

BACHELOROPPGAVE

TITTEL Effektivisering av optimaliseringsalgoritmer på VVS-tekniske problemer?	DATO 22.05.2019
	ANTALL SIDER / ANTALL VEDLEGG 49 / 25
VEILEDER Eirik Grønmo, Audun Johansen	VEILEDER Marius Lysebo

ID MED	KONTAKTPERSON Guillermo Duran Moro
--------	---------------------------------------

finne ut hvordan «Particle swarm optimization»-algoritmen presterer på VVS-tekniske problemer. i programmeringsspråket Python og kvalitetssjekkes ved hjelp av flere kjente standardfunksjoner. Videre løse et problem innenfor varmetransport. I siste del av oppgaven løser vi et problem presentert av vår håndball Norge AS. Her undersøkes optimal plassering av et antall rør med ulik radius i en sjakt. Tilslutt besvarer vi spørsmål om hvordan algoritmen presterer på VVS-tekniske problemer.
--

erkoet med denne oppgaven har vi jobbet systematisk fra første dag med å bygge opp
le skript, samtidig som vi har tilegnet oss kunnskap med tilhørende teori. Prosessen
vende, det ble brukt mye tid på å teste og feilsøke skriptet. Ingen i gruppen hadde
er kompleks programmering fra tidligere, og vi har derfor hatt en bratt læringskurve
med oppgaven.

veileder Marius Lysebo for veiledning og diskusjoner utover i de sene nattetimer
e mesteret. Det gjelder spesielt for hjelp til å løse feil og problemer godt gjemt i
vordan strukturere arbeidet. Vi vil også takke Emil Nygård, Bendik Tilley Solhaug og
n Moro hos Rambøll Norge for hjelp til å koble oppgaven opp mot mulig praktisk
enfor VVS.

problem innenfor varmetransport som omhandler varmeoverføring gjennom en vegg, et på et problem som dreier seg om plassering av rør i føringsveier tildelt av vår samarbeidspartner Rambøll Norge AS

Vår versjon av algoritmen ved hjelp av programmeringsspråket Python. Dette krever en lang utviklingsprosess og mye programmering. Algoritmen blir så validert ved å løse flere kjente problemer. Etter å ha testet algoritmen flere ganger og gjort små endringer, er vi fornøyd med at skriptet presterer.

Med hjelp av skriptet et problem innenfor varmetransport. Vi studerer en vegg med et rørnettverk, betong og et varmeelement. Algoritmen skal plassere varmeelementet i en slik posisjon at den oppnår høyest mulig overflatetemperatur på *begge* sider av veggen. Resultatet av vår optimering bekreftes med analytiske beregninger.

Oppgaven presenterer og løser vi et problem fra vår samarbeidspartner, Rambøll Norge AS. Vi undersøker mulighetene for å optimalisere plasseringen av et antall rør med ulik radius i en kanal. I tillegg er det optimale å plassere rørene så tett som mulig uten at de overlapper. Vi presenterer resultatet i kapittel 4 ved å plassere rørene ved hjelp av en videreutviklet variant av PSO.

De nevnte resultatene, mener vi at GCPSO er et godt verktøy til å optimalisere VVS-prosjekter.

on tllPSO.....	10
matisk optima lisering	11
av nettverksstruktur	12
ammet.....	13
an beveger egentlig en partikkel seg?.....	17
mmelse av v , $c1$ og $c2$	18
jonstester	19
ekriterier i algoritmen	20
Antall iterasjoner er nådd.....	20
ngen forbedring observert	20
Avstand mellom partiklene	21
valg av stoppekriterium	21
etssikring av skript med standardfunksjoner	21
Ackleys funksjon	22
Sfærefunksjon.....	23
casoms funksjon	24
McCormicks funksjon	25
tat.....	26
av varmeelement i vegg	27
mentering	28
etinger på hver side	29

r	48
.....	49
gg 1	49
gg 2	53
gg 3	60
gg 4	62
gg 5	68

20 partikler som konvergerer mot optimal løsning til funksjonen x^2 . På x-aksen ser vi
 og på y-aksen ser vi antall iterasjoner og hver farge representerer en partikkel. Hver
 representerer ferdien til en partikkel på vei mot funksjonens minimumspunkt 18

ikt over forsøk som viser presisjonen ved forskjellige verdier av v , $c1$ og $c2$. På y-aksen
 $n = \log_{10}(\epsilon)$ med $\epsilon = y_n - f_{min}$ der y_n er funksjonens verdi etter n-iterasjoner og
 onens kjente globale minimumsverdi. En høyere verdi på y-aksen representerer en
 n. Forsøk nr. 1 har den høyeste presisjonen. 19

ned 1000 iterasjoner. På y-aksen har vi $P_{presisjon} = \log_{10}(\epsilon)$ med $\epsilon = y_n - f_{min}$
 onens verdi etter n-iterasjoner og f_{min} er funksjonens globale minimumsverdi. En
 y-aksen representerer en høyere presisjon. Når antall ukjente øker, reduseres
 20

enhengen mellom tid, presisjon og antall iterasjoner (fra 100 til 10000). Presisjonen
 en til høyre og er absoluttverdien av den Briggske logaritmen til feilmarginen. 21

s funksjon 22

funksjon av to variabler 23

s funksjon 24

rmick funksjon 25

n viser snitt av en vegg delt i tre sjikt. Sjikt I og III grenser mot henholdsvis inn- og
 olig. Sjikt II er varmelementet. Elementet vil i virkeligheten være tynn som en
 e. Det er varmelementets plassering i veggen (langs x) som skal bestemmes. Veggen
 g er i scenario 2 bygd opp som følger: sjikt 1 er treverk, sjikt II er aluminium og sjikt III
 e i figuren viser varmestrømmen fra varmelementet og ut til omgivelsene. 27

eraturprofil for $b = L/2$. Like betingelser på hver side av varmelementet. Blå farge
 urprofilen i sjikt 1. Rød farge viser temperaturprofilen i varmelementets sjikt. Gul
 peraturprofilen i sjikt 2. Her er $L = 0,3m$, $c = 0,05m$, $egen = 200W/m^3$, $k1 = k3 =$
 $2W/mK$, $h1 = h2 = 20W/m^2K$ og $T_{\infty,1} = T_{\infty,2} = 20^\circ C$. 30

eraturprofiler ved forskjellige plasseringer av varmelementet. Blå linje er
 ilen med varmelementet plassert ved $L/4$. Svart linje er temperaturprofilen med

er plassert så tett som mulig i en firkant	35
er plassert så tett som mulig i en firkant	36
er plassert så tett som mulig i en firkant	36
er sirkler tangerer eller overlapper hverandre får funksjonen en straff på 10000 slik at	
derer dette som en dårlig løsning	38
en viser kreftene som påvirker sirklene når $F = E_{\text{fastøring}} + E_{\text{straff}} + E_{\text{tiltrekning, y}}$	39
er plassert med maksimal avstand mellom dem i en firkant. Posisjonene er (0.46,	
4), (0.04, 0.46), (0.04, 0.04) med funksjonsverdien 12,89	41
er plassert så langt det lar seg gjøre mot figurens y-akse (venstre side). Sirklene er	
over i y-retning slik at avstanden mellom dem fremdeles er størst mulig.	
en er -3969,12	41
er som overlapper hverandre for å minimere funksjonsverdien som er -7772,81	42
er som har plassert seg så langt det lar seg gjøre mot figurens y-akse (venstre side)	
er. Her er det innført en drastisk økning i funksjonsverdien for hver sirkel som	
funksjonsverdien er -6237,61	42
er plassert så nærme som mulig y-aksen	42
er plassert så nærme som mulig y-aksen	42
er plassert så nærme som mulig y-aksen	43
er plassert så nærme som mulig y-aksen	43
er plassert så nærme som mulig y-aksen	43
er plassert så nærme som mulig y-aksen	43
er plassert så nærme som mulig y-aksen	44
er plassert så nærme som mulig y-aksen	44
er plassert så nærme som mulig y-aksen	44
er plassert så nærme som mulig y-aksen	44
er tiltrukket øvre venstre hjørne	45
er tiltrukket øvre høyre hjørne	45
er tiltrukket nedre venstre hjørne	45

algoritmen, heretter omtalt som
engang skal vi implementere og
men på noen kjente funksjoner.
forsikret oss om at algoritmen
testet den på et



Figur 1 Fugler som flyr i flokk (Foto: Manuel Presti/Science
Photo Library/Corbis)

problem, skal vi bruke den på et problem innenfor varme, ventilasjon og sanitær.

Instilling

er «Particle swarm optimization»-algoritmen på VVS-tekniske problemer?

entere en intelligent algoritme i programmeringsspråket Python som skal testes på
problemer i form av matematiske funksjoner. Deretter skal skriptet implementeres for
isk problem innenfor varme transport som har direkte anvendelse i VVS-bransjen. I
Rambøll har vi også utformet et praktisk VVS-teknisk problem. Problemet dreier seg
n plasserer et gitt antall med rør optimalt i en firkantet sjakt for at rørene skal oppta
al i sjakten.

ens oppbygning

es skriptet med standardfunksjoner. Som man kan se av resultatene, fungerer skriptet
g det finner minimumspunktene til de fire testfunksjonene med stor nøyaktighet.

vi fastslå plasseringen til et varmeelement i en vegg. Funksjonen vi skal optimalisere

Resultatet av forsøkene til Kennedy og Eberhart ble en algoritme som egner seg til
av funksjoner. Fugleflokken ønsker man å gjenspeile med en flokk partikler i et gitt
posisjon er bestemt av antall variable. Partiklene er betrakte masseløse objekter som

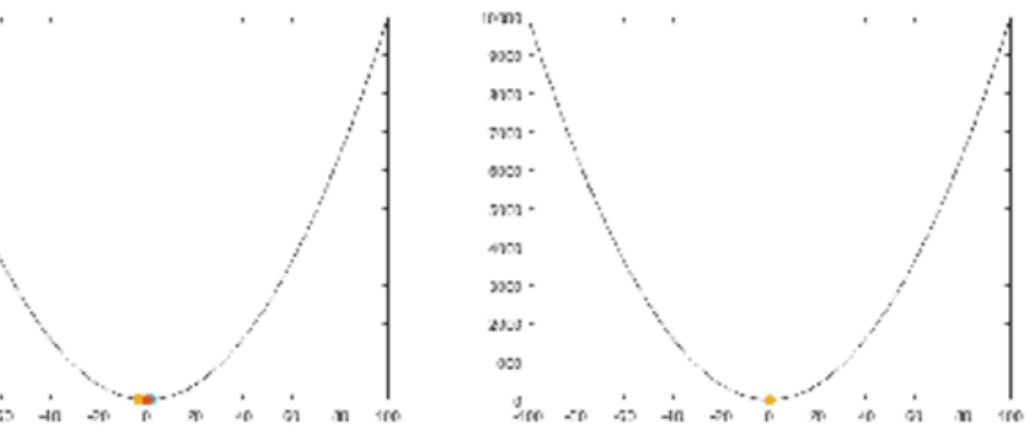
opererer ved at partiklene starter sin bevegelse fra tilfeldige posisjoner i et gitt område.
Startposisjonene til partiklene på eksempelfunksjonen x^2 . Denne funksjonen er benyttet
partiklenes bevegelsesmønster mot dens minimumspunkt. Området vi ser i figuren går
0 på x -aksen, hvor hver x -verdi har en tilhørende funksjonsverdi y . Algoritmen
funksjonsverdien ut ifra alle partiklenes x -verdi. Partiklene begynner så å bevege seg, og
else de gjør, evalueres funksjonen på nytt. Hver bevegelse kalles for en iterasjon.

kommuniserer med hverandre og trekkes mot den i gruppen som har den mest optimale
Figur 3 og Figur 4 viser situasjonen etter henholdsvis 5 og 8 iterasjoner.

resultatet etter 20 iterasjoner. Partiklene har plassert seg i beste posisjon. Alle
plassert i punkt $x = 0$. Bevegelsesmønsteret til partiklene forklares i figurene nedenfor.



Figur 3: 5 iterasjoner



Figur 5: 20 iterasjoner. Alle partiklene er samlet i samme punkt

Matisk optimalisering

Optimalisering handler om å finne maksimumspunkt eller minimumspunkt til en funksjon

å finne verdien til variabelen eller variablene til f som gir høyeste eller laveste

Et ekstremalpunkt i en funksjon f kan enten være lokalt, lavest eller høyest

i et begrenset område, eller globalt, lavest eller høyeste funksjonsverdi til hele

[p. 487].

algoritmen er implementert med stjernestruktur.

est generelle formen, hvor alle partiklene

med hverandre, som vist i Figur 6. At alle

muniserer med hverandre gjør at algoritmen

ere mot én løsning. Dette går på bekostning av søkeområdet [4, p. 301].

nettverksstrukturer er ringstruktur, hjulstruktur og Von Neumann-struktur.

r kommuniserer partiklene kun med sine n

kler. Ved å endre n forandres det sosiale

om partiklene. I tilfellet $n = 2$ ser det sosiale

om i Figur 7. Her kommuniserer hver partikkel kun

oer. Dette fører til at informasjon blir delt med en

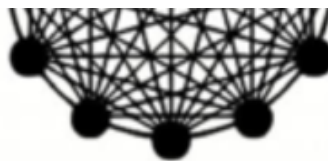
t gjennom nettverket av partikler. Hver partikkel

ot sin nabo med beste løsning. Fordelen med en

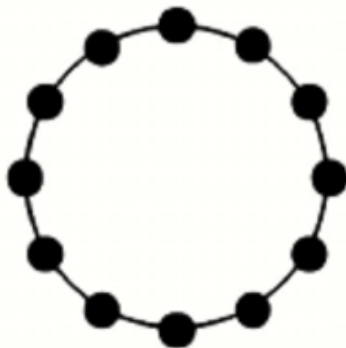
at søkeområdet blir dekket i en større grad enn

struktur. Denne strukturen er bedre egnet til å

nsjoner som har flere globale topp- eller bunnpunkt [4, p. 301].



