

INF-MAT-5380

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

Leksjon 4

Leksjon 3 - Oversikt

- Tilfeldig søk
- Simulert størkning
- Terskelakseptanse

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
- I praksis:
 - beregningskrevende
 - finjustering kan gi gode resultater
 - god der det er vanskelig å lage robuste heuristikker som baseres på problemstruktur

Leksjon 4 - Oversikt

- Tabusøk

Tabusøk (TS) - Bakgrunn

- Fred Glover 1986: "Future paths for integer programming and links to artificial intelligence"
- Pierre Hansen 1986: "The Steepest Ascent/Mildest Descent Heuristic for Combinatorial Optimization"
- Forskning innen diskret optimering – operasjonsanalyse og kunstig intelligens
- Metoder som bryter barrierer, undersøker ubrukbare områder
- Surrogatføringer
- Kuttplanmetoder
- Læring, kognitiv vitenskap

Tabu (Taboo)

”Ladet med farlig, unaturlig kraft”

”Bannlyst på grunn av moral, smak eller risiko”

*Tabusøk: Styring av søk mot ellers
utilgjengelige områder i søkerommet ved bruk
av restriksjoner*

- Prinsipper for intelligent problemløsning
- Strukturer som utnytter historie (“læring”)

Tabusøk - Hovedidéer

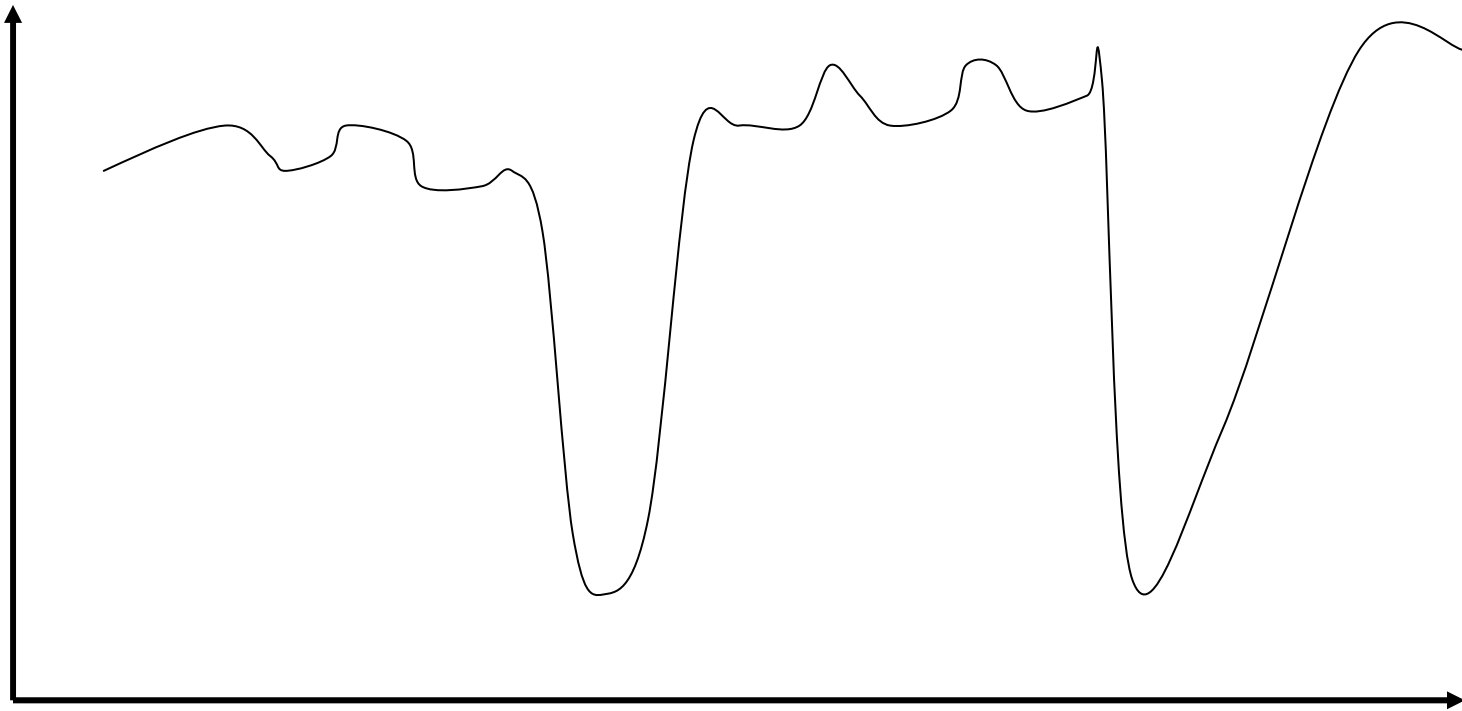
- Basert på lokalsøk
- Søker å unngå lokale optima ved å tillate ikke-forbedrende flytt
- Bruk av minne for å
 - unngå løkker
 - spre søket (diversifisering)
- Generell strategi for å styre nabolagssøk/"indre" heuristikker
- *Meta-heuristikk* (Glover)

