Optimisation de fonctions à haut coût

Ulysse Côté Allard, Gabriel Dubé, Richard Khoury, Luc Lamontagne, Benoit Gosselin et François Laviolette.

Time Adaptive dual particle swarm optimization

But

Minimisation globale d'une fonction quelconque.

Tour d'horizon des algos PSO et présentation de celui que nous avons développé.

Particle Swarm Optimization

Méthode d'optimisation basée sur le mouvement des groupes d'oiseaux/poissons.

Proposé par Kennedy en 1995.

Terminologie

Fonction d'évaluation : La fonction à minimiser. Elle prend un vecteur de D dimensions

Position : Point dans l'espace de recherche. C'est l'entrée de la fonction d'évaluation.

Aptitude (fitness) : Résultat de la fonction d'évaluation en un point

Maison: Meilleure position trouvée par une particule (fitness la plus basse)

Particule : Un ensemble comportant la position actuelle, la fitness actuelle, la maison et la meilleure fitness trouvée ainsi qu'une vélocité.

Population : Regroupement de particules

Maison du « rich kid »: Meilleure position trouvée par l'ensemble de la swarm

Exemplar: Une position vers laquelle une créature se dirige.

Particle Swarm Optimization

Générer N particules avec des positions et vélocité aléatoires. Évaluer leur fitness.

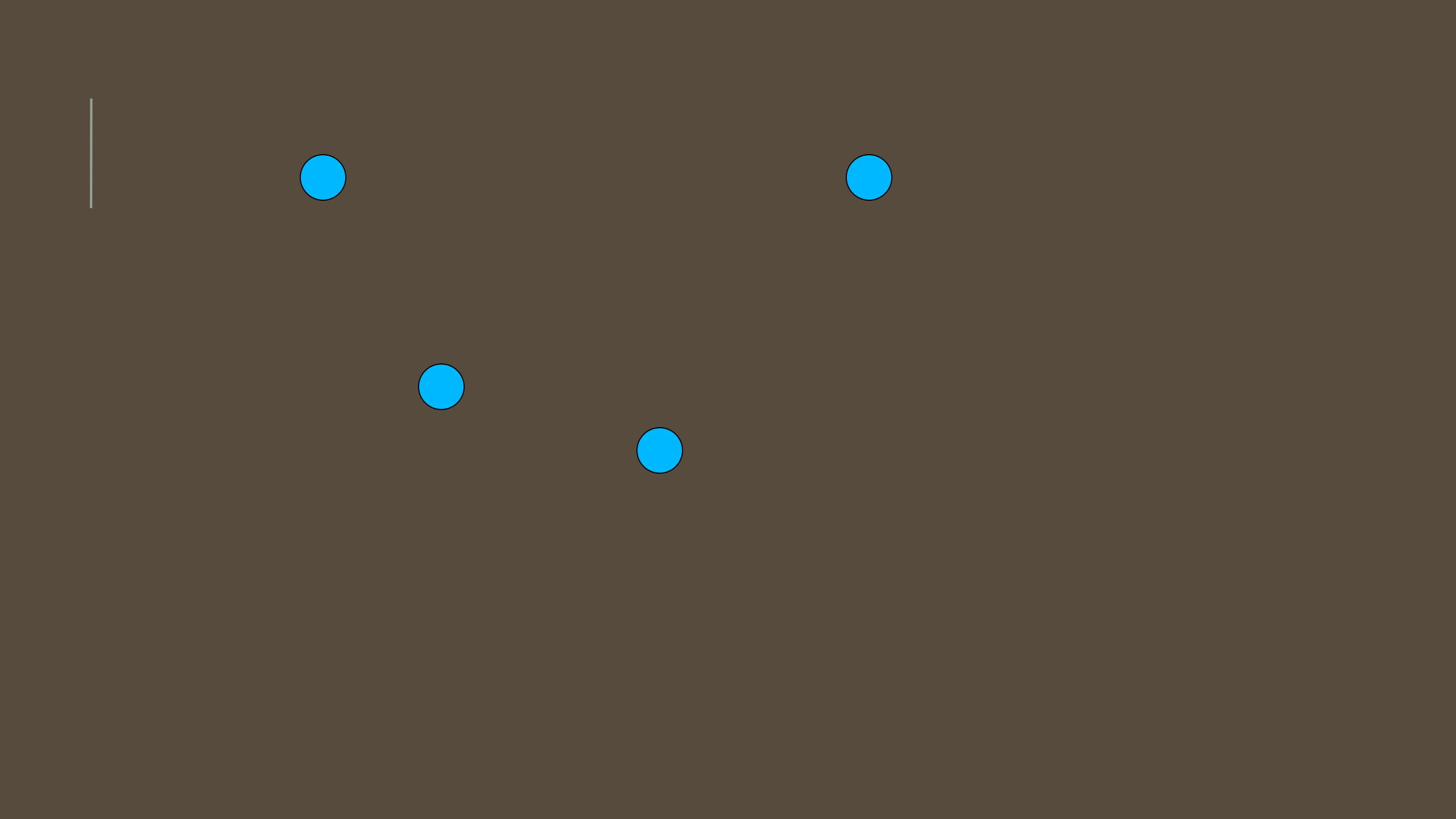
À chaque itération: Calculer la nouvelle vélocité:

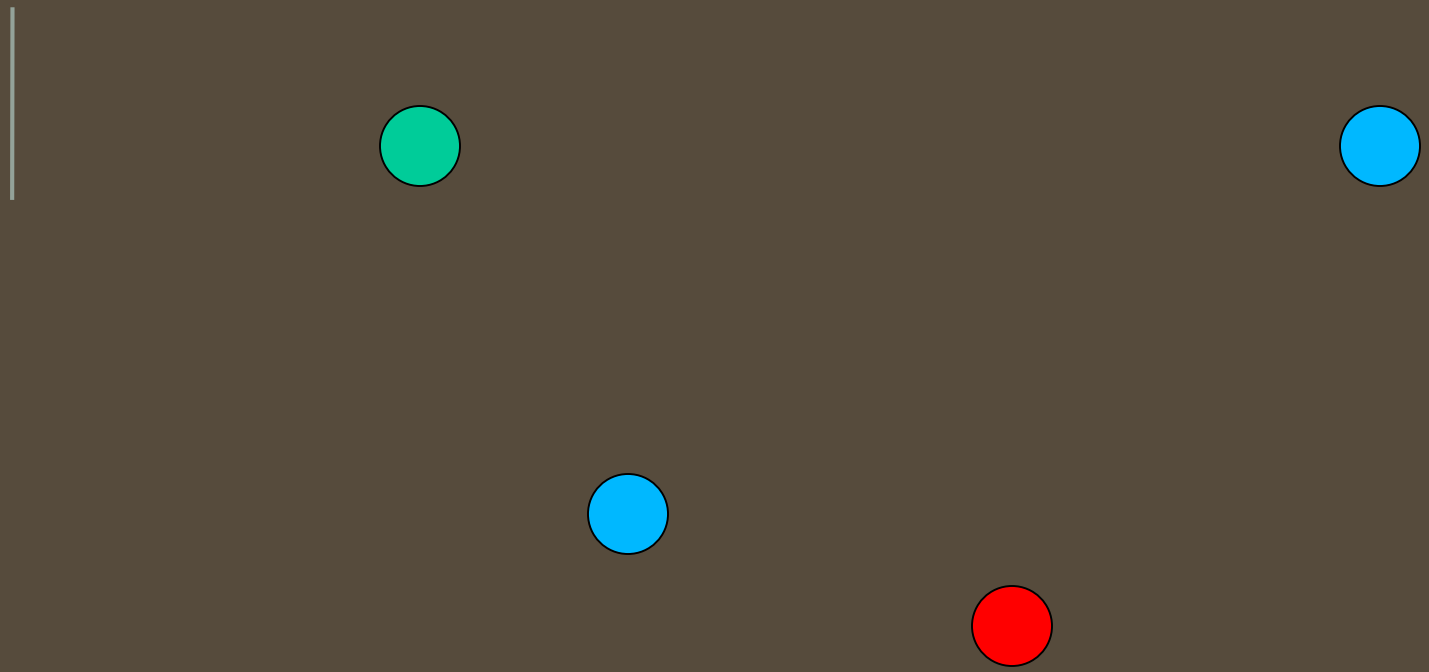
$$V_{n+1} = c*V_n + confiance\_personnelle*rand()*Maison + confiance\_swarm*rand()*Rich\_Kid\_maison$$

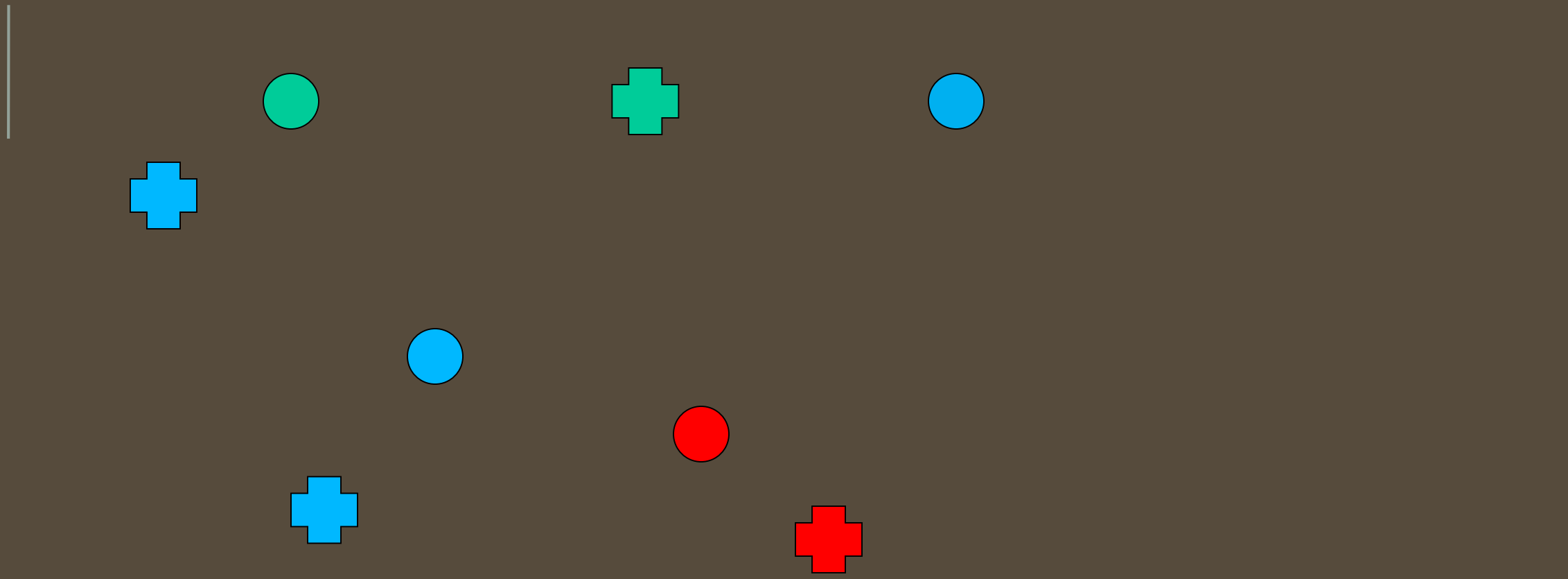
c, confiance_personnelle, confiance_swarm: des nombres réels.

