

INF-MAT-5380

<http://www.uio.no/studier/emner/matnat/ifi/INF-MAT5380/>

Leksjon 3

Leksjon 2 - Oppsummering

- Eksempler på DOP
- Alternative formuleringer
- Definisjon nabolag, -operator
- Lokalsøk
- Definisjon lokalt optimum
- Eksakt nabolag
- Prosedyre for lokalsøk
- Traversering av nabolagsgraf
- Kommentarer, ulemper, hovedutfordring

Forenklet variant av

Local_Search (S,f,N,strategy)

*/ strategy is "First Accept" or "Best Accept"

current=Init_Solution(S)

*/ incumbent:=current unødvendig i rent LS

local_optimum:=false

while not local_optimum **do**

 (current,local_optimum):=

 Search_the_Neighborhood

 (current,N(current),f,strategy)

if local_optimum **return** current

od

Search_the_Neighborhood (current,Neighbors,f,strategy)

best_neighbor:=current

for s **in** Neighbors **do**

if $f(s) < f(\text{best_neighbor})$ **then**

if strategy="First Accept" **then**

return (s,false) **else**

 best_neighbor:=s */ strategy is "Best Accept"

fi

fi

od

return (best_neighbor,best_neighbor=current)

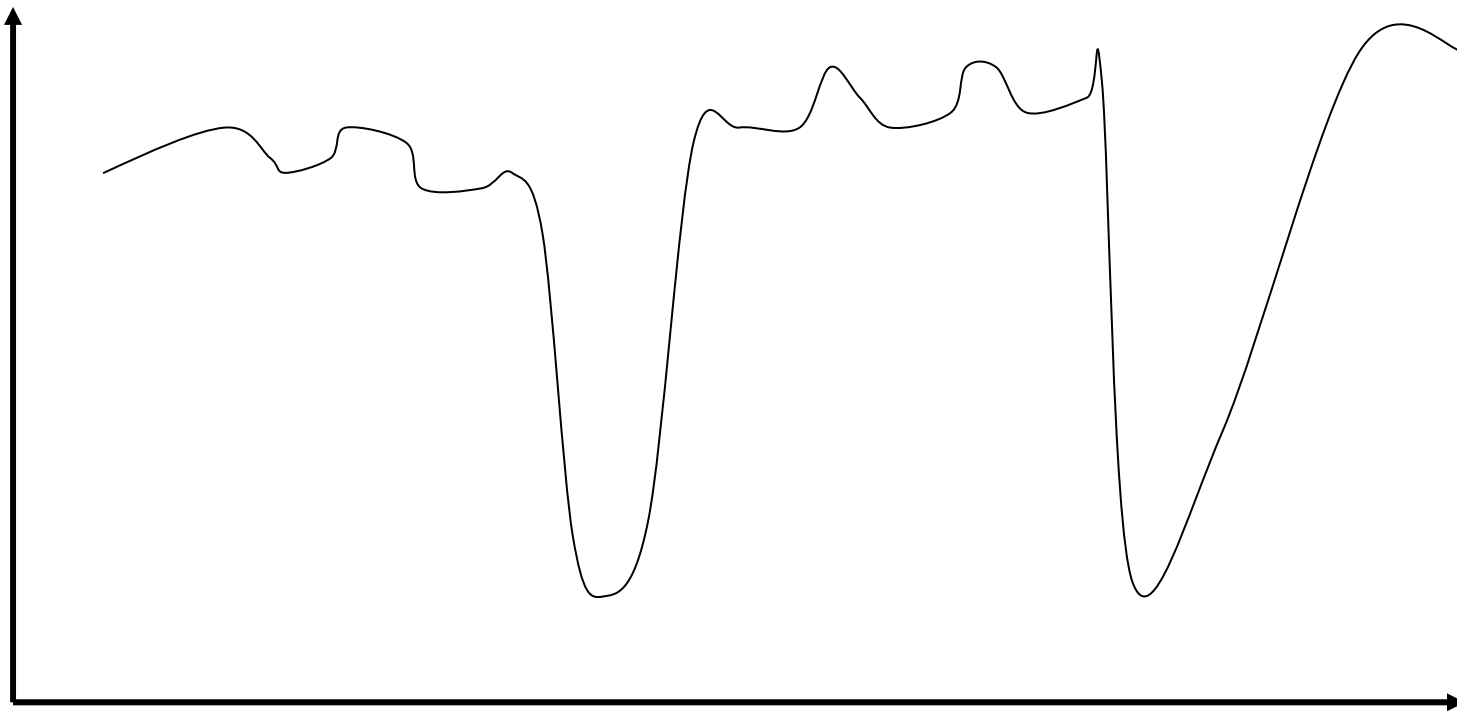
*/ (current,local_optimum)