Grunnleggende Tabusøk

- Lokalsøk med "Beste Nabo"-strategi
- Alltid flytt til beste nabo ("aggressivt lokalsøk")
 - Fordel/ulempe?
- Men: noen naboer er tabu
 - defineres av tabu-kriterier
- Men: noen er tilgjengelige (admissible) likevel
 - bestemmes av admissibilitetskriterier
 - typisk eksempel: beste løsning hittil
- Minne (korttids): Tabuliste

Lokale og globale optima

Kostnad



Løsningsrom

Taburestriksjoner

- defineres på egenskaper ved løsninger eller flytt - *attributter*
- hvor ofte – eller hvor nylig - (frequency, recency) har denne attributten vært involvert i (å generere) løsning
- datastruktur: tabu-liste

Local_Search (S,f,N,strategy)

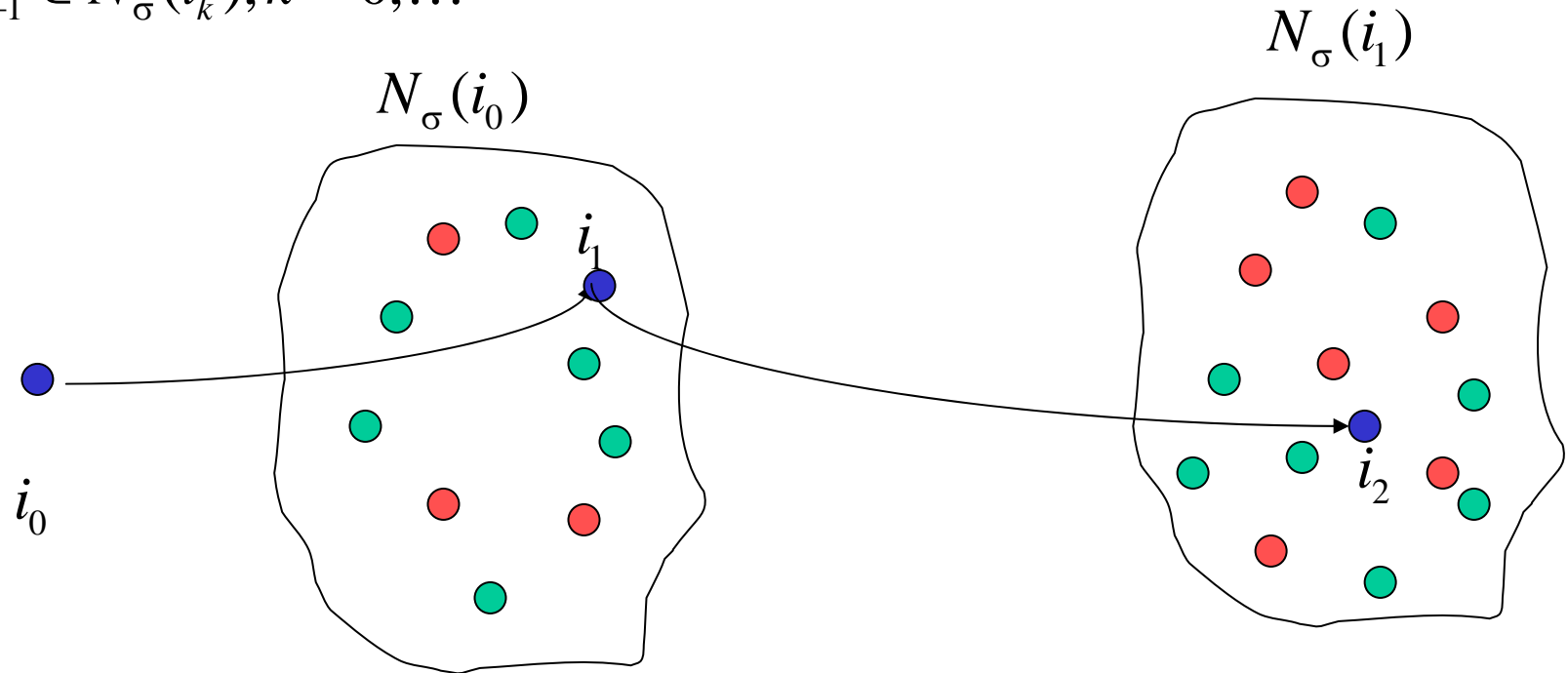
```
incumbent:=current=Init_Solution(S)
*/ best solution until now
local_optimum:=false
while not local_optimum do
    (current,incumbent,local_optimum):=
        Search_the_Neighborhood
        (current,N,f,strategy,incumbent)
    if local_optimum return incumbent
od
```