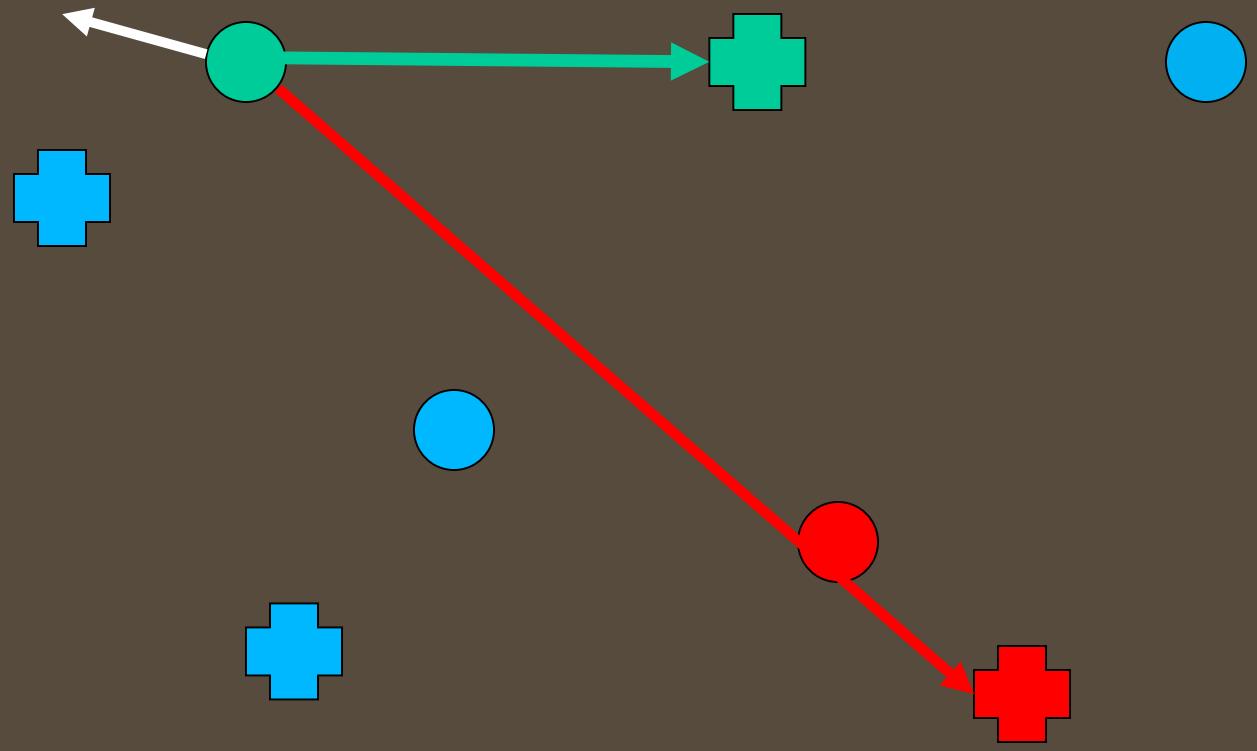
Rand() un nombre aléatoire pigé d'une distribution uniforme entre 0 et 1.

Mettre à jour la position de la créature avec la vélocité.









Topologie
(Comment les particules communiquent)

Global: Toutes les particules se voient et se parlent.

Local: Une particule ne peut voir que son voisin d'index de droite, de gauche et elle-même dans la population. (Topologie anneau)

Global (-G): Convergence plus rapide. Plus grand risque d'être bloqué dans un minimum local.
Local (-L): Convergence plus lente. Reste moins facilement bloqué dans un minimum local.

En pratique: Impossible de savoir d'avance quelle topologie est meilleure.

W-PSO

Exactement comme le PSO, mais remplace c par W, un réel, qui diminue linéairement avec le temps.

Compromis Exploration/Exploitation.

En pratique: mieux que PSO.

Problèmes avec PSO/w-pso

Utilisation de deux exemplars (maison et maison du « rich kid »)

Deux problèmes:

1) Oscillation

2) Deux pas en avant, un pas en arrière.

Comprehensive learning particle swarm optimization (CLPSO)

But: enlever les problèmes associés aux deux exemplars.

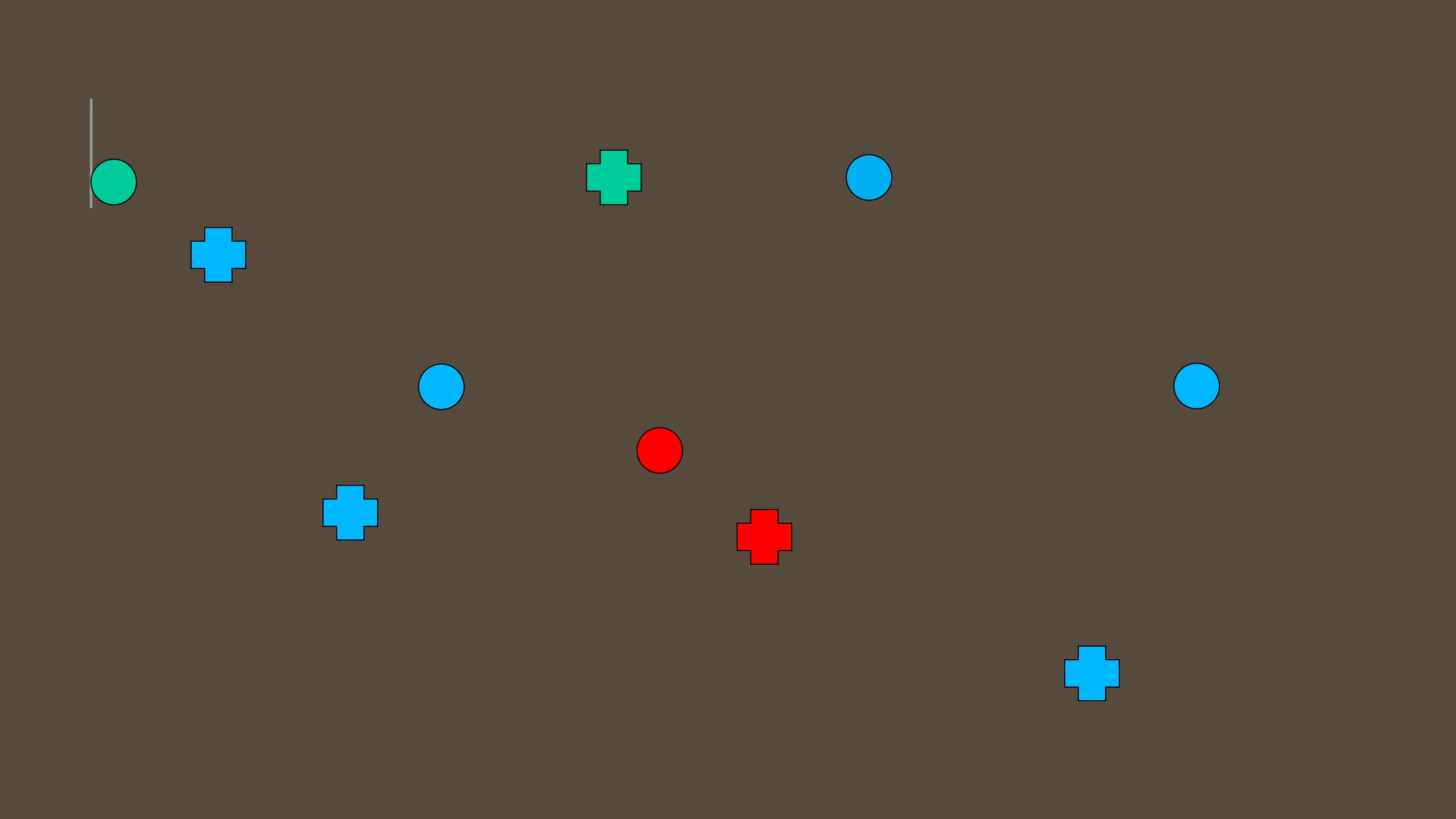
CLPSO

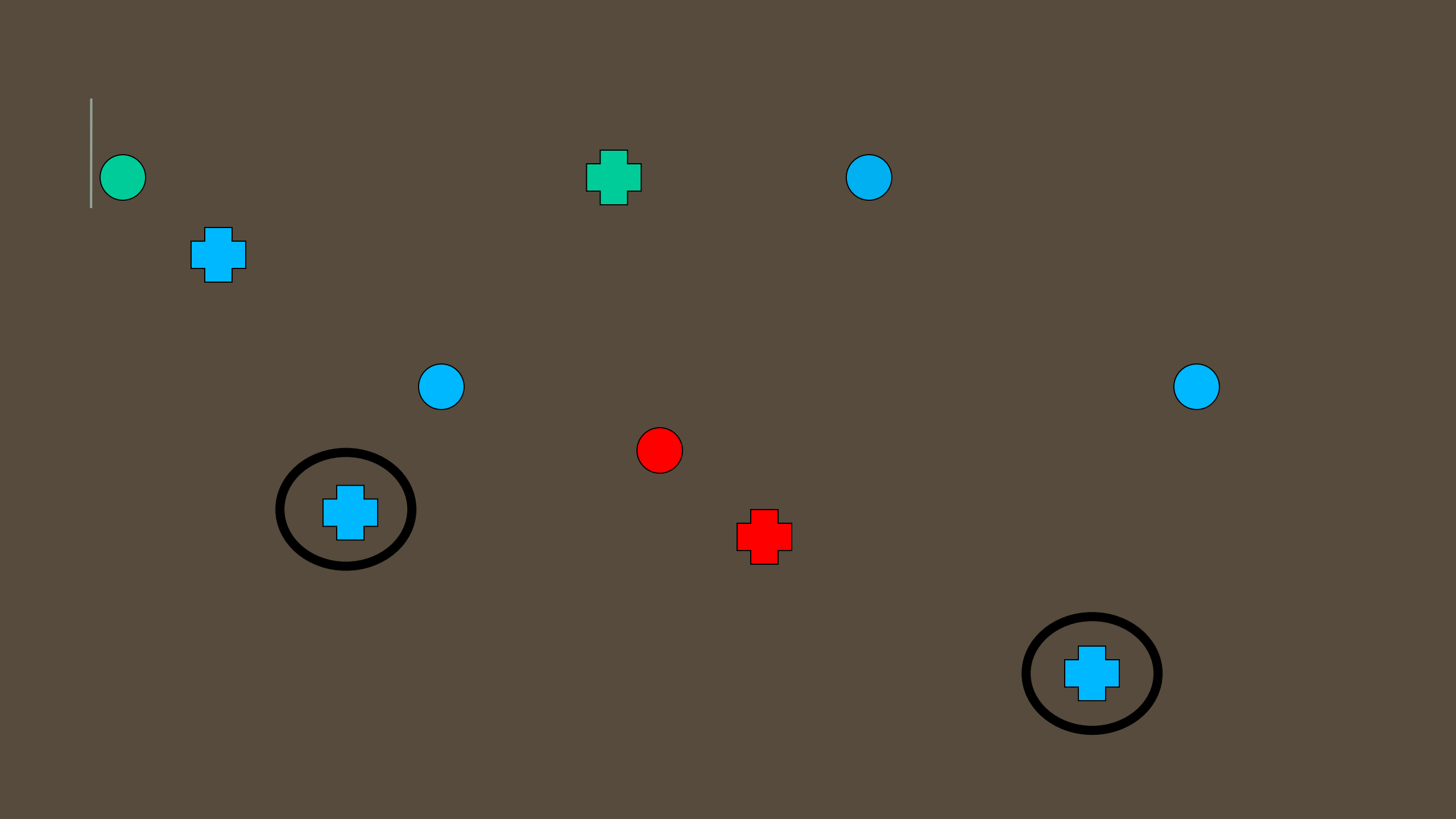
Nouvelle équation pour la vélocité:

