

INF-MAT-5380

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

Leksjon 6

Leksjon 5 - Oversikt

- Styrt lokalsøk (Guided Local Search, GLS)

Guided Local Search (GLS) - Bakgrunn

- GENET (neural network)
 - Prosjekt for løsning av 'Constraint Satisfaction' Problemer (CSP) Fra tidlig på 90-tallet
 - Tsang, Voudouris m.fl. (Davenport)
- Edward Tsang og Chris Voudouris 1994
 - videreutvikling av GENET fra CSP til 'Partial CSP' (nesten-løsninger når problemet er overbeskranket): satisfaction \Rightarrow optimering
 - ledet til 'the Tunnelling Algorithm' (94) som igjen ledet til GLS (95)

Hovedidé

- Generell strategi, metaheuristikk, for å styre nabolagssøk/"indre" heuristikker
- Straffer uønskede egenskaper ('features') i løsninger
- Fokuserer på lovende deler av søkerommet
- Søker å unngå lokal optima ved å jobbe med en utvidet objektivfunksjon (original + straffeledd)
- Kjører lokalsøk til (lokalt) minimum, straffer egenskaper, lokalsøk til minimum, straffer,

GLS vs. SA

- I SA er det vanskelig å finne riktig nedkjølingsskjema (problem-avhengig)
 - Høy temperatur gir dårlige løsninger
 - Lav temperatur gir konvergens til lokalt minimum
 - SA er ikke-deterministisk
- GLS besøker lokale minima, men kan komme unna.
 - Ikke tilfeldige "oppover"-flytt som i SA
 - GLS er deterministisk
 - Konvergerer ikke i lokalt minimum; straff tillegges helt til man unnslipper

Minne - GLS vs. Tabusøk

- I tabusøk benyttes frekvensbasert (langtids)minne til å straffe egenskaper som opptrer ofte.
- GLS bruker sitt minne (p_i) gjennom hele søket, ikke bare i egne faser som i tabusøk.
- GLS straffer på bakgrunn av både kostnad og frekvens av en løsning.
- Tabusøk straffer bare på bakgrunn av frekvens og kan dermed søke å fjerne “gode” egenskaper.
- GLS unngår dette ved å benytte domenekunnskap (c_i)
- I GLS synker sannsynligheten for senere å bli straffet jo mer en egenskap er blitt straffet ($c_i/[1+p_i]$)

GLS vs. Tabusøk

	Tabusøk	GLS
Informasjon som brukes	Modifisert nabolag	Modifisert objektivfunksjon
Minne	Tabu liste, frekvensbasert minne	Straff av egenskaper
Når?	Hver iterasjon eller hver N'te iterasjon.	I lokalt minimum i utvidet objektivfunksjon
Søkets "natur"	- Unngå stopp i lokale minima og reverserende flytt - Diversifisering; straffe flytt som gjøres ofte eller attributter som opptrer ofte i løsninger	- Unnslippe lokale minima - Distribuere søke-innsatsen avhengig av kostnad på egenskaper
Intensifisering / diversifisering	< >	Parameteren λ
Reduksjon av nabolag	Kandidatliste	Fast Local Search - FLS

GLS - Oppsummering

- Kan betraktes som "mykere" form for tabusøk
- Straffer (uønskede) egenskaper i løsninger
 - i lokale optima
 - den/de egenskaper som har størst "utility"
- Lokalsøk med utvidet objektfunksjon
 - endrer "topografien"
 - unnslipper derved lokale optima

Leksjon 6 - Oversikt

- Genetiske algoritmer (GA)
- Øvrige populasjonsbaserte metoder
 - Memetiske algoritmer
 - Maurkolonioptimering

Genetiske Algoritmer (GA)

- Holland et al 1960-1970
- Funksjonsoptimering
- AI (spill, mønstergjenkjenning, ...)
- OR etter hvert
- Basisidé:
 - intelligent utforskning av søkerom basert på tilfeldig søk
 - analogier fra biologi