Search_the_Neighborhood (current,N,f,strategy,incumbent)

```
best_neighbor:=current
neighbors=N(current)
for i in neighbors do
    if f(i) < f(best_neighbor) then best_neighbor:=i
    if f(i) < f(incumbent) then
        if strategy="First Accept" then
            return (i,i,false) else
                incumbent:=i          */ strategy is "Best Accept"
        fi
    fi
od
return (best_neighbor,incumbent,best_neighbor=current)
*/      (current,incumbent,local_optimum)
```

Lokalsøk: Traversering av nabolagsgraf

$$i_{k+1} \in N_{\sigma}(i_k), k = 0, \dots$$



Et *flytt* er prosessen å velge en gitt løsning i nabolaget til nåværende løsning - som nåværende løsning for neste iterasjon

Local_Search (S,f,N,Basic_Tabu_Search)

incumbent:=current:=Init_Solution(S)

*/ best solution until now

*/ local_optimum:=false

while not Stopping_Criterion() **do**

 current:=Search_the_Neighborhood

 (current,N,f,Basic_Tabu_Search,incumbent)

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

 */ **if** local_optimum **return** incumbent

od

return incumbent

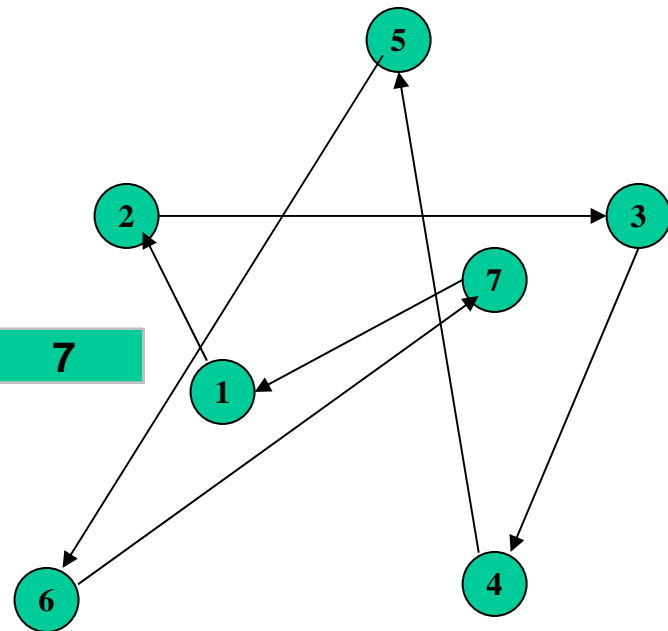
Search_the_Neighborhood (current, N, f, strategy, incumbent)

```
*/ Assuming strategy=Basic_Tabu_Search
*/ best_neighbor:=current
best_acceptable_neighbor:=Really_Bad_Solution()
Neighbors=N(current)
for i in Neighbors do
    if f(i) < f(best_acceptable_neighbor)
        and (not Tabu(i,Tabu_List) or Admissible(i)) then
        best_acceptable_neighbor:=i
od
Update_Tabu_List(best_acceptable_neighbor,Tabu_List)
return best_acceptable_neighbor
```