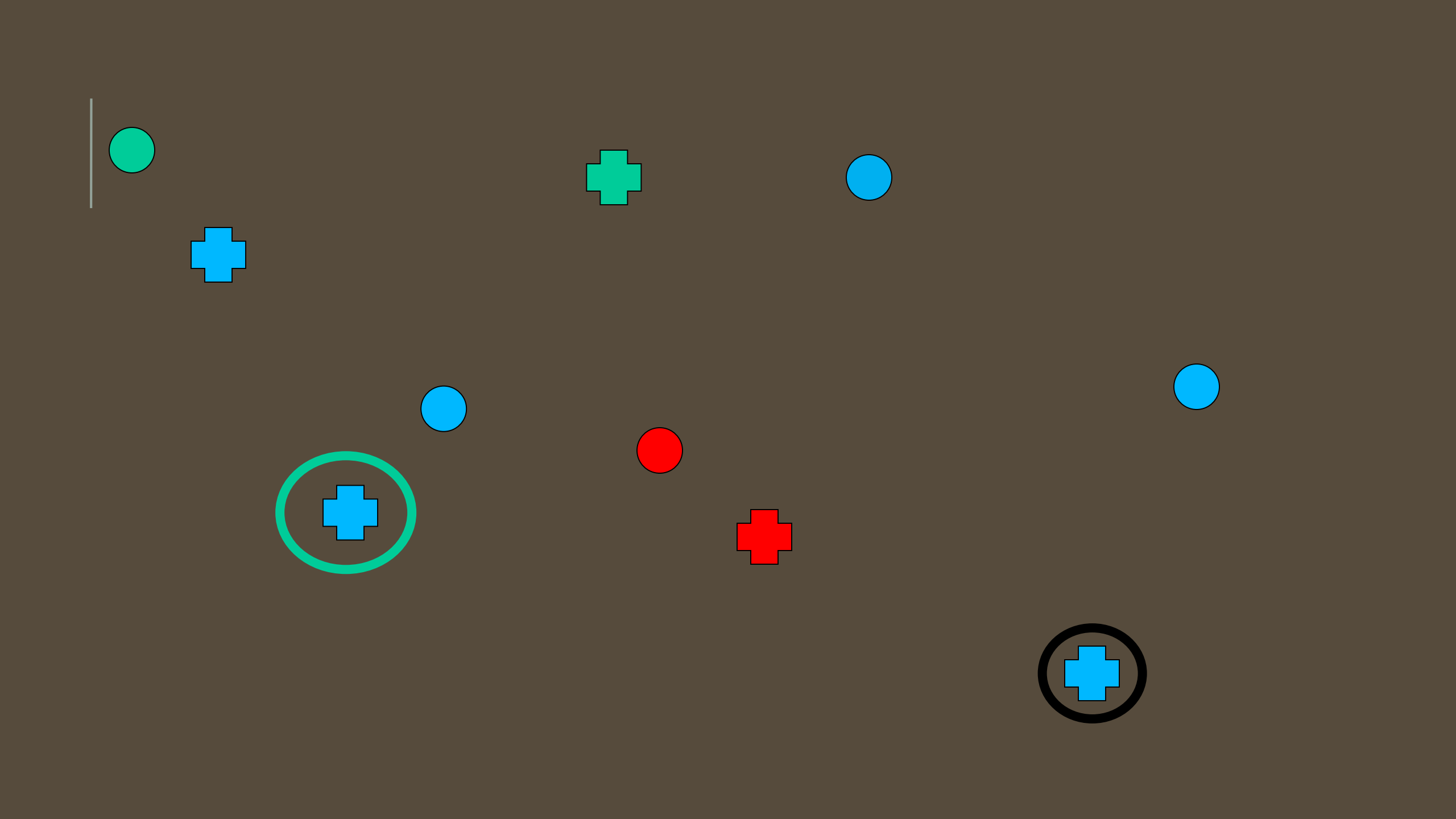
$$V_{n+1} = w*V_n + confiance*rand()*clpso\_exemplar$$

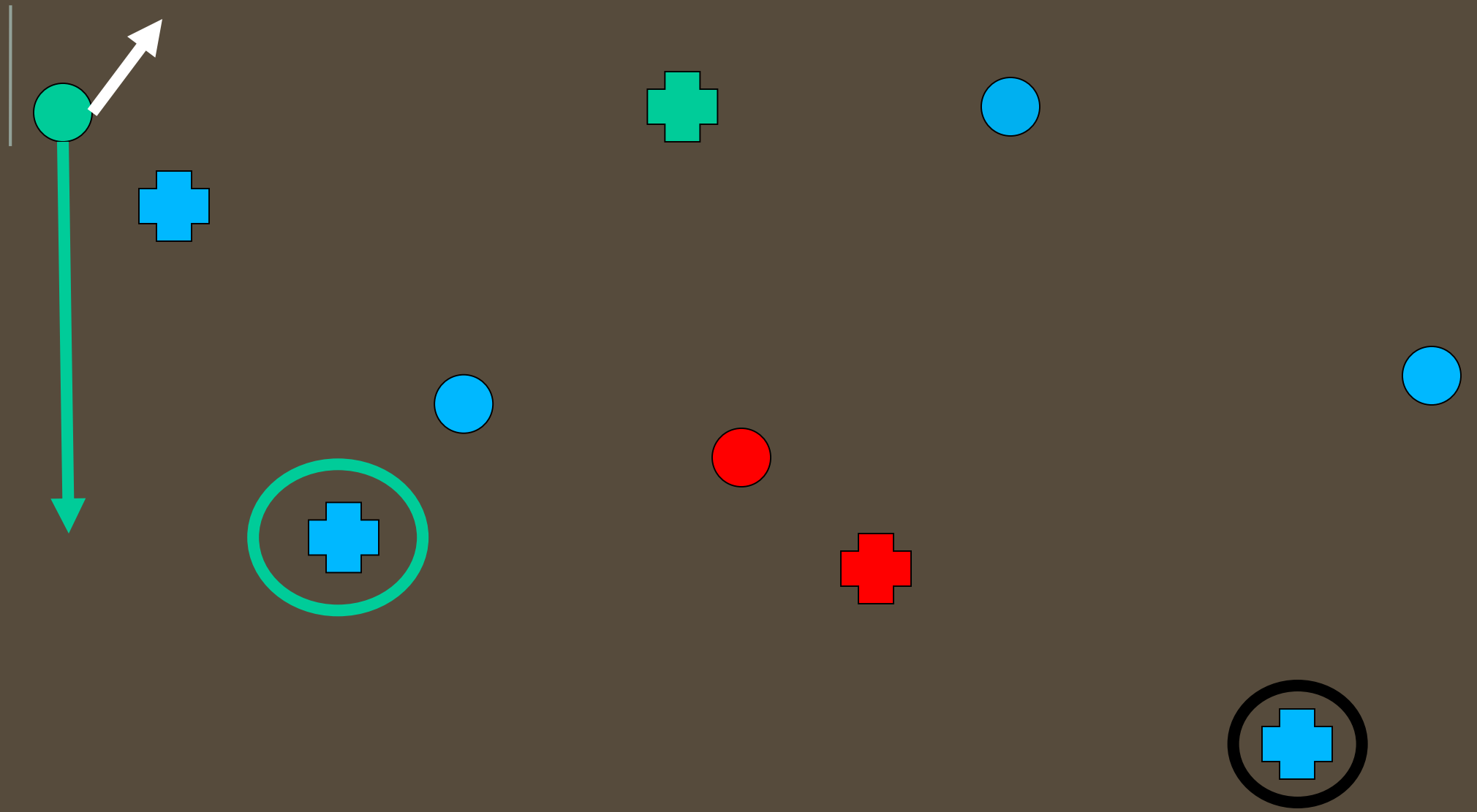
Confiance: Un nombre réel constant

Clpso_exemplar: Un vecteur position.









CLPSO

-Clpso_exemplar n'est pas très informatif.

-Pas de topographie locale explicite. L'exemplar fait une émulation d'une topographie locale.

OLPSO

Même idée que CLPSO. Mais la conception de l'exemplar se fait de façon « intelligente » en utilisant l'orthogonal learning (OL).

Considérer la maison (1) de la créature et la maison du « rich kid » (2) de la population.

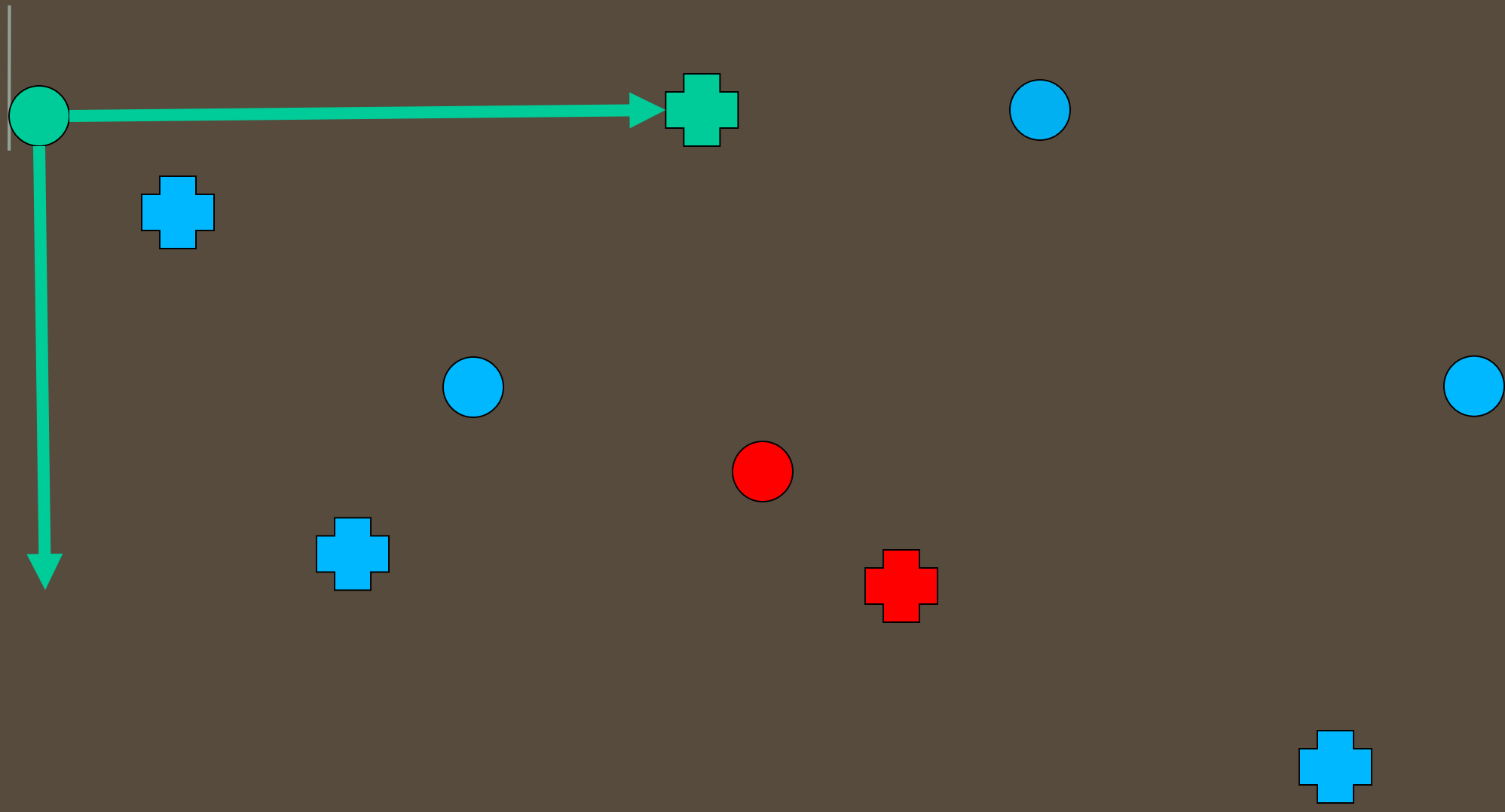
Trouver pour chaque dimension si la particule se dirige vers (1) ou (2).