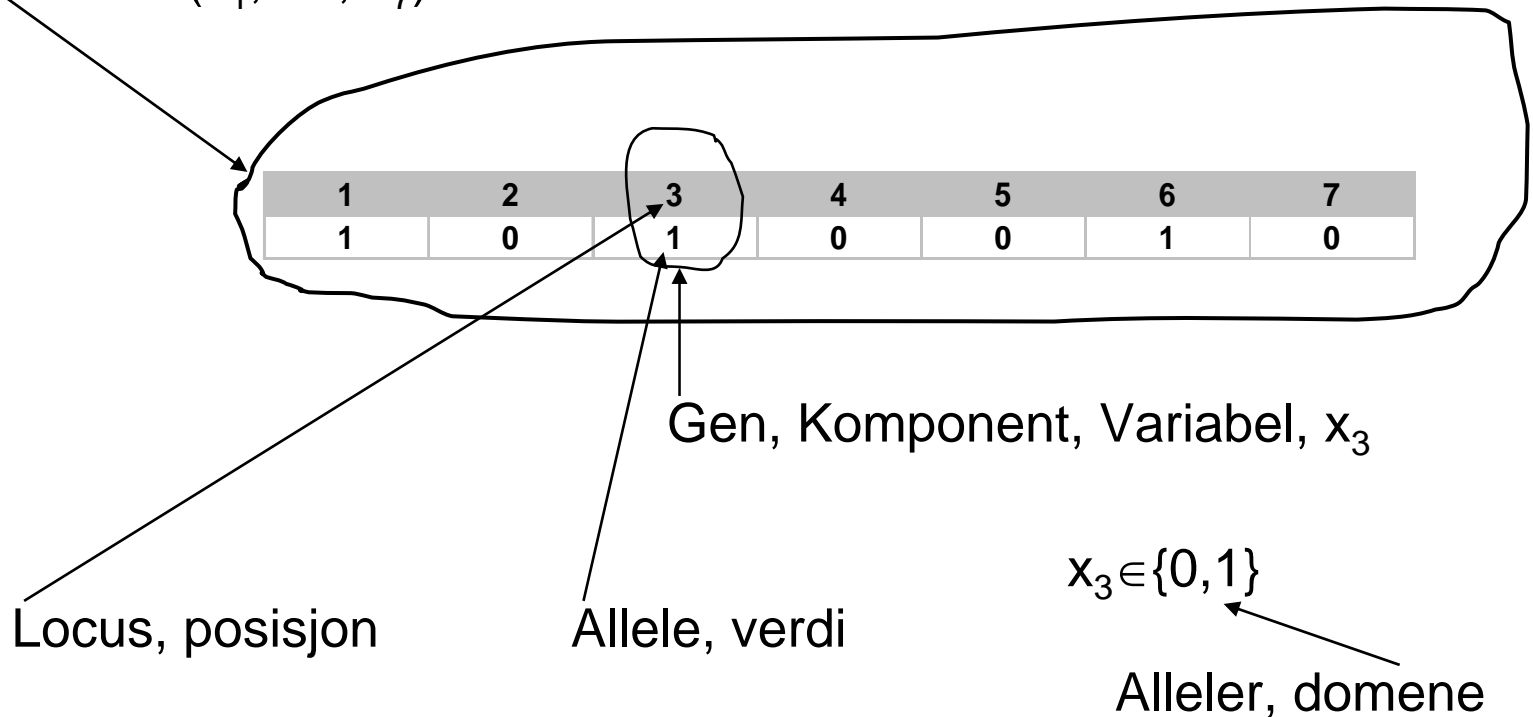
GA - Analogier med biologi

- Representasjon av komplekse objekter ved en vektor av enkle komponenter
- Kromosomer
- Selektiv forplantning
- Darwinistisk evolusjon

- Klassisk GA: Binær enkoding

Klassisk GA: Binære kromosomer

Kromosom, komponentvektor, vektor, streng, løsning,
individ $\mathbf{x}=(x_1, \dots, x_7)$



Genotype, Phenotype, Populasjon

- Genotype
 - kromosom
 - samling av kromosomer
 - kodet streng, samling av kodete strenger
- Phenotype
 - det fysiske uttrykk
 - egenskapene som en mengde løsninger har
- Populasjon - en mengde løsninger

Genetiske operatører

- Manipulerer kromosomer/løsninger
- Mutasjon: Unær operator
- Krysning (crossover): Binær operator
- Invertering
- ...

Evaluering av individer

- Tilpasning, tilpasningsdyktighet – "fitness"
- Relateres til objektfunksjon for DOP
- Maksimaliseres
- Brukes i seleksjon ("Survival of the fittest")
- Normaliseres gjerne