Eksempel: TSP

- Representasjon: permutasjonsvektor
- Flytt: parvis bytte

$$(i, j) \quad i < j \quad i, j \in [1, n]$$

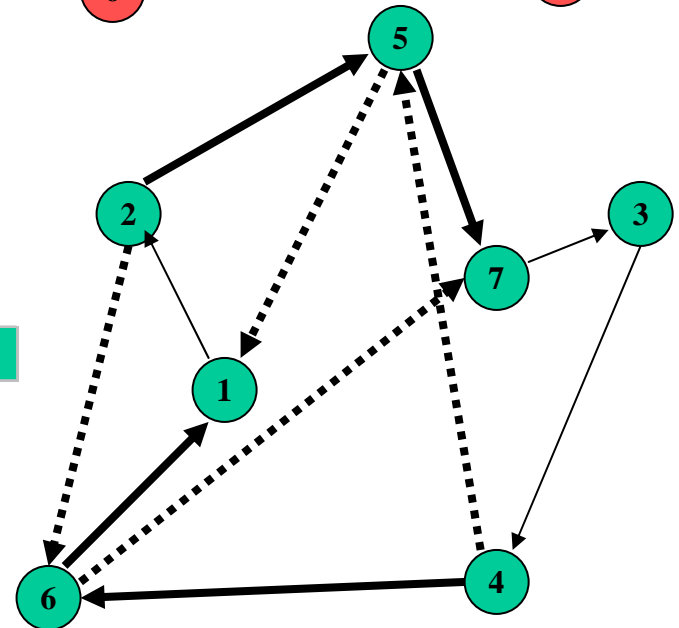
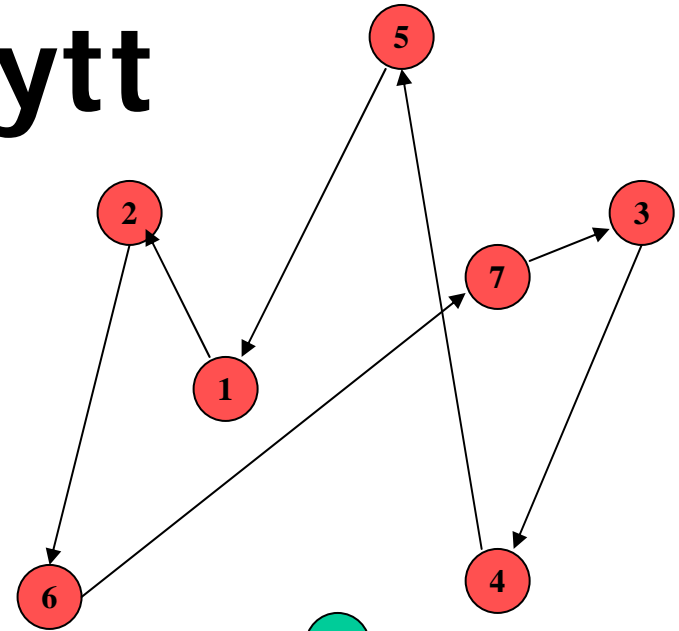


TSP-eksempel - Flytt

Bytte i permutasjonsvektor



Flytt: *Bytt(5,6)*



TSP-eksempel

- Antall flytt: $\binom{n}{2}$ $i_{k+1} \in N_{\sigma}(i_k), k = 0, \dots$

- Til hvert flytt: *flyttverdi*

$$\Delta_{k+1} = f(i_{k+1}) - f(i_k), i_{k+1} \in N(i_k)$$

- Valg av taburestriksjon
 - attributt: de byer som inngår i flytt
 - det er tabu å foreta flytt som involverer samme byer
 - de siste k iterasjoner
 - k=3 (tabu-varighet, "tabu tenure")
- Valg av aspirasjonskriterium
 - beste løsning hittil

TSP-eksempel

- Tabukriterier og datastruktur

- Attributt: paret av byer som er involvert i flyttet
- Tabu: flytt som involverer samme par
- Antall iterasjoner til flytt blir lovlig igjen (tabuvarighet): 3
- Datastruktur: triangulær tabell, antall iterasjoner til trekk blir lovlig igjen
- Oppdateres for hvert flytt

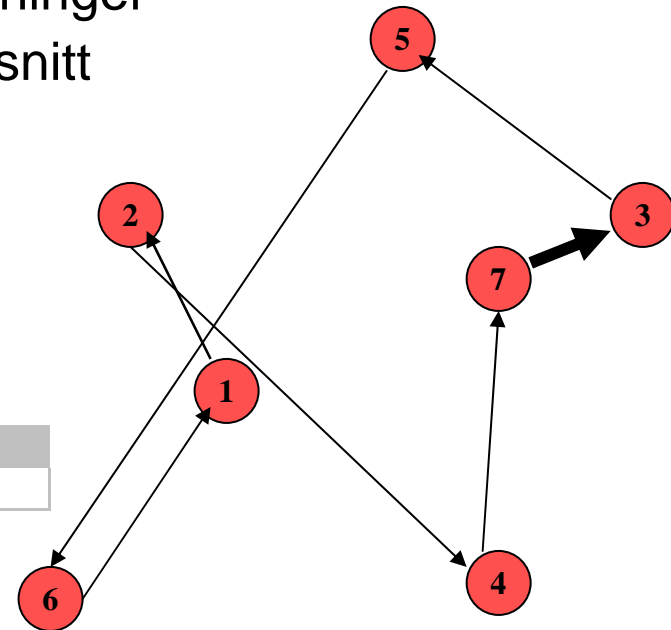
	2	3	4	5	6	7
1	0	2	0	0	0	0
	2	0	3	0	0	0
		3	0	0	0	0
			4	1	0	0
				5	0	0
					6	0
						0

