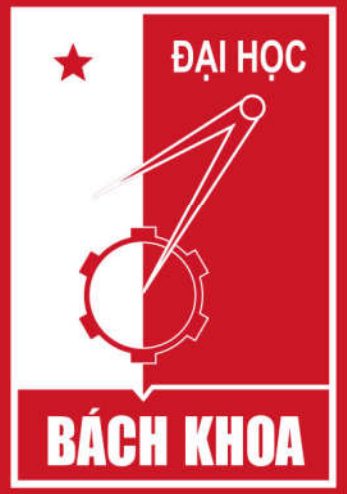


HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

HA NOI UNIVERSITY OF SCIENCE AND TECHNOLOGY
SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

FUNDAMENTALS OF OPTIMIZATION

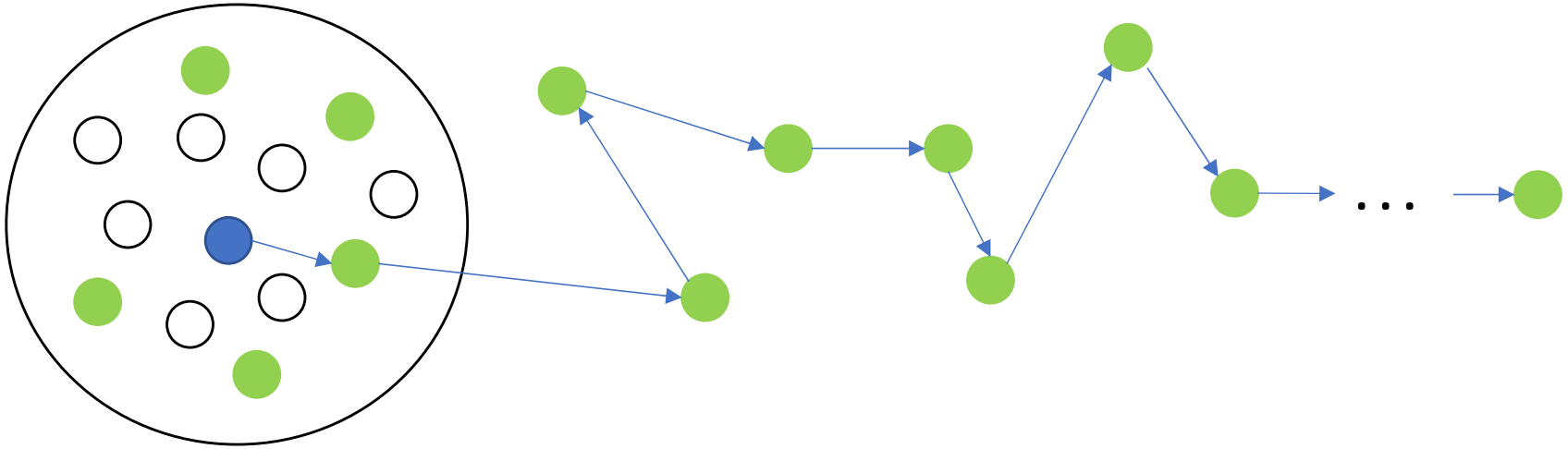
Constraint-based local search methods

Content

- Overview
- Examples

Overview

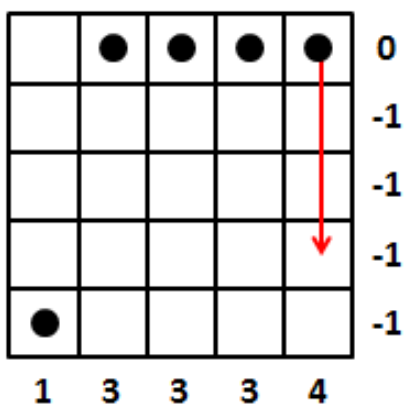
- Start with an initial solution
- Iteration:
  - Replace the current solution by one of its neighbors
- Stop condition
  - Time limit expired
  - Maximum number of iterations
  - Optimal solution found
  - . . .

