

INFO 180 – Metodar i kunstig intelligens

Seminaroppgåve 4 – 16.sept – 20. sept 2024

1. Program for grådig søk og simulert herding

I denne oppgåva skal de jobbe med ei ny formulering av invitasjonsproblemet.

På Mitt UiB under kursets filkatalog fins ei zip-fil som inneheld Java-filer (Oppg4.zip) for invitasjonsproblemet (i ny formulering). Det er 4 filer der. **opt_candidate.py** representerer alle mulige gjester til eit selskap. **opt_invitation_problem.py** inneheld spesifikasjonen av sjølve problemet med vilkår. **opt_invitation_assignment.py** representerer ei mulig løysing på invitasjonsproblemet, og **opt_invitation_solver.py** inneheld algoritmane som løyser problemet.

Problemet er formulert slik at ein ønskjer å invitere til eit optimalt selskap der ein har 30 kandidatar til å vere gjester. Kvar potensiell gjest har ein score for kor godt ein likar personen. I tillegg er det lagt in 4 vilkår (sjå programkoden korleis dette er gjort)

1. Det er best med om lag 12 gjester. 12 er topp, men litt avvik kan og vere ok.
2. Det er best med om lag like mange kvinner som menn, men litt avvik kan vere ok.
3. Anne mistrivast sterkt i selskap med Ola
4. Rune, Helge, Ivar og Lars har ein tendens til å ta kontroll over musikken og vel musikk som få andre likar. To av dei er ok, men tre eller fire går ikkje så bra.

`opt_invitation_solver` er køyrbar, så de kan prøve den med ein gong. Slik `main`-funksjonen er no køyrer den først ein grådig **maksimerings**-algoritme, og så ein simulert herding-algoritme (sjå læreboka og forelesingsnotatar til kurset). Parametrane til simulert herding er gjevne på toppen av `opt_invitation_solver` og viser til ein initiell temperatur og nedkjølingsfaktor.

Oppgåve 1.a: du skal programmere to nye mjuke vilkår (legg til i `opt_invitation_problem.py`). Desse vilkåra er

5. Sofie og Tom er einslege og du syns dei passar godt saman, så det ville vere fint om dei begge kom slik at du fikk fungere som «Kirsten Giftekniv».
6. Du ønskjer å gjennomføre ein selskapsleik der det er best at deltakarane har ulike førebokstavar, så det er fint om at alle dei inviterte gjestene har ulike førebokstavar.

Oppgåve 2.b: Du skal modifisere `main`-funksjonen og køyre med ulike verdier for `COOLING_FACTOR` i `opt_invitation_solver`. Programmet skal køyre problemet 100 gonger og legg saman scorane for kvar av

- a) Den grådige algoritmen
- b) Simulert herding med `COOLING_FACTOR` lik 0.8
- c) Simulert herding med `COOLING_FACTOR` lik 0.95
- d) Simulert herding med `COOLING_FACTOR` lik 0.99

Kva for algoritme av dei 4 er best?

2. Populasjonsbaserte algoritmar

Anta du har eit lite optimeringsproblem med 6 variable $X = [x_1, x_2, x_3, x_4, x_5, x_6]$ og domene $\{1, 2, \dots, 10\}$ for alle variable. Objektivfunksjonen som skal **minimerast** er

$$\text{obj}(X) = \frac{x_1 x_2 - x_3 x_4 + x_4 x_6}{x_5 x_6 + 3x_1 x_6 - 3x_2}$$

Du har starta eit enkelt **strålesøk** med 6 individ og fikk følgande startpopulasjon frå bruk av slumptalsgeneratoren.

Individ nr	x_1	x_2	x_3	x_4	x_5	x_6
1	10	1	8	9	5	1
2	8	7	5	2	9	5
3	4	9	10	8	6	9
4	9	7	4	3	5	2
5	5	2	3	4	3	1
6	10	9	10	4	3	5

Du vil prøve strålesøk der 3 individ går vidare i kvar runde. Kva for 3 individ vert det i dette tilfellet?

For kvart «overlevande» individ lag to nye individ ved å auke x_4 med 1 for det første barnet, og minke x_3 med 1 for det andre barnet. Ingen av dei gamle individa overlever. Kva vert den nye populasjonen?

Du prøver også **genetiske algoritmar**.

Du brukar eitt-punkts-kryssing, og lar dei to beste individa «pare» seg og få to nye barn med kryssing mellom x_1 og x_5 . Korleis ser dei nye individa ut? Kva er deira verdi på objektiv-funksjonen. Du muterer det «svakaste» av desse to borna ved å redusere x_1 med 1. Kva vert no objektiv-funksjonen for dette bornet?

Du bør kunne løyse desse problema med å jobbe i eit rekneark, så du treng ikkje programmere noko.