Figur 6: Stjernestruktur



Figur 7: Ringstruktur

en presenteres de overordnede funksjonene i
sten av oppgaven bygger på. Dette krever en
programmeringsspråket Python. Hvis leseren
eg inn i dette, anbefaler vi nettsiden «Python tutorial» hos W3Schools.com [5].

```

    by as np
    e

    l
    s = 10
    s = 100
    = 1

    # Velocity
    # Cognitive component

```

```

- 1)) or 1))
np.row_stack((upper_boundary, lower_boundary))

```

er som begrenser partiklenes bevegelser opprettes

ersikrer vi oss om at partiklene ikke er utenfor angitte området i kodesnutt 3.

```

def boundaries(particle):
    boundaries = []
    for j in range(n_unknowns):
        if (boundaries[0][j] >= particle[j]) and (particle[j] >=
            boundaries[1][j]):
            boolBoundaries.append(True)
        else:
            boolBoundaries.append(False)
    return boolBoundaries

```

all av grenser

oppretter vi vektorer som inneholder partiklenes grenser, posisjon, bevegelsesretning,

partiklenes lokalt og globalt beste posisjoner. Siden vi søker etter et globalt

t, er partiklenes lokalt og globalt beste posisjoner initiert med en uendelig høy verdi.

les jevnt utover hele søkeområdet.

```

matrix with the particles best position
best_position = np.copy(particle_position)
particle_best_position)

best value with inf boundary
best_F_value = np.array([np.float64('inf') for _ in
particles]))

float with the groups best value. Starting value is 'inf'
_F_value = np.float64('inf')

matrix with groups best position
c_position = np.array([np.float64('inf') for _ in range
n_groups]))

velocity vector of the particles. Starting velocity is zero
vector = np.zeros((n_particles, n_unknowns),
dtype=np.float64)

```

Initialise av nødvendige variabler med tilhørende verdier

```

0
iteration < n_iterations:
    for i in range(n_particles):
        check_boundaries(particle_position[i]):
            F_candidate = F(particle_position[i])
            # print(particle_position[i])

        # Personal best position
        if (particle_best_F_value[i] > F_candidate):
            particle_best_F_value[i] = F_candidate
            particle_best_position[i] = particle_position[i]
            # print(iteration, " ", particle_best_position[:,11])

        # Global best position
        if (global_best_F_value > F_candidate):
            global_best_F_value = F_candidate
            global_best_position =
particle_position[i])

# Update velocity
for i in range(n_particles):
    new_velocity = (w * velocity_vector[i,j]) + (c1 *
np.random()) * (particle_best_position[i,j] -
particle_position[i,j]) + (c2 * np.random.random()) *
(global_best_position[j] - particle_position[i,j])
    velocity_vector[i,j] = new_velocity
    particle_position[i,j] += new_velocity
    iteration += 1

```

er kjernen av PSO-algoritmen. Her oppdateres funksjonsverdier, posisjoner og hastigheter

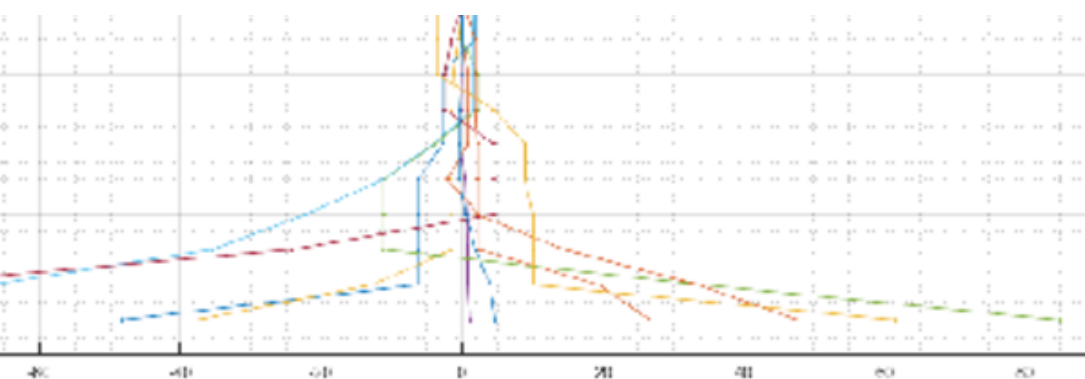
den beste funksjonsverdien oppnådd ut, samt partiklens posisjon:

v er en fysisk komponent og c_2 er en sosial komponent. Konstanten c_1 avgjør i hvor stor grad partiklene beveger seg mot sin tidligere beste posisjon, mens c_2 er en konstant som påvirker i hvor stor grad partiklene beveger seg og setter fart mot den beste posisjonen som så langt er oppnådd av partiklene.

Konstantene nevnt ovenfor er ikke opplagt og tidligere forskning [4, pp. 306-309] enes ikke på hvilke verdier er de beste verdiene for v , c_1 og c_2 . Det har blitt vist at v , c_1 og c_2 må velges med forsiktighet. En mulig regel man kan benytte seg av er at forholdet mellom konstantene er $v > c_1 > c_2$ [4, p. 307].

Partiklene som multipliseres med c_1 og c_2 skal være mellom 0 og 1. Farten til hver partikkel bestemmes ved å generere et tilfeldig tall for å få innført en grad av tilfeldighet inn i bevegelsen. Dette fører til at søkeområdet dekkers. Dette forhindrer også til en viss grad at partiklene låser seg i et lokalt minimum, og blir værende i denne posisjonen. Hvordan algoritmen skal oppføre seg hvis partiklene blir værende ved samme posisjon forklarer vi i seksjon 2.7.

Figur 2.1 viser, ved hjelp av de fargede linjene, partiklenes ferd mot funksjonens globale minimum. På x-aksen ser vi søkeområdet og på y-aksen ser vi antall iterasjoner. Man ser at partiklene beveger seg om $x = 0$ når antall iterasjoner øker.



partikler som konvergerer mot optimal løsning til funksjonen x^2 . På x-aksen ser vi søkeområdet og på y-aksen ser vi antall iterasjoner og hver farge representerer en partikkel. Hver fargete linje representerer ferden til en partikkel mot minimumspunktet.

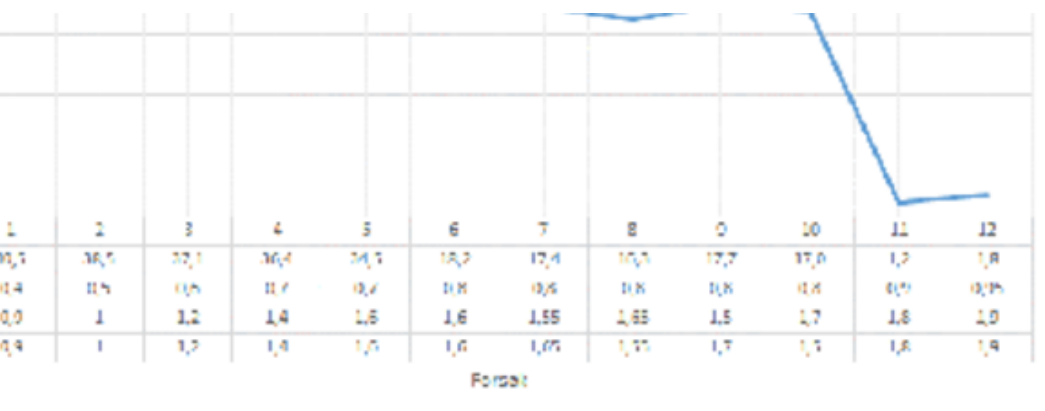
Valg av v , $c1$ og $c2$

Etter at vi har valgt v , $c1$ og $c2$ målte vi presisjonen til algoritmen ved å beregne funksjonen $f(x_i) =$

gjennomsnittet av $f(x_i)$ over 1000 iterasjoner. Denne funksjonen har et kjent globalt minimumspunkt i $x_i = 0$, der

der $f = 0$. Av Figur 10 kommer det frem at forsøk nummer én, hvor det ble benyttet

$c1 = 0,9$ og $c2 = 0,9$ ga den beste presisjonen.



over forsøk som viser presisjonen ved forskjellige verdier av v , c_1 og c_2 . På y-aksen har vi $Precision = |y_n - f_{min}|$ der y_n er funksjonens verdi etter n -iterasjoner og f_{min} er funksjonens kjente globale høyere verdi på y-aksen representerer en høyere presisjon. Forsøk nr. 1 har den høyeste presisjonen.

0,4, $c_1 = 0,9$ og $c_2 = 0,9$ er benyttet videre i testingen av funksjonene i seksjonene 2.3 og 2.8.4, da de ga høyest presisjon.

Testester

vi ikke begrunnet hvorfor vi bruker 50 partikler og 1000 iterasjoner. I Figur 11

interessammenhengen mellom presisjon og antall dimensjoner. Testene er utført

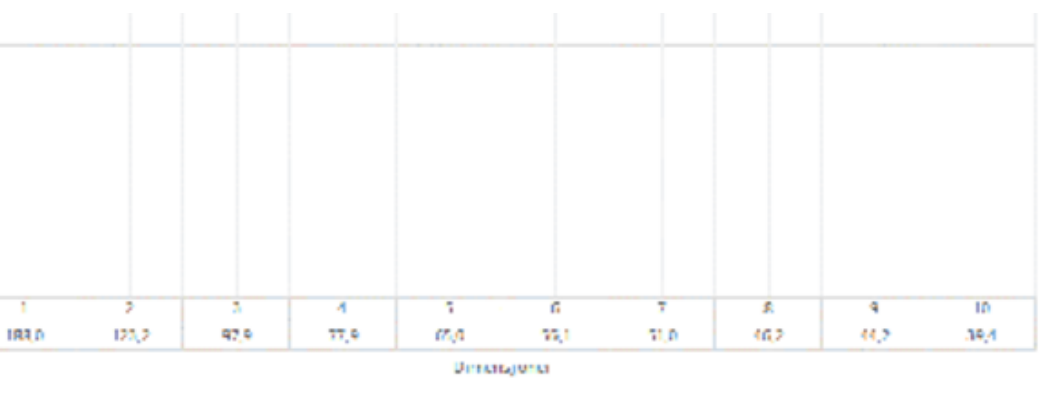
h $f(x_i) = \sum_{i=1}^{10} x_i^2$, da det er enkelt å øke antall ukjente fra én til ti, slik som vi gjør i

testene er gjennomført med verdiene $v = 0,4$, $c_1 = 0,9$ og $c_2 = 0,9$.

reken til algoritmen er satt til 50 partikler. Litteratur sier at det er liten gevinst ved å

kler til mer enn 50 [4, p. 312]. Man må være klar over at et større antall partikler

sinne tilskrevet. Dette er fordi et 11-barnesinne datamaskinen må gjennomføre



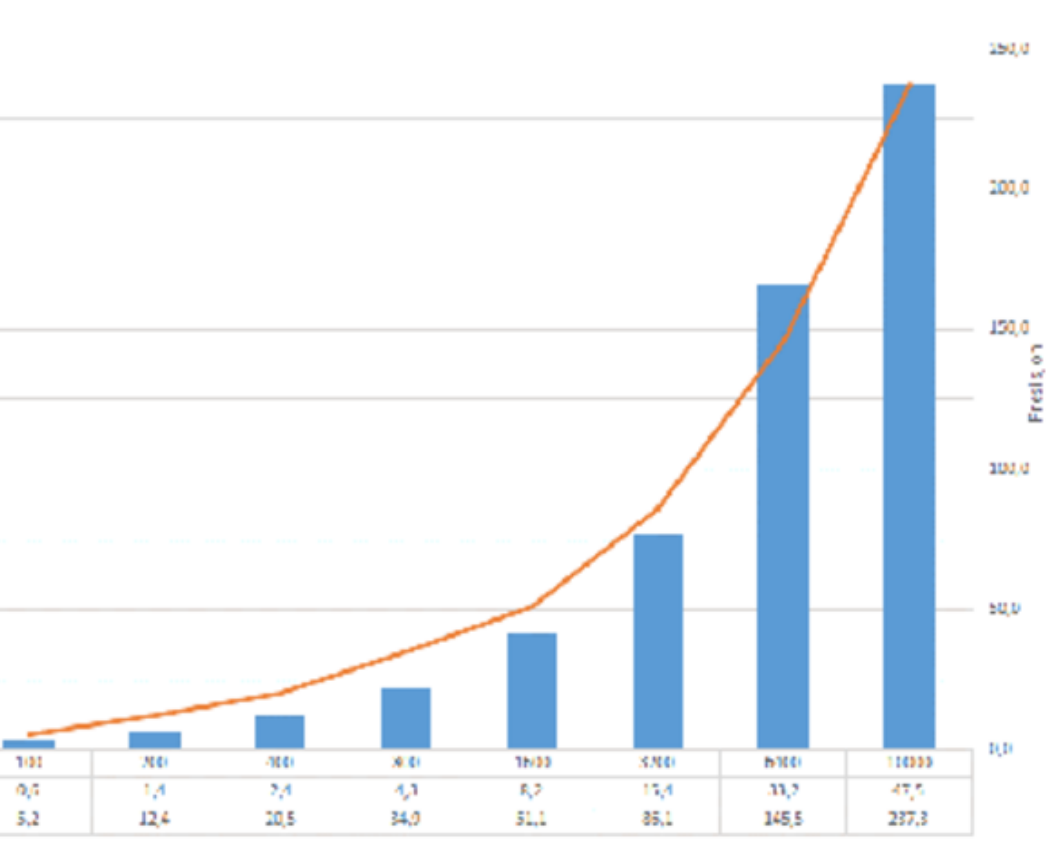
1000 iterasjoner. På y-aksen har vi $Precision = |g_{10}(z)|$ med $z = |y_n - f_{min}|$ der y_n er funksjonens verdier og f_{min} er funksjonens globale minimumsverdi. En høyere verdi på y-aksen representerer en høyere ukjent, reduseres presisjonen.

