Alternative Nelder Mead Algorithm

Code based on "Evolving a Nelder-Mead Algorithm for Optimization with Genetic Programming", by Iztok Fajfar, Janez Puhan & Árpád Bűrmen

https://direct.mit.edu/evco/article/25/3/351/1046/Evolving-a-Nelder-Mead-Algorithm-for-Optimization

Alternate Nelder Mead

NelderMead Alternative code

Using 4 starting points instead the classical 3 just because it's easy relate the simplex with a box for plots.

```
NMA(f(1), B, O) :=
    S := [ [B_1; B_3]  [B_1; B_4]  [B_2; B_3]  [B_2; B_4] ]
    [ κ := [1 1.375 2 0.625]   [εy εx h MI] := O ]
    [ n := length(S)   r := [1..n]   C(x#) := 1/n · (Σ x#) ]
    [ x := S_r   F_r := f(x_r)   X := 0 ]
    for iter ∈ [1..MI]
        Fx := csort(augment(F, x), 1)
        [ F := col(Fx, 1)   x := col(Fx, 2) ]
        if (normi(F − C(F)) < εy) ∧ (normi(x_1 − x_n) < εx)
            break
        else
            xc := C(x)
            if f(xc + κ_1·(xc − x_n)) < F_n
                if f(xc + κ_3·(xc − x_n)) < f(xc)
                    xn := xc + κ_2·(xc − x_n)
                else
                    xn := xc + κ_1·(xc − x_n)
            else
                xn := xc − κ_4·(xc − x_n)
            [ x_n := xn   F_n := f(xn) ]
    x_1
```

Pseudocode from the above reference

Order the simplex vertices.
if $f(\boldsymbol{c}+1(\boldsymbol{c}-\boldsymbol{v}_{\mathrm{w}})) < f(\boldsymbol{v}_{\mathrm{w}})$ **then**
  if $f(\boldsymbol{c}+2(\boldsymbol{c}-\boldsymbol{v}_{\mathrm{w}})) < f(\boldsymbol{c})$ **then**
    $\boldsymbol{v}_{\mathrm{new}} = \boldsymbol{c}+1.375(\boldsymbol{c}-\boldsymbol{v}_{\mathrm{w}})$
  else
    $\boldsymbol{v}_{\mathrm{new}} = \boldsymbol{c}+1(\boldsymbol{c}-\boldsymbol{v}_{\mathrm{w}})$
  end if
else
  $\boldsymbol{v}_{\mathrm{new}} = \boldsymbol{c}-0.625(\boldsymbol{c}-\boldsymbol{v}_{\mathrm{w}})$
end if
Replace $\boldsymbol{v}_{\mathrm{w}}$ with $\boldsymbol{v}_{\mathrm{new}}$.

Plot Utilities

Examples

Examples

$B = \begin{bmatrix} x1 & x2 \\ y1 & y2 \end{bmatrix}$ Box for plots and guess values

$NP := \begin{bmatrix} 80 \\ 80 \end{bmatrix}$ Points for contours $NP = \begin{bmatrix} nx \\ ny \end{bmatrix}$

$NSol := \begin{bmatrix} 10^{-15} & 10^{-9} & 0 & 200 \end{bmatrix}$ Options for NM Methods $O = \begin{bmatrix} \varepsilon y & \varepsilon x & h & MI \end{bmatrix}$

$g := pCmapJet(200, 0.9)$ Colormap

zo Exact solution for the example

Shorthands

$\varphi(x) := f(x_1, x_2)$

```
Plot(f, zo, za, zc) := { pFillContour(pGrid(f, B, NP), g)
                         augment(za_1, za_2, "x", 8, "yellow")
                         augment(zc_1, zc_2, "x", 8, "white")
                         augment(zo_1, zo_2, "o", 8, "red") }
```

$NMC(\varphi(1), B, O) := \left|\ \text{NelderMead}\left(\varphi(x\#),\ \text{eval}\left(\frac{1}{2}\cdot\begin{bmatrix} B_1 + B_2 \\ B_3 + B_4 \end{bmatrix}\right),\ O_1\right)\right.$

Classical NelderMead Method from Nonlinear Solvers plugin using the center of the box as guess instead a simplex.

For an actual comparation between codes they must to start from the same simplex, but it isn't the purpose here, it just to introduce the new algorithm for the method and see if it "works".