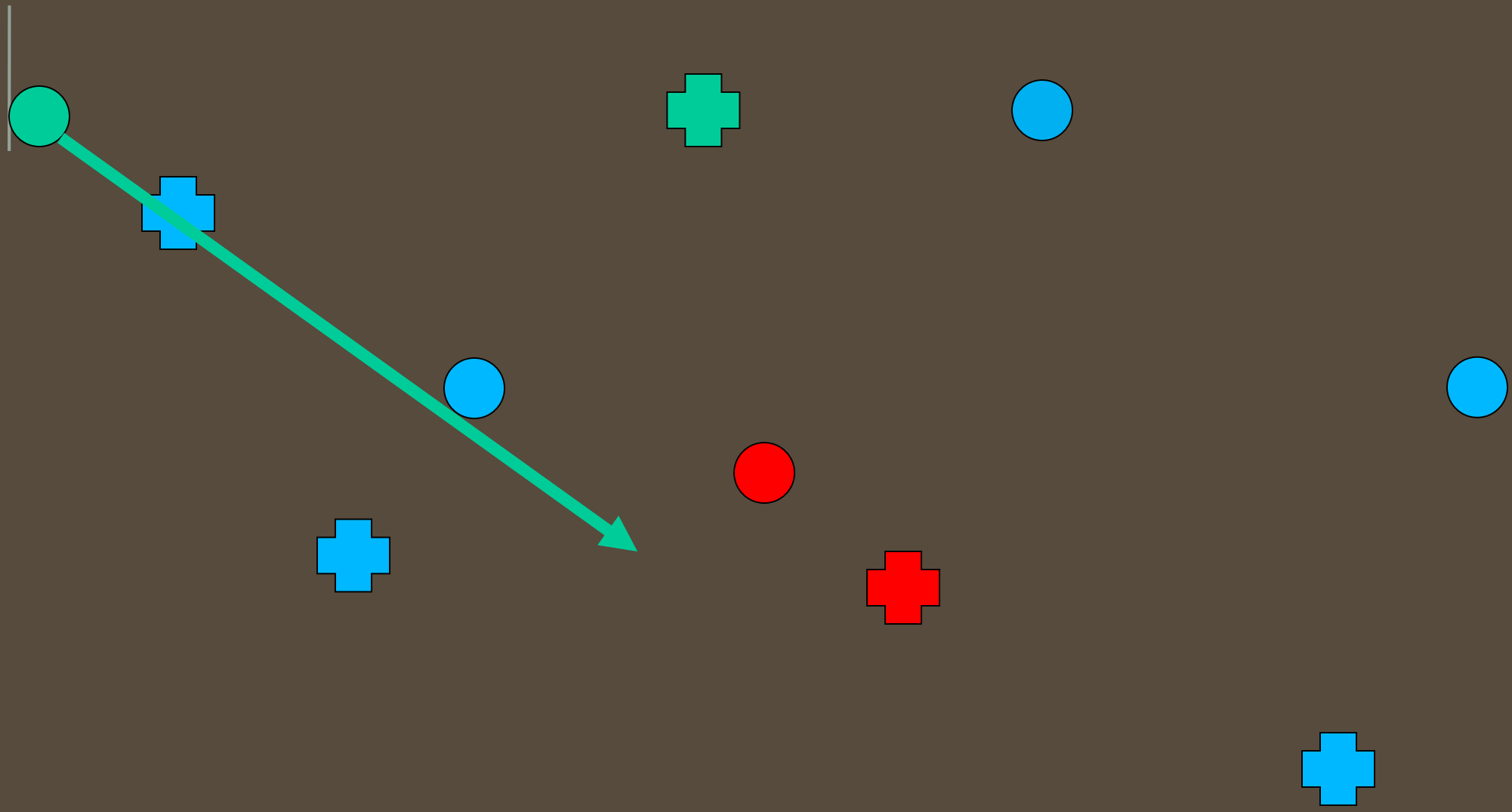
Approche exhaustive: Tester toutes les combinaisons (2^D FE).

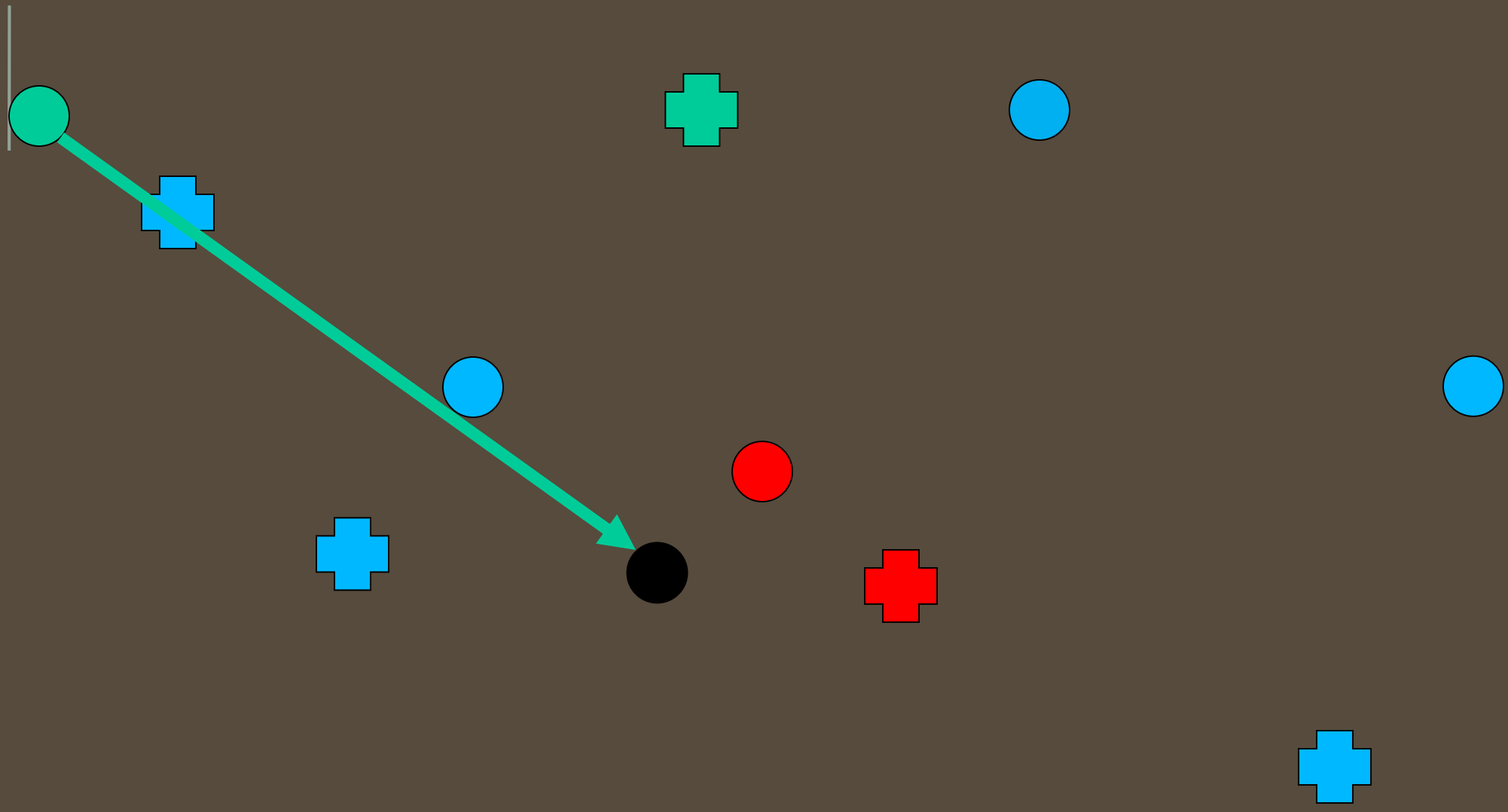
L'OL permet d'estimer pour chaque dimension la valeur parmis ces deux positions qui est la plus prometteuse (2D FE).

Avantage: Exemplar beaucoup plus intelligent et meilleure convergence.

Désavantage: Calcul exemplar coûteux.







OLPSO

Avantages: Exemplar beaucoup plus intelligent et meilleure convergence.

Désavantages: Calcul exemplar coûteux. Qualité de la solution finale dépends de la topologie.

Time adaptive dual particle swarm optimization (TAD-PSO)

Contribution

TAD-PSO (Contribution)

Idée: Séparer l'**exploration** et l'**exploitation** en deux populations (**Population Principale (exploration)** et **Population Auxiliaire (exploitation)** respectivement).

Par rapport à OLPSO: N'est plus dépendant à l'hyperparamètre de la topologie (comme CLPSO).

Par rapport à CLPSO: Exemplar calculé de façon beaucoup plus intelligente.

Main population

Topologie globale.

Utilise OLPSO, mais l'exemplar est calculé entre la maison et l'exemplar de CLPSO.

Calculer un nouvel exemplar à toutes les N générations sans amélioration.

Favorise l'exploration. Utilise l'information de l'ensemble de la population.

Main population

Calcul exemplar = 2*D.

Explosion du nombre d'évaluations lorsque la population converge.

Désactivation des particules pendant l'optimisation (favoriser l'exploitation).

Auxiliary population

Utilise un algorithme nommé PSO-TVAC.

Comme W-PSO:

$V_{n+1} = w*V_n + confiance\_personnelle*rand()*Maison + confiance\_swarm*rand()*meilleure\_maison\_swarm$

Mais confiance_personnelle diminue dans le temps et confiance swarm augmente dans le temps.

Algorithme qui converge extrêmement rapidement (presque pas d'exploration) et est mauvais lorsqu'utilisé seul.

Topologie meta-globale (connexion globale sur sa population et sur la main population)

Benchmarks

	Benchmark Problem	Search Range (R)	Global Opt. $\mathbf{x}$	$f_{\min}$	Name
Unimodal	$f_1(\mathbf{x}) = \sum_{i=1}^{D} \left(\sum_{j=1}^{i} x_j \right)^2$	$[-10, 10]^D$	$\{0\}^D$	0	Schwefel's P1.2 [18]
	$f_2(\mathbf{x}) = \sum_{i=1}^{D-1} [100(x_{i+1} - x_i^2)^2 + (x_i - 1)^2]$	$[-10, 10]^D$	$\{1\}^D$	0	Rosenbrock [18]*
	$f_3(\mathbf{x}) = \sum_{i=1}^{D} (10^6)^{(i-1)/(D-1)} x_i^2$	$[-100, 100]^D$	$\{0\}^D$	0	Elliptic [18]
	$f_4(\mathbf{x}) = \sum_{i=1}^{D} x_i^2$	$[-100, 100]^D$	$\{0\}^D$	0	Sphere [28]
Multimodal	$f_5(\mathbf{x}) = \sum_{i=1}^{D} [x_i^2 - 10\cos(2\pi x_i) + 10]$	$[-5.12, 5.12]^D$	$\{0\}^D$	0	Rastrigin [18]
	$f_6(\mathbf{x}) = -20\exp\left(-0.2\sqrt{\frac{1}{D}\sum_{i=1}^{D} x_i^2}\right) - \exp\left(\frac{1}{D}\sum_{i=1}^{D}\cos(2\pi x_i)\right) + 20 + e$	$[-32, 32]^D$	$\{0\}^D$	0	Ackley [18]
	$f_7(\mathbf{x}) = 418.982887272 \times D - \sum_{i=1}^{D} x_i \sin(\sqrt{\lvert x_i \rvert})$	$[-500, 500]^D$	$\{420.96\}^D$	0	Schwefel [18]
	$f_8(\mathbf{x}) = \sum_{i=1}^{D} \lvert x_i \sin(x_i) + 0.1 x_i \rvert$	$[-10, 10]^D$	$\{0\}^D$	0	Alpine [18]
	$f_9(\mathbf{x}) = \frac{1}{4000}\sum_{i=1}^{D} x_i^2 - \prod_{i=1}^{D} \cos\left(\frac{x_i}{\sqrt{i}}\right) + 1$	$[-600, 600]^D$	$\{0\}^D$	0	Griewank [18]
	$f_{10}(\mathbf{x}) = \frac{\pi}{D}\left(10\sin^2(\pi y_1) + \sum_{i=1}^{D-1}(y_i - 1)^2[1 + 10\sin^2(\pi y_{i+1})] + (y_D - 1)^2\right) + \sum_{i=1}^{D} u(x_i, 10, 100, 4)$ Where $y_i = 1 + \frac{1}{4}(x_i + 1)$, $u(x_i, a, k, m) = \begin{cases} k(x_i - a)^m, & x_i > a \\ 0, & -a \le x_i \le a \\ k(-x_i - a)^m, & x_i < -a \end{cases}$	$[-50, 50]^D$	$\{0\}^D$	0	Generalized Penalized [18]
Rotated and Shifted	$f_{11}(\mathbf{x}) = 418.982887272 \times D - \sum_{i=1}^{D} z_i$ Where $z_i = \begin{cases} y_i \sin\sqrt{\lvert y_i \rvert}, & \lvert y_i \rvert \le 500.0 \\ -.001(\lvert y_i \rvert - 500)^2, & \text{otherwise} \end{cases}$ $\mathbf{y} = \mathbf{y}' + \{420.96\}^D$, $\mathbf{y}' = M(\mathbf{x} - \{420.96\}^D)$	$[-500, 500]^D$	$\{420.96\}^D$	0	Rotated Schwefel [6]
	$f_{12}(\mathbf{x}) = \sum_{i=1}^{D-1} [100(y_{i+1} - y_i^2)^2 + (y_i - 1)^2]$, where $\mathbf{y} = M\mathbf{x}$	$[-10, 10]^D$	$\{0\}^D$	0	Rotated Rosenbrock [18]
	$f_{13}(\mathbf{x}) = \sum_{i=1}^{D} [y_i^2 - 10\cos(2\pi y_i) + 10]$, where $\mathbf{y} = M\mathbf{x}$	$[-5.12, 5.12]^D$	$\{0\}^D$	0	Rotated Rastrigin [18]
	$f_{14}(\mathbf{x}) = \sum_{i=1}^{D-1} [100(y_{i+1} - y_i^2)^2 + (y_i - 1)^2]$, where $\mathbf{y} = \mathbf{x} - \mathbf{c}$	$[-10, 10]^D$	$\{1\}^D + \mathbf{c}$	0	Shifted Rosenbrock [29]
	$f_{15}(\mathbf{x}) = \sum_{i=1}^{D} [y_i^2 - 10\cos(2\pi y_i) + 10]$, where $\mathbf{y} = \mathbf{x} - \mathbf{c}$	$[-5.12, 5.12]^D$	$\mathbf{c}$	0	Shifted Rastrigin [29]
	$f_{16}(\mathbf{x}) = \sum_{i=1}^{D} [y_i^2 - 10\cos(2\pi y_i) + 10]$, where $\mathbf{y} = M(\mathbf{x} - \mathbf{c})$	$[-5.12, 5.12]^D$	$\mathbf{c}$	0	Shifted Rotated Rastrigin [18]

