Designovervejelser:

Selektion

Roulette metoden

Man kan forestille sig at hvert individ er tildelt et stykke på en roulette og størrelsen på stykket er proportional med individets fitness. Roulette bliver spinnet n antal gange, det vil tage for at vælge forældrene til den næste generation. Under hvert spin bliver individet under roulettens markør valgt til, at være en del af en gruppe af forældre til den næste generation. En kandidat kan godt blive valgt til, at være forældre flere gange, dette er okay, da vi vælge forældre til næste generation og ikke selve individerne i generationen. Formålet med denne metode er, at få valgt de forældre med den største fitness til næste generation, da de har større sandsynlig for, at skabe individer større fitness. Problemet med denne metode er dog, at den genetiske algoritme hurtigt vil stå fast i den ene del af fitness rummet, da det er muligt at vælge den samme forældre flere gange og derved kan der blive skabt en meget ens artet population, som gør at der kun vil blive udforsket et bestemt område af rummet i stedet for at udforske hele rummet.

Rank metoden

Metoden har ligheder med roulette metoden med i stedet for at proportionel med den absolute fitness er den proportionel med den relative fitness. Der er altså ligegyldigt om den fitteste har 10 gange højere fitness en den næste i rangen eller om den har 0.0001% højere fitness. I begge cases vil sandsynligheden for den den fitteste være den samme.

Tournament metoden

2 tilfældige individer bliver valgt fra populationen. Man generer en tilfældig værdi fra 0-1 for sammenligner den med valgte sandsynligheds værdi. Hvis værdien er mindre eller lige med sandsynlighedsværdien bliver det individ med højst fitness valgt ellers bliver individet med den lavere fitness valgt. Sandsynlighedsværdien bliver altid sat højere end 0.5 for at favorisere individet med den højeste fitness.

”Seach space” referer til en gruppe af kandidat løsning til et problem, hvor der er en ”distance” i mellem kandidaterne. For eksempel lad os tage vigtigt problem indenfor bioengeering, hvordan man designer et protein. Antaget at man vil søge efter et protein som er en sekvens af aminosyre som kan blive brugt til at bekæmpe en virus. ”Seach space” vil være en kollektion af alle mulige proteiner. Dette vil give os uendelige mange muligheder derfor begrænser vi længden af proteinet til længden 50 som stadig vil være et stort ”seach space” siden der er 20 mulige aminosyre i hver position i proteinet. Hvis vi repræsenter aminosyrerne i form af alfabetet vil et muligt protein se ledes ud.

ASDKEGHB…. Vi definer distance mellem proteinerne som forskellen i alfabetet på den tilsvarende position i et andet protein fx ASDKEGHB og BSDKEGHB er distance 1 og distance mellem ASDKEGHB og GCCHAKAA er 8.