$f(x, y) := 4\cdot(x-5)^2 + (y-6)^2$

$B := \begin{bmatrix} -5 & 10 \\ -5 & 10 \end{bmatrix} \quad zo := \begin{bmatrix} 5 \\ 6 \end{bmatrix}$

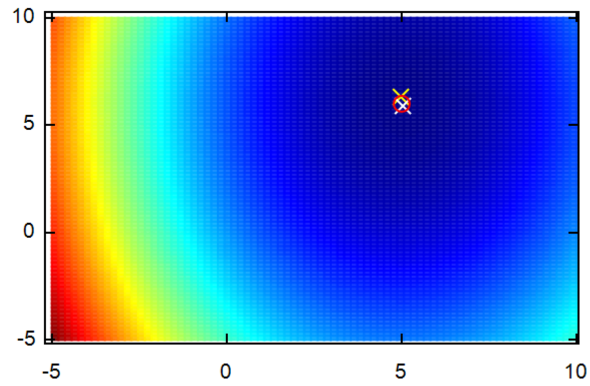
$za := NMA(\varphi(x), B, \text{NSol})$

$zc := NMC(\varphi(x), B, \text{NSol})$

$\varphi(zo) = 0$

$\varphi(za) = 0.103281213545949$

$\varphi(zc) = 0.015543958361777$



Plot ("f", zo, za, zc)

$z_1 := -13 + x - 2\cdot y + 5\cdot y^2 - y^3$

$z_2 := -29 + x - 14\cdot y + y^2 + y^3$

$f(x, y) := z_1^2 + z_2^2$

$B := \begin{bmatrix} 3 & 6 \\ 3 & 6 \end{bmatrix} \quad zo := \begin{bmatrix} 5 \\ 4 \end{bmatrix}$

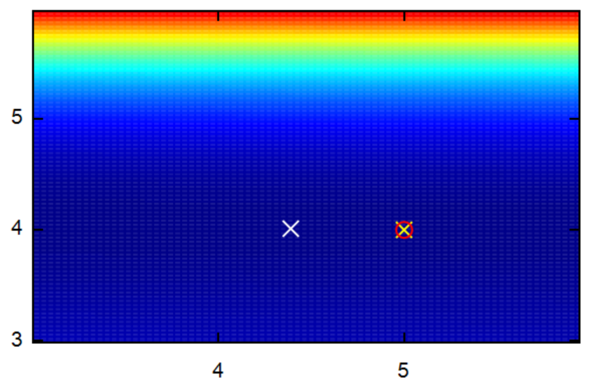
$za := NMA(\varphi(x), B, \text{NSol})$

$zc := NMC(\varphi(x), B, \text{NSol})$

$\varphi(zo) = 0$

$\varphi(za) = 0.000000088906783$

$\varphi(zc) = 0.543022665741093$



Plot ("f", zo, za, zc)

$z_1 := (1.5 - x\cdot(1-y))^2$

$z_2 := \left(2.25 - x\cdot\left(1-y^2\right)\right)^2 + \left(2.625 - x\cdot\left(1-y^3\right)\right)^2$

$f(x, y) := z_1 + z_2$

$B := \begin{bmatrix} 0 & 4 \\ 0 & 4 \end{bmatrix} \quad zo := \begin{bmatrix} 3 \\ 0.5 \end{bmatrix}$

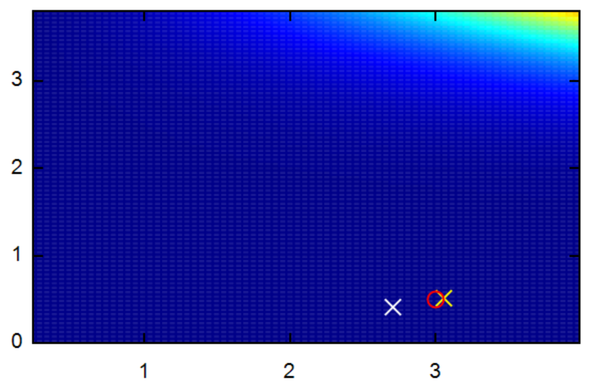
$za := NMA(\varphi(x), B, \text{NSol})$

$zc := NMC(\varphi(x), B, \text{NSol})$

$\varphi(zo) = 0$

$\varphi(za) = 0.000480493753523$

$\varphi(zc) = 0.020060444564884$



Plot ("f", zo, za, zc)

$f(x, y) := \left(x - y^2\right)^2 + 100 \cdot (1 - x)^2$

$B := \begin{bmatrix} 0 & 3 \\ 0 & 3 \end{bmatrix}$ $zo := \begin{bmatrix} 1 \\ 1 \end{bmatrix}$

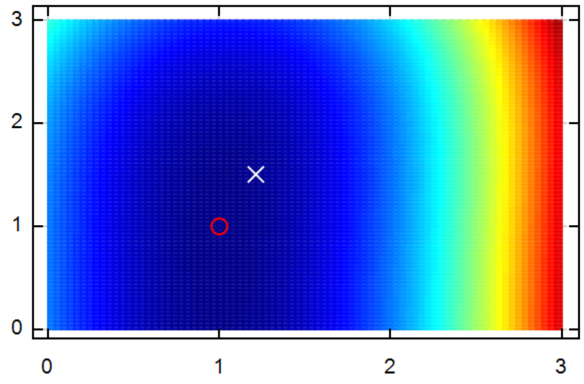
$za := NMA(\varphi(x), B, \text{NSol})$

$zc := NMC(\varphi(x), B, \text{NSol})$

$\varphi(zo) = 0$

$\varphi(za) = 1.61304251090827 \cdot 10^{-15}$

$\varphi(zc) = 5.65334548004554$



Plot("f", zo, za, zc)

Alvaro