TSP-eksempel

- Alternative tabukriterier/attributter

- ulovlig å operere på gitt(e) byer
- ulovlig å endre på byen med posisjon k i vektor
- kriterier knyttet til lenker mellom byer
 - lenker som forekommer ofte i gode løsninger
 - lengde på lenker - i forhold til gjennomsnitt
- for permutasjonsproblemer:
 - attributter knyttet til forrige/neste virker ofte bra

1	2	3	4	5	6	7
2	4	7	3	5	6	1



TSP-eksempel

- Kandidatliste for flytt

1	2	3	4	5	6	7
2	4	7	3	5	6	1

Gjeldende løsning

Verdi: 123

Bytt	Verdi
3,1	-2
2,3	-1
3,6	1
7,1	2
6,1	4

Kandidatliste

TSP-eksempel

- iterasjon 0

1	2	3	4	5	6	7
2	5	7	3	4	6	1

Gjeldende løsning

Verdi: 234

	1	2	3	4	5	6	7
1	1	0	0	0	0	0	0
2		2	0	0	0	0	0
3			3	0	0	0	0
4				4	0	0	0
5					5	0	0
6						6	0
7							7

Bytt	Verdi
5,4	-34
7,4	-4
3,6	-2
2,3	0
4,1	4

← Velger flytt *Bytt(5,4)*

Kandidatliste

Tabu-liste

TSP-eksempel

- iterasjon 1 (etter flytt (4,5))

Bytte	Verdi
5,4	-34
7,4	-4
3,6	-2
2,3	0
4,1	4

Kandidatliste

1	2	3	4	5	6	7
2	4	7	3	5	6	1

Gjeldende løsning

Verdi: 200

	2	3	4	5	6	7
1	0	0	0	0	0	0
2		0	0	0	0	0
3			0	0	0	0
4				3	0	0
5					0	0
6						0

Tabu-liste

TSP-eksempel

- iterasjon 2

Verdi: 200

Gjeldende løsning

1	2	3	4	5	6	7
2	4	7	3	5	6	1

Bytte	Verdi
3,1	-2
2,3	-1
3,6	1
7,1	2
6,1	4

← Velger flytt: (3,1)

Kandidatliste

	1	2	3	4	5	6	7
1		0	0	0	0	0	0
2			0	0	0	0	0
3				0	0	0	0
4					3	0	0
5						0	0
6							0
7							

Tabu-liste

TSP-eksempel

- iterasjon 3

Verdi: 198

Gjeldende løsning

1	2	3	4	5	6	7
2	4	7	1	5	6	3

Bytte	Verdi
1,3	2
2,4	4
7,6	6
4,5	7
5,3	9

← Tabu!

← Velger flytt: (2,4) (forverrende)

Kandidatliste

	1	2	3	4	5	6	7
1		0	3	0	0	0	0
2			0	0	0	0	0
3				0	0	0	0
4					2	0	0
5						0	0
6							0
7							

Tabu-liste

TSP-eksempel

- iterasjon 4

Verdi: 200

Gjeldende løsning

1	2	3	4	5	6	7
4	2	7	1	5	6	3

Bytte	Verdi
4,5	-6
5,3	-2
7,1	0
1,3	3
2,6	6

← Tabu, men gir beste verdi hittil (194).
Velger (4,5)

Kandidatliste

	1	2	3	4	5	6	7
1		0	2	0	0	0	0
2			0	3	0	0	0
3				0	0	0	0
4					1	0	0
5						0	0
6							0
7							

Tabu-liste

Observasjoner

- I eksemplet forbys 3 av 21 flytt
- Strengere tabukriterium oppnås ved
 - øke tabu-varighet
 - styrke tabu-restriksjon
- Dynamisk tabu-varighet (Reaktivt Tabusøk) virker ofte bedre
- Tabu-liste krever plass
- I eksemplet: tabukriteriet basert på nærhet i tid
- Langtidsminne: frekvens

TSP-eksempel

- Frekvensbasert langtidsminne

- Brukes typisk til å diversifisere (spre) søket
- Aktiveres f. eks. når det ikke er forbedrende flytt
- Vanlig å bruke straff for flytt som er utført ofte i langtidsstrategi for diversifisering

Tabu-status (nærhet i tid)

	1	2	3	4	5	6	7	
1			2					
2				3				
3	3							
4	1	5			1			
5		4		4				
6			1		2			
7	4			3				