Leksjon 3 - Oversikt

- Tilfeldig søk
- Simulert størkning
- Terskelakseptanseteknikker

Lokale og globale optima

Kostnad



Løsningsrom

Unnslippe lokale optima i LS

- noen mulige strategier

- Restart av lokalsøk fra andre løsninger
- Tillate flytt til dårligere løsninger
 - deterministisk
 - probabilistisk
- Minne
 - hvilke løsninger er besøkt før?
 - unngå å besøke dem igjen
- Variere mellom ulike nabolag

Metaheuristikker (Generelle heuristikker)

- søkestrategier for å unnsnippe lokale optima
- introdusert tidlig 80-tallet
- betydelig suksess i løsning av DOP
- analogier fra fysikk, biologi, menneskelige hjerne, menneskelig problemløsning
- mange varianter

Noen metaheuristikker

- Simulated Annealing (SA, simulert størkning)
- Threshold Accepting (TA, terskelakseptanse)
- Genetic Algorithms (GA, genetiske algoritmer)
- Memetic Algorithms (MA, memetiske algoritmer)
- Evolutionary Algorithms (EA, Evolusjonære algoritmer)
- Differential Evolution (DE, differensiell evolusjon)
- Ant Colony Optimization (ACO, maurkolonioptimering)
- Scatter Search (SS, spredningssøk)
- Path Relinking (PR, ?)
- Tabu Search (TS, tabusøk)
- Guided Local Search (GLS, styrt lokalsøk)
- Greedy Randomized Adaptive Search (GRASP, ?)
- Iterated Local Search (ILS, iterert lokalsøk)
- Variable Neighborhood Descent / Search (VND/VNS, variabelt nabolagsnedstigning / søk)
- Neural Networks (NN, nevralt nett)

"Definisjon" av metaheuristikk (a la Osman & Kelly)

En metaheuristikk er
en iterativ genereringsprosess
som styrer en underliggende heuristikk
ved å kombinere
(på en intelligent måte)
ulike strategier for å utforske og utnytte søkerom
(og læringsstrategier)
for å finne nær-optimale løsninger på en effektiv måte

"Definisjon" av metaheuristikk (a la Glover & Kochenberger)

Løsningsmetoder som benytter **interaksjon** mellom **lokale forbedringsprosedyrer** (lokalsøk) og **strategier** på høyere nivå for å **unnsnippe lokale optima** og sørge for et **robust søk** i et søkerom

Variant av lokalsøk: Tilfeldig søk ("Random Search")

```
Procedure Random_Search(f,N,Stop,initial)
begin
  current:=incumbent:=initial;
  while not Stop() do
  begin
    current:=Random_Solution(N(current))
    if f(current) < f(incumbent) then
    begin
      incumbent:=current;
    end
  end
  end
  return incumbent;
end
```

Stoppkriterier?

Variant av lokalsøk:

Tilfeldig nedstigning ("Random Descent")

```
Procedure Random_Descent(f,N,Stop,initial)
begin
  new_solution:=current:=incumbent:=initial
  while not Stop() do
    begin
      Neighbors:=N(current)
      while not Stop() and f(new_solution) >=f(current) do
        begin
          new_solution:=Random_Solution(Neighbors)
        end
        current:=new_solution
        if f(current) < f(incumbent) then incumbent:=current
      end
    return incumbent
  end
```

Problemer med lokalsøk

- Mulige strategier

- Mål
 - unnsnippe lokale optima
 - unngå løkker
- Akseptere dårligere løsning
- Tilfeldighet

- Metaheuristikken simulert størkning bruker disse strategiene

Metaheuristikken Simulert størkning (Simulated Annealing, SA)

- Inspirert av statistisk termodynamikk (nedkjøling av smeltet materiale)
- Brukt i optimering i 20 år (Kirkpatrick et al 1983)
- Bygget på lokalsøk (variant av Random Search/Random Descent)
- Enkel å implementere
- Mye litteratur
- Konvergerer mot globalt optimum under svake antakelser
- Men som oftest sakte....