TABLE VI

10 DIMENSIONS

Function		PSO-G	PSO-L	CLPSO	HPSO-TVAC	OLPSO-G	OLPSO-L	TAD-PSO
Schwefel's P1.2 f_1	Mean	9.418e-5	6.806e-3	3.122e-2	**4.808e-19**	244.837	475.505	6.837e-13
	SD	1.722e-4	1.003e-2	1.854e-2	**1.753e-18**	65.929	128.818	3.682e-12
	Rank	3	4	5	**1**	6	7	2
Rosenbrock f_2	Mean	5.672	7.074	0.564	0.745	46.267	3.213	**0.427**
	SD	1.977	1.455	0.334	1.163	54.048	12.352	**0.472**
	Rank	5	6	2	3	7	4	**1**
Elliptic f_3	Mean	1.289	1.799	4.558e-12	1.991e-21	2.442e-44	2.489e-19	**2.639e-56**
	SD	2.888	2.238	6.158e-12	2.024e-21	9.298e-44	3.877e-19	**1.083e-55**
	Rank	6	7	5	3	2	4	**1**
Sphere f_4	Mean	7.838e-5	2.996e-2	4.684e-13	2.261e-23	6.784e-47	1.751e-21	**2.946e-58**
	SD	2.626e-4	4.768e-2	5.025e-13	2.876e-23	1.751e-21	3.848	**1.559e-57**
	Rank	6	7	5	3	2	4	**1**
Rastrigin f_5	Mean	8.592	3.266	**4.770e-8**	0.332	13.830	0.133	6.633e-2
	SD	3.353	1.283	**8.183e-8**	0.469	6.183	0.425	0.248
	Rank	6	5	**1**	4	7	3	2
Ackley f_6	Mean	1.119	0.470	4.006e-7	4.765e-12	9.711e-15	1.533e-9	**3.553e-15**
	SD	0.823	0.551	2.541e-7	1.573e-12	3.299e-15	8.286e-10	**9.173e-16**
	Rank	7	6	5	3	2	4	**1**
Schwefel f_7	Mean	1408.290	1293.514	1.901e-11	524.419	502.047	51.323	**0.0**
	SD	263.002	267.523	1.761e-11	154.734	226.545	58.690	**0.0**
	Rank	7	6	2	5	4	3	**1**
Alpine f_8	Mean	4.526e-2	1.010e-2	4.749e-5	6.338e-12	**1.130e-14**	8.932e-4	1.177e-14
	SD	4.511e-2	1.895e-2	3.217e-5	2.107e-11	**8.935e-15**	7.004e-4	6.031e-14
	Rank	7	6	4	3	**1**	5	2
Griewank f_9	Mean	0.166	0.199	2.432e-3	0.105	1.307e-3	**0.0**	2.710e-3
	SD	7.453e-2	8.400e-2	3.869e-3	7.437e-2	7.043e-3	**0.0**	5.631e-3
	Rank	6	7	3	5	2	**1**	4
Generalized Penalized f_{10}	Mean	5.654e-32	**0.0**	**0.0**	1.912e-30	6.219e-32	**0.0**	**0.0**
	SD	2.747e-31	**0.0.**	**0.0**	4.080e30	3.021e-31	**0.0**	**0.0**
	Rank	5	**1**	**1**	7	6	**1**	**1**
Rotated Schwefel f_{11}	Mean	379.021	397.082	575.940	1404.767	198.536	178.524	**23.030**
	SD	442.856	394.378	216.545	454.710	257.727	230.253	**53.960**
	Rank	4	5	6	7	3	2	**1**
Rotated Rosenbrock f_{12}	Mean	11.030	7.321	4.464	2.953	5.211	17.125	**1.671**
	SD	21.871	1.372	2.901	2.291	2.062	28.674	**1.163**
	Rank	6	5	3	2	4	7	**1**
Rotated Rastrigin f_{13}	Mean	13.929	9.394	6.868	6.136	9.220	7.929	**3.756**
	SD	7.175	2.919	1.819	2.073	3.954	1.752	**2.145**
	Rank	7	6	3	2	5	4	**1**
Shifted Rosenbrock f_{14}	Mean	7.382	8.397	0.732	0.825	3.395	2.270	**0.435**
	SD	13.117	4.291	0.784	1.223	2.732	1.949	**0.519**
	Rank	6	7	2	3	5	4	**1**
Shifted Rastrigin f_{15}	Mean	14.297	6.681	1.546e-8	0.895	1.227	**0.0**	9.950e-2
	SD	4.408	2.740	3.399e-8	0.827	0.984	**0.0**	0.298
	Rank	7	6	2	4	5	**1**	3
Shifted Rotated Rastrigin f_{16}	Mean	26.035	15.526	6.924	23.524	11.011	9.493	**3.283**
	SD	9.616	5.777	2.102	8.457	6.649	4.392	**2.180**
	Rank	7	5	2	6	4	3	**1**
Ave. rank		5.938	5.563	3.188	3.813	4.063	3.563	**1.5**
Final rank		7	6	2	4	5	3	**1**
Algorithms		PSO-G	PSO-L	CLPSO	HPSO-TVAC	OLPSO-G	OLPSO-L	TAD-PSO

30 Dimensions

Function		PSO-G	PSO-L	CLPSO	HPSO-TVAC	OLPSO-G	OLPSO-L	TAD-PSO
Schwefel's P1.2 f_1	Mean	1.012	2.991	8.123	**9.736e-10**	3.250e-3	0.823	1.032e-4
	SD	0.353	1.055	1.946	**1.662e-9**	5.562e-3	0.684	1.841e-4
	Rank	5	6	7	**1**	3	4	2
Rosenbrock f_2	Mean	53.055	55.865	14.536	6.509	21.610	3.640	**0.021**
	SD	33.507	23.640	9.79	5.125	30.465	6.420	**0.021**
	Rank	6	7	4	3	5	2	**1**
Elliptic f_3	Mean	40.780	126.296	6.671e-12	1.455e-19	**1.068e-69**	3.175e-33	1.472e-62
	SD	22.676	46.415	5.046e-12	6.345e-20	**2.742e-69**	1.093e-32	3.455e-62
	Rank	6	7	5	4	**1**	3	2
Sphere f_4	Mean	0.920	8.402	2.419e-12	8.328e-21	**6.990e-72**	1.624e-35	2.128e-63
	SD	0.641	3.522	1.132e-12	3.396e-21	**1.741e-71**	5.162e-35	8.885e-63
	Rank	6	7	5	4	**1**	3	2
Rastrigin f_5	Mean	48.426	19.707	9.471e-5	0.365	3.051	1.895e-15	**0.0**
	SD	10.620	4.353	1.128e-4	0.748	1.703	1.020e-14	**0.0**
	Rank	7	6	3	4	5	2	**1**
Ackley f_6	Mean	3.126	1.951	5.050e-7	6.522e-11	5.566e-15	6.869e-15	**4.974e-15**
	SD	0.576	0.450	1.345e-7	1.157e-11	1.760e-15	8.862e-16	**1.740e-15**
	Rank	7	6	5	4	2	3	**1**
Schwefel f_7	Mean	5631.682	5320.552	2.581e-10	1396.914	435.606	2.583e-4	**1.213e-12**
	SD	652.454	644.091	1.714e-10	258.064	265.949	2.156e-4	**1.182e-12**
	Rank	7	6	2	5	4	3	**1**
Alpine f_8	Mean	1.742	0.597	6.878e-4	2.784e-10	**2.598e-15**	7.979e-8	5.727e-12
	SD	0.832	0.342	2.029e-4	1.597e-10	**2.202e-15**	2.862e-7	3.083e-11
	Rank	7	6	5	3	**1**	4	2
Griewank f_9	Mean	0.701	1.070	6.816e-9	0.015	4.503e-3	**0.0**	**0.0**
	SD	0.159	0.039	6.545e-9	0.019	0.010	**0.0**	**0.0**
	Rank	6	7	3	5	4	**1**	**1**
Generalized Penalized f_{10}	Mean	3.704e-31	**0.0**	**0.0**	5.664e-31	1.215e-32	**0.0**	**0.0**
	SD	6.389e-31	**0.0**	**0.0**	9.110e-31	4.544e-32	**0.0**	**0.0**
	Rank	6	**1**	**1**	7	5	**1**	**1**
Rotated Schwefel f_{11}	Mean	2205.323	3676.135	3208.419	5385.639	688.816	433.874	**159.444**
	SD	647.201	976.877	384.556	662.571	441.160	486.607	**235.615**
	Rank	4	6	5	7	3	2	**1**
Rotated Rosenbrock f_{12}	Mean	74.403	51.407	44.847	**11.653**	28.366	22.758	20.067
	SD	57.820	18.033	5.630	**8.875**	10.1663	11.639	3.170
	Rank	7	6	5	**1**	4	3	2
Rotated Rastrigin f_{13}	Mean	64.094	51.655	78.979	25.814	38.568	44.740	**5.845**
	SD	19.643	12.082	9.006	6.936	8.868	10.239	**3.242**
	Rank	6	5	7	2	3	4	**1**
Shifted Rosenbrock f_{14}	Mean	63.512	101.083	56.093	4.812	11.601	1.462	**0.038**
	SD	61.965	87.208	20.043	11.892	23.620	3.868	**0.069**
	Rank	6	7	5	3	4	2	**1**
Shifted Rastrigin f_{15}	Mean	100.851	63.089	3.273	0.929	6.832	0.265	**0.0**
	SD	21.350	15.242	0.985	1.333	2.447	0.440	**0.0**
	Rank	7	6	4	3	5	2	**1**
Shifted Rotated Rastrigin f_{16}	Mean	166.826	109.473	75.110	119.592	48.486	53.560	**6.314**
	SD	32.494	22.604	7.638	25.990	11.150	12.469	**1.720**
	Rank	7	5	4	6	2	3	**1**
Ave. rank		6.250	5.875	4.375	3.875	3.250	2.625	**1.313**
Final rank		7	6	5	4	3	2	**1**
Algorithms		PSO-G	PSO-L	CLPSO	HPSO-TVAC	OLPSO-G	OLPSO-L	TAD-PSO