$$f : S \rightarrow [0,1]$$

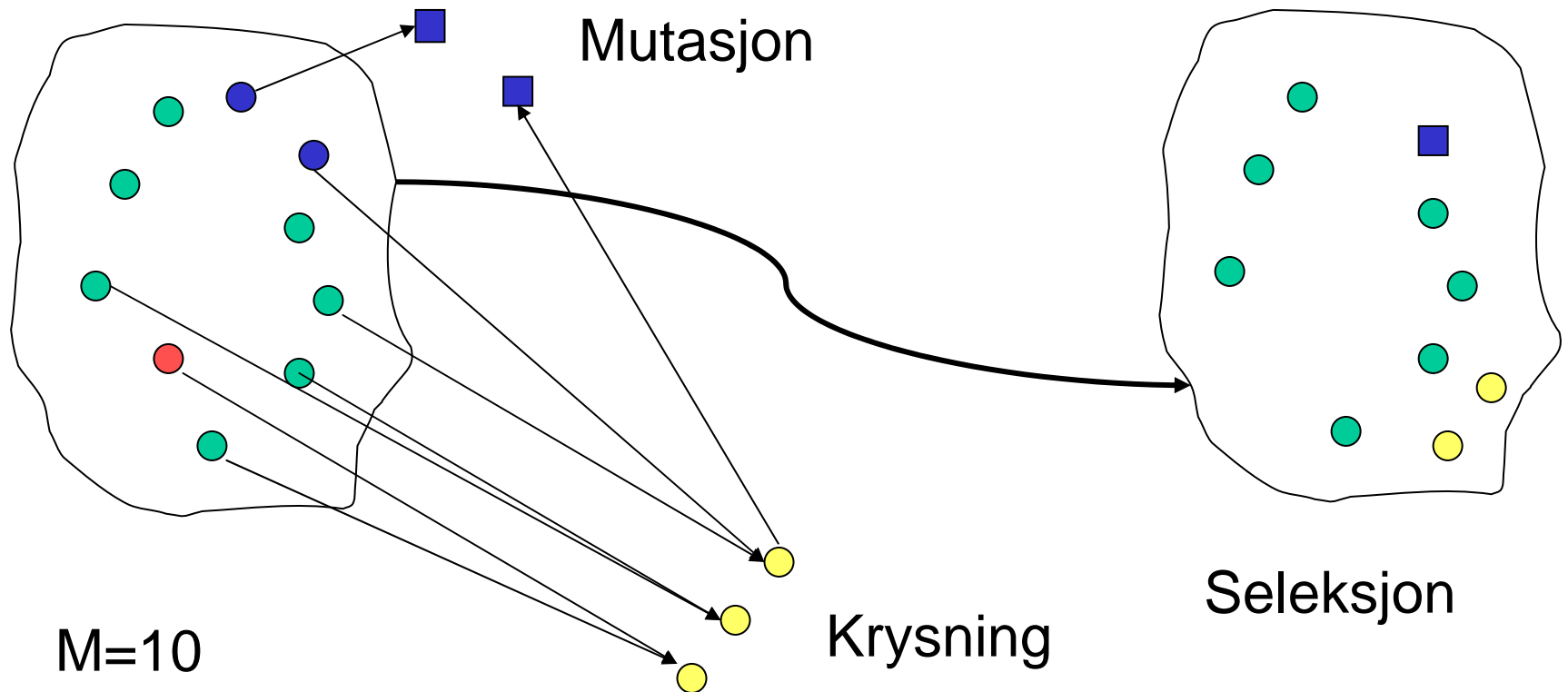
GA - Evolusjon

- N Generasjoner av populasjoner
- For hvert steg i evolusjonen
 - seleksjon av individer for genetiske operasjoner
 - formasjon av nye individer
 - seleksjon av individer som skal overleve
- Fast populasjonsstørrelse M

GA - Evolusjon

Generasjon X

Generasjon X+1



Klassisk GA: Binære kromosomer

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

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

- Funksjonsoptimering
 - kromosom svarer til binær enkoding av reelt tall - min/maks av vilkårlig funksjon
- DOP, eksempelvis TSP
 - binær enkoding av løsning
 - ofte bedre med mer direkte representasjon (f. eks. sekvensrepresentasjon)