Simulert størkning

- Metropolis et al (1953)
 - algoritme for å simulere energiendringer av fysiske systemer under nedkjøling
- Kirkpatrick et al (1983)
 - foreslo å bruke samme type simulering til å søke i brukbare løsninger for DOP

SA - Analogi

Termodynamikk

- Systemtilstand
- Energi
- Tilstandsending
- Temperatur
- Slutt-tilstand

Diskret optimering

- Tillatt løsning
- Kostnad
- Flytt til naboløsning
- Kontrollparameter
- Slutt-løsning

Simulert størkning (SA)

Kan uttrykkes som strategi for valg i nabolag i basalt lokalsøk:

Procedure Local_Search(Init_Sol,N,f,Strategy,Stop_Criterion)

*/ Strategy = SA

incumbent:=current:= Init_Sol

Repeat

 current:=Select_SA_Neighbor(f,current,N(current),Stop_Criterion)

if f(current)< f(incumbent) **then** incumbent :=current

Until Stop_Criterion()

return incumbent

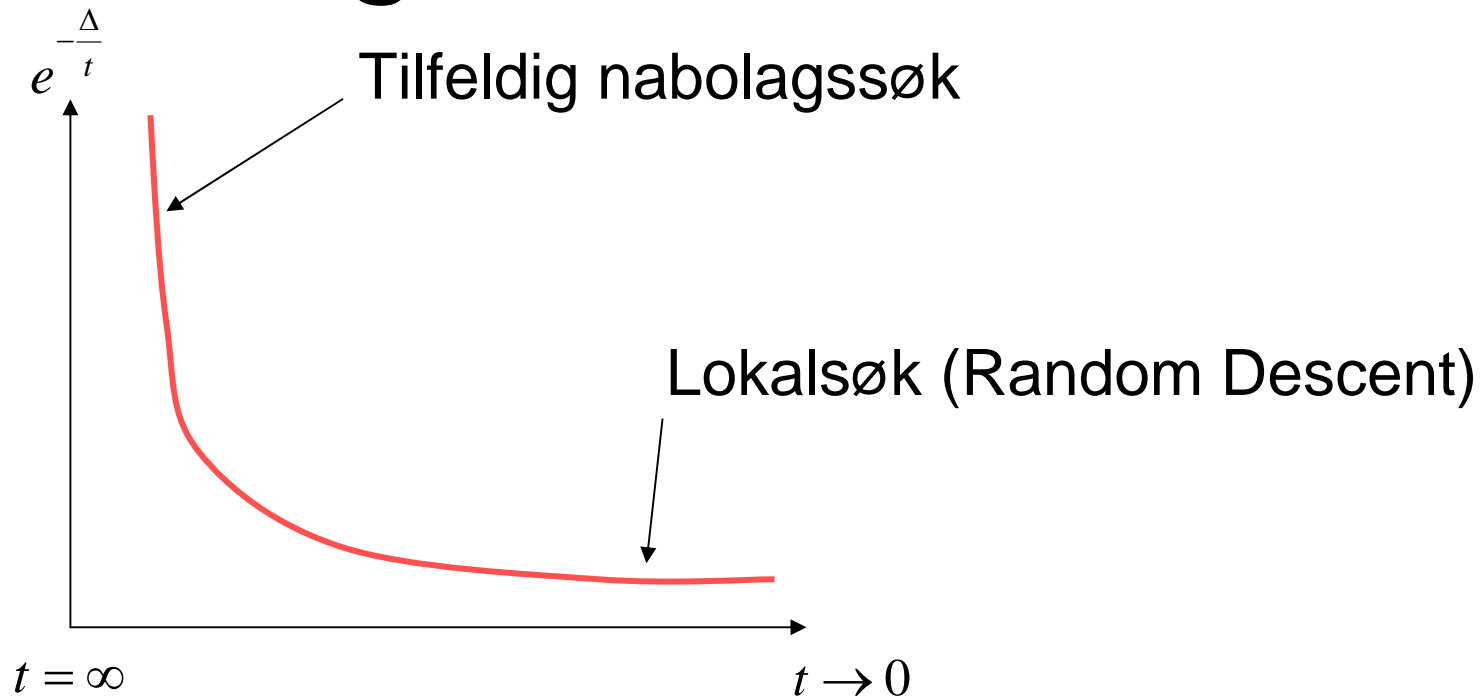
Valg av flytt i SA

- Modifisert "Random Descent"
- Velger tilfeldig løsning i nabolag
- Aksepterer denne
 - ubetinget, hvis bedre løsning enn nåværende
 - med viss, endelig sannsynlighet hvis dårligere
- Sannsynlighet styres av kontrollparameter (temperatur)
- Unngår å sitte fast i lokalt optimum

Valg av flytt i SA

```
Procedure Select_SA_Neighbor  
  (f,current,Neighbors,Stop_Criterion)  
  */ Strategy is Simulated Annealing  
while not Stop_Criterion() do  
begin  
  i:=Random_Element(Neighbors)  
  delta := f(i) - f(current)  
  if delta < 0 or Random(0,1) < exp(-delta/t) then  
    return i  
end
```

Akseptanse av dårligere løsning



SA - overordnet struktur

- initiell kontrollvariabel t_0 (høy verdi)
- et antall iterasjoner med samme kontrollvariabel
- så reduksjon $t_{i+1} = \alpha(t_i)$
- ”nedkjølingsplan” (”cooling schedule”)
- stoppkriterier – f.eks ”minimumstemperatur”
- gjentakelse
- løsningskvalitet og hastighet avhengig av valg
- nabolagsstruktur viktig

SA – overordnet prosedyre

Procedure Simulated_Annealing

(f,N,Stop_Criterion,T0,Nrep,Alpha)

incumbent:=current:= Find_Initial_Solution()

