación.

- $hijo(i)$: hijo del nodo $i - 1$
- $prod(i)$: producción del nodo $i - 1$
- $C(i, :)$: coordenadas del nodo $i - 1$
- $D(i, j)$: distancia entre los nodos $i - 1$ y $j - 1$
- $A(i, j)$: incidencia del nodo $i - 1$ en el nodo $j - 1$
- N número total de nodos

Datos caso 1

```
% Encabezado
str = 'CASO_1';

% hijo del nodo i-1
% nodo: 0 1 2 3 4 5 6
hijo = [0 5 4 6 1 5 6];

% Producción del nodo i-1
% nodo: 0 1 2 3 4 5 6
prod = [0 10 5 3 4 9 8];

% Coordenadas del nodo i-1
C = [0 0; 3 2; 6 2; 6 0; 3 -2; 6 -2; 9 -2];

N = length(hijo);
```

Conversión de padre a matriz de incidencias

```
A = zeros(N);  
for i=1:N  
    if hijo(i)~=0  
        A(i,hijo(i)) = 1;  
    end  
end  
% Matriz de incidencias  
display(A)
```

Cálculo de corriente en cada arista del árbol

```
Aux = A;
envia = zeros(size(hijo));           % potencia que envía cada nodo a su padre
while 1
    hojas = find(sum(Aux,1)==0);      % si el nodo raíz es hoja,
    if isequal(hojas,1)              % salir
        break
    end
    envia(hojas) = envia(hojas) + prod(hojas);
    for i=1:length(hojas)
        envia(hijo(hojas(i))) = envia(hijo(hojas(i))) + envia(hojas(i));
        Aux(hojas(i),:) = zeros(size(hijo));
    end
    Aux(1,hojas) = 1;
end
display(envia)
```

Cálculo de costos

```
% Función de costo unitario para caso 1 (debe cambiarse para los casos 2 y 3)
costoUnitario = @(capacidad) 2 + capacidad.^0.6;
```

```
% La matriz de distancias contiene la distancia del nodo i-1 al nodo j-i
```

```
D = zeros(N);
```

```
for i=1:N
```

```
    for j=i+1:N
```

```
        D(i,j) = norm(C(i,:)-C(j,:));
```

```
        D(j,i) = D(i,j);
```

```
    end
```

```
end
```

```
% Matriz de distancias
```

```
display(D)
```

```
costo = zeros(1,N);
```

```
for i=2:N
```

```
    costo(i) = costoUnitario(envia(i))*D(i,hijo(i));
```

```
end
```

```
% Costo por arista
```

```
display(costo)
```

```
fprintf('El costo total es: %f\n', sum(costo))
```

Cálculo de costos (en forma vectorial)

```
% Esto corre más rápido porque utiliza operaciones matriciales

% Función de costo unitario para caso 1 (debe cambiarse para el caso 2 y 3)
costoUnitario = @(capacidad) 2 + capacidad.^0.6;

% La matriz de distancias contiene la distancia del nodo i-1 al nodo j-i
D = zeros(N);
for i=1:N
    for j=i+1:N
        D(i,j) = norm(C(i,:)-C(j,:));
        D(j,i) = D(i,j);
    end
end
% Matriz de distancias
display(D)

costo = costoUnitario(envia).*[0;diag(D(2:N,hiho(2:N)))']';

% Costo por arista
display(costo)
fprintf('El costo total es: %f\n', sum(costo))
```

Gráfica de red con producción y cantidad que se envía

```
delta = 0.1; % desplazamiento de las etiquetas
plot(C(1,1),C(1,2),'sr',C(2:end,1),C(2:end,2),'ob')
xlim([min(C(:,1))-0.5 max(C(:,1)+0.5)]);
ylim([min(C(:,2))-0.5 max(C(:,2)+0.5)]);
hold on
for p=2:N
    h = hijo(p);
    text(C(p,1)+delta,C(p,2)+delta,sprintf('%d:%d/%d',p-1,prod(p),envia(p)))
    plot([C(p,1) C(h,1)],[C(p,2) C(h,2)],'-b')
end
text(C(1,1)+delta,C(1,2)+delta,'0')
hold off
title(sprintf('%s: Red eléctrica (nodo: producción/potencia)',str))
xlabel('x')
ylabel('y')
```