GA - Mutasjon

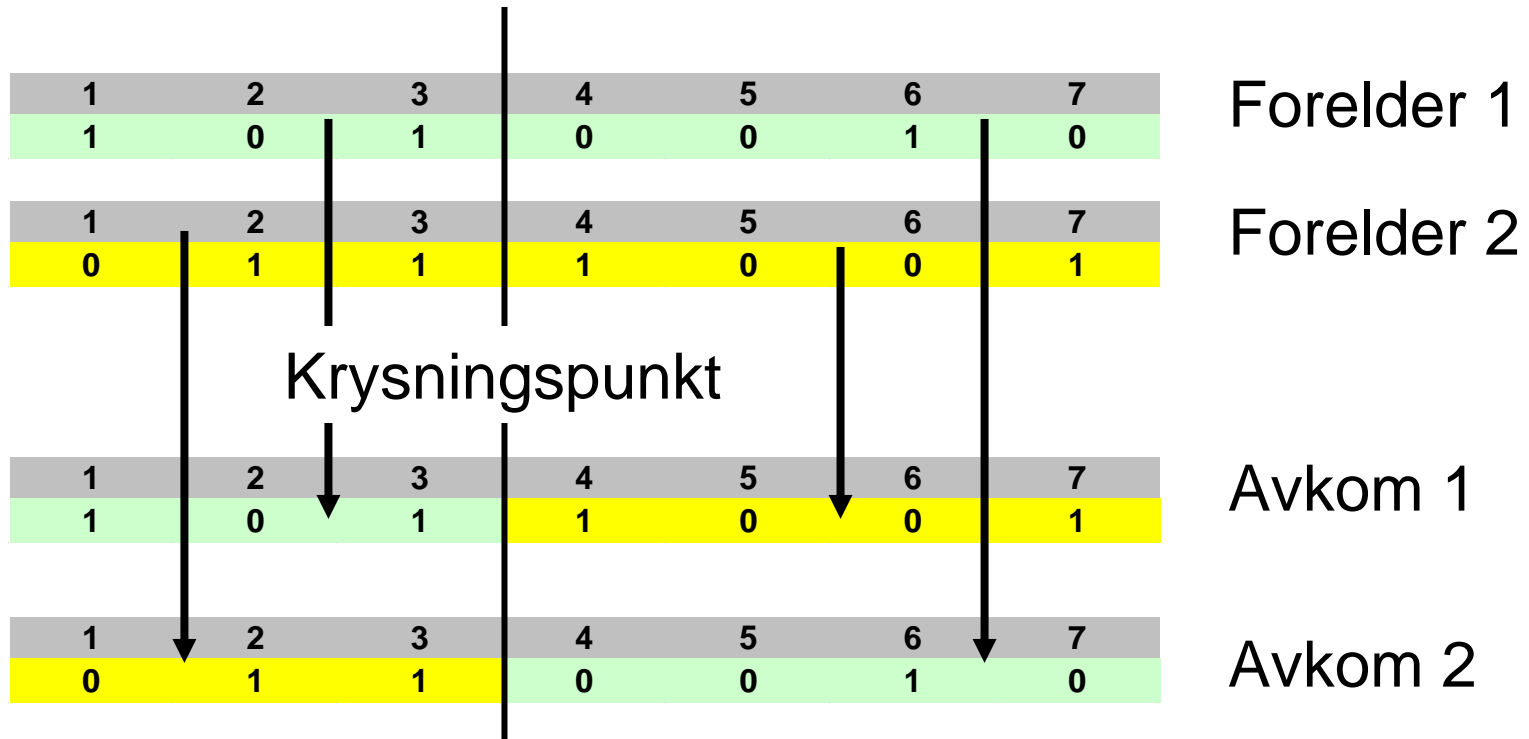
1	2	3	4	5	6	7
1	0	1	0	0	1	0



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

GA - Klassisk kryssning (1-punkt)

- En forelder velges basert på fitness
- Den andre forelder velges tilfeldig
- Tilfeldig valg av krysningspunkt



GA – klassisk kryssning forts.

- Vilkårlig individ i populasjonen byttes ut med ett av de to avkom
- Reproduksjon så lenge du orker
- Kan betraktes som sekvens av nesten like populasjoner
- Alternativt:
 - en av foreldrene velges strengt etter fitness
 - kryssning skjer inntil M avkom er dannet
 - den nye populasjonen består av avkommet
- Mange andre muligheter ...
- Grunnleggende GA med klassisk kryssning og mutasjon virker ofte bra

GA – standard reproduksjonsplan

- Fast populasjonsstørrelse
- Standard krysning
 - en forelder velges tilfeldig etter fitness
 - den andre velges tilfeldig
 - tilfeldig krysningspunkt
 - et tilfeldig individ byttes ut med ett av avkommene
- Mutasjon
 - en viss sannsynlighet for at individene muteres
 - tilfeldig hvilket gen som muteres
 - standard: mutasjon av avkom

Teoretisk analyse av GA (Holland)

- Skjema: delmengder av lignende kromosomer
- samme alleler i visse loki

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

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

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

- Et gitt kromosom er med i mange skjema
- Hvor mange?

