

# Metoda hill-climbing

Joanna Kołodziejczyk

2016

## Prosty hill climbing

```
s ← s0
eval ← f(s)
k ← 0
while k < kmax do
  sn ← neighbour(s)
  evaln ← f(sn)
  if evaln < eval then
    s ← sn
    eval ← evaln
  end if
  k ← k + 1
end while
return s
```

# Steepest hill climbing

```
 $s \leftarrow s_0; s_{best} \leftarrow s_0$   
 $eval \leftarrow f(s); eval_{best} \leftarrow f(s)$   
loop  
   $s_{list} \leftarrow neighbours(s)$   
  for all  $x$  in  $s_{list}$  do  
     $eval_n \leftarrow f(x)$   
    if  $eval_n < eval_{best}$  then  
       $s_{best} \leftarrow x$   
       $eval_{best} \leftarrow eval_n$   
    end if  
  end for  
  if  $eval_{best} \geq eval$  then  
    return  $s$   
  end if  
   $s \leftarrow s_{best}; eval \leftarrow eval_{best}$   
end loop
```

# Co musi być dane przed uruchomieniem algorytmu

- problem optymalizacyjny, ze zdefiniowaną, minimalizowaną funkcją celu  $f$
- `neighbour()` lub `neighbours()` — funkcja generowania sąsiada lub wszystkich sąsiadów stanu bieżącego, która powinna wprowadzać małe losowe zmiany i być tak skonstruowana, by każde możliwe rozwiązanie było osiągalne
- ograniczenie liczby wykonanych kroków  $k_{max}$