100 Dimensions

Function		PSO-G	PSO-L	CLPSO	HPSO-TVAC	OLPSO-G	OLPSO-L	TAD-PSO
Schwefel's P1.2 f_1	Mean	118.532	62.458	1051.594	**0.157**	244.837	475.505	91.964
	SD	21.095	13.763	86.545	**0.097**	65.929	128.818	31.589
	Rank	4	2	7	**1**	5	6	3
Rosenbrock f_2	Mean	633.846	756.390	191.807	113.688	46.267	3.213	**0.166**
	SD	137.158	108.231	52.727	38.956	54.048	12.352	**0.174**
	Rank	6	7	5	4	3	2	**1**
Elliptic f_3	Mean	976.225	2036.556	6.251e-7	8.849e-17	**2.442e-44**	2.489e-19	4.282e-37
	SD	229.203	296.693	1.561e-7	2.916e-17	**9.298e-44**	3.877e-19	1.245e-36
	Rank	6	7	5	4	**1**	3	2
Sphere f_4	Mean	78.290	330.663	2.335e-7	2.413e-17	**6.784e-47**	1.751e-21	1.861e-38
	SD	19.391	49.578	5.860e-8	8.693e-18	**1.715e-46**	3.848e-21	4.676e-38
	Rank	6	7	5	4	**1**	3	2
Rastrigin f_5	Mean	224.640	126.230	142.416	11.542	13.830	0.133	**0.0**
	SD	33.143	15.243	9.993	8.081	6.183	0.425	**0.0**
	Rank	7	5	6	3	4	2	**1**
Ackley f_6	Mean	7.248	4.633	1.002e-4	1.976e-9	**9.711e-15**	1.533e-9	6.241e-14
	SD	0.668	0.240	9.679e-6	4.340e-10	**3.299e-15**	8.286e-10	1.013e-13
	Rank	7	6	5	4	**1**	3	2
Schwefel f_7	Mean	23174.825	22936.395	771.640	4267.070	502.047	51.323	**9.944e-12**
	SD	1827.283	1380.293	377.452	598.737	226.545	58.690	**6.366e-12**
	Rank	7	6	4	5	3	2	**1**
Alpine f_8	Mean	24.499	9.075	0.085	3.620e-8	**1.130e-14**	8.931e-4	1.064e-4
	SD	4.563	2.548	0.025	2.985e-8	**8.935e-15**	7.004e-4	6.641e-5
	Rank	7	6	5	2	**1**	4	3
Griewank f_9	Mean	1.821	3.955	4.495e-7	0.011	0.001	**0.0**	**0.0**
	SD	0.173	0.489	2.541e-7	0.013	0.007	**0.0**	**0.0**
	Rank	6	7	3	5	4	**1**	**1**
Generalized Penalized f_{10}	Mean	0.996	5.597e-32	**0.0**	5.387e-31	6.219e-32	**0.0**	**0.0**
	SD	2.540	3.014e-31	**0.0**	7.072e-31	3.021e-31	**0.0**	**0.0**
	Rank	7	4	**1**	6	5	**1**	**1**
Rotated Schwefel f_{11}	Mean	2205.323	3676.135	18563.120	19887.658	1040.930	**339.398**	575.033
	SD	647.201	976.877	616.851	3073.267	769.272	**318.779**	627.482
	Rank	4	5	6	7	3	**1**	2
Rotated Rosenbrock f_{12}	Mean	662.811	803.915	97.184	**81.361**	122.521	169.373	131.502
	SD	147.654	150.490	2.142	**45.348**	26.917	47.979	52.283
	Rank	6	7	2	**1**	3	5	4
Rotated Rastrigin f_{13}	Mean	237.936	307.793	430.041	71.754	104.504	155.127	**11.174**
	SD	43.782	47.917	22.097	13.534	21.374	28.639	**3.118**
	Rank	5	6	7	2	3	4	**1**
Shifted Rosenbrock f_{14}	Mean	633.873	3512.113	248.419	107.850	35.497	7.813	**1.196**
	SD	343.957	1159.389	64.070	47.962	54.836	20.464	**1.226**
	Rank	6	7	5	4	3	2	**1**
Shifted Rastrigin f_{15}	Mean	741.799	521.107	180.615	29.116	37.444	1.459	**0.531**
	SD	102.652	44.060	12.429	25.712	8.778	0.986	**0.559**
	Rank	7	6	5	3	4	2	**1**
Shifted Rotated Rastrigin f_{16}	Mean	795.473	731.826	468.640	304.366	160.613	179.744	**16.541**
	SD	127.527	71.654	25.879	49.098	27.785	33.770	**4.211**
	Rank	7	6	5	4	2	3	**1**
Ave. rank		6.125	5.875	4.750	3.688	2.875	2.750	**1.688**
Final rank		7	6	5	4	3	2	**1**
Algorithms		PSO-G	PSO-L	CLPSO	HPSO-TVAC	OLPSO-G	OLPSO-L	TAD-PSO

Test statistique

Friedman -> Post Hoc (Holm).

Changement de dimensions -> Change benchmark.

Average Ranking From Friedman's Test and Holm Null Hypothesis Testing

		PSO-G	PSO-L	CLPSO	HPSO-TVAC	OLPSO-G	OLPSO-L	TAD-PSO
10D& 30D& 100D	Rank	6.187	5.927	4.240	3.583	3.292	3.135	**1.635**
	h	**1**	**1**	**1**	**1**	**1**	**1**	-
10D	Rank	6.188	5.906	3.469	3.813	3.688	3.344	**1.594**
	h	**1**	**1**	**1**	**1**	**1**	**1**	-
30D	Rank	6.188	5.938	4.375	3.875	3.188	3.031	**1.406**
	h	**1**	**1**	**1**	**1**	**1**	**1**	-
100D	Rank	6.133	5.933	4.867	3.133	3.000	2.967	**1.967**
	h	**1**	**1**	**1**	0	0	0	-

*Reject null hypothesis ($h = 1$), accept null hypothesis ($h = 0$).

Giant Squid Swarm Optimization

Optimisation de fonctions à haut coûts (Work in progress)

Fonctions à haut coûts

Une fonction qui nécessite beaucoup de ressources à évaluer (temps, argent, éthique).

Algorithmes d'optimisation normaux mal adaptés (e.g. descente de gradient, Algorithmes évolutionnaires, PSO).

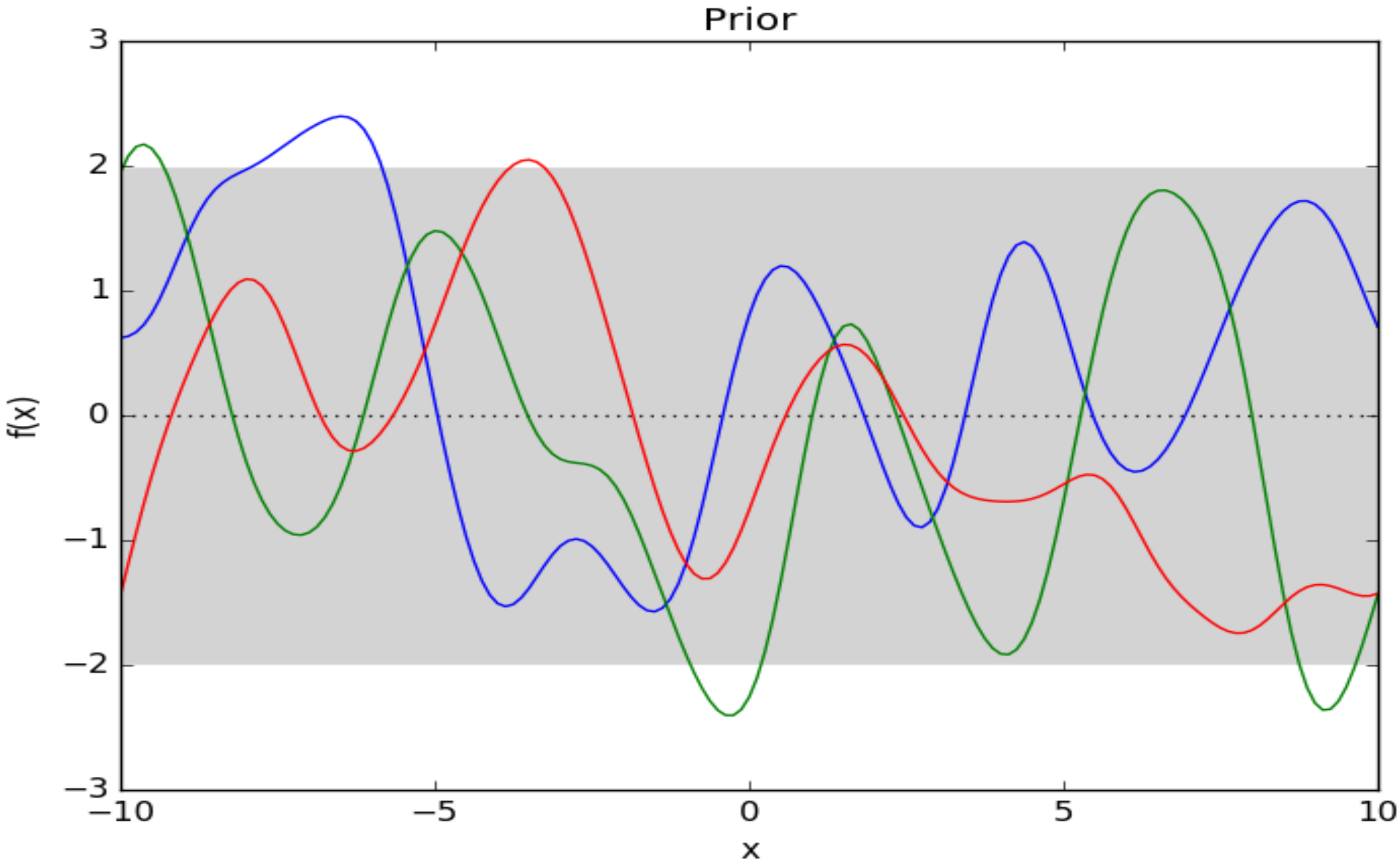
Alternative: Grid Search, Random Search, « Expert » Guessing