Skjema - fitness

- Hver gang vi evaluerer fitness til et kromosom samles informasjon om gjennomsnittlig fitness til hvert av skjemaene
- Teoretisk: Populasjon kan inneholde ($M 2^n$) skjema
- I praksis: overlapp

GA – Iboende parallelitet

- En rekke skjema evalueres i parallell
- Hvor mange?
- Under rimelige antakelser: $O(M^3)$
- Ved anvendelse av genetiske operasjoner vil de skjema som representeres i populasjonen øke eller minke i forhold til relativ fitness
- Skjemateoremet

Lengde og orden for skjema

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

- Lengde: *Avstanden* mellom første og siste definerte posisjon
- Orden: Antall definerte posisjoner
- Eksempel:
 - Lengde: 2
 - Orden: 2

Fitness-ratio

Forholdet mellom gjennomsnittlig fitness av et skjema $e(S)$ og gjennomsnittlig fitness til populasjonen P

$$e(S) = \frac{\frac{\sum_{s \in S} f(s)}{N(S)}}{\frac{\sum_{s \in P} f(s)}{M}}$$

Lemma 1

Under en reproduksjonsplan der en forelder selekteres ut fra fitness, er det forventede antall instanser av et skjema S i generasjon $t+1$

$$E(S,t+1)=e(S,t)N(S,t)$$

der $e(S,t)$ er fitness-forholdet for skjema S og $N(S,t)$ er antall instanser av S i generasjon t

Lemma 2

Hvis kryssning anvendes i generasjon t med sannsynlighet P_c for et skjema S av lengde $l(S)$ så er sannsynligheten $P(S, t+1)$ for at S vil representeres i generasjon $t+1$ begrenset av:

$$P(S, t+1) \geq 1 - P_c \frac{l(S)}{n-1} (1 - P(S, t))$$

Sannsynlighet for at S blir ødelagt ved kryssningen

Sannsynlighet for at den andre forelderen ikke er av skjema S

Lemma 3

Hvis mutasjon anvendes i generasjon t , med sannsynlighet P_m for at et vilkårlig bit muteres, på et skjema S med orden $k(S)$ så er sannsynligheten for at S vil representeres i populasjonen ved tid $t+1$ være begrenset nedentil:

$$P(S, t+1) = (1 - P_m)^{k(S)} \geq 1 - P_m k(S)$$

Sannsynlighet
for at S
overlever
mutasjon

Skjemateoremet

I GA med standard reproduksjonsplan der sannsynlighet for krysning og mutasjon er P_c og P_m , respektive, og skjema S med orden $k(S)$ og lengde $l(S)$ har fitness-forhold $e(S,t)$ i generasjon t , så er forventet antall representanter for skjema S i generasjon $t+1$ begrenset av:

$$E(S, t+1) \geq \left[1 - P_c \frac{l(S)}{n-1} (1 - P(S, t)) - P_m k(S) \right] e(S, t) N(S, t)$$

S overlever
krysning og
mutasjon

Korollar

Representasjonen av skjema S vil i gjennomsnitt øke dersom

$$e(S, t) \geq 1 + P_c \frac{l(S)}{n-1} + P_m k(S)$$

Korte, lavordens skjema vil øke sin representasjon forutsatt at deres fitness-ratio er litt større enn 1, mens lengre, høyere-orden skjema må jobbe hardere for å overleve

GA – Byggeblokkhypotesen

- Goldberg (1989)
- Korte, lavordens skjema kombineres til bedre og bedre løsninger

Senere utvikling av teori

- Skjemateorem for
 - uniformt valg av foreldre i kryssning
 - valg av begge foreldre basert på fitness
- Eksakt uttrykk i skjemateoremet
- Analyse ved bruk av Walsh-funksjoner (fra signalanalyse)
- generalisering av skjema – form
 - design av ønskete operatorer

GA- Utvidelser og modifikasjoner

- mange variasjonsmuligheter
- mye litteratur, kaotisk
- uklar terminologi
- modifikasjoner
 - populasjon
 - enkoding
 - operatorer
 - hybridisering, parallellisering

GA – Evaluering av ytelse

- 3 viktige mål
 - basert på objektiv $f(x)$
 - indeks t ("tid") angir når løsning er generert
- Beste løsning hittil $f^*(x_t)$

- On-line
$$f^{online}(T) = \frac{\sum_{t=1}^T f(x_t)}{T}$$

- Off-line
$$f^{offline}(T) = \frac{\sum_{t=1}^T f^*(x_t)}{T}$$

GA – Utvidelser

- Populasjonsstørrelse

- Små populasjoner - underdekning
- Store populasjoner – beregningsmessig krevende
- Optimal størrelse øker eksponensielt med strenglengden i binære enkodinger
- Ofte kan størrelser på 30 virke bra
- mellom n og $2n$ (Alander)