Frekvens av flytt

Ideer bak Tabusøk

- Mindre vekt på randomisering (enn SA)
- “Intelligent” søk må baseres på mer systematisk styring
- Mer vekt på fleksible minnestrukturer
- Nabolag modifieres på basis av minne (ekskludering av løsninger som er tabu, korttidsstrategi)
- Også tillegg av løsninger (eliteløsninger, gode lokale optima, gjerne brukt i langtidsstrategi)
- Bruk av historie for å modifisere evaluering av løsninger
- TS kan kombineres med straff for brudd på føringer (a la Lagrange-relaksering)

Ideer bak Tabusøk

- “Aggressivt søk”: velg god løsning i nabolag
- Store problemer, kostbar evaluering av løsninger:
 - delta-evaluering
 - approksimasjon av kostnadsfunksjon
 - isolere gode kandidater i nabolag
 - kandidatliste av trekk, utvides ved behov
 - ovenstående er generelle teknikker i lokalsøk-baserte metoder!
- De fleste TS-implementasjoner er deterministiske
- Probabilistisk Tabusøk
 - flytt velges probabilistisk, men basert på basale TS prinsipper
- De fleste TS-implementasjoner er enkle
 - basalt TS med korttidshukommelse, statisk tabu-varighet

TS - Generell formulering

- Finn initiell løsning x
- Initialiser minne H
- Iterer inntil stoppkriterium er oppfylt:
 - Finn kandidatliste $Candidate_N(x)$ blant akseptable løsninger i $N(x)$
 - Velg (med begrensede ressurser) den løsning i kandidatlisten som minimerer en utvidet kostnadsfunksjon $c(H,x)$
 - Oppdater lokalsøk-strukturer og minnet H

Kritiske valg i TS

- Valg av nabolag N
- Definisjon av minnet H
- Hvordan kandidatlisten $Candidate_N(x)$ bestemmes
 - tabukriterium, attributt
 - aspirasjonskriterium
 - langtidsstrategi (inkludering av eliteløsninger)
- Evalueringsfunksjon $c(H,x)$

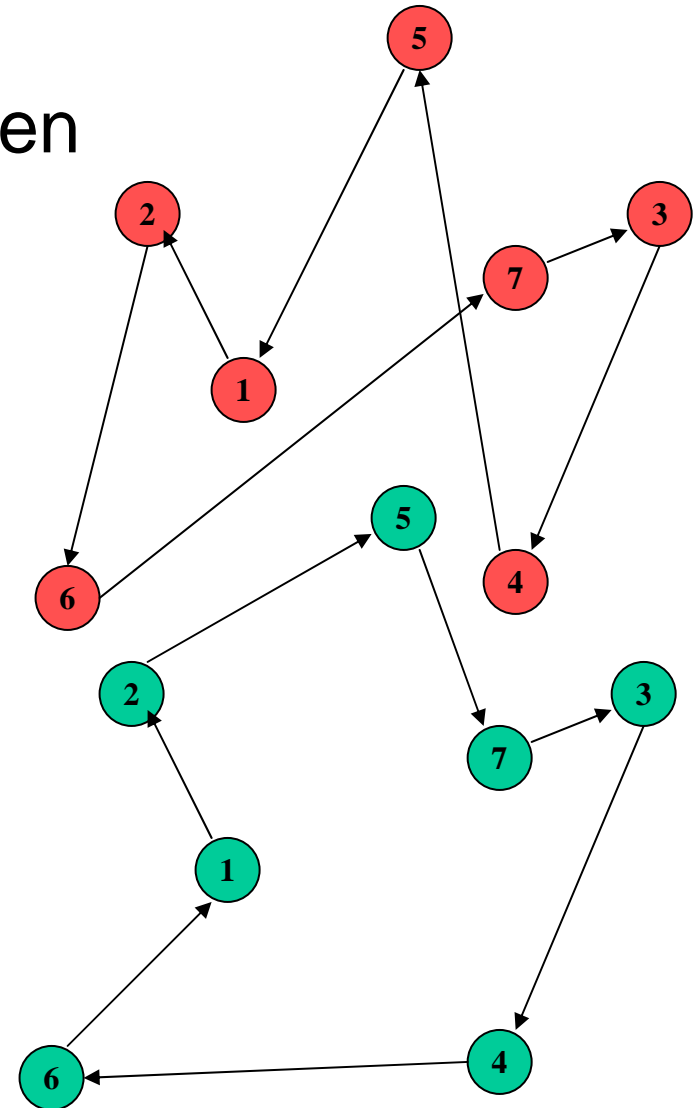
- Ofte implementeres kun grunnleggende tabusøk

