Algoritmo de Ford y Fulkerson

```
Establecer la fuente s y el sumidero t en el grafo G
para todas las aristas (i, j)∈E hacer
   Residuo[i, j] ← cij - fij + fji
Flujo_Max ← 0
para i ← 1 hasta n hacer
   Flujo_Max ← Flujo_Max + fsi
Camino ← TRUE
mientras Camino = TRUE hacer
   { para i ← 1 hasta n hacer
       { Predecesor[i] ← 0
         Marca[i] ← FALSE
       }
     Marca[s] ← TRUE
     Predecesor[s] ← s
     Incremento[s] ← ∞
     mientras (∃x tal que Marca[x] = TRUE) y (Predecesor[t] = 0)
     hacer
        { Elegir un vértice i tal que Marca[i] = TRUE
          Marca[i] ← FALSE
          para j ← 1 hasta n hacer
             { si (Predecesor[j] = 0) y (Residuo[i, j] > 0)
               entonces
                  { Predecesor[j] ← i
                    Incremento[j] ← min(Incremento[i],
                                        Residuo[i, j])
                    Marca[j] ← TRUE
                  }
             }
        }
     si Predecesor[t] > 0 entonces
        { Δ ← Incremento[t]
          i ← t
          mientras i ≠ s hacer
             { Residuo[Predecesor[i], i] ←
                                   Residuo[Predecesor[i], i] - Δ
               Residuo[i, Predecesor[i]] ←
                                   Residuo[i, Predecesor[i]] + Δ
               i ← Predecesor[i]
             }
          Flujo_Max ← Flujo_Max + Δ
        }
     en otro caso
        Camino ← FALSE
   }
para i ← 1 hasta n hacer
   { para j ← 1 hasta n hacer
        fij ← max(cij - Residuo[i, j], 0)
   }
```

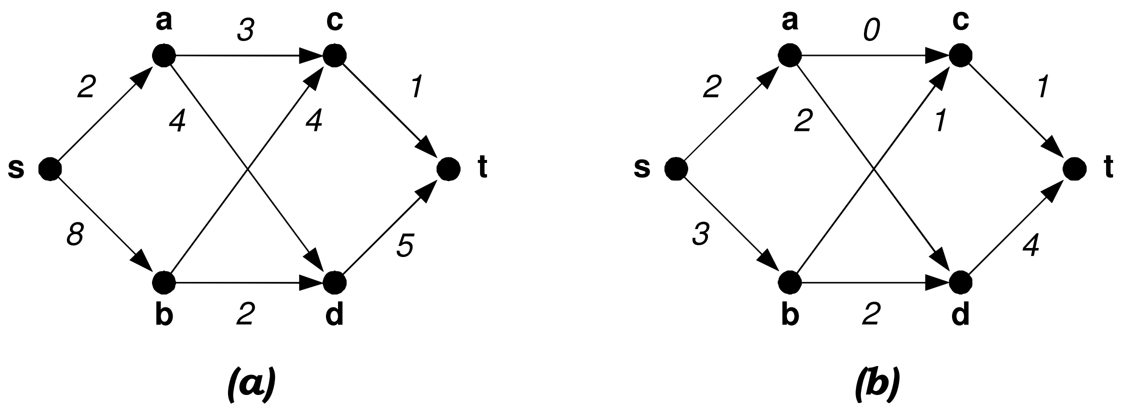

Figura 1: (a) Grafo inicial; (b) Patrón de flujo final.

Iteración	Camino incremental	Cuello de botella	Flujo
0	—	—	*0*
1	$\mathbf{s} \xrightarrow{2} \mathbf{a} \xrightarrow{3} \mathbf{c} \xrightarrow{1} \mathbf{t}$	*1*	*0+1=1*
2	$\mathbf{s} \xrightarrow{1} \mathbf{a} \xrightarrow{4} \mathbf{d} \xrightarrow{5} \mathbf{t}$	*1*	*1+1=2*
3	$\mathbf{s} \xrightarrow{8} \mathbf{b} \xrightarrow{2} \mathbf{d} \xrightarrow{4} \mathbf{t}$	*2*	*2+2=4*
4	$\mathbf{s} \xrightarrow{6} \mathbf{b} \xrightarrow{4} \mathbf{c} \xrightarrow{1} \mathbf{a} \xrightarrow{3} \mathbf{d} \xrightarrow{2} \mathbf{t}$	*1*	*4+1=5*
5	*No existe ningún camino desde s a t*	—	***Flujo máximo=5***

Tabla 1: Traza del grafo anterior.

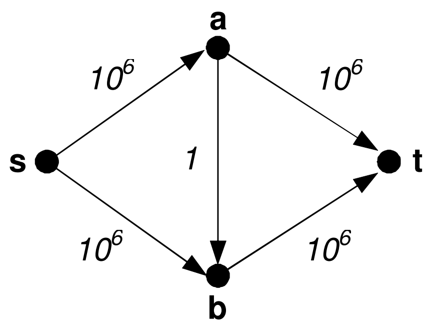

Figura 2: Caso patológico.

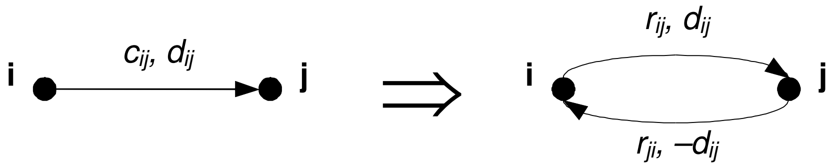

Figura 3: Ejemplo de residuo.