GA – Utvidelser

- Initiell populasjon

- vanlig: tilfeldig trukne strenger
- alternativ: "såkorn": gode løsninger
 - raskere konvergens
 - prematur konvergens

GA – Utvidelser

- Seleksjonsmekanismer

- Generasjonsgap/overlapp
- Andel av populasjonen velges til reproduksjon
- Avkommet erstatter tilfeldig trukne individer
- Tvilstomt om dette virker bedre
- Inkrementell seleksjon virker bra i flere anvendelser

- Duplisering bør unngås
- Usikkert om beste løsning hittil overlever
- Elitisme
- Selektiv død

GA – Utvidelser

- Fitness

- Objektivfunksjonen sjelden egnet
- Naivt mål gir konvergens til like individer
- For tidlig konvergens
- Skalering
 - begrenset konkurranse i tidlige generasjoner
 - økt konkurranse etter hvert

$$f(x) = \alpha g(x) + \beta$$

GA – Utvidelser

- Fitness/Seleksjon

- Bruke rangering i stedet for objektiv
- Turneringsseleksjon
 - tilfeldig utvalg av grupper
 - beste i gruppa går videre til forplantning

GA – Utvidelser

- Operatører

- Mutasjon – bevarer diversitet
 - Valg av mutasjonsrate ikke kritisk
- Krysning – ofte effektivt
 - sent i søket: krysning gir ingen effekt
 - selektivt valg av krysningspunkt
- Fler-punkts krysning
 - 2-punkts har gitt bedre ytelse
 - 8-punkts krysning har gitt best resultater

GA – Utvidelser

- Generalisert kryssning

- Bit-streng sier hvor genene skal hentes fra

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

P1

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

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

P0

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

Avkom

GA – Utvidelser

- Invertering

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



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

GA

– Enkoding og representasjon

- Ikke-binær enkoding
- Sekvensrepresentasjon
- PMX (Partially Mapped Crossover)
 - 2-punkts krysning

P1	1	2	3	4	5	6	7	
	2	1	3	4	5	6	7	3 ↔ 1
P2	1	2	3	4	5	6	7	4 ↔ 2
	4	3	1	2	5	7	6	5 ↔ 5
O1	1	2	3	4	5	6	7	
	3	4	2	1	5	6	7	
O2	1	2	3	4	5	6	7	
	1	2	4	3	5	7	6	

GA - Hybridisering

- GAs styrke og svakhet:
domeneuavhengighet
- Hybridisering
 - Såing, gode individer i initialpopulasjon
 - Lokalsøk på individer
 - Kombinasjon med andre metaheuristikker

GA - parallellisering

- Fitness-evaluering
- Finkornet parallellisering
 - hver løsning sin prosessor
 - asynkron parallellisering
- Grovkornet parallellisering
 - del-populasjoner

Anvendelser

- AI og OR
- TSP og VRP
- Sekvensering og tidsplanlegging
- Graph coloring
- Ryggsekkproblemer, bin-packing
- Mengdedekning, mengdepartisjonering

GA - Oppsummering

- Viktige karakteristika
 - populasjon av løsninger
 - domeneuavhengighet – enkoding
 - mangel på utnyttelse av struktur
 - iboende parallellitet – skjema, vokabular
 - robusthet
 - gode mekanismer for intensifisering
 - mangler noe diversifisering
- Hybrid med lokalsøk: Memetic algorithms

Memetisk - meme

- Introdusert av Dawkins (1976): "The Selfish Gene"
- Analogt til gen i kulturell evolusjon

"Examples of memes are tunes, ideas, catch-phrases, clothes fashions, ways of making pots or building arches."

Nye trender

GA- Evolusjonsmetoder

- Mer direkte representasjon av løsninger
 - mer naturlig enkoding av løsning
 - spesielle krysnings- og mutasjonsoperatorer
- Mer intensivering
 - lokalsøk (spesielle mutasjonsoperatorer) til lokalt optimum
 - Memetiske algoritmer

Maurkoloni-optimering (Ant Colony Optimization – ACO)

- Presentasjon Marc Reimann, Univ. Wien
- Populasjonsbasert metode
- Inspirert av maurs ”læring” og kommunikasjon ved bruk av luktstoffer (feromoner)
- Flere ”agenter” (maur) konstruerer løsninger ved:
 - konstruksjonsheuristikker
 - tilfeldighet
 - informasjon fra andre agenter

Maur (forts.)

- Sosiale insekter
- Selvorganiserende kollektiv adferd
- Kompleks dynamikk fremstår fra enkle individer
 - „emergent behaviour“
 - „swarm intelligence“
- Interessante sosiale fenomen
 - arbeidsdeling
 - allokering av oppgaver
 - organisering av „gravsteder“

Maur (forts.)