Tabusøk - Attributter

- Attributt: Egenskap ved løsning eller flytt
- Kan baseres på ethvert aspekt av løsningen som endres ved flytt
- Utgangspunkt for definisjon av taburestriksjoner
- Ett flytt kan endre mer enn ett attributt

Eksempel: TSP

- Attributt basert på kantene i turen
 - *A1*: Kant legges til i løsningen
 - *A2*: Kant fjernes fra løsningen
- Flytt av type *Bytt*:
 - 4 kanter fjernes
 - 4 kanter legges til
 - *Bytt*(5,6)
 - *A1*:(2,5),(5,7),(4,6),(6,1)
 - *A2*:(2,6),(6,7),(4,5),(5,1)



TS - Naturlige attributter

- Antar binær encoding av løsning
- A1: $x_m^{(k)} : 0 \rightarrow 1$
- A2: $x_n^{(k)} : 1 \rightarrow 0$
- A3: A1 og A2
- A4: Endring av objektiv $f(x)$
- A5: Endring av verdi på annen funksjon $g(x)$

Fra- og til-attributter

- Hvert attributt A_i kan relateres til de to løsninger som inngår i flyttet $x^{(k)} \rightarrow x^{(k+1)}$
- *Bytt(5,6)* i TSP med permutasjonsvektor
 - $A1: (2,5), (5,7), (4,6), (6,1)$
 - $A2: (2,6), (6,7), (4,5), (5,1)$
- $x_{2,5}(0,1), x_{5,7}(0,1), x_{4,6}(0,1), x_{6,1}(0,1)$
- $x_{2,6}(1,0), x_{5,7}(1,0), x_{4,5}(1,0), x_{5,1}(1,0)$

Bruk av attributter i taburestriksjoner

- Anta at flyttet $x^{(k)} \rightarrow x^{(k+1)}$ involverer attributtet A
- Vanlig taburestriksjon:
ikke lov å utføre flytt som reverserer status for A
- TSP-eksempelet:
 - $x_{2,5}(0,1)$
 - $x_{2,5}(1,0)$ er tabu (i noen iterasjoner)

TS - Naturlige Taburestriksjoner

- Antar binær enkoding av løsning
- $R1: x_m^{(k)} : 1 \rightarrow 0$, der tidligere $x_m^{(k-i)} : 0 \rightarrow 1$
- $R2: x_n^{(k)} : 0 \rightarrow 1$, der tidligere $x_n^{(k-j)} : 1 \rightarrow 0$
- $R3$: Minst en av $R1, R2$
- $R4$: Både $R1$ og $R2$
- $R5$: Som $R4$, men med tilleggskravet at den foregående endring skjedde samtidig ($i=j$)
- $R6$: Funksjonen $g(x)$ får en verdi som er oppnådd tidligere
- $R7$: Funksjonen $g(x)$ går fra verdi v_2 til v_1 der $g(x)$ er endret fra verdi v_1 til v_2 i en tidligere iterasjon

Taburestriksjoners rolle

- Vanlig: Unngå reverserende flytt
- Alternativt: unngå repetisjon av stier i søkerommet
 - $R1: x_m^{(k)} : 1 \rightarrow 0$, der tidligere $x_m^{(k-i)} : 1 \rightarrow 0$
- Taburestriksjoner aktiveres kun dersom betingelse om nærhet i tid (eller hyppighet) er oppnådd
- *Tabu-status* for et attributt:
 - attributtet er *tabu-aktivt* dersom betingelsene om nærhet eller frekvens er oppfylt
 - ellers *tabu-inaktivt*
- Unngå av sykler i søkerommet
 - kan kreve avanserte tabukriterier
 - ikke nødvendigvis et mål

Bestemmelse av tabuvarighet

- Statisk
 - $t=7$
 - $t=\sqrt{n}$ der n er mål for problemstørrelse
- Dynamisk
 - $t \in [5, 11]$
 - $t \in [0.9\sqrt{n}, 1.1\sqrt{n}]$
- Avhengighet av attributt
 - TSP-eksempel
 - kant-attributt, tabukriterier for kant inn og ut
 - færre kanter inne enn ute
 - samme tabuvarighet vil være ubalansert

Aspirasjonskriterier

- Når kan taburestriksjoner ignoreres?
- Kan være viktig for ytelse
- Klassisk aspirasjonskriterium:
Aksepter tabu flytt som gir beste løsning hittil
- Andre muligheter:
 - kriterier basert på grad av endring
 - grad av brukbarhet
 - *Influens* av flytt: Grad av strukturell endring av løsning
 - avstandsmål for løsninger
 - Eksempel: Hamming-avstand
- Flytt med høy influens kan være viktige å forfølge når søket er nær et lokalt optimum

