

Blatt5

Henri

30.5.2020

```
#f - die zu evaluierende Funktion
#x0 - die Startindividuen, anzugeben als Matrix, wobei eine Zeile für ein Individuum steht
#lower - die untere Grenze
#upper - die obere Grenze
#evals - die Anzahl der Evaluationen (Budget)
#mu - Anzahl der Eltern pro Generation
#lambda - Anzahl der Kinder pro Generation
#selektionsmethode - die Methode mit der die neue Generation bestimmt wird, zu wählen zwischen "plus" und "minus"
#bei komma lambda >= mu notwendig
#
#

f <- function(x){
  ergebnis = 1.5*x[1]**2+x[2]**2+21*sin(x[1])*cos(x[2])+0.5*(abs(x[1])**2+abs(x[2])**2)
  return(ergebnis)
}

g <- function(x){
  ergebnis = x[1]**3-x[2]**3+x[2]**2+1000*cos(x[1])*sin(x[2])
  return(ergebnis)
}

h <- function(x){
  ergebnis = x[1]**2+x[2]**2+x[3]**2
  return(ergebnis)
}

ea <- function(f, x0, lower = -10, upper = 10, evals = 200, mu = 1, lambda = 10,
               selektionsmethode = "plus", sigma = 0.1, tau = 0.1){

  sol = rep(0, each = mu)
  for(p in 1:mu){
    sol[p] = f(x0)
  }
  #füge ergebnisse an die elternmatrix an
  parent = cbind(x0, sol)

  #erzeuge hilfsmatrix für kinder
  children = matrix(rep(0, each = (ncol(parent)*lambda)), nrow = lambda)

  if(mu > 1){
```

```

#bei zwei oder mehr Eltern, nutze intermediate crossover und mutation

for(i in 1:evals){

  #generiere offspring durch crossover
  offspring = crossover(parent, mu, lambda)

  #mutiere offspring zu kindern
  for(o in 1:nrow(offspring)){
    children[o,] = mutate(matrix(offspring[o,], nrow = 1), 1, sigma)
  }

  #Anpassung der Schrittweitensteuerung
  sigma = sigma*exp(rnorm(1, mean = 0, sd = tau**2))

  #anpassen der Werte auf lower/upper
  for(x in children){
    if(x > upper) {
      x = upper
    }
    else if(x < lower){
      x = lower
    }
  }

  #berechnen der Fitness
  for(i in 1:lambda){
    children[i, ncol(children)] = f(children[i,])
  }

  #auswahl der Individuen für die nächste Generation
  if(selektionsmethode == "komma"){
    population = rbind(parent, children)
    children.sorted = children[order(children[,ncol(children)]),]
    for(v in 1:mu){
      parent[v,] = children.sorted[v,]
    }
  }
  else{
    population = rbind(parent, children)
    population.sorted = population[order(population[,ncol(population)]),]
    for(v in 1:mu){
      parent[v,] = population.sorted[v,]
    }
  }
}

}
else if(mu == 1){
  #bei einem elternteil nutze nur mutation

```

```

for(i in 1:evals){
  children = mutate(parent, lambda, sigma)
  #Anpassung der Schrittweitensteuerung
  sigma = sigma*exp(rnorm(1, mean = 0, sd = tau**2))

  #anpassen der Werte auf lower/upper
  for(x in children){
    if(x > upper) {
      x = upper
    }
    else if(x < lower){
      x = lower
    }
  }

  #berechnen der Fitness
  for(i in 1:lambda){
    children[i, ncol(children)] = f(children[i,])
  }

  #auswahl der Individuen für die nächste Generation
  if(selektionsmethode == "komma"){
    population = rbind(parent, children)
    children.sorted = children[order(children[,ncol(children)]),]
    parent[1,] = children.sorted[1,]
  }
  else{
    population = rbind(parent, children)
    population.sorted = population[order(population[,ncol(population)]),]
    parent[1,] = population.sorted[1,]
  }
}
return(population)
}

#parent - zu mutierendes Individuum
#lambda - Anzahl der Nachkommen
#sigma - Wert zur Schrittweitensteuerung

mutate <- function(parent, lambda, sigma){

  #erstelle child matrix
  children = matrix(rep(parent, each=lambda), nrow = lambda)

  #Mutiere die Eltern lambda-mal
  for(i in 1:lambda){
    for(x in 1:(ncol(children)-1)){#letzte Spalte enthält das ergebniss der funktion
      children[i, x] = rnorm(1, mean = parent[1,x], sd = sigma)
    }
  }
}

```

```

    }
  }
  return(children)
}

crossover <- function(parent, mu, lambda){

  children = matrix(rep(0, each = (ncol(parent)*lambda)), nrow=lambda)

  for(i in 1:lambda){
    w = runif(mu, min = 0, max = 1)
    w = w/sum(w)
    for(x in 1:mu){
      children[i,] = children[i,]+(w[x]*parent[x,])
    }
  }

  return(children)
}

```

Der EA nutzt Mutation mit Schrittweitensteuerung nach Schwefel und nutzt als Standardwerte eine untere/obere Grenze von -10/10, $\mu = 1$, $\lambda = 10$, $\sigma = 0.1$, $\tau = 0.1$ und als Selektionsmethode $\mu + \lambda$, da sie diese Komponenten und Defaultwerte auf dem vierten Übungsblatt für ähnliche Funktionen im Bereich $[-10;10]$ bewährt haben. Falls zwei oder mehr Eltern pro Generation vorhanden sind, wird außerdem das intermediate crossover genutzt, damit alle Eltern gleichmäßig einfließen.