BIO2045 – Séance 4

Optimisation par colonies de fourmis

Contenu

Concepts principaux

Commentaires

Documentation

Passage par référence

Fonctions qui modifient leurs arguments

Modèle

```julia
using CairoMakie
using Statistics
using ProgressMeter
using StatsBase
CairoMakie.activate!(px_per_unit = 6.0)
```

LOL

```julia
function disposition_points(n)
    angles = rand(n) .* 2π
    radii = sqrt.(rand(length(angles)) .* 3.0 .+ 1.0)
    x = cos.(angles) .* radii
    y = sin.(angles) .* radii

    stops = permutedims(hcat(x, y))
    return stops
end

points = 100

P = zeros(Float64, points, points)
D = zeros(Float64, points, points)

xy = disposition_points(points)

for i in 1:points
    for j in 1:points
        D[i,j] = sqrt(sum((xy[:,i] .- xy[:,j]) .^2.0))
    end
end

scatter(xy)

function walk_on_graph(D, P, i)
    α = 0.9
    β = 1.5
    n_sites = size(D, 1)

    visites = zeros(Bool, n_sites)
    visites[i] = true

    chemin = [i]

    while sum(visites) != n_sites
        voisins = Int64[]
        cible = Float64[]
        for voisin in setdiff(1:n_sites, chemin)
            pheromones = max(P[last(chemin), voisin], 1e-5)
            poids = (pheromones^α)/D[last(chemin), voisin]^β
            push!(cible, poids)
            push!(voisins, voisin)
        end

        next_site = sample(voisins, Weights(cible))
        push!(chemin, next_site)
        visites[next_site] = true
    end
    return chemin
end

"""
    chemin_distance(chemin, D)

Distance totale du chemin, incluant le retour au point initial

- `chemin`: vecteur de positions qui indique l'ordre de visite
des sites
- `D`: matrice de distance entre les sites
"""
function chemin_distance(chemin, D)
```

```
ERROR: ParseError:
# Error @ /home/tpoisot/Documents/SimulerLeVivant/
output/04_fourmis.md:61:36
"""
function chemin_distance(chemin, D)
#                                  └ ── premature end of input
```

d représente la distance pour revenir au début du cycle

on veut créer un circuit, et le chemin s'arrête au dernier site visité

le chemin indique l'ordre de visite des sites

D est la matrice de distance entre tous les points

```
    d = D[chemin[end], chemin[begin]]
```

```
ERROR: UndefVarError: `chemin` not defined in `Main.var"##295"`
Suggestion: check for spelling errors or missing imports.
```

on va calculer la distance totale du reste de chemin on a deja la distance du retour vers le premier site

```
    for i in 2:length(chemin)
```

```
ERROR: ParseError:
# Error @ /home/tpoisot/Documents/SimulerLeVivant/
output/04_fourmis.md:1:30
    for i in 2:length(chemin)
#                            └ ── premature end of input
```

on lit dans D la distance entre le site visité à la position i et le site visité juste avant (position i-1)

on ajouter cette distance à d

```
        d += D[chemin[i-1], chemin[i]]
    end
```

```
ERROR: UndefVarError: `d` not defined in `Main.var"##295"`
Suggestion: add an appropriate import or assignment. This global
was declared but not assigned.
```

d contient la distance totale du cycle distance du chemin + distance du retour au premier point on renvoie d

```
    return d
end

function pheromones!(P, chemin, D)
    Q = size(P, 1)
    score = Q / chemin_distance(chemin, D)
    P[chemin[end], chemin[1]] += score
    for i in 2:length(chemin)
        P[chemin[i-1], chemin[i]] += score
    end
    return P
end

track = zeros(Float64, 120)
n_fourmis = 50
evaporation_rate = 0.99

@showprogress for i in 1:length(track)

    chemins = [walk_on_graph(D, P, rand(1:points)) for _ in
1:n_fourmis]
```

```
ERROR: UndefVarError: `d` not defined in `Main.var"##295"`
Suggestion: add an appropriate import or assignment. This global
was declared but not assigned.
```

Remove the chemins with more than the median chemin_distance

```
    distances = [chemin_distance(chemin, D) for chemin in
chemins]
    median_distance = median(distances)
    chemins = chemins[findall(distances .<= median_distance)]

    for chemin in chemins
        pheromones!(P, chemin, D)
    end
    P ./= length(chemins)
    #P .+= rand(size(P)).*0.02 .- 0.01
    P .*= evaporation_rate

    track[i] = minimum(distances)
```

```
end
```

```
ERROR: UndefVarError: `chemins` not defined in `Main.var"##295"`
Suggestion: check for spelling errors or missing imports.
```

Plotting the points and lines colored by the value of P

```
fig = Figure(size=(600, 680))
gl = fig[1,1] = GridLayout()
ax = Axis(gl[1, 1], aspect=1)
ax2 = Axis(gl[2,1], yscale=sqrt)
lines!(ax2, track, color=:purple, linewidth=2)
hidedecorations!(ax2)

scatter!(ax, xy[1, :], xy[2, :], color=:black)
hidespines!(ax)
hidedecorations!(ax)

rowsize!(gl, 2, Relative(0.2))
fig
```

```
ERROR: UndefVarError: `track` not defined in `Main.var"##295"`
Suggestion: check for spelling errors or missing imports.
```