Eksempler på aspirasjonskriterier

- Enkelt: Hvis alle trekk er tabu - velg det som er “minst tabu”
- Basert på objektiv: $f(x) < f(x_{incumbent})$
- Basert på søkeretning:
forbedrende vs. forverrende trekk
- Basert på influens: f.eks. Hamming-avstand

Frekvensbasert minne

- Komplementært til korttidsminne
- Brukes i langtidsstrategier i søket
- Frekvenstellere
 - *residens*-basert
 - *transisjons*-basert
- TSP-eksempel
 - hvor ofte har en gitt kant vært med i løsningen? (*residens-basert frekvensmål*)
 - hvor ofte har kantens status vært endret? (*transisjons-basert frekvensmål*)

Intensifisering og diversifisering

- *Intensifisering*: Aggressiv prioritering av gode løsningsattributter i ny løsning
 - kort sikt: basert direkte på attributtene
 - lengre sikt: bruk av eliteløsninger, deler av eliteløsninger (vokabularer)
- *Diversifisering*: Spredning av søk, ved at en prioriterer flytt som gir løsninger med nye sammensetninger av attributter

Intensifisering og diversifisering

- enkle mekanismer

- bruk av frekvensbasert minne
- basert på delmengde S av alle løsninger
- diversifisering:
 - S velges slik at den inneholder stor andel av genererte løsninger (f. eks. alle lokale optima)
- intensifisering:
 - S velges som liten delmengde av eliteløsninger som
 - i stor grad har sammenfallende attributter
 - har liten avstand i søkerommet
 - Flere slike delmengder - partisjonering, cluster-analyse

Straff og oppmuntring (Pisk og gulrot)

- Brukes i evaluering av flytt, tillegg til objektiv
- Gulrot for intensifisering vil være pisk for diversifisering (og omvendt)
- Diversifisering
 - flytt til løsning som inneholder attributter med høy frekvens straffes
 - TSP-eksempel: $g(x)=f(x)+w_1\sum\omega_{ij}$
- Intensifisering
 - flytt til løsning som inneholder attributter med høy frekvens blant eliteløsningene oppmuntres
 - TSP-eksempel: $g(x)=f(x)-w_2\sum\gamma_{ij}$

Path relinking

- Velger to eliteløsninger x' og x''
- Lager (kortest mulig) sti av løsninger som forbinder dem
 $x' \rightarrow x'(1) \rightarrow \dots \rightarrow x'(r) = x''$
- En eller flere av løsningene på stien velges som ny initiell løsning

Kandidatliste

- Strategi for å begrense evaluering av nabolag
- Begrenset liste av trekk som synes lovende
 - tilfeldig valg
 - partisjonering
- Kandidatlisten kan utvides
- Gjenbruk av informasjon
- Parallellprosessering

- Generell teknikk i lokalsøk-baserte metoder!

Strategisk oscillering

- Mål: effektivt samspill mellom intensifisering og diversifisering
- Oppdeling av søket i faser
 - brukbare løsninger / ubrukbare løsning
 - kan realiseres med modifisert evaluering
- Eksempel: Ryggsekkproblemet

Tabusøk - Anvendelser

- Gode resultater for mange problemtyper
- Tidsplanlegging
 - jobshop, flowshop, enmaskin, turnusplanlegging
- Ruteplanlegging
 - TSP, VRP
- Layout, VLSI-design
- Ryggsekkproblemer

- Ikke så gode resultater for hardt beskrankede problemer?

Tabusøk - Sammendrag

- Inspirert fra matematisk optimering og kunstig intelligens/kognitiv vitenskap
- Fokus på bruk av minne fremfor bruk av tilfeldighet
- Basert på “aggressivt” lokalsøk, aksepterer flytt til dårligere løsning
- Kandidatliste-strategier etc. for billigere evaluering av nabolag
- Bruk av korttidsminne
 - attributter
 - unngå reversering og repetisjon
 - tabukriterier, tabuliste
- Aspirasjonskriterier - tabuer er til for å brytes ...
- Diversifisering og intensifisering
- Path relinking
- Strategisk oscillering

- Gode resultater for mange, harde DOP

Leksjon 5 - Oversikt

- Styrt lokalsøk (Guided Local Search, GLS)
- Genetiske algoritmer (GA)

INF-MAT-5380

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

Leksjon 4