```
begin
    using HiGHS
    using JuMP    , BenchmarkTools
    using PlotlyJS    , LinearAlgebra    , Distributions    ,Random
end
```

Considere uma rede 2D com n nós beacon em posições conhecidas $x_i \in \mathbb{R}^2, i = 1, \ldots, n$

Assumimos uma coordenada $x^k = (x, y)$ como uma posição desconhecida inicial

O nó alvo x^k vai transmitir um sinal no instante t_0 e os nós beacon irão receber o sinal no instante t_i conforme a expressão abaixo

$$t_i = t_0 + \frac{d_i}{v} + \omega_i$$

```
md"
$tᵢ = t₀ + \frac{dᵢ}{v} + \omega ᵢ$
"
```

em que $d_i = \sqrt[2]{(x_i - x)^2 + (y_i - y)^2}$ conforme a distancia euclidiana entre o beacon i e o ponto x, sendo v a velocidade de propagação do sinal

$\omega_i \approx n(0, \frac{\sigma_i{}^2}{v^2})$

Supondo que t_0 é um parâmetro desconhecido fixo durante o processo. Para evitar a influência deste parâmetro, subtraimos todas as medições t_i da última medição. Consequentemente, pode ser expresso como

$$t_i - t_n = (t_0 + \frac{d_i}{v} + \omega_i) - (t_0 + \frac{d_n}{v} + \omega_n)$$

$$t_i - t_n = \frac{d_i + d_n}{v} + \omega_i - \omega_n$$

$$v(t_i - t_n) = d_i + d_n + v(\omega_i - \omega_n)$$

incluindo variaveis para cada parte da equação

$$r_i = v(t_i - t_n)$$

$$d_{i,n} = d_i + d_n$$

$$X_i = v(\omega_i - \omega_n)$$

obtivemos a seguinte expressão

$$R = D + X$$

sendo R a distância entre os pontos fixos e o ponto alvo (x_k) considerando os erros, obtemos

$$R = [r_1, r_2, \ldots, r_n]$$

$$D = [d_{1,n}, \ldots, d_{n-1,n}]$$

$$X = [X_1, \ldots, X_{n-1}]$$

consideramos a minimização da norma ι_1 abaixo

$$\underset{x}{\text{minimize}}\left\| R - D \right\|_1$$

```
md"
$\underset{x}{\text{minimize}}\left \| R - D \right \|_1$
"
```

A norma ι_1 é um problema de otimização não convexo e difícil de resolver, se propõe uma solução eficiente com um método subótimo para resolver este problema. Descrevemos o procedimento de aproximação da norma ι_1 com minimização de resíduos em um problema de otimização convexo, usando um vetor fictício s.

$$s = [s_1, \ldots, s_{n-1}]^T$$

$$\underset{x,s}{\text{minimize}} \sum_{n-1}^{i=1} s_i$$

sujeito a $s_i = r_i + \|x - x_n\|_2 - \|x - x_i\|_2$

O problema em questão é não convexo e não linear. No caso de relações sinal-ruído suficientemente altas, a restrição acima pode ser aproximado usando a expansão em série de Taylor de primeira ordem, de modo que podemos estudar como a função f se comporta em torno de um ponto fixo x^k

$$\begin{aligned} \min \quad & \sum_{i=1}^{n-1} s_i \\ \text{s. a.} \quad & A_i x + C_i - s_i = 0 \end{aligned}$$

conforme a expansão de taylor de primeira ordem:

$$f(x) = f(x_0) + \frac{f'(x_0)}{1!}(x - x_0)$$

Sendo assim temos $s_i(x_k) = r_i + \|x_k - x_n\|_2 - \|x_k - x_i\|_2$

$$s_i(x_k)' = \frac{x_k - x_n}{\|x_k - x_n\|_2} - \frac{x_k - x_i}{\|x_k - x_i\|_2}$$

substituindo na expansão de taylor:

$$f(x) = r_i + \|x_k - x_n\|_2 - \|x_k - x_i\|_2 + (\frac{x_k - x_n}{\|x_k - x_n\|_2} - \frac{x_k - x_i}{\|x_k - x_i\|_2})(x - x_k)$$

$$f(x) = r_i + \|x_k - x_n\|_2 - \|x_k - x_i\|_2 + x\frac{x_k - x_n}{\|x_k - x_n\|_2} - x\frac{x_k - x_i}{\|x_k - x_i\|_2}$$

$$-x_k\frac{x_k - x_n}{\|x_k - x_n\|_2} + x_k\frac{x_k - x_i}{\|x_k - x_i\|_2}$$

incluindo variaveis para cada parte da equação

$$A_i = (\frac{x_k - x_n}{\|x_k - x_n\|_2} - \frac{x_k - x_i}{\|x_k - x_i\|_2})$$

$$C_i = r_i + \|x_k - x_n\|_2 - \|x_k - x_i\|_2 - x_k\frac{x_k - x_n}{\|x_k - x_n\|_2} + x_k\frac{x_k - x_i}{\|x_k - x_i\|_2}$$