T:=T0

Repeat

for i:=1 **to** Nrep **do** */ Several iterations with one T value

begin

current :=Select_SA_Neighbor(f, current,N(sol),incumbent,T)

if f(current) < f(incumbent) **then** incumbent:= current

end

T:=Alpha(T) */ Cooling

Until Stop_Criterion()

return incumbent

Statistisk analyse av SA

- Modell: tilstandsoverganger i søkerommet
- Overgangssannsynligheter $[p_{ij}]$ mellom løsninger
- Kun avhengig av i og j : homogen Markovkjede

Når alle overgangssannsynligheter er endelige, vil SA-lokalsøket konvergere mot en stasjonær fordeling som er uavhengig av startløsningen. Når temperaturen går mot null vil denne fordelingen gå mot en uniform fordeling over de globale optima.

- Statistisk garanti for at SA finner globalt optimum
- I praksis: eksponensiell kjøretid for å garantere optimum

SA i praksis

- heuristisk approksimasjonsalgoritme
- oppførsel sterkt avhengig av nedkjølingsplan
- teori: ønskelig med eksponensielt antall iterasjoner ved hver temperatur
- i praksis:
 - stort antall iterasjoner, få temperaturer
 - lite antall iterasjoner, mange temperaturer

SA i praksis – "heksekunst"

- geometrisk rekke

$$t_{i+1} = at_i, \quad i = 0, \dots, K \quad a < 1 \quad (0.8 - 0.99)$$

- antall repetisjoner kan varieres
- adaptivitet: variabelt antall trekk før temperaturreduksjon
- eksperimentering nødvendig

SA – Generelle beslutninger

- Nedkjølingsplan
 - basert på maksimal forskjell mellom løsninger i nabolag
 - antall repetisjoner ved hver temperatur
 - reduksjonsraten
- Adaptivt antall repetisjoner
 - flere repetisjoner for lavere temperaturer
 - antall aksepterte flytt, men maksimalgrense
- Svært lave temperaturer er unødvendig
- Nedkjølingsrate er viktigere enn nedkjølingsmetode

SA – Problemspesifikke beslutninger

- Viktige mål
 - Responstid
 - Kvalitet på løsning
- Viktige valg
 - Løsningsrom
 - hva med ikke-tillatte løsninger?
 - Nabolagsstruktur
 - Kostnadsfunksjon
 - straff for brudd på føringer
 - approksimasjon
 - Nedkjølingsplan