Kriterier i algoritmen

Blir værende i samme posisjon over lengre tid, tyder dette på at de har samlet seg i et minimum. Hvis dette skjer før algoritmen har gjort alle iterasjonene, bør prosessen skal avbrytes for å spare maskinkraft og tid.

Iterasjoner er nådd

Et skriptet skal stanse er at antall iterasjoner er oppnådd. Antall iterasjoner påvirker konvergens av en funksjon, og algoritmens presisjon, når algoritmen stoppes prematurt. Antall iterasjoner er korrekt for å sikre at det oppnås konvergens til riktig tid. Hva som er riktig antall iterasjoner avhenger av funksjonen som skal optimaliseres. I Figur 12, nedenfor, ser vi sammenheng mellom tidsbruk, presisjon og antall iterasjoner.



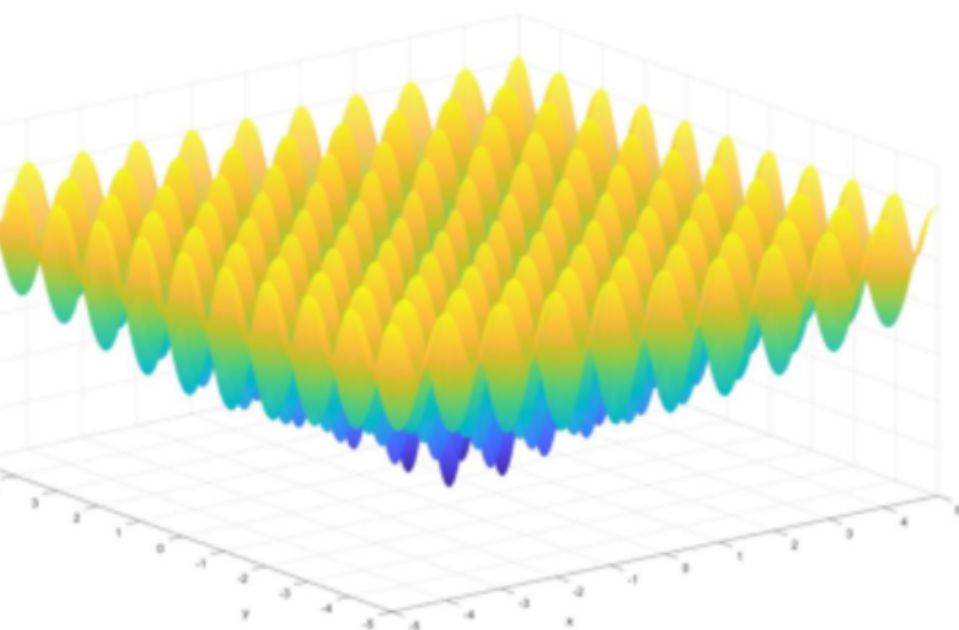
Sammenheng mellom tid, presisjon og antall iterasjoner (fra 100 til 10000). Presisjonen er vist på y-aksen til høyre, og tiden er vist på x-aksen til venstre. Tallet på x-aksen er verdien av den Briggske logaritme n til feilmarginen.

Vi ser at 10000 iterasjoner gir høyest presisjon. 10000 iterasjoner blir derfor benyttet i de fleste av funksjonene.

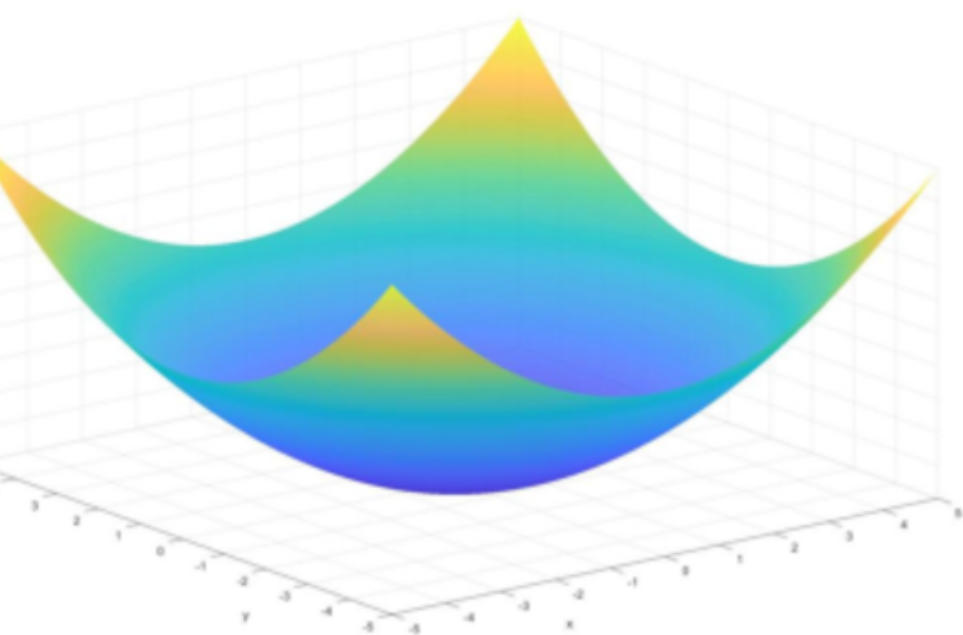
Test av skript med standardfunksjoner

ter. I tillegg av de vanlige optimeringsalgoritmene ville gjort akkurat det.

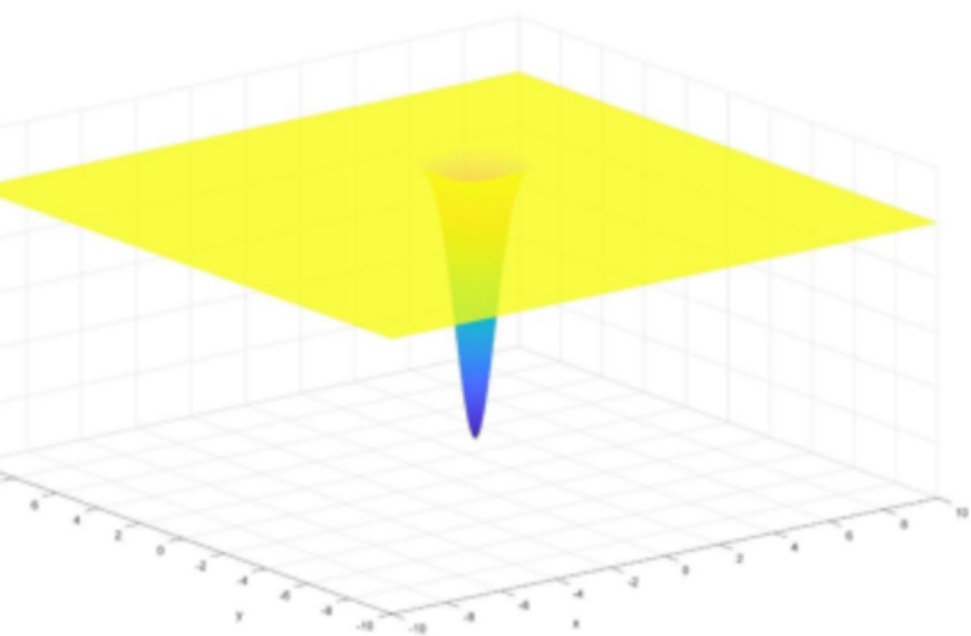
$f(x, y) = -20^{-0,2\sqrt{0,5(x^2+y^2)}} - e^{\sqrt{0,5 \times (\cos 2\pi x + \cos 2\pi y)}} + e + 20$ og det globale
 punkt $(0,0)$. Vi tester funksjonen med definisjonsområde mellom -5 og 5 for både x og
 y . Resultatet: $f(-6,98849930E-17, 6,07057124E-18) = 0$



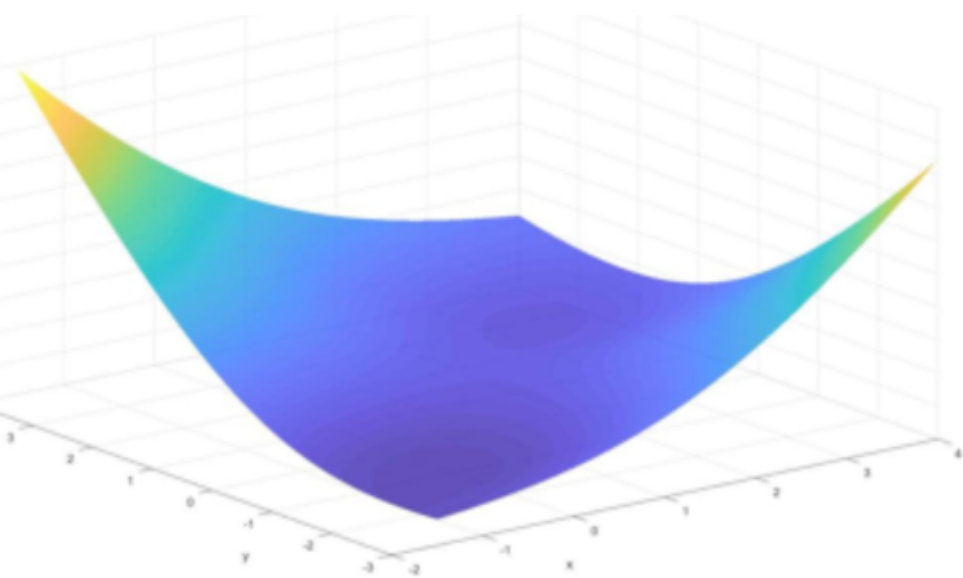
funksjon



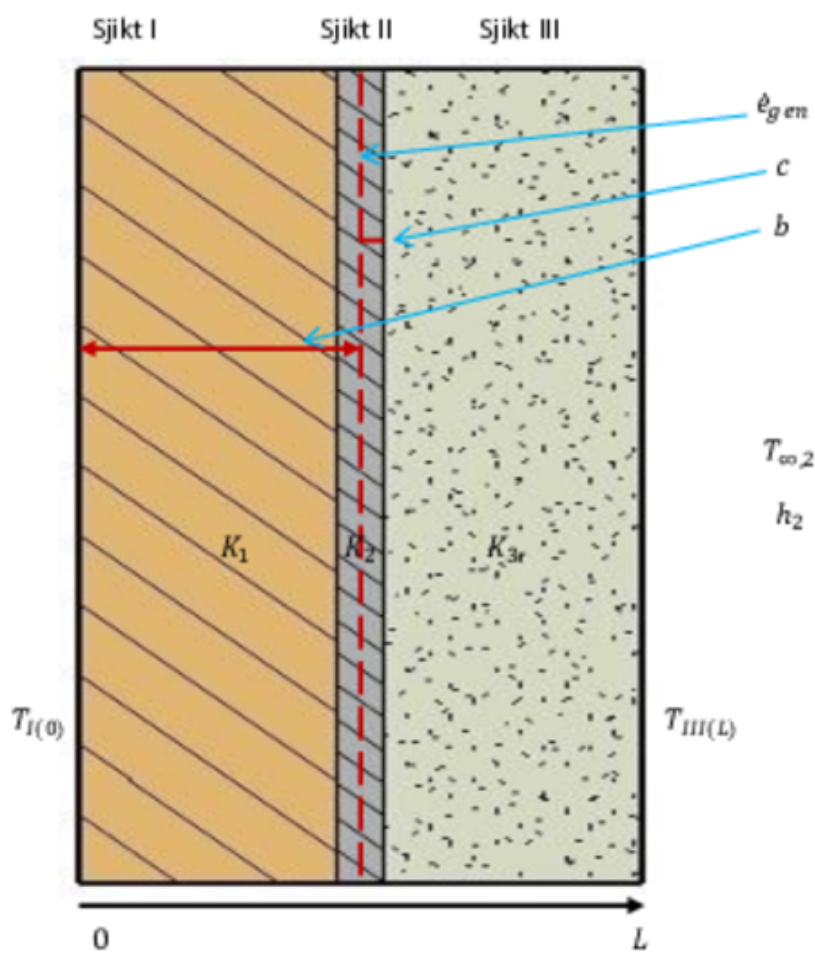
tsjon av to variable r



fungsi



ek funksjon



en viser snitt av en vegg delt i tre sjikt. Sjikt I og III grenser mot henholdsvis inn- og utsiden av en varmeelementet. Elementet vil i virkelighet være tynn som en aluminiumsplate. Det er

te nedsenkes i vann eller i en annen kjølemedium. Veggens er 30 cm tykk og er i en varmeledning

ario der vi har like betingelser for varmetransport på hver side av varmeelementet.

il si at det er samme materiale på begge sider av sjikt II, og temperaturene, samt

sjonkoeffisientene på hver side av veggen er de samme.

ario der vi har ulike verdier på hver side av varmeelementet. I dette eksempelet har vi

med treverk i innvendig sjikt, betong utvendig og varmeelement i midten.

aturene og konveksjonskoeffisientene er forskjellige på hver side.

entering

n funksjon algoritmen kan optimalisere problemet har vi benyttet oss av ligningene i

beskriver varmefluksen gjennom veggen. Ved å bearbeide ligningene får vi

filene til de forskjellige sjiktene i veggen. Ligningene kombineres slik at man får et

nen av overflatetemperaturen på hver side av veggen. Dette gir en funksjon av

ed å sette funksjonen inn i skriptet kan en optimal verdi for b bestemmes. Optimal

nnen av overflatetemperaturene på begge sider av veggen blir høyest mulig.

som beskriver varmestrømningen gjennom veggen. Disse kommer fra Fauriers konduksjonslov og
gjennom de forskjellige sjiktene.