Relaxando a minimização anterior obtivemos:

$$A_i x + C_i - S_i \leq 0$$

calcula_rᵢ (generic function with 1 method)

```
function calcula_rᵢ(n,σ,xₖ,X,v)
    ω = similar(σ)
    for (index,σi) in enumerate(σ)
        ω[index] = rand(Normal(0,σi/v))
    end
    ωₙ = pop!(ω)
    ω .-= ωₙ
    ω *= v
    D = ((norm.(X .- [xₖ]))[1:end-1].-(norm.(X .- [xₖ]))[end])
    R = D + ω
    return R
end
```

solucao_LP (generic function with 1 method)

```
function solucao_LP(A, C, n)
    IC = Model(HiGHS.Optimizer)
    @variable(IC,x[1:2]>=0)
    @variable(IC,sᵢ[1:n-1]>=0)
    @objective(IC, Min,sum(sᵢ))
    @constraint(IC,(A*x)+C-sᵢ .== 0)
    print(IC)
    optimize!(IC)
    println("Termination status: $(termination_status(IC))")
    if termination_status(IC) == MOI.OPTIMAL
        println("Optimal objective value: $(objective_value(IC))")
        println("x: ",value.(x))
    else
        @constraint(IC,(A*x)+C-sᵢ .<= 0)
        print(IC)
        optimize!(IC)
        println("Termination status: $(termination_status(IC))")
        if termination_status(IC) == MOI.OPTIMAL
            println("Optimal objective value: $(objective_value(IC))")
            println("x: ",value.(x))
            println("s: ",value.(sᵢ))
        else
            println("No optimal solution available")
        end
    end
end
```

```
let
function Input(prompt)
    print(prompt)
    readline()
end

n = Input("valores de X, digitar [[12.,12],[12.,12]]\n")
println("Your name is $n.")
end
```

```
valores de X, digitar [[12.,12],[12.,12]]
Your name is .
```

```
let

    ######MAIN###########
    Random.seed!(10)
    X = [[0.,0],[100,100.],[0,100.],[100,0.]]
    n = length(X)
    xₖ = [111,99]
    v = 20
    σ = Float64[.5,2,2.0,1]
    R = calcula_rᵢ(n,σ,xₖ,X,v)
    Aᵢ(xᵢ) = ((xₖ - X[end])/norm(xₖ - X[end]))  - ((xₖ - xᵢ)/norm(xₖ - xᵢ))
    Cᵢ(xᵢ) = norm(xₖ - X[end]) -  norm(xₖ - xᵢ) + dot(Aᵢ(xᵢ),-xₖ)
    C = R +  Cᵢ.(X[1:end-1])
    A = Aᵢ.(X[1:end-1])
    A = permutedims(hcat(A...))
    solucao_LP(A,C,n)

end
```

```
Min sᵢ[1] + sᵢ[2] + sᵢ[3]
Subject to
 -0.6358639083936645 x[1] + 0.32826888771521723 x[2] - sᵢ[1] = -38.480839190
290226
 -0.8854616803928573 x[1] + 1.0844194807161374 x[2] - sᵢ[2] = 8.951405820934
113
 -0.8895278952735653 x[1] + 1.00289237810919 x[2] - sᵢ[3] = -0.6220942608272
448
 x[1] ≥ 0.0
 x[2] ≥ 0.0
 sᵢ[1] ≥ 0.0
 sᵢ[2] ≥ 0.0
 sᵢ[3] ≥ 0.0
Running HiGHS 1.3.0 [date: 1970-01-01, git hash: e5004072b-dirty]
Copyright (c) 2022 ERGO-Code under MIT licence terms
Presolving model
3 rows, 5 cols, 9 nonzeros
2 rows, 4 cols, 6 nonzeros
2 rows, 4 cols, 6 nonzeros
Presolve : Reductions: rows 2(-1); columns 4(-1); elements 6(-3)
Solving the presolved LP
Using EKK dual simplex solver - serial
  Iteration        Objective     Infeasibilities num(sum)
          0     0.0000000000e+00 Pr: 2(253.011) 0s
          2     9.9040873089e-01 Pr: 0(0) 0s
Solving the original LP from the solution after postsolve
Model   status      : Optimal
Simplex   iterations: 2
Objective value     :  9.9040873089e-01
HiGHS run time      :          0.00
Termination status: OPTIMAL
Optimal objective value: 0.9904087308934351
x: [111.98492400616007, 99.6936764163446]
```

```
##http://www.cnmac.org.br/novo/index.php/CNMAC/conteudo/2022/53/99
```