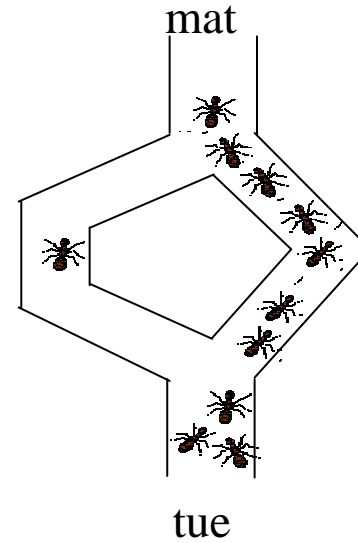
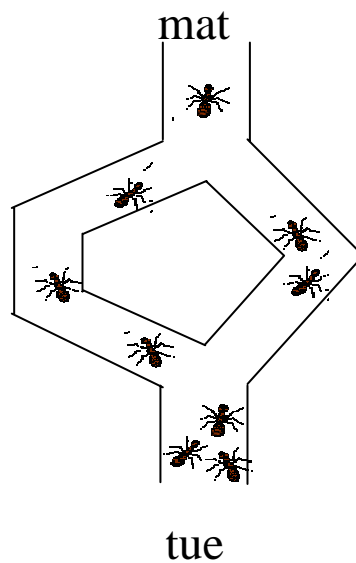
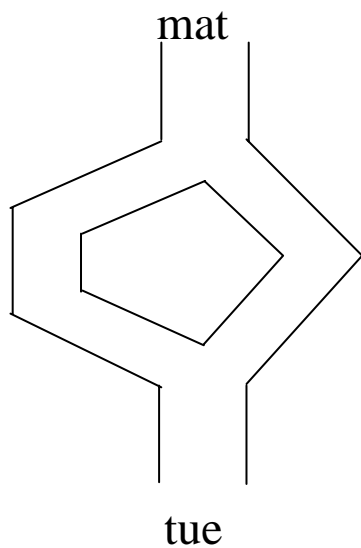
- Flere interessante fenomener
 - Samarbeid om transport
 - **Fôrfinning**
- Fôrfinning (e.g. *Lasius niger*)
 - Hver maur legger spor etter seg (feromoner)
 - Maur følger andres spor

Maur og stier

- Stidannelse og stifølging ved enkle lokale regler
- Maur legger fra seg feromoner for å kunne orientere seg (og andre) senere
- Maur følger tidligere stier ifølge feromonintensitet
- Tydelige stier har en tendens til å bli enda tydeligere, forsterkning (reinforcement)
- Feromon fordamper, lite brukte stier har tendens til å forsvinne

Maur - Stidannelse og stifølging

- „Binær bro“-eksperimentet
- Transportoptimering



Maurkolonioptimering for DOP

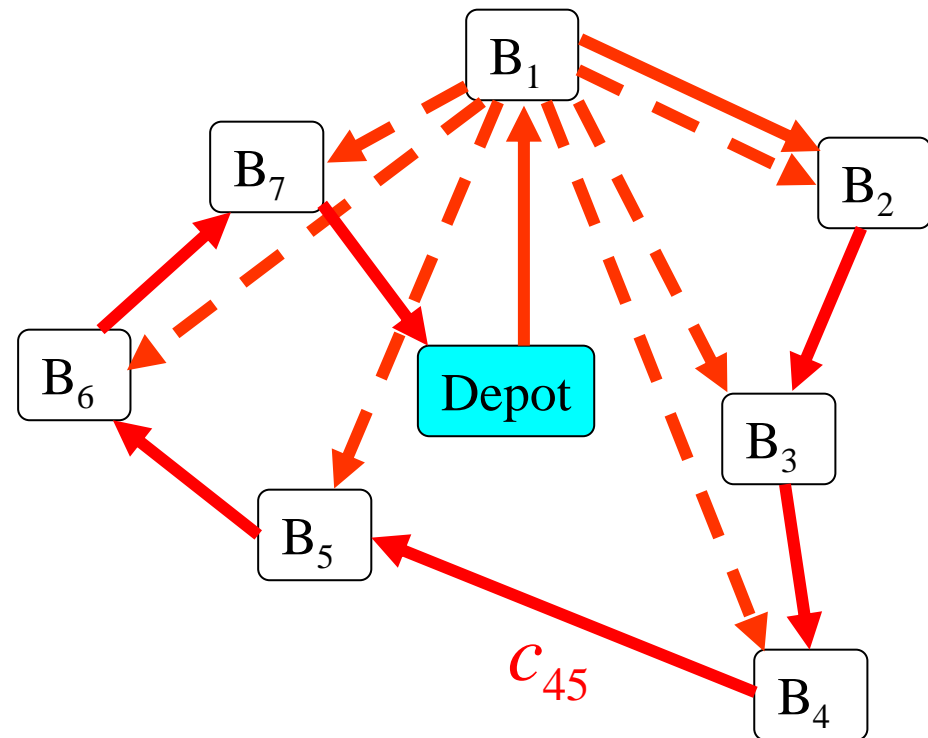
- Ant Colony Optimization (ACO)