Ligninger til kapittel 3

	Sjikt I	Sjikt II	Sjikt III
	(1)	(2)	(3)
t	$\frac{dT}{dx} = \frac{\dot{q}_0}{k}$	$\frac{dT}{x} = -\frac{\dot{q}_{gen}}{k}x + B_1$	$\frac{dT}{dx} = -\frac{\dot{q}_L}{k}$

$$A_2 = -\frac{\dot{e}_{g,em}(b-c)}{h_2} + B_1 \frac{k_2}{h_2} + T_{\infty,1}, \quad (8)$$

$$C_2 = \frac{\dot{e}_{g,em}(b+c)}{h_2} - B_1 \frac{k_2}{h_2} + T_{\infty,2}, \quad (9)$$

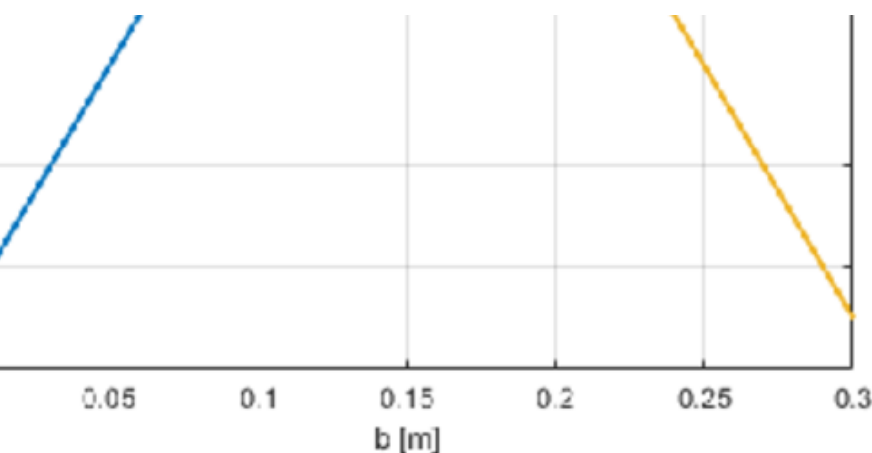
$$B_1 = \frac{\left(\frac{2bc}{k_2} - \frac{(b+c)(b+c-L)}{k_3} + \frac{(b-c)^2}{k_1} + \frac{(b+c)}{h_2} + \frac{(b-c)}{h_1} \right) + T_{\infty,2} - T_{\infty,1}}{2c - \frac{k_2}{k_3}(b+c-L) + \frac{k_2}{k_1}(b-c) + \frac{k_2}{h_2} + \frac{k_2}{h_1}} \quad (10)$$

(9) og (10) kommer fra ligningene i tabell 1 og gir oss grunnlaget for å undersøke hvordan varmeelementet må plasseres.

I utledning av ligning (8), (9) og (10).

[Betingelser på hver side](#)

Her bruker vi for å få klarhet i hvordan temperaturprofilen ser ut i en situasjon med like betingelser på hver side av varmeelementet. Vi ønsker samtidig å sjekke om skriptets resultat stemmer overens med våre håndberegninger. Vi ønsker å danne oss et bilde av hvordan temperaturprofilen ser ut når varmeelementet er plassert i midten av veggen. Vi forventer da at temperaturprofilen vil være symmetrisk om midten ved de gitte betingelsene. De konduktive egenskapene og tykkelsen til varmeelementet er valgt slik at formen til temperaturprofilen



temperaturprofil for $b = L/2$. Like betingelser på hver side av varmeelementet. Blå farge viser
 i sjikt 1. Rød farge viser temperaturprofilen i varmeelementets sjikt. Gul farge viser
 i sjikt 2. Her er $L = 0,3\text{m}$, $c = 0,05\text{m}$, $\dot{e}_{gen} = 200 \frac{\text{W}}{\text{m}^2}$, $k_1 = k_3 = 1 \frac{\text{W}}{\text{mk}}$, $k_2 = 2 \frac{\text{W}}{\text{mk}}$, $h_1 =$
 $h_3 = T_{\infty,2} = 20^\circ\text{C}$.

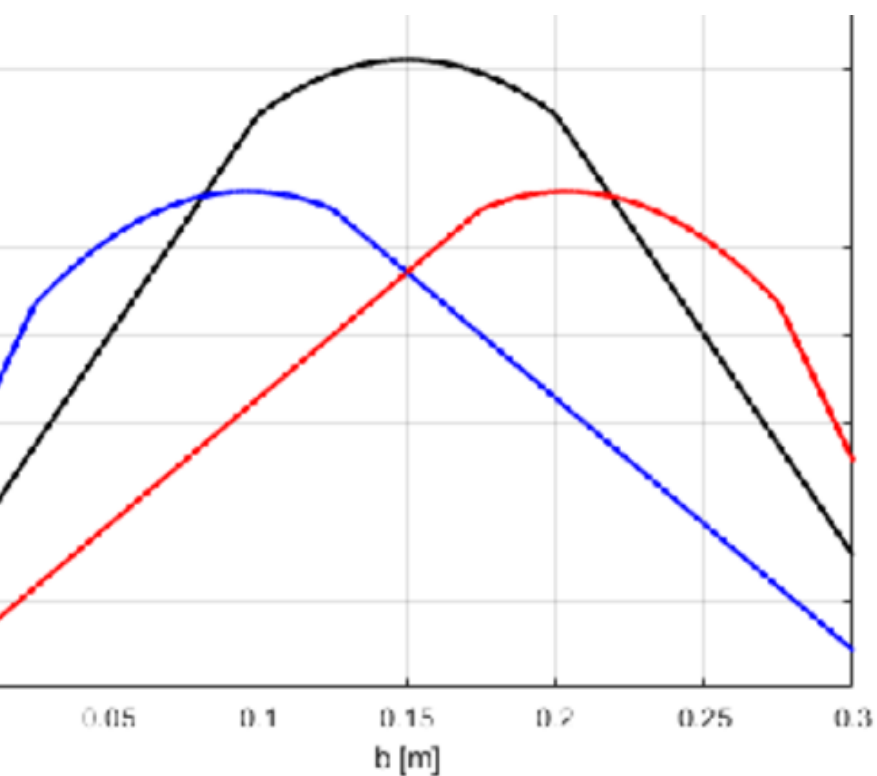
der $k_1 = k_3$ er den førstederiverte av funksjonen:

$$f'(b) = \dot{e}_{gen} \left(\frac{1}{h_2} - \frac{1}{h_1} \right) + k_2 \left(\frac{1}{h_1} - \frac{1}{h_2} \right) \frac{\dot{e}_{gen} \left(2 \left(\frac{1}{k_2} - \frac{2}{k_1} \right) c + \frac{L}{k_1} + \frac{1}{h_2} + \frac{1}{h_1} \right)}{2 \left(1 - \frac{k_2}{k_1} \right) c + k_2 \left(\frac{L}{k_1} + \frac{1}{h_2} + \frac{1}{h_1} \right)} \quad (11)$$

ifølge (11) at betingelsen $h_1 = h_2$ medfører at $f'(b) = 0$. Dette resultatet forteller oss

at plasseringen i veggens er irrelevant for å oppnå høyest mulig temperatur. Videre

er de endepunktene i funksjonen. Funksjonen ved betingelsene $k_1 = k_3$ og $h_1 = h_2$ er:

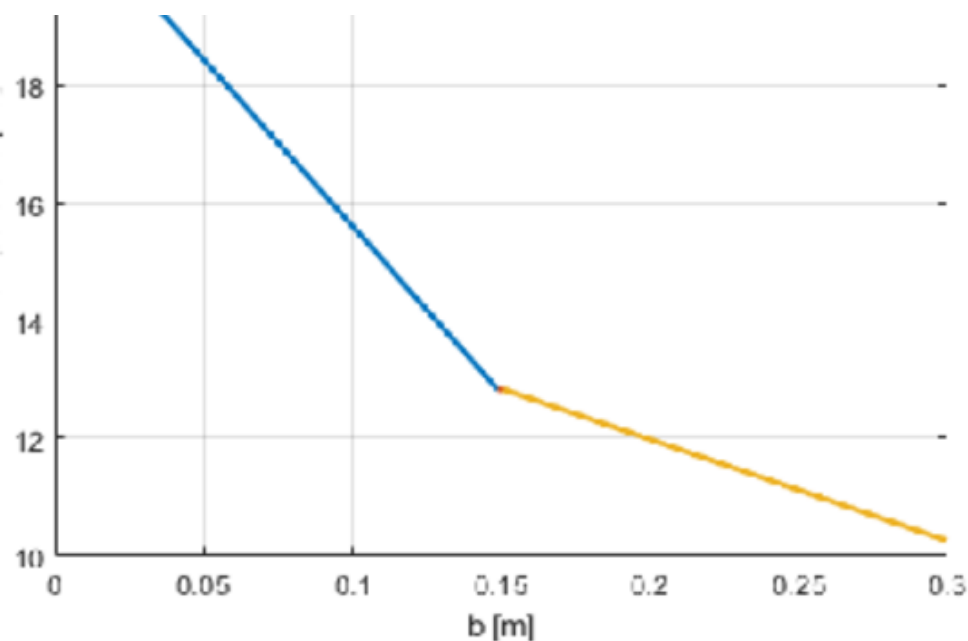


temperaturprofiler ved forskjellige plasseringer av varmeelementet. Blå linje er temperaturprofilen med varmeelementet plassert ved $L/4$. Svart linje er temperaturprofilen med varmeelementet plassert ved $L/2$. Rød linje er temperaturprofilen med varmeelementet plassert ved $3L/4$. Summen av temperaturene på begge sider av veggens tykkelse er det samme for disse temperaturprofilene.

Oppgave 1: Bestem størrelsen på varmeelementet på hver side.

Oppgave 2: Bestem størrelsen på varmeelementet på hver side ved en vegg med bredde 30 cm hvor et varmeelement skal plasseres slik at

temperaturene på hver side av veggens tykkelse blir høvest mulig. Veggens tykkelse er 30 cm. Siktet

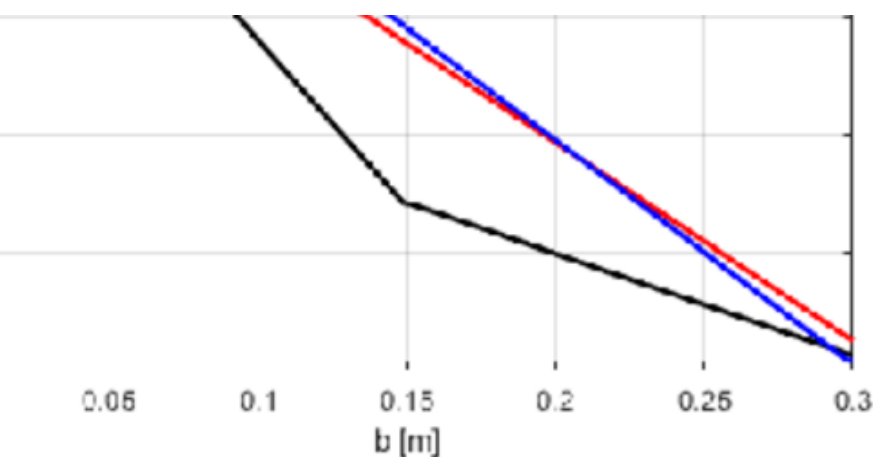


temperaturprofil gjennom vegg i scenario 2 med varmeelement plassert ved $b = L/2$. Blå linje er viser
 i sjikt I, gjennom treverket. Oransje linje (i midten) viser temperaturprofilen gjennom varmeelementet.
 r temperaturprofilen i sjikt III, gjennom betongen. I figuren er $L = 0,3m$, $c = 0,001m$, $\epsilon_{geu} = 200 \frac{W}{m^2}$
 $= 160 \frac{W}{mK}$, $k_3 = 0,45 \frac{W}{mK}$, $h_1 = 10 \frac{W}{m^2K}$, $h_2 = 30 \frac{W}{m^2K}$, $T_{en,1} = 22^\circ C$ og $T_{en,2} = 10^\circ C$.

ing (7) inn i skriptet foreslår algoritmen å plassere varmeelementet i veggens

t III. Summen av overflatetemperaturene blir da tilnærmet lik $31^\circ C$. Dette er vår

ring av varmeelementet.