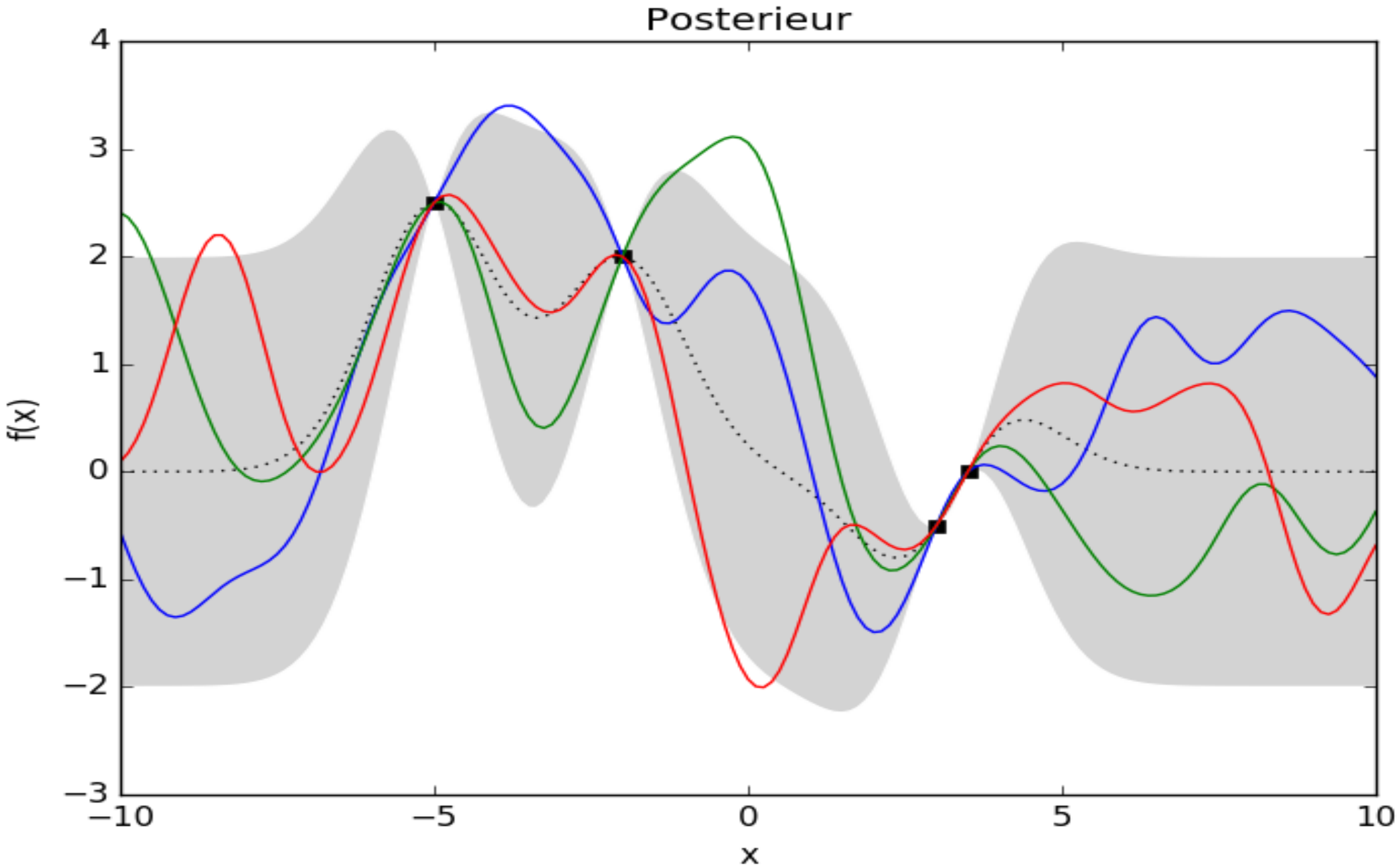
Random Search

Bayésien Optimisation

State of the art

Processus Gaussien


Prior
3
2
1
0
−1
−2
−3
f(x)
−10
−5
0
5
10
x


Posterieur
4
3
2
1
0
−1
−2
−3
f(x)
−10
−5
0
5
10
x

Fonctions à haut coûts (State of the art)

Optimisation bayésienne (e.g. Spearmint, SigOpt)

Avec un peu de chance…

Giant Squid Swarm Optimization (Notre Algorithme)

Philosophie

Comme la fonction est inconnue. On utilise un régresseur pour l'approximer (Processus Gaussien).

Processus Gaussien:

On traite la fonction comme une fonction aléatoire et on y appose un aprioris. (Prior Gaussien).

On évalue un point. On update l'aprioris qui devient un postériore de la fonction. On utilise la distribution à postériori pour créer une fonction d'acquisition qui permet d'évaluer le prochain point.

Avantage: Mesure d'incertitude sur les prédictions, calcul de gradient facile.

Désavantage: Hypothèses forte. Mesure d'incertitude « weird »

Optimisation bayésienne

1. Représenter la fonction inconnue par un processus Gaussien.
2. Fonction de fitness (e.g. Expected Improvement, Upper Confidence Bound, Improvement Probabilty)
3. Descente de gradient pour trouver le minimum.
4. Évaluation du point
5. Ré-entraînement du GP
6. Retour #3 jusqu'à nombre max itérations.

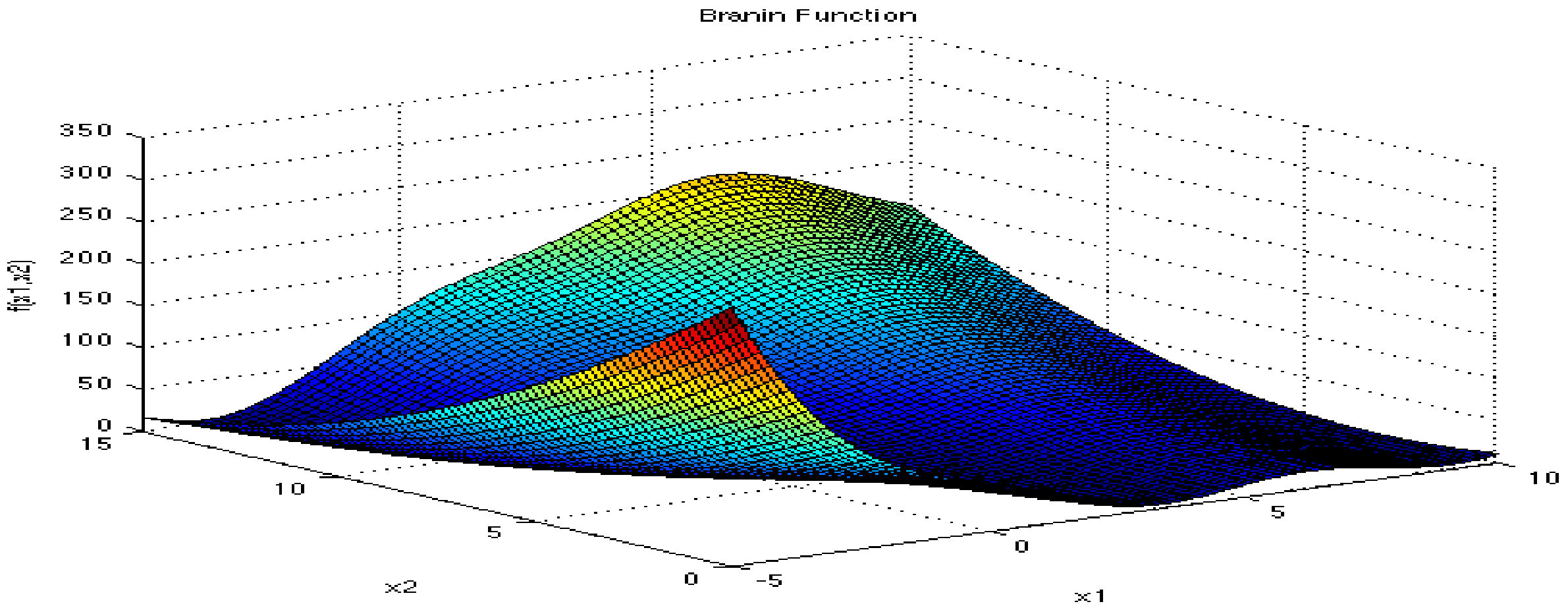
SigOpt

Une librairie pour l'optimisation bayésien comme Spearmint.

Compare favorablement à Spearmint. Entre autre, meilleur sur la fonction de Branin (nous y viendrons).

(A stratified analysis of Bayesian Optimization Methods. Dwancker et al.)

Résultats avant présentation algorithmes.