Algoritmo de Busacker y Gowen

```
Establecer la fuente s y el sumidero t en el grafo G y el flujo
objetivo θ
para todas las aristas (i, j)∈E hacer
   { Residuo[i, j] ← cij
     Costo[i, j] ← dij
   }
para todas las aristas (i, j)∉E hacer
   { Residuo[i, j] ← 0
     Costo[i, j] ← ∞
   }
Flujo_Actual ← 0
Camino ← TRUE
mientras (Flujo_Actual < θ) y (Camino = TRUE) hacer
   { Encontrar un camino mínimo P desde s a t
     si no existe camino P entonces
        Camino ← FALSE
     si Camino = TRUE entonces
        { Δ ← Capacidad mínima del camino P
          si Flujo_Actual + Δ > θ entonces
             Δ ← θ - Flujo_Actual
          para cada arista (i, j)∈P hacer
             { Residuo[i, j] ← Residuo[i, j] - Δ
               Residuo[j, i] ← Residuo[j, i] + Δ
               Costo[j, i] ← -Costo[i, j]
               si Residuo[i, j] = 0 entonces
                  Costo[i, j] ← ∞
             }
          Flujo_Actual ← Flujo_Actual + Δ
        }
   }
Costo_Total ← 0
para i ← 1 hasta n hacer
   { para j ← 1 hasta n hacer
        { fij ← max(cij - Residuo[i, j], 0)
          Costo_Total ← Costo_Total + fij * Costo[i, j]
        }
   }
```

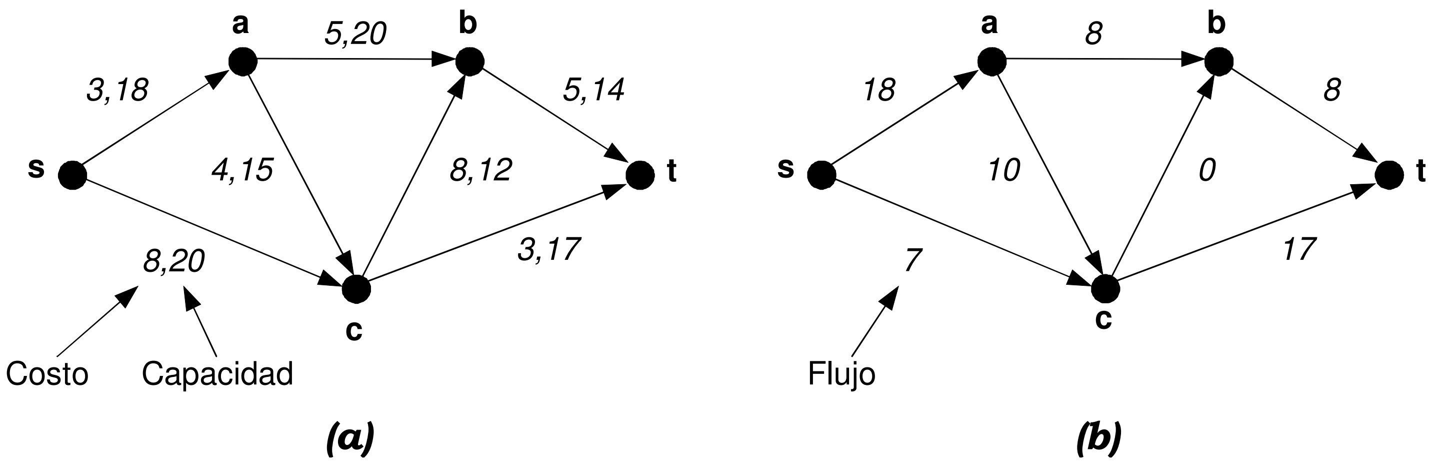

Figura 4: (a) Grafo inicial; (b) Patrón de flujo final.

Iteración	*Camino P*	$Costo_P=\Sigma d_{ij}$	$\Delta_P=min\{c_{ij}\}$	*Flujo actual*
0	$\mathbf{i}\xrightarrow{d_{ij},c_{ij}}\mathbf{j}$	—	—	*0*
1	$\mathbf{s}\xrightarrow{3,18}\mathbf{a}\xrightarrow{4,15}\mathbf{c}\xrightarrow{3,17}\mathbf{t}$	*10*	*15*	*0+15=15*
2	$\mathbf{s}\xrightarrow{8,20}\mathbf{c}\xrightarrow{3,2}\mathbf{t}$	*11*	*2*	*15+2=17*
3	$\mathbf{s}\xrightarrow{3,3}\mathbf{a}\xrightarrow{5,20}\mathbf{b}\xrightarrow{5,14}\mathbf{t}$	*13*	*3*	*17+3=20*
4	$\mathbf{s}\xrightarrow{8,18}\mathbf{c}\xrightarrow{-4,15}\mathbf{a}\xrightarrow{5,17}\mathbf{b}\xrightarrow{5,11}\mathbf{t}$	*14*	*11*	*20+5=**25**=θ*

Tabla 2: Traza del algoritmo sobre el grafo de la Figura 4 (a).