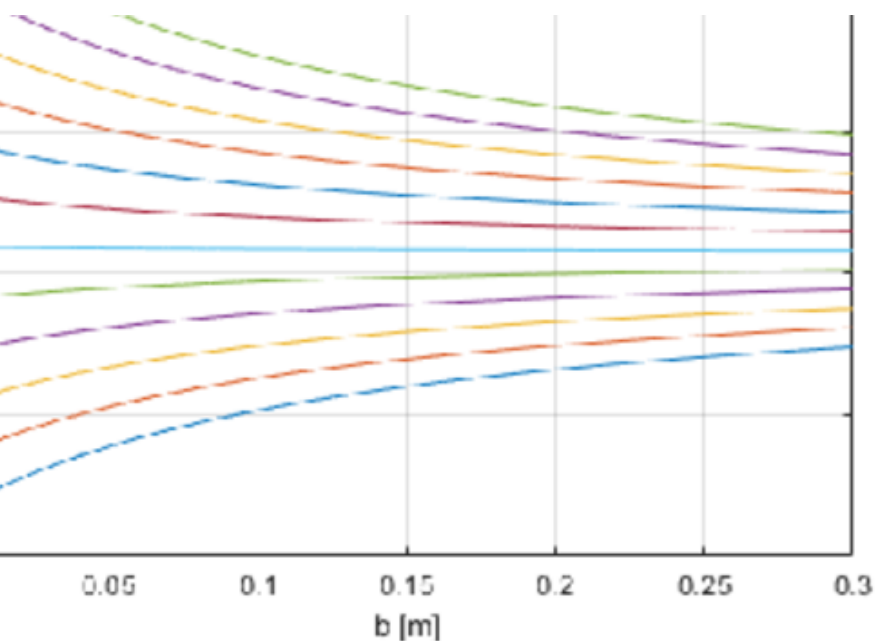


Temperaturprofil gjennom veggen i scenario 2 med varmeelement plassert ved $b = c$ (rød), $b = L/2$ (svart)
 Dette representerer henholdsvis varmeelementets plassering på venstre side av veggen, i midten av
 høyre side av veggen

For varmelementets plassering på den siden av veggen hvor det er lavest
 temperatur. Vi ønsket å undersøke om varmeelementets plassering stemte ved å analysere
 (1) og dens førstederiverte.

Resultatene er i samsvar med hva algoritmen foreslår av løsninger. Disse finner man i Vedlegg
 1. Vi trygt konkludere med at varmeelementets plassering må være i ytterkant av
 veggen for å oppnå høyest mulig overflatetemperatur. I tillegg til dette har vi fått enda en

temperaturprofil gjennom veggen i scenario 2 med varmeelement plassert ved $b = c$ (rød), $b = L/2$ (svart)



funksjonen $f(b)$. Hver linje viser summen av temperaturene på hver side av veggen ved forskjellige varmeelementer ved gitte effektnivåer. Av plottet kan man se at beste plassering av varmeelementet går fra $L - c$ (grafens venstre side) til $L + c$ (grafens høyre side) ved en økning av effekten. Dette kan man tydelig se ved skrumming. Beste plassering finner man ved den høyeste verdien til hver linje. Effektnivået til plottet går fra 1000 W til 12000 W med steg på 1000 W

et ved en gitt effekt i varmeelementet endrer varmeelementets beste plassering seg fra ene side av veggen. Det er viktig at man her er klar over at det kun er θ_{gen} som endres

Vårt ønske er å finne et generelt uttrykk for når den beste plasseringen av varmeelementet endres.

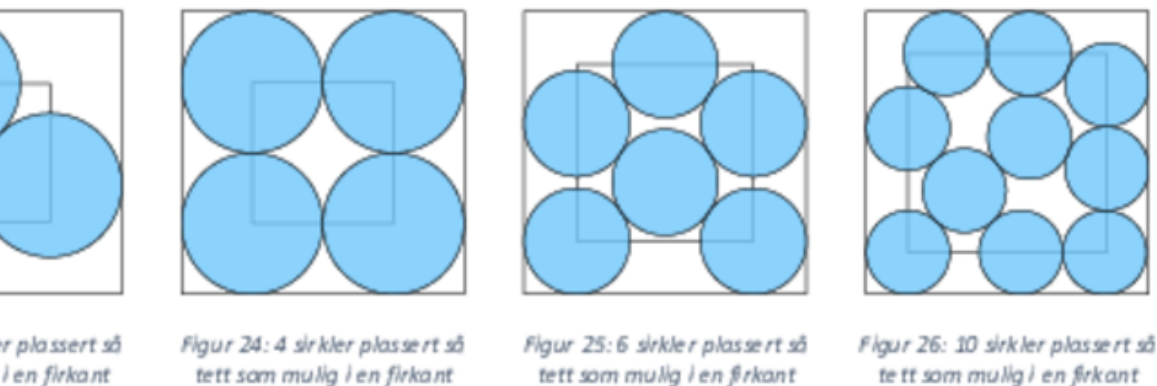
fra seksjon 3.3 gir dette $\dot{e}_{gen} \approx 5889,5 \text{ W}$. Vi ønsker nå å undersøke igjen om vår
k foreslår riktig løsning. Ved å sette $\dot{e}_{gen} = 5889 \text{ W}$ og $\dot{e}_{gen} = 5890 \text{ W}$ inn i skriptet,
t går fra å foreslå $L - c$ til c som optimal plassering for varmeelementet. Dette
varmeelementets plassering endres mellom disse to verdiene og beviser at vår algoritme
sning.

r utledning av ligning (13).

anvendelser i VVS-bransjen

Eksempelet ovenfor er i utgangspunktet helt teoretisk, men kan imidlertid overføres
de konstruksjoner som for eksempel gulvvarme, takvarme og snøsmelteanlegg. Man
blemet og kun se på et varmeelements plassering for å bestemme materialtykkelser
ntets effekt.

et eget felt innenfor matematikk [13]. Det er, av andre, beregnet optimal plassering av radius opp til $N \leq 10000$ for å minimere arealet til det omsluttende kvadratet. Her viser sirkler plassert slik at det omsluttede arealet blir minst mulig [13].



Vår løsning skal tjene som et praktisk hjelpemiddel til de prosjekterende, slik at de kan sette av arealet og volumet som må settes av til sjakter.

Vi plasserer våre rør nærmest mulig en av sjaktens vegger, i vårt tilfelle y-aksen.

Vi ønsker at rørene vil skje i en firkantet sjakt. Samtidig ønsker vi at rørene ikke kommer i kontakt med

å oppnå dette trenger vi en matematisk funksjon som vil diktere rørenes

et xy -plan.

nen

ergien mellom to partikler med samme ladning. Potensialet til kraften F er

$$E_{potensial} = k_c \frac{q_1 q_2}{r}, \quad (23)$$

den mellom to partikler, og q_1 og q_2 er ladningene til partiklene. Vi velger å se bort fra de fysiske konstanten k_c da den ikke har relevans i vårt tilfelle. Vi anser også sirklene som har samme ladning. Som resultat av dette, blir frastøtningskraften mellom to sirkler

$$E_{frastotning, 2 \text{ sirkler}} = \frac{1}{\sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}}. \quad (24)$$

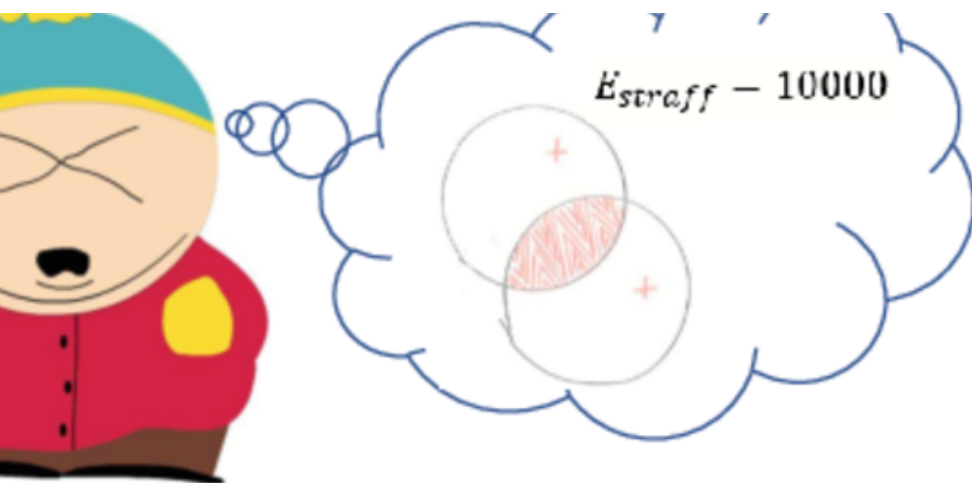
Avstanden (24) er avstanden mellom senterne til to sirkler beregnet ved hjelp av Pytagoras'. For å beregne avstander mellom sirkler trenger vi sirklens posisjon og radius. Sirklens posisjon er gitt ved x_n og y_n , og radiusen er gitt ved r_n . n er antall sirkler.

Den totale frastøtningen ved å summere frastøtningskraften for hvert sirkelpar

$$E_{frastotning} = \sum_{i=1}^n \sum_{j=i+1}^n \frac{1}{\sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}} \quad (25)$$

Avstanden mellom to partikler øker, blir $E_{frastotning}$ mindre og vil derfor gi en lavere

Siden vi velger å anvende PSOen vår til å finne minimumsverdien til vår funksjon, vil vi presentere frastøtning.



Sirkler tangerer eller overlapper hverandre får funksjonen en straff på 10000 slik at algoritmen vurderer en annen løsning.

ning

klene plasseres nærmest mulig en flate som man kan anse som en av sjaktens vegger.

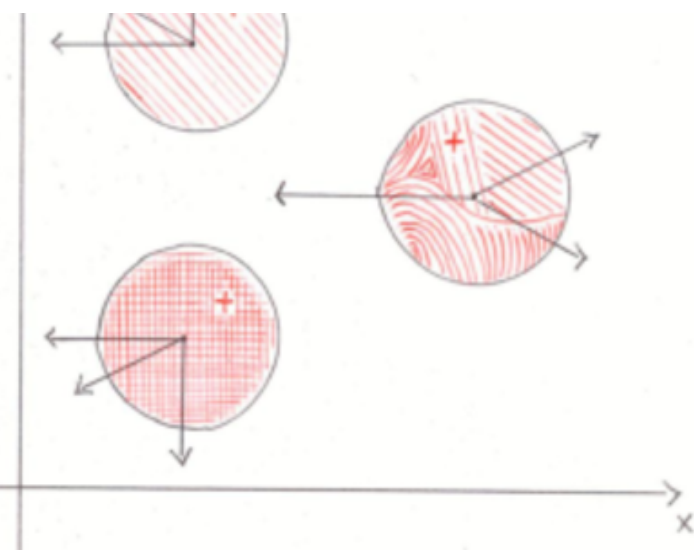
tt inspirere av Coulombs lov og tenker oss at veggten til sjakten representerer en plate

t elektrisk ladning til partiklene. Det vil si at veggten tiltrekker seg sirklene. Hvis vi for

r y-aksen som vegg, vil tiltrekningen i vår funksjon se slik ut:

$$E_{\text{tiltrekning } y} = - \sum_{i=1}^n \frac{k_1}{\sqrt{x^2}} \quad (27)$$

konstanten $k_1 = 50$. Ved å endre fortegnet til $E_{\text{tiltrekning } y} = \dots$ vil veggten ikke lenger



Den viser kreftene som påvirker sirklene når $F = E_{\text{frougpløring}} + E_{\text{sløyf}} + E_{\text{sløretning, y}}$

Guaranteed Convergence PSO

Men fant vi ut at vi trengte en kraftigere algoritme enn den generelle algoritmen nevnt i lemet som oppsto var at algoritmen fant et lokalt minimum etter forholdsvis få
 sluttet å lete etter lavere funksjonsverdier. Den endelige plasseringen ble da ikke
 tre partiklene følger partikkelen med beste posisjon, og søket etter en enda bedre
 er opp. Valget falt derfor på en utvidet versjon av PSO, kalt Guaranteed Convergence
 kalt GCPSO [14]. Denne algoritmen garanterer konvergens ved at partikkelen med den
 n alltid skal bevege seg i en tilfeldig retning i nærheten av den globale beste

Verdien til *feil* settes til 0

partikkel har funnet en bedre funksjonsverdi og blitt den berste partikkelen

til suksess settes til 0

Verdien til *feil* settes til 0

partikkel har funnet en bedre funksjonsverdi

Verdien til suksess settes til 0

Verdien til *feil* øker med 1

suksess er større enn 15, blir ρ multiplisert med 2. Hvis verdien til *feil* er større enn

isert med 0,5. Og til slutt, hvis samme partikkel ikke finner noen bedre funksjonsverdi

rende verdi forbli uendret.

n ρ påvirker bevegelsen til den beste partikkelen, se GCP50-skriptet i Vedlegg 1.

av skriptet

ksjonen å undersøke om funksjonen F (ligning (28)) opptre som vi ønsker. Dette

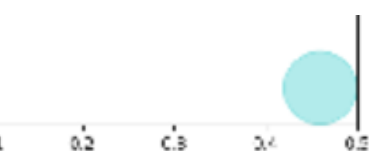
å inspisere leddene $E_{\text{frastotning}}$ og $E_{\text{tiltrekning}}$ for seg selv, og så benytte dem

tt kontrollerer vi om E_{straff} fungerer som den skal. Vi velger nedenfor å bare vise

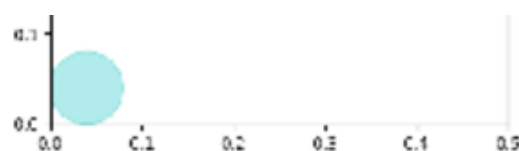
et og resulta tene med rø r av samme radius. Dette er gjort for enkelhetens skyld.

tte $E_{\text{frastotning}}$ i funksjonen, blir alle sirklene plassert som vist i Figur 29, i hvert sitt

ette leddet maksimerer avstanden mellom alle sirklene er dette den forventede



Figur 29: En sirkel plassert med maksimal avstand fra kant. Posisjonene er $(0.46, 0.46)$ og $(0.46, 0.04)$ med funksjonsverdien 12,89



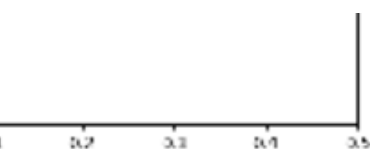
Figur 30: Fire sirkler plassert så langt det lar seg gjøre mot figurens y-akse (venstre side). Sirklene er fordelt jevnt utover i y-retning slik at avstanden mellom dem fremdeles er størst mulig. Funksjonsverdien er -3969,12

vi å framprovosere en situasjon der E_{straff} må benyttes. Vi øker derfor antall sirkler

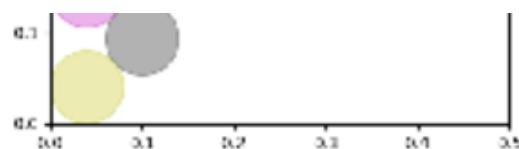
ved å øke antallet sirkler. Ved å øke antallet sirkler gjør de at sirklene overlapper, som vist i Figur 31.