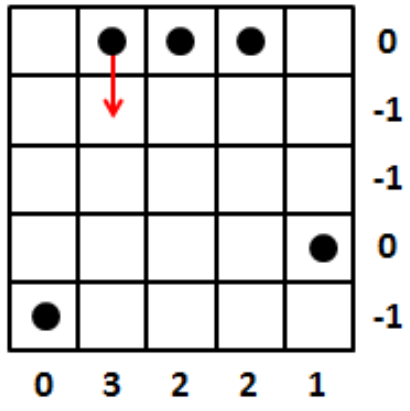


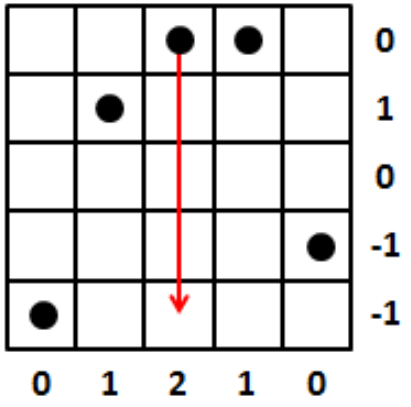
Neighbors and selection

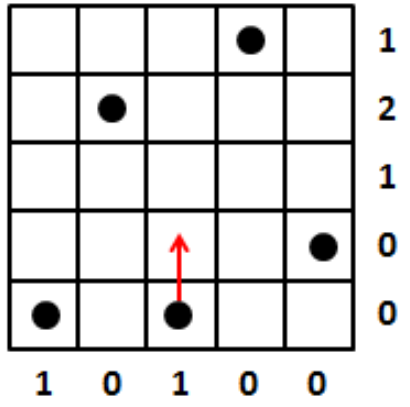
- Neighbors: Change some decision variables on the current solution
- Neighbor selection: Base on some quality function

N-queens problem

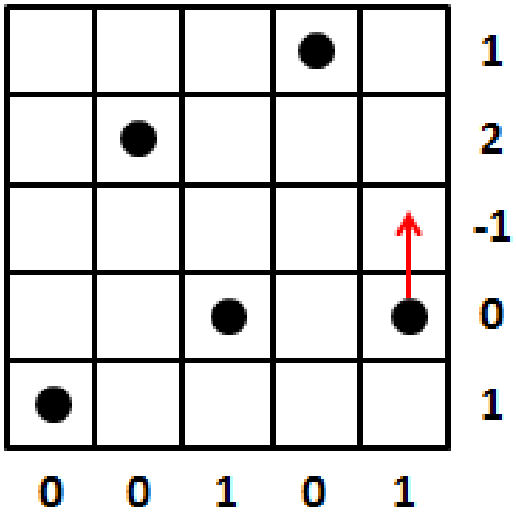


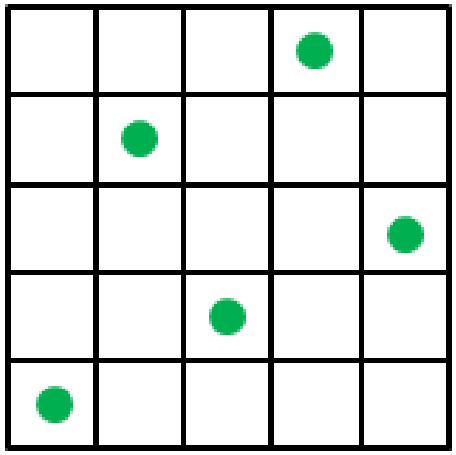






N-queens problem





Constraint-based local search

- Decision variables
  - Model the problem
- Invariants
  - Maintain properties of the problem during the search
- Constraints and functions
  - Maintain properties of the problem and feature differentiation
- Search heuristics
  - Hill climbing
  - Tabu Search
  - Simulated annealing
  - …

Constraint-based local search: N-queens

```
from VarIntLS import VarIntLS
from LocalSearchManager import LocalSearchManager
from NotEqual import NotEqual
from NotEqualFunction import NotEqualFunction
from AllDifferentFunction import AllDifferentFunction
from ConstraintSystem import ConstraintSystem
from PlusVarConst import PlusVarConst
from HillClimbingSearch import HillClimbingSearch
import random as rd
```

Constraint-based local search : N-queens

```
n = 100
mgr = LocalSearchManager()
x = [VarIntLS(mgr,0,n-1,0,'x[' + str(i) + ']') for i in range(n)]
constraints = []
f0 = [PlusVarConst(x[i],0,'f0[' + str(i) + ']') for i in range(n)]
f1 = [PlusVarConst(x[i],i,'f1[' + str(i) + ']') for i in range(n)]
f2 = [PlusVarConst(x[i],-i,'f2[' + str(i) + ']') for i in range(n)]



constraints.append(AllDifferentFunction(f0,'AllDifferentFunction0'))
constraints.append(AllDifferentFunction(f1,'AllDifferentFunction1'))
constraints.append(AllDifferentFunction(f2,'AllDifferentFunction2'))

CS = ConstraintSystem(constraints)
mgr.close()
```

Constraint-based local search : N-queens

```
for iter in range(100000):
        cand = []
        minD = 1000000
        for i in range(n):
                for v in range(x[i].getMinValue(), x[i].getMaxValue() + 1):
                        d = CS.getAssignDelta(x[i],v)
                        if d < minD:
                                cand = []
                                cand.append([i,v])
                                minD = d
                        elif d == minD:
                                cand.append([i,v])
```

Constraint-based local search : N-queens

```
        idx = rd.randint(0,len(cand)-1)


        i,v = cand[idx]
        oldValue = x[i].getValue()
        x[i].setValuePropagate(v)
        print('x = ',[x[i].getValue() for i in range(n)])
        if CS.violations() == 0:
                break
        cur = CS.violations()
```

ĐẠI HỌC BÁCH KHOA

25 YEARS ANNIVERSARY SOICT

VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG

SCHOOL OF INFORMATION AND COMMUNICATION TECHNOLOGY

Thank you for your attentions!

soict.hust.edu.vn/ fb.com/groups/soict