- Introdusert av Dorigo, Maniezzo & Colorni 1992
- Populasjonsbasert metaheuristikk
- Hver „maur“ i populasjonen **konstruerer** en løsning
- Når alle er ferdige oppdateres et **minne** (kunstig feromon)
- Løsningskonstruksjon og minneoppdatering repeteres inntil stoppkriterium er oppfylt

Maursystem for DOP

- Ant System (AS)

- Første ACO-metode (Dorigo 1992)
- Løsningskonstruksjon (for hver maur)
- Beslutninger tas probabilistisk
- e.g.: TSP (første anvendelse)
- Konstruksjonsmekanisme
 - „Nærmeste nabo“
 - tilfeldighet
- Tilfeldig valgt startby



Maursystem - Konstruksjonsmekanisme

Lokale beslutninger under konstruksjon baseres på:

- en konstruktiv heuristisk (grådig) regel
- et lokalt kvalitetskriterium σ (*a priori heuristisk informasjon*)
- et adaptivt minne (et dynamisk, globalt kvalitetskriterium τ)
- tilfeldighet

$$p_{ij} = \begin{cases} \frac{[\sigma_{ij}]^\beta \cdot [\tau_{ij}]^\alpha}{\sum_{h \in \Omega_i} [\sigma_{ih}]^\beta \cdot [\tau_{ih}]^\alpha} & \text{hvis } h \in \Omega_i \quad \sigma_{ij} = \frac{1}{c_{ij}} \text{ for TSP} \\ 0 & \text{ellers} \end{cases}$$

der Ω_i er mengden av tillatte alternativer

Maursystem (forts.)

Oppdatering av dynamisk informasjon (feromon)

TSP-eksempelet: Alle kanter oppdateres etter hver iterasjon, for hver maur

$$\tau_{ij} = (1 - \rho) \cdot \tau_{ij} + \sum_{m \in M} \Delta \tau_{ij}^{(m)}, \quad 0 < \rho \leq 1$$

$(1 - \rho) \cdot \tau_{ij}$ er feromon på kant ij etter fordampning

$\sum_{m \in M} \Delta \tau_{ij}^m$ er forsterkningen av kant ij (feromonutslipp, maur m)

$$\Delta \tau_{ij}^m = \begin{cases} \frac{1}{f(s^m)}, & \text{hvis } (i, j) \in s^m \\ 0 & \text{ellers} \end{cases}$$

Maursystem (forts.)

- Virker bare sånn passe ...
- Biologisk analogi delvis forlatt
- 3 videreutviklinger
 - Ant Colony System – (ACS, Maurkolonioptimering)
 - Max-Min Ant System
 - Rank Based Ant System
- Alle inkluderer lokalsøk for å forbedre løsning
- ACS
 - Modifisert global og lokal oppdatering
 - Globalt og lokalt feromon
 - Elitisme: kun „beste maur“ får oppdatere globalt feromon
 - Endret randomisering, sannsynlighet i to trinn
 - velge grådig
 - velge probabilistisk blant alle tillatte alternativer

Leksjon 6 - Oppsummering

- Genetiske algoritmer (GA)
 - inspirert av biologisk evolusjon
 - populasjon av løsninger (kromosomer)
 - rekombinering ved kryssning
 - mutasjon, seleksjon
- Øvrige populasjonsbaserte metoder
 - Memetiske algoritmer ("GA+lokalsøk")
 - Maurkolonioptimering

Leksjon 7 - Oversikt

- Kategorisering av metaheuristikker
- Konstruktive heuristikker
- Multi-start baserte metaheuristikker
 - Variabelt nabolagssøk (VND/VNS)
 - Grådig Adaptivt Randomisert Søk (Greedy Randomized Adaptive Search, GRASP)
 - Iterert lokalsøk (Iterated Local Search, ILS)
- GUT (Grand Unifying Theory)?
- Hyperheuristikker