Ved å øke antallet sirkler, forhindrer vi at sirklene overlapper. Dette ser vi i Figur 32, der sirklene

er plassert så langt det lar seg gjøre mot figurens venstre side, uten å overlappe.



som overlapper hverandre for å
funksjonsverdien som er -7772,81



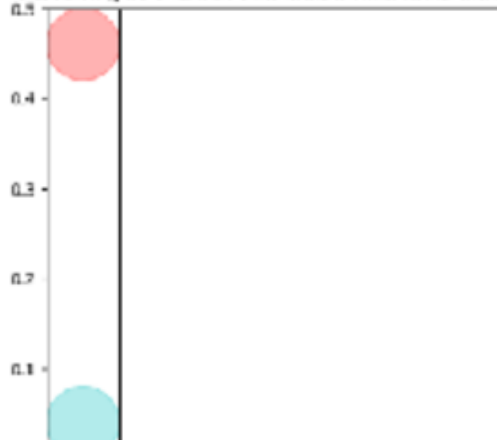
Figur 32: 8 sirkler som har plassert seg så langt det lar seg
gjøre mot figurens y-akse (venstre side) uten å overlape.
Her er det innført en drastisk økning i funksjonsverdien for
hver sirkel som overlapper. Funksjonsverdien er -6237,61

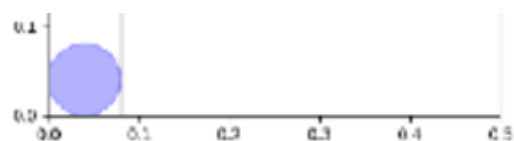
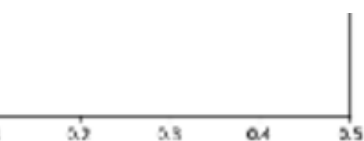
t
ne i seksjon 4.3 ser vi at funksjonen gjør at sirklene plasseres slik vi ønsker. Vi kan nå
re sirkler inntil y-aksen som vist i figurene nedenfor. Figurene viser optimal plassering
med samme radius.

1 sirkler i en firkant med areal 0.04 m



Plassering av 2 sirkler i en firkant med areal 0.04 m





er plassert så nærme som mulig y-aksen

Figur 36: Fire sirkler plassert så nærme som mulig y-aksen

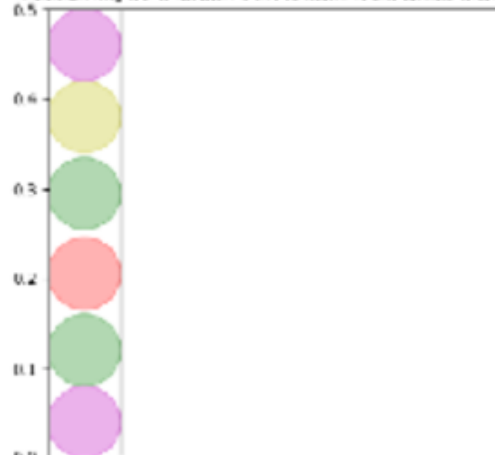
alle sirklene inntil y-aksen, samtidig som de er så langt unna hverandre som mulig. I dette et bilde.

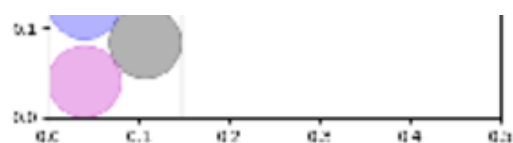
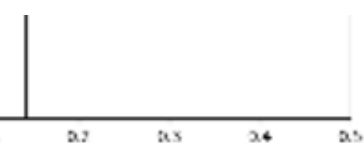
også at alle sirklene ligger inntil y-aksen. Her er imidlertid sirklene ikke fordelt jevnt i to sirklene er plassert en anelse tettere sammen mot hver sin ende av y-aksen. Dette sirklene i midten blir frastøtt av et ulikt antall sirklene på hver sin side i y-retning. For en grå sirkelen én sirkel på over-, og to på undersiden. Dette fører til at kraften fra undersiden er større enn kraften fra oversiden.

5 sirklene i en firkant med areal 0.04 m



Plassering av 8 sirklene i en firkant med areal 0.04 m





er plassert så nærme som mulig y-aksen

Figur 40: Åtte sirkler plassert så nærme som mulig y-aksen

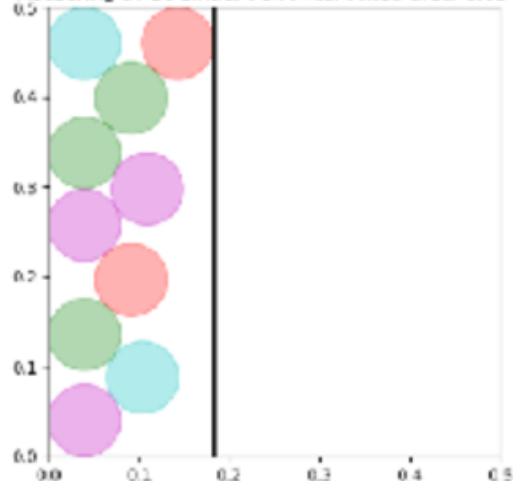
en situasjon der det ikke er tilstrekkelig plass til alle sirklene i y-retning. Vi ser at én), ligger lengre unna y-aksen enn de andre sirklene. Dette er det nærmeste den n, uten å overlape andre sirkler. Samtidig er avstanden til flest mulig sirkler at frastøtningskraften fra disse blir minst mulig.

en situasjon som ligner Figur 39. Vi har et partall rør og ser en symmetri som ligner har sett i figurene ovenfor, hvor sirklene blir fordelt jevnt på hver side av $y = 0,25$.

9 sirkler i en firkant med areal 0,07 m



Plassering av 10 sirkler i en firkant med areal 0,09 m



edenfor.

Plestoring av 3 sirkler i en firkant

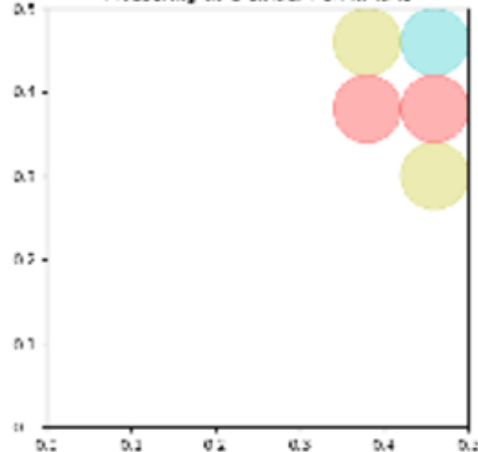


er tiltrukket øvre venstre hjørne

Plestoring av 7 sirkler i en firkant



Plestoring av 5 sirkler i en firkant



Figur 44: Fem sirkler tiltrukket øvre høyre hjørne

Plestoring av 9 sirkler i en firkant



lasseringen.

ligger vedlagt som Vedlegg 1.

Arbeid

n av GCPSO kan videreutvikles og brukes som grunnlag for lignende

beregninger. Et naturlig nestesteg er å innføre et nytt ledd i funksjonen F , som

arealet til den omsluttende firkanten rundt sirklene. Dette vil gi mulighet til å

ktens areal. Denne vil antakeligvis gjøre algoritmen til et enda bedre verktøy for VVS-

reloptimalisering er implementert vil en naturlig videre vei være å koble skriptet

et Dynamo, for å videre benytte det i Revit. Tanken er at man skal kunne hente

en, diameterne til rørene som skal plasseres i sjakten, og så la algoritmen og Dynamo

automatisk i bygningsmodellen.

nskelig å måle prestasjonen til GCPSO mot én eller flere andre

algoritmer. Ved å gjøre dette vil man kunne bekrefte eller avkrefte om GCPSO er den

de algoritmen til å løse slike optimaliseringsproblemer.

nde og lærerike prosesser.

GCP SO mot andre optimaliseringsalgoritmer, og kan derfor heller ikke si hvordan

er i forhold til disse. Det vi derimot kan si, er at vår algoritme er god på numerisk

Dette til tross for at optimalisering generelt er et vanskelig felt, ettersom man ofte

ningen er et lokalt eller globalt ekstremalpunkt.

de nevnte resultatene ovenfor har vi konstatert at skriptet fungerer bra. Vi er

den algoritmen bruker på å gi oss et resultat, og oppnår høy presisjon på kort tid. Ved

er får vi gode resultater på under et minutt. Denne tiden er i en

sammenheng ubetydelig.

te, viser GCP SO seg å være et godt verktøy til å optimalisere VVS-tekniske problemer.

nt Laboratory Systems , 28 April 2015.

brecht, «Particle Swarm Optimization,» *Computational Intelligence* , pp. 289-358 ,

ols, «Python tutorial,» [Internett]. Available:

www.w3schools.com/PYTHON/default.asp. [Funnet 14 Februar 2019].

arkFcns,» [Internett]. Available: <http://benchmarkfcns.xyz/fcns>. [Funnet 28 Januar

Function,» [Internett]. Available:

enchmarkfcns.xyz/benchmarkfcns/ackleyfcn.html. [Funnet 28 Januar 2019].

Function,» 2019. [Internett]. Available:

enchmarkfcns.xyz/benchmarkfcns/spherefcn.html. [Funnet 28 Januar 2019].

Function,» [Internett]. Available:

enchmarkfcns.xyz/benchmarkfcns/easomfcn.html. [Funnet 28 Januar 2019].

nick Function,» [Internett]. Available:

enchmarkfcns.xyz/benchmarkfcns/mccormickfcn.html. [Funnet 28 Januar 2019].

k, «Varmekonduktivitet og varmemotstand for bygningsmaterialer,» 2003.

t]. Available:

www.byggforsk.no/dokument/209/varmekonduktivitet_og_varmemotstand_for_bygn
materialer. [Funnet 28 Mars 2019].

gerstad, Tekniske føringsveier i større bygninger, Oslo: RIF - organisasjonen for

```

0.025, 0.020, 0.020, 0.020, 0.020, 0.020]
ance = 0.0

= 50
e = 4000
len(r)
= 2 * n_centers
ty = ('inf')

and penalty constants
‡ Velocity
‡ Cognitive component
‡ Social component
‡ Penalty constant for individual distance too small
‡ Component for attraction to y_axis
‡ Component for attraction to x_axis

m):

the function to be solved.
: unknown[0] ** 2
an int

_distances = distances(unknown)
_width = width(unknown)
= fitness_distances ‡+fitness_width
fitness

es(unknown):

tes the total and individual distance between all particles and
.....