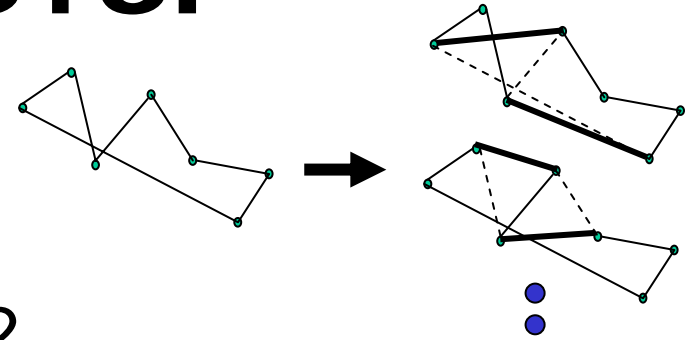
SA – Valg av nabolag

- Størrelse
- Variasjon i størrelse
- Topologi
 - Symmetri
 - Sammenheng: Enhver løsning kan nås fra enhver annen
- "Topografi"
 - "Spikre"
 - "Platåer"
 - Dype lokale optima
- Nabolag og kostnadsfunksjon

SA - Hastighet

- Tilfeldig valg av nabo
 - reduksjon av nabolag
- Kostnad av kandidatløsning
 - differanse uten full beregning
 - approksimasjon
- Akseptansebeslutning
 - forenklinger

SA – Eksempel: STSP



- Søkerom $(n-1)!/2$
- Nabolag: 2-opt $n(n-1)/2$
- Sammenhengende
- Enkel representasjon av flytt
- Naturlig kostnadsfunksjon
- Differanse enkel å beregne
- Generalisering: k-opt

SA – Finjustering

- Testproblemer
- Testbenk
- Visualisering av løsning
- Verdier på
 - kostnad / straff
 - temperatur
 - antall / andel aksepterte flytt
 - iterasjon / CPU
- Sammenheng med SA-parametre
- Fare for overtilpasning

SA – Modifikasjoner og utvidelser

- Probabilistisk
 - Endret akseptansesannsynlighet
 - Forenklet kostnadsfunksjon
 - Approksimasjon av eksponensialfunksjon / tabell
 - Få temperaturer
 - Omstart
- Deterministisk
 - Terskelakseptanse (Threshold Accepting, TA)
 - Record-to-Record Travel
 - Nedkjølingsplan
 - Omstart

Valg av flytt

- Threshold Accepting (TA)

Procedure Select_TA_Neighbor (f,current,Neighbors,incumbent,theta1)

*/ Strategy is Threshold Accepting

while true do

begin

i:=Random_Element(Neighbors)

delta := f(i) - f(current)

