

---

**Algorithmus 1** Berechnung eines Guten-FAA

---

```
1: procedure GFAA( $G, f_{aus}, \{a_1, a_2, a_3\}$ )
2:   if FAA( $G, \{a_1, a_2, a_3\}$ )  $\neq$  None then
3:      $\{d_1, d_2\} \leftarrow$  demands for  $\mathcal{N}_G$ 
4:     initialize  $\mathcal{N}_G$ 
5:      $\varphi = (\varphi_1, \varphi_2) \leftarrow$  two-flow( $\mathcal{N}_G$ ) ▷ Netzwerk ??
6:     if  $\varphi \neq$  None then ▷ Fluss muss nicht ganzzahlig sein
7:       if  $\varphi$  is non-integer then
8:          $\varphi_z \leftarrow$  FAA-flow( $\varphi_2$ )
9:          $\mathcal{N} \leftarrow \mathcal{N}_G \setminus \{\text{edges used by } \varphi_z\}$ 
10:         $\tilde{\varphi} \leftarrow$  int-one-flow( $\mathcal{N}$ ) ▷ Ecken- und Schnyder-Fluss zu  $\varphi_z$ 
11:        if  $|\tilde{\varphi}| = d_1 + d_2 - |\varphi_z|$  then
12:           $\phi \leftarrow$  FAA( $\varphi_z$ )
13:          return  $\phi$ 
14:        else ▷ Nur erreichbar, falls Vermutung ?? falsch
15:           $\varphi = (\varphi_1, \varphi_2) \leftarrow$  int-two-flow( $\mathcal{N}_G$ )
16:          if  $\varphi \neq$  None then ▷ Sonst Gegenbeispiel zu Vermutung ??
17:             $\phi \leftarrow$  FAA( $\varphi_2$ )
18:            return  $\phi$ 
19:          end if
20:        end if
21:      else
22:         $\phi \leftarrow$  FAA( $\varphi_2$ )
23:      end if
24:    end if
25:    return  $\phi$ 
26:  end if
27: end procedure
```

---