```

```

ctraction_y = 0
for i in range(n_centers):
    distance = (unknown[i*2])
    distance = (unknown[(i*2)-1])
    attraction_x += 50 / np.sqrt((y_distance + 0.01)**2) * c5
    attraction_y += 50 / np.sqrt((x_distance + 0.01)**2) * c4
    total_distance_penalty = 0
for k in range(len(individual_distances)):
    if (individual_distances[k] < individual_distances_min[k]):
        individual_distance_penalty += c3
    rejection = E_attraction_x - E_attraction_y +
    total_distance_penalty

def get_x_position(radius):
    #position and radius and input and returns necessary x-axis

x_position = np.array(unknown[0::2])
x_position = radius + x_position
x_max = np.max(x_width)
x_width_max

def is_within_boundaries(particle):
    #if an particle is within it's boundaries. Returns false

boundaries = []
for j in range(n_unknowns):
    if (boundaries[0][j] >= particle[j] and (particle[j] >=
    boundaries[1][j])):
        boolBoundaries.append(True)
    else:
        boolBoundaries.append(False)

```

```

ance, high=y_max - r[int(K/4)] - safety_distance)

# matrix with the particles best position
best_position = np.copy(particle_position)

# matrix with groups best position
best_position = np.array([np.float64('inf') for _ in range
                           n_particles)])

# best value with inf boundary
best_F_value = np.array([np.float64('inf') for _ in
                           range(n_particles)])

# best value with the groups best value. Starting value is 'inf'
best_F_value = np.float64('inf')

# velocity vector of the particles. Starting velocity is zero
velocity_vector = np.zeros((n_particles, n_unknowns), dtype=np.float64)

# vector of radiuses
radii_vector = np.array(r, dtype=float)

# particle index
particle_index = None
best_particle_index = None
best_F_value = np.float64('inf')

# 0 † Highest acceptable value is 15
# 0 † Highest acceptable value is 5
# 0

while iteration < n_iterations:
    for i in range(n_particles):
        check_boundaries(particle_position[i]):
            # candidate = particle_position[i]

```

```

global_best_F_value = global_best_F_value
global_best_particle_index = new_best_particle_index

if iterations > 15:
    rho = 2 * rho
if failures > 5:
    rho = 0.5 * rho

rho = rho

def velocity_vector(i):
    for j in range(n_particles):
        if j != new_best_particle_index:
            for j in range(n_unknowns):
                new_velocity = (v * velocity_vector[i,j]) + (c1 *
                    np.random.random()) * (particle_best_position[i,j] -
                    particle_position[i,j]) + (c2 * np.random.random()) *
                    (global_best_position[j] - particle_position[i,j])
                velocity_vector[i,j] = new_velocity
                particle_position[i,j] += new_velocity
            else:
                for j in range(n_unknowns):
                    new_velocity = (v * velocity_vector[i,j]) -
                        (particle_position[i,j] - particle_best_position[i,j]) + rho *
                        np.random.uniform(low=-1, high=1)
                    velocity_vector[i,j] = new_velocity
                    particle_position[i,j] += new_velocity

    iteration += 1

print("The best global minimum is", global_best_F_value, "with positions",
      global_best_position)
if c5 == 0:
    print("The minimum area is", y_max*width(global_best_position), "with
          width", "and width", width(global_best_position))

```

$$\frac{\dot{q}_L}{K_3} (x-L) + C_2$$

$$T_m'(x) = - \frac{\dot{q}_L}{K_3}$$

g an temperature:

$$(b-c) + A_2 = - \frac{\dot{q}_{gen}}{2K_2} (b-c)^2 + B_1(b-c) + B_2$$

$$(b+c-L) + C_2 = - \frac{\dot{q}_{gen}}{2K_2} (b+c)^2 + B_1(b+c) + B_2$$

and the Kijou):

$$- \dot{q}_{gen} (b-c) + B_1 K_2$$

$$\dot{q}_{gen} (b+c) - B_1 K_2 \quad \text{and} \quad - \dot{q}_{gen} (b+c) - B_1 K_2$$

$$(A_2 - T_{\infty,1})(b-c) + A_2 = -\frac{e_{\text{gen}}}{2k_2}(b-c)^2 + B_1(b-c) + B_2$$

$$(C_2 - T_{\infty,2})(b+c-L) + C_2 = -\frac{e_{\text{gen}}}{2k_2}(b+c)^2 + B_1(b+c) + B_2$$

мы знаем A_2 и C_2

$$= -\frac{e_{\text{gen}}(b-c)}{h_1} + B_1 \frac{k_2}{h_1} + T_{\infty,1}$$

$$= \frac{e_{\text{gen}}(b+c)}{h_2} - B_1 \frac{k_2}{h_2} + T_{\infty,2}$$

$$\frac{e_{\text{gas}}(b+c)}{K_3} + B_1 \frac{K_2}{K_3} \left((b+c-L) + \frac{e_{\text{gas}}(b+c)}{h_2} - B_1 \frac{K_2}{h_2} + T_{\infty,2} \right)$$

$$= - \frac{e_{\text{gas}}}{2K_2} (b+c)^2 + B_1(b+c) + B_2$$

KKer (7) fms (8)

$$\frac{e_{\text{gas}}}{K_2} \left((b+c)^2 - (b-c)^2 \right) + B_1 \underbrace{\left((b+c) - (b-c) \right)}_{B_1(2c)} + \cancel{B_2} - \cancel{B_2}$$

...

$$2c - \frac{K_2}{K_3} (b+c-L) + \frac{K_2}{h_2} + \frac{K_2}{h_1} + \frac{K_2}{K_1} (b-c)$$

$$\frac{e_{\text{gas}}}{K_2} (b+c)^2 - \frac{e_{\text{gas}}}{K_2} (b+c)(b+c-L) + \frac{e_{\text{gas}}}{K_2} (b-c)^2$$