***/ SA: if** delta < 0 **or** Random(0,1) < exp(-delta/t)

if delta < theta1 */ Positive Threshold w.r.t. current

then return i

end

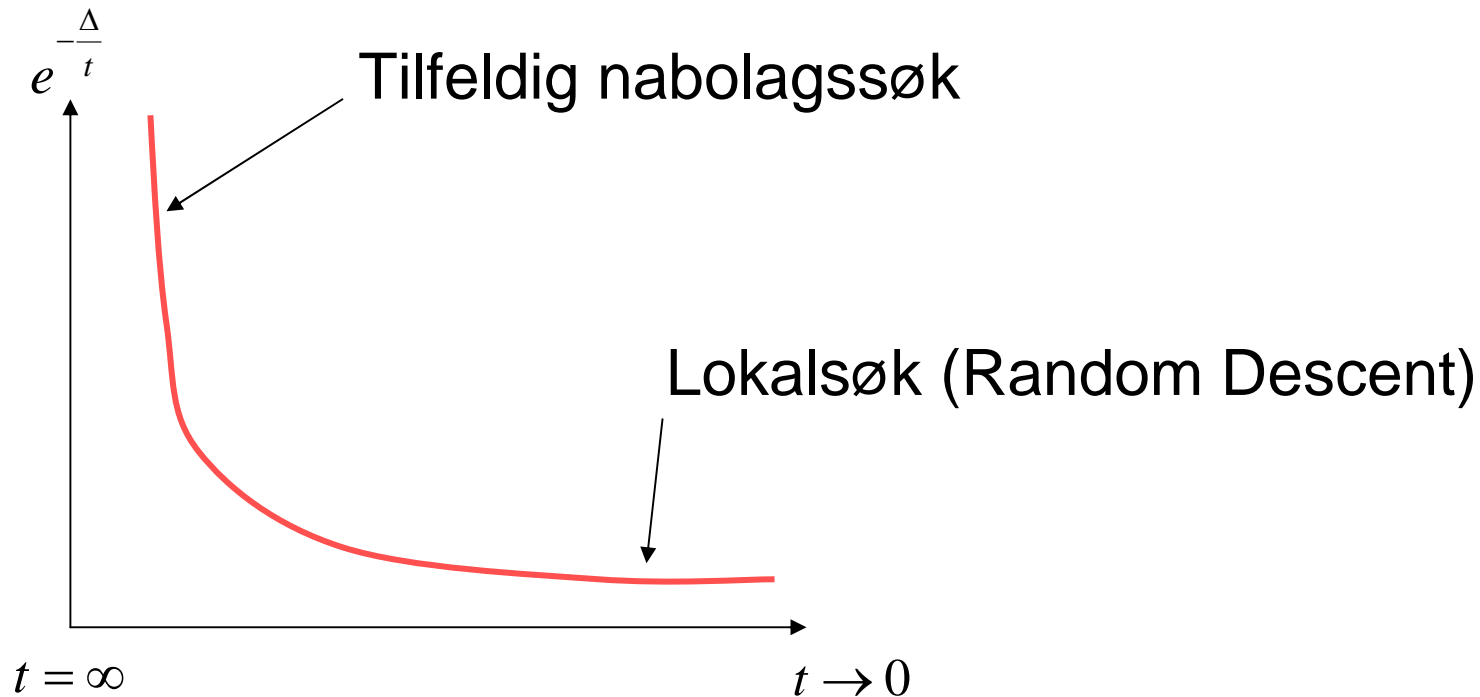
- Problem med denne prosedyren?

Valg av flytt

- Record-to-Record Travel

```
Procedure Select_RRT_Neighbor  
  (f,current,Neighbors,incumbent,theta2)  
  /* Strategy is Record-to-Record Travel  
while true do  
begin  
  i:=Random_Element(Neighbors)  
  /* SA, TA: delta := f(i) - f(current)  
  /* SA: if delta < 0 or Random(0,1) < exp(-delta/t)  
    if f(i) < theta2*incumbent */ theta2 > 1  
    then return i  
end
```

SA/TA – "Nedkjølingsplan"



Remedier:

- godt valg av starttemperatur
- godt stoppkriterium
- hurtigere nedkjøling først og sist
- testkjøring

SA – Valg av flytt

- Standard: Tilfeldig trekk i nabolag
 - Problematisk rundt lokale optima
 - Remedie: Syklisk valg av nabo
- Standard: Lav akseptanserate ved lave temperaturer
 - mye unødig beregning
 - mulige remedier:
 - Akseptansesannsynlighet
 - Valg av nabo basert på veiet trekning
 - Deterministisk akseptanse

SA – Kombinasjon med andre metoder

- Preprossesering – god startløsning
- Standard lokalsøk underveis
 - hvert aksepterte flytt
 - hver forbedrende flytt
- SA i konstruksjonsheuristikker

SA - Oppsummering

- Inspirert av statistisk mekanikk - nedkjøling
- Metaheuristikk
 - lokalsøk
 - tilfeldig nedstigning
 - tilfeldighet for å unngå lokalt optimum
- Enkel og robust metode, raskt å komme i gang
- Bevis for konvergens til globalt optimum
 - verre enn komplett søk
- I praksis:
 - beregningskrevende
 - finjustering kan gi gode resultater
 - god der det er vanskelig å lage robuste heuristikker som baseres på problemstruktur

Leksjon 3 - Oppsummering

- Lokalsøk – hovedutfordring
- Mulige strategier for å unnsnippe lokale optima
- Definisjon metaheuristikk
- Tilfeldig lokalsøk
- Simulert størkning (Simulated Annealing, SA)
- Terskelakseptanse (Threshold Accepting, TA)
- Record-to-Record Travel (?, RRT)

Leksjon 4 - Oversikt

- Tabusøk