τ_{13} τ_{14}
 $\uparrow$

$$\underline{+\tau_{12} - \tau_{11}}$$

$$= A_2 + C_2$$

~~~~~

$$x \quad - \frac{(b+c)(b+c-L)}{K_3} = - \frac{(b+c)^2}{K_3} + \frac{(b+c)L}{K_3}$$

$$\frac{e_{gen}(b+c)}{h_2} + \frac{e_{gen}(b-c)}{h_1} - B_1 \frac{K_2}{h_2} - B_1 \frac{K_2}{h_1} + T_{\infty,2} - T_{\infty,1}$$

$$\underbrace{\hspace{10em}}$$

$$- B_1 \left( \frac{K_2}{h_2} + \frac{K_2}{h_1} \right)$$

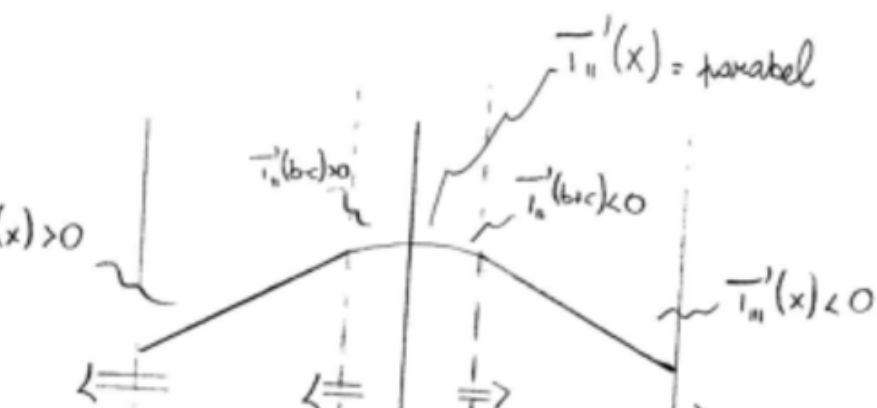
$$\frac{e_{gen}(b+c)(b+c-L)}{K_3} + B_1 \frac{K_2}{K_3} (b+c-L) + \frac{e_{gen}(b-c)^2}{K_1} - B_1 \frac{K_2}{K_1} (b-c)$$

$$e_{gen} \left( \frac{(b+c)}{h_2} + \frac{(b-c)}{h_1} \right) - B_1 \left( \frac{K_2}{h_2} + \frac{K_2}{h_1} \right) + T_{\infty,2} - T_{\infty,1}$$

$$+ 2bc + \cancel{c^2} - \cancel{b^2} + 2bc - \cancel{c^2} = 4bc$$

$$\frac{1}{K_1} T_2(b-c) = \frac{n_1}{K_1} T_{p,1}(b-c) + A_2 + \frac{e_{\text{gau}}}{2K_2} (b-c)^2 - B_1(b-c)$$

$$= \frac{e_{\text{gau}}}{2K_2} (b-c)^2 + \left( \frac{n_1}{K_1} (b-c) + 1 \right) A_2 - \frac{n_1}{K_1} T_{p,1}(b-c) - B_1(b-c)$$



dx

$$\frac{q_L}{K_2} = \underbrace{-\frac{\dot{e}_{\text{gen}}}{K_2}(b+c) + B_1}_{\frac{dT_h}{dx} < 0 \text{ melBmi } b \text{ og } b+c}$$

$$\Rightarrow -\dot{q}_L = -\dot{e}_{\text{gen}}(b+c) + B_1 K_2$$

$$\dot{q}_L = \dot{e}_{\text{gen}}(b+c) - B_1 K_2$$

$$\frac{(b-c)}{h_1} + \bar{T}_{\infty,2} - \bar{T}_{\infty,1}$$

$$\left( \frac{2bc}{K_2} + \frac{\cancel{b^2} - 2bc + \cancel{c^2} - \cancel{b^2} - \cancel{c^2} + bL + cL}{K_1} + \frac{b+c}{h_2} + \frac{b-c}{h_1} \right) + \bar{T}_{\infty,2} - \bar{T}_{\infty,1}$$

$$2c - \frac{K_2}{K_1} (\cancel{b} - \cancel{b} + 2c - L) + \frac{K_2}{h_2} + \frac{K_2}{h_1}$$

$$\left( \frac{2c}{K_2} \cdot b - \frac{4c}{K_1} \cdot b + \frac{L}{K_1} \cdot b + \frac{cL}{K_1} + \frac{b}{h_2} + \frac{c}{h_2} + \frac{b}{h_1} - \frac{c}{h_1} \right) + \bar{T}_{\infty,2} - \bar{T}_{\infty,1}$$

$$2c - \frac{K_2}{K_1} (2c - L) + \frac{K_2}{h_2} + \frac{K_2}{h_1}$$

$$\left( \frac{2c}{K_2} - \frac{4c}{K_1} + \frac{L}{K_1} + \frac{1}{h_2} + \frac{1}{h_1} \right) b + \frac{c}{K_1} \left( \frac{L}{K_1} + \frac{1}{h_2} - \frac{1}{h_1} \right) + \bar{T}_{\infty,2} - \bar{T}_{\infty,1}$$

$$2 \left( 1 - \frac{K_2}{K_1} \right) c + \frac{K_2}{K_1} L + \frac{K_2}{h_2} + \frac{K_2}{h_1}$$

$$\frac{1}{h_1 h_2} = \frac{1}{h_1} + \frac{1}{h_2} \quad c + \infty, 1 + \infty, 2$$

$$= e_{\text{gen}} \left( \frac{1}{h_2} - \frac{1}{h_1} \right) b + K_2 \left( \frac{1}{h_1} - \frac{1}{h_2} \right) \cdot \frac{e_{\text{gen}} \left( 2 \left( \frac{1}{K_2} - \frac{2}{K_1} \right) c + \frac{L}{K_1} + \frac{1}{h_2} + \frac{1}{h_1} \right) b}{2 \left( 1 - \frac{K_2}{K_1} \right) c + K_2 \left( \frac{L}{K_1} + \frac{1}{h_2} + \frac{1}{h_1} \right)}$$

$$+ K_2 \left( \frac{1}{h_1} - \frac{1}{h_2} \right) \cdot \frac{e_{\text{gen}} \left( \frac{L}{K_1} + \frac{1}{h_2} - \frac{1}{h_1} \right) + \overline{T_{\infty, 2}} - \overline{T_{\infty, 1}}}{2 \left( 1 - \frac{K_2}{K_1} \right) c + K_2 \left( \frac{L}{K_1} + \frac{1}{h_2} + \frac{1}{h_1} \right)}$$

$$+ e_{\text{gen}} \left( \frac{1}{h_1} + \frac{1}{h_2} \right) c + \overline{T_{\infty, 1}} + \overline{T_{\infty, 2}}$$

$$= e_{\text{gen}} \left( \frac{1}{h_2} - \frac{1}{h_1} \right) + K_2 \left( \frac{1}{h_1} - \frac{1}{h_2} \right) \cdot \frac{e_{\text{gen}} \left( 2 \left( \frac{1}{K_2} - \frac{2}{K_1} \right) c + \frac{L}{K_1} + \frac{1}{h_2} + \frac{1}{h_1} \right)}{2 \left( 1 - \frac{K_2}{K_1} \right) c + K_2 \left( \frac{L}{K_1} + \frac{1}{h_2} + \frac{1}{h_1} \right)}$$

$$f' = \frac{2AD(Db+E) - AD(D)}{(Db+E)^2} = \frac{2ADb^2 + 2AEb - ADb^2}{(Db+E)^2}$$

$$= \frac{ADb^2 + 2AEb}{(Db+E)^2}$$

$$g' = \frac{3(Db+E) - 3b(D)}{(Db+E)^2} = \frac{3Db^2 + 3E - 3Db^2}{(Db+E)^2} = \frac{3E}{(Db+E)^2}$$

$$h' = \frac{0(Db+E) - F(D)}{(Db+E)^2} = -\frac{DF}{(Db+E)^2}$$

$$\frac{ADb^2 + 2AEb + 3E - DF}{(Db+E)^2}$$

$$= -\frac{\dot{e}_{gen}}{h_1} + \frac{K_2}{h_1} \left( \frac{ADb^2 + 2AEB + BE - DF}{(Db + E)^2} \right)$$

$$= \frac{\dot{e}_{gen}}{h_2} - \frac{K_2}{h_2} \left( \frac{ADb^2 + 2AEB + BE - DF}{(Db + E)^2} \right)$$

$$0 = \underbrace{\dot{e}_{gen} \left( -\frac{1}{h_1} + \frac{1}{h_2} \right)}_G + \underbrace{K_2 \left( \frac{1}{h_1} - \frac{1}{h_2} \right)}_H \left( \frac{ADb^2 + 2AEB + BE - DF}{(Db + E)^2} \right) =$$

$$= G + \frac{\underbrace{(ADH)}_P b^2 + \underbrace{(2AEH)}_Q b + \underbrace{(BEH - DFH)}_R}{(Db + E)^2} = 0$$

$$D^2G)b^2 + (Q + 2DEG)b + (R + GE^2) = 0$$

$$\frac{-(Q + 2DEG) \pm \sqrt{(Q + 2DEG)^2 - 4 \cdot (P + D^2G)(R + GE^2)}}{2(P + D^2G)}$$

$$\underbrace{\left[ e_{\text{gas}} \left( \frac{1}{K_1} - \frac{1}{K_3} \right) K_2 \left( \frac{1}{h_1} - \frac{1}{h_2} \right) + K_2 \left( \frac{1}{K_1} - \frac{1}{K_3} \right) e_{\text{gas}} \left( -\frac{1}{h_1} + \frac{1}{h_2} \right) \right]}_0$$

$$= ADH + \left[ K_2 \left( \frac{1}{K_1} - \frac{1}{K_3} \right) \right]^2 e_{\text{gas}} \left( -\frac{1}{h_1} + \frac{1}{h_2} \right)$$

$$\left( \frac{1}{K_1} - \frac{1}{K_3} \right) K_2 \left( \frac{1}{K_1} - \frac{1}{K_3} \right) K_2 \left( \frac{1}{h_1} - \frac{1}{h_2} \right) + e_{\text{gas}} \left( -\frac{1}{h_1} + \frac{1}{h_2} \right) \left[ K_2 \left( \frac{1}{K_1} - \frac{1}{K_3} \right) \right]^2$$

$$\underbrace{\left( \frac{1}{h_1} - \frac{1}{h_2} \right) \left[ K_2 \left( \frac{1}{K_1} - \frac{1}{K_3} \right) \right]^2 + e_{\text{gas}} \left( -\frac{1}{h_1} + \frac{1}{h_2} \right) \left[ K_2 \left( \frac{1}{K_1} - \frac{1}{K_3} \right) \right]^2}_0$$

$$\frac{1}{K_3} \left[ e_{\text{gen}} \left( \left( \frac{1}{K_1} - \frac{1}{K_3} \right) c + \frac{L}{K_3} - \frac{1}{h_1} + \frac{1}{h_2} \right) c + \overline{T}_{\infty,2} - \overline{T}_{\infty,1} \right] K_2 \left( \frac{1}{h_1} - \frac{1}{h_2} \right)$$

$$\frac{1}{2} \left\{ e_{\text{gen}} \left( 2 \left( -\frac{1}{K_1} + \frac{1}{K_2} - \frac{1}{K_3} \right) c + \frac{L}{K_3} + \frac{1}{h_1} + \frac{1}{h_2} \right) \left( 2 - K_2 \left( \frac{1}{K_1} + \frac{1}{K_3} \right) \right) c \right.$$

$$\left. + e_{\text{gen}} \left( 2 \left( -\frac{1}{K_1} + \frac{1}{K_2} - \frac{1}{K_3} \right) c + \frac{L}{K_3} + \frac{1}{h_1} + \frac{1}{h_2} \right) K_2 \left( \frac{L}{K_3} + \frac{1}{h_1} + \frac{1}{h_2} \right) \right]$$

$$- \left[ e_{\text{gen}} \left( \left( \frac{1}{K_1} - \frac{1}{K_3} \right) c + \frac{L}{3} - \frac{1}{h_1} + \frac{1}{h_2} \right) c \cdot K_2 \left( \frac{1}{K_1} - \frac{1}{K_3} \right) \right.$$

$$\left. + K_2 \left( \frac{1}{K_1} - \frac{1}{K_3} \right) (\overline{T}_{\infty,2} - \overline{T}_{\infty,1}) \right]$$

$$\left( \frac{1}{K_1} - \frac{1}{K_3} \right) \left( \frac{1}{h_1} - \frac{1}{h_2} \right)$$

$$\left[ \dots \right]^2$$

$n_3 \quad n_4 \quad n_2 / \quad (K_1 \quad K_3) \sim$

$$+ e_{gen} \left( 2 \left( -\frac{1}{K_1} + \frac{1}{K_2} - \frac{1}{K_3} \right) c + \frac{L}{K_3} + \frac{1}{h_1} + \frac{1}{h_2} \right) K_2 \left( \frac{L}{K_3} + \frac{1}{h_1} + \frac{1}{h_2} \right)$$

$$- e_{gen} \left( \left( \frac{1}{K_1} - \frac{1}{K_3} \right) c + \frac{L}{K_3} - \frac{1}{h_1} + \frac{1}{h_2} \right) K_2 \left( \frac{1}{K_1} - \frac{1}{K_3} \right) c$$

$$- K_2 \left( \frac{1}{K_1} - \frac{1}{K_3} \right) (\overline{T}_{\infty,2} - \overline{T}_{\infty,1}) \Bigg]$$

$$\frac{1}{h_2} \Bigg) \left[ 2 e_{gen} \left( -\frac{2}{K_1} + \frac{2}{K_2} - \frac{2}{K_3} \right) c + \frac{L}{K_3} + \frac{1}{h_1} + \frac{1}{h_2} \right)$$

$$- e_{gen} \left( \left( -\frac{2}{K_1} + \frac{1}{K_1} + \frac{1}{K_2} - \frac{2}{K_3} - \frac{1}{K_3} \right) c \right.$$

$$1 \quad h_1 \quad h_2 / \quad 1 \quad \dots \quad 27$$

$$a \left( b \left( -\frac{1}{h_1} + \frac{1}{h_2} \right) + c \left( \frac{1}{h_1} + \frac{1}{h_2} \right) \right) + \beta_2 \left( K_2 \left( \frac{1}{h_1} - \frac{1}{h_2} \right) \right) + \overline{T}_{\infty,1} + \overline{T}_{\infty,2}$$

$$a \left( -b \left( \frac{1}{h_1} - \frac{1}{h_2} \right) + c \left( \frac{1}{h_1} + \frac{1}{h_2} \right) \right) +$$

$$a \left( \frac{2bc}{K_2} - \frac{(b+c)^2}{K_3} + \frac{(b+c)L}{K_3} + \frac{(b-c)^2}{K_1} + \frac{(b+c)}{h_2} + \frac{(b-c)}{h_1} \right) - \overline{T}_{\infty,1} + \overline{T}_{\infty,2}$$

$$\underbrace{2c - \frac{K_2(b+c)}{K_3} + \frac{K_2 L}{K_3} + \frac{K_2(b-c)}{K_1} + K_2 \left( \frac{1}{h_1} + \frac{1}{h_2} \right)}_{\text{}} + K_2 \left( -\frac{(b+c)}{K_3} + \frac{L}{K_3} + \frac{(b-c)}{K_1} \right)$$

$$K_2 \left( \frac{1}{h_1} - \frac{1}{h_2} \right) \cdot \overline{T}_{\infty,1} - \overline{T}_{\infty,2}$$

---


$$\frac{\frac{L}{h_2} + c\left(\frac{1}{h_1} + \frac{1}{h_2}\right) + \left(\frac{\frac{2bc}{k_3} - \frac{(b+c)^2}{k_3} + \frac{(b+c)}{k_3} L + \frac{(b-c)^2}{k_1} + \frac{(b+c)}{h_2}\right) + \left(\frac{\frac{2c}{k_2} - \frac{(b+c)}{k_3} + \frac{L}{k_3} + \frac{(b-c)}{k_1} + \left(\frac{1}{h_1} - \frac{1}{h_2}\right)\right)}{+ \overline{1\omega_1,1} + \overline{1\omega_1,2}}$$

$$\frac{\left(\frac{1}{h_1} - \frac{1}{h_2}\right)}{+ \frac{L}{k_3} + \frac{(b-c)}{k_1} + \left(\frac{1}{h_1} + \frac{1}{h_2}\right)}$$

$$b \rightarrow b = L - c :$$

$$\left( \frac{1}{h_1} - \frac{1}{h_2} \right) + c \left( \frac{1}{h_1} + \frac{1}{h_2} \right) + \left( \frac{2(L-c)c}{K_2} - \frac{(L-\cancel{\infty+c})^2}{K_3} + \frac{(L-\cancel{\infty+c})}{K_3} L + \frac{(L-c)}{K_1} \right)$$

$$\frac{2c}{K_2} - \cancel{\frac{1}{K_3}} + \cancel{\frac{1}{K_3}} + \frac{(L-2c)}{K_1}$$

$$+ (-1 \omega_{1,1} + 1 \omega_{1,2}) \left[ \frac{\left( \frac{1}{h_1} - \frac{1}{h_2} \right)}{\frac{2c}{K_2} - \cancel{\frac{1}{K_3}} + \cancel{\frac{1}{K_3}} + \frac{(L-2c)}{K_1} + \left( \frac{1}{h_1} + \frac{1}{h_2} \right)} \right] + 1 \omega$$

$$\left( \frac{1}{h_2} \right) + \frac{2c}{h_1} + \left( \frac{2Lc - 2c^2 + (L-2c)^2}{K_2 K_2 K_1} + \frac{L}{h_2} + \frac{(L-2c)}{h_1} \right) \left( \frac{1}{h_1} - \frac{1}{h_2} \right) + \left( \frac{L}{K_1} + 2c \left( -\frac{1}{h_1} + \frac{1}{h_2} \right) + \left( \frac{1}{h_1} + \frac{1}{h_2} \right) \right)$$

+

$$= + \frac{Lc^2}{K_2} + \frac{2Lc - 2c^2}{K_2} + \frac{L - 2c}{h_1} + \frac{L}{h_2}$$

2 and  $\rightarrow b = c :$

$$\left( \frac{1}{h_1} - \frac{1}{h_2} \right) + c \left( \frac{1}{h_1} + \frac{1}{h_2} \right) + \left( \frac{\frac{2c^2}{k_2} - \frac{4c^2}{k_3}}{k_3} + \frac{\frac{2c}{k_3} L}{k_3} + 0 + \frac{2c}{h_2} + 0 \right) \left( \frac{\frac{2c}{k_2} - \frac{2c}{k_3} + \frac{L}{k_3} + 0 + \left( \frac{1}{h_1} + \frac{1}{h_2} \right)}{\right)$$

$$\left( \frac{1}{h_1} - \frac{1}{h_2} \right) \left[ \frac{\frac{2c}{k_2} - \frac{2c}{k_3} + \frac{L}{k_3} + 0 + \left( \frac{1}{h_1} + \frac{1}{h_2} \right)}{\right] + \overline{1_{\infty,1}} + \overline{1_{\infty,2}}$$

$$\left( \frac{2c}{k_2} + \frac{L - 2c}{k_3} + \frac{1}{h_2} \right) \left( \frac{1}{h_1} - \frac{1}{h_2} \right) \left[ \frac{2c \left( \frac{1}{k_2} - \frac{1}{k_3} \right) + \frac{L}{k_3} + \left( \frac{1}{h_1} + \frac{1}{h_2} \right)}{\right] + \left( \overline{1_{\infty,1}} + \overline{1_{\infty,2}} \right)$$

$$+ \overline{1_{\infty,1}} + \overline{1_{\infty,2}}$$

Wiggen plem on  $f(c) = f(L-c)$ . Da:

$$\left( \frac{1}{h_1} - \frac{1}{h_2} \right) + \left( \frac{\frac{2c^2}{k_2} + 2Lc - Lc^2 + \frac{2c}{h_2}}{k_2 k_3} - \frac{\frac{L^2 - 4Lc + 4c^2}{k_1} + \frac{2Lc}{k_1}}{2c \left( \frac{1}{k_2} - \frac{1}{k_3} \right) + \frac{L}{k_3} + \left( \frac{1}{h_1} + \frac{1}{h_2} \right)} \right)$$

$$\left( \frac{1}{h_1} - \frac{1}{h_2} \right) - \frac{\frac{1}{h_1} - \frac{1}{h_2}}{2c \left( \frac{1}{k_1} + \frac{1}{k_2} \right) + \frac{L}{k_1} + \left( \frac{1}{h_1} + \frac{1}{h_2} \right)} - \frac{\frac{1}{h_1} - \frac{1}{h_2}}{2c \left( \frac{1}{k_2} - \frac{1}{k_3} \right) + \frac{L}{k_3} + \left( \frac{1}{h_1} + \frac{1}{h_2} \right)}$$

$$\left( \frac{1}{h} \right) \left[ \frac{1}{2c \left( -\frac{1}{k_1} + \frac{1}{k_2} \right) + \frac{L}{k_1} + \left( \frac{1}{h_1} + \frac{1}{h_2} \right)} - \frac{1}{2c \left( \frac{1}{k_2} - \frac{1}{k_3} \right) + \frac{L}{k_3} + \left( \frac{1}{h_1} + \frac{1}{h_2} \right)} \right]$$

$$\overline{1_{\infty,2}} \left[ \frac{1}{2c \left( \frac{1}{k_2} - \frac{1}{k_1} \right) + \frac{L}{h_1} + \left( \frac{1}{h_1} + \frac{1}{h_2} \right)} - \frac{1}{2c \left( \frac{1}{k_2} - \frac{1}{k_3} \right) + \frac{L}{h_3} + \left( \frac{1}{h_1} + \frac{1}{h_2} \right)} \right]$$

$$-2c) + \frac{\frac{2c^2}{k_2} + \frac{2Lc - 4c^2}{k_3} + \frac{2c}{h_2}}{\frac{2c \left( \frac{1}{k_2} - \frac{1}{k_3} \right) + \frac{L}{h_3} + \left( \frac{1}{h_1} + \frac{1}{h_2} \right)}} - \frac{L^2 - 4Lc + 4c^2}{k_1} \cdot \frac{2c \left( \frac{1}{k_2} - \frac{1}{k_1} \right) + \frac{L}{k_1}}{2c \left( \frac{1}{k_2} - \frac{1}{k_3} \right) + \frac{L}{h_3} + \left( \frac{1}{h_1} + \frac{1}{h_2} \right)}$$