Branin-Hoo

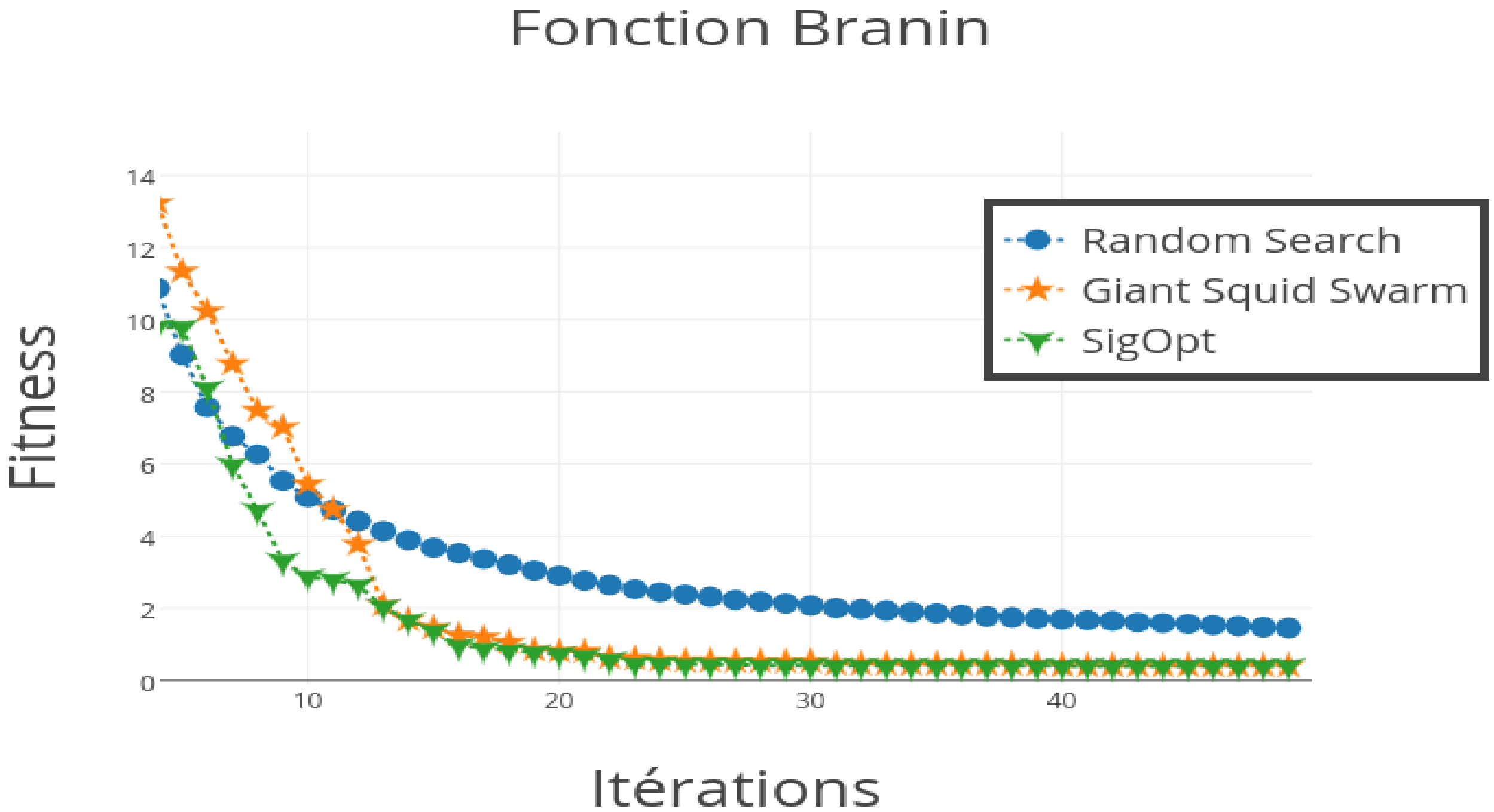
2 dimensions

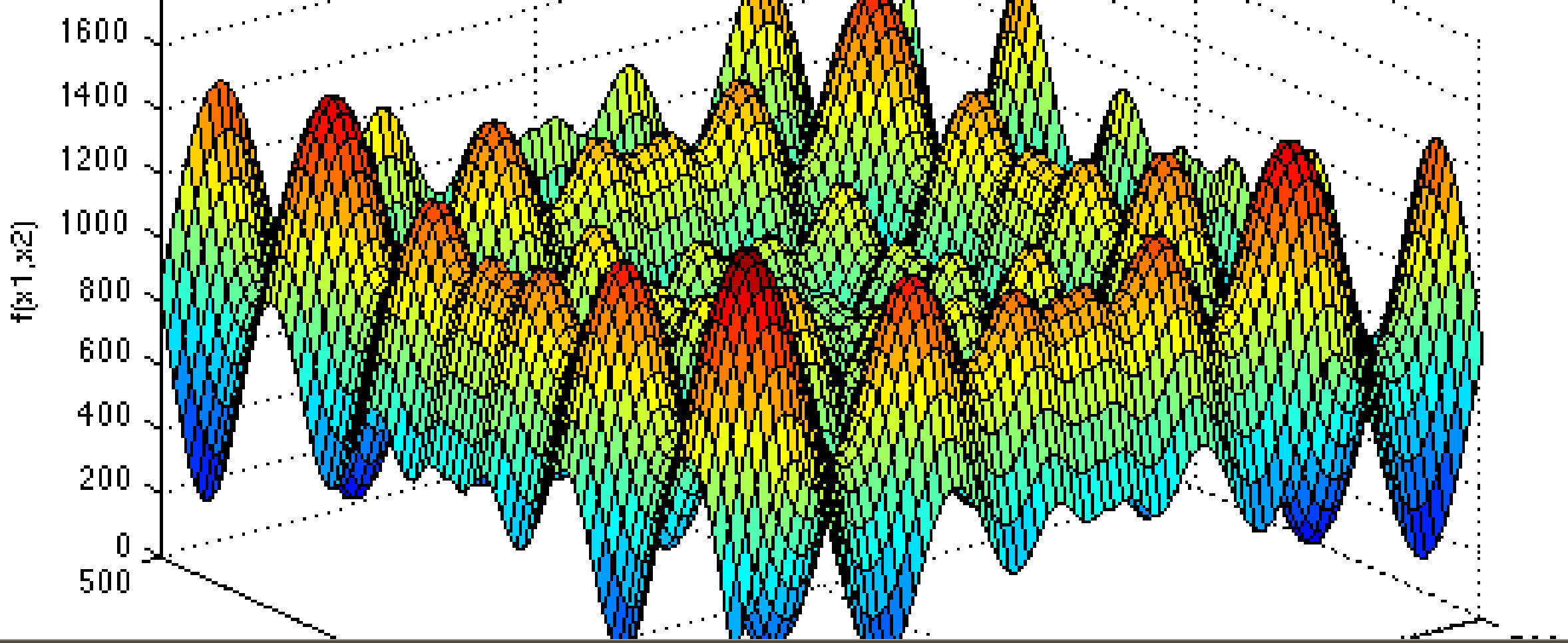
-5.0 <= x <= 10.0

0.0 <= y <= 15.0

3 Minimum globaux = 0.397887

$$f_{\text{Branin01}}(\mathbf{x}) = \left(-1.275\frac{x_1^2}{\pi^2} + 5\frac{x_1}{\pi} + x_2 - 6\right)^2 + \left(10 - \frac{5}{4\pi}\right)\cos(x_1) + 10$$

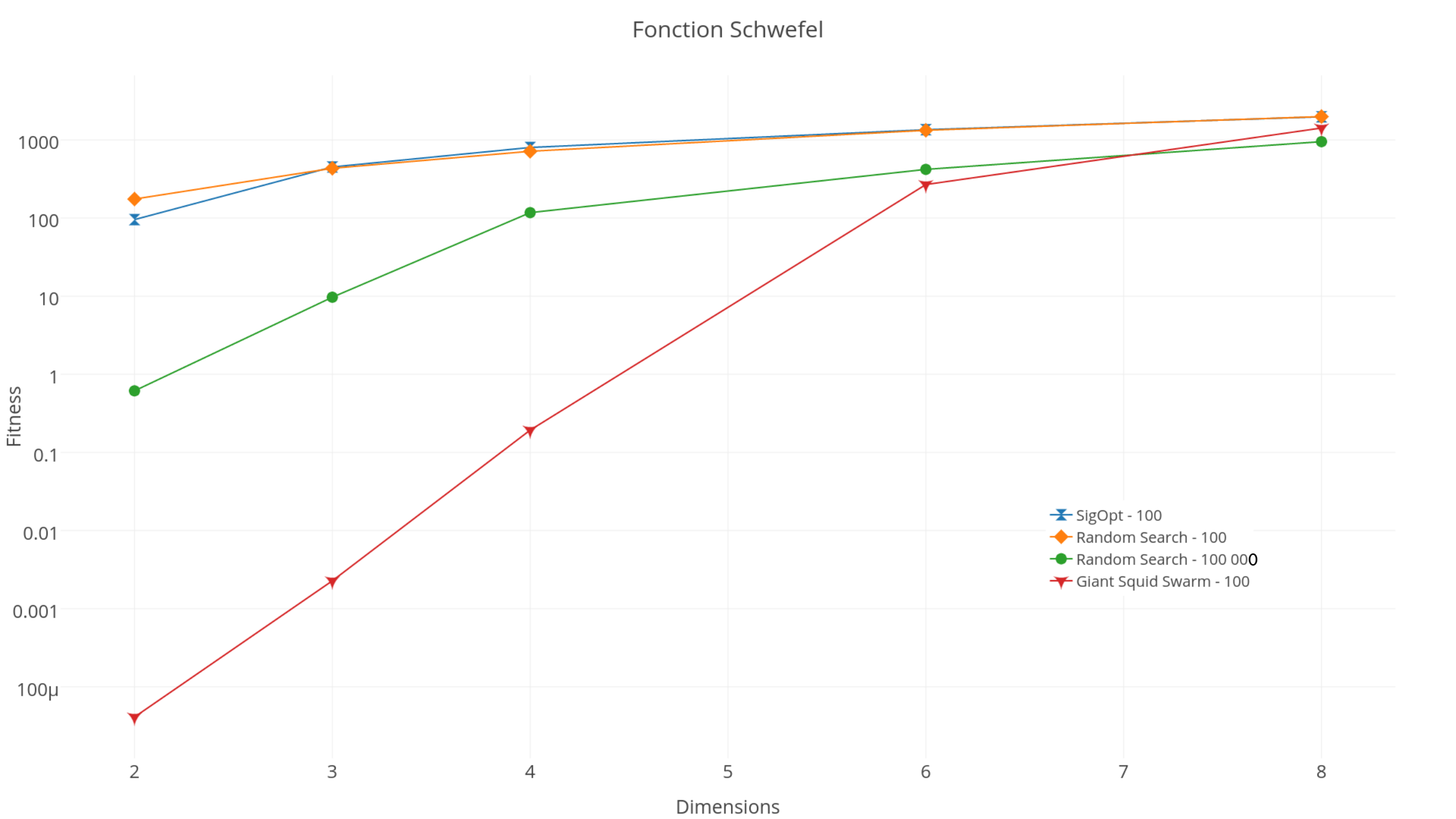

Fonction Branin
Fitness
Itérations
Random Search
Giant Squid Swarm
SigOpt
0
2
4
6
8
10
12
14
10
20
30
40



$$f(\mathbf{x}) = 418.9829d - \sum_{i=1}^{d} x_i \sin(\sqrt{|x_i|})$$

Schwefel

X entre -500 et 500,
minimisation, D-dimensions
1 Minimum global = 0.0


Fonction Schwefel
1000
100
10
1
0.1
0.01
0.001
100µ
Fitness
2
3
4
5
6
7
8
Dimensions
SigOpt - 100
Random Search - 100
Random Search - 100 000
Giant Squid Swarm - 100

Giant Squid Swarm

Pseudo-Code

1. Initialisation: Générer N positions aléatoires. K-means avec K = 2. Évaluer les centres trouver.
2. Déterminer exploration/exploitation. Construire le régresseur approprié avec tout les points réellement évalué.
3. Initialiser TAD-PSO pour trouver le minimum de la régression.
4. Évalué le minimum trouvé par TAD-PSO et l'évalué avec la fonction à haut coût.
5. Retour à #2 jusqu'au nombre max d'itérations atteint.

Nerf de la guerre

Exploitation

But: Trouver un point à évaluer qui permettra au régresseur de mieux évaluer la function

Exploration

But: Trouver une nouvelle position sur la fonction à haut coût qui soit meilleur que celle trouvé jusqu'à maintenant

Exploitation

Observation. Le régresseur est à son plus fiable proche des points déjà évalués.

Idée: Créer N TAD-PSO initialisé autour des meilleurs points trouvé et de X autres points aléatoirement choisit.

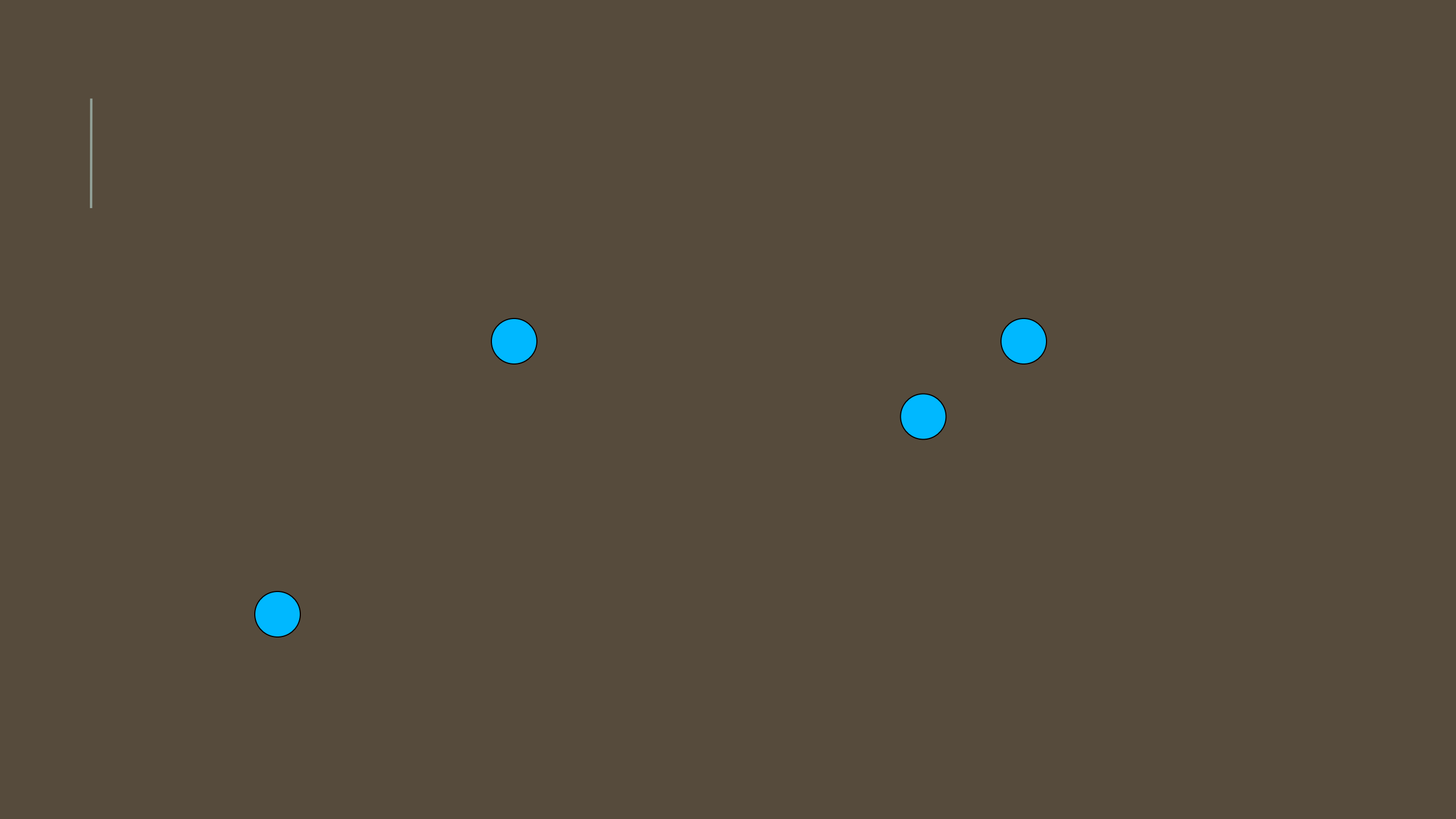
De plus, restreindre l'aire de recherche pour avoir une certaine assurance du point qu'on évaluera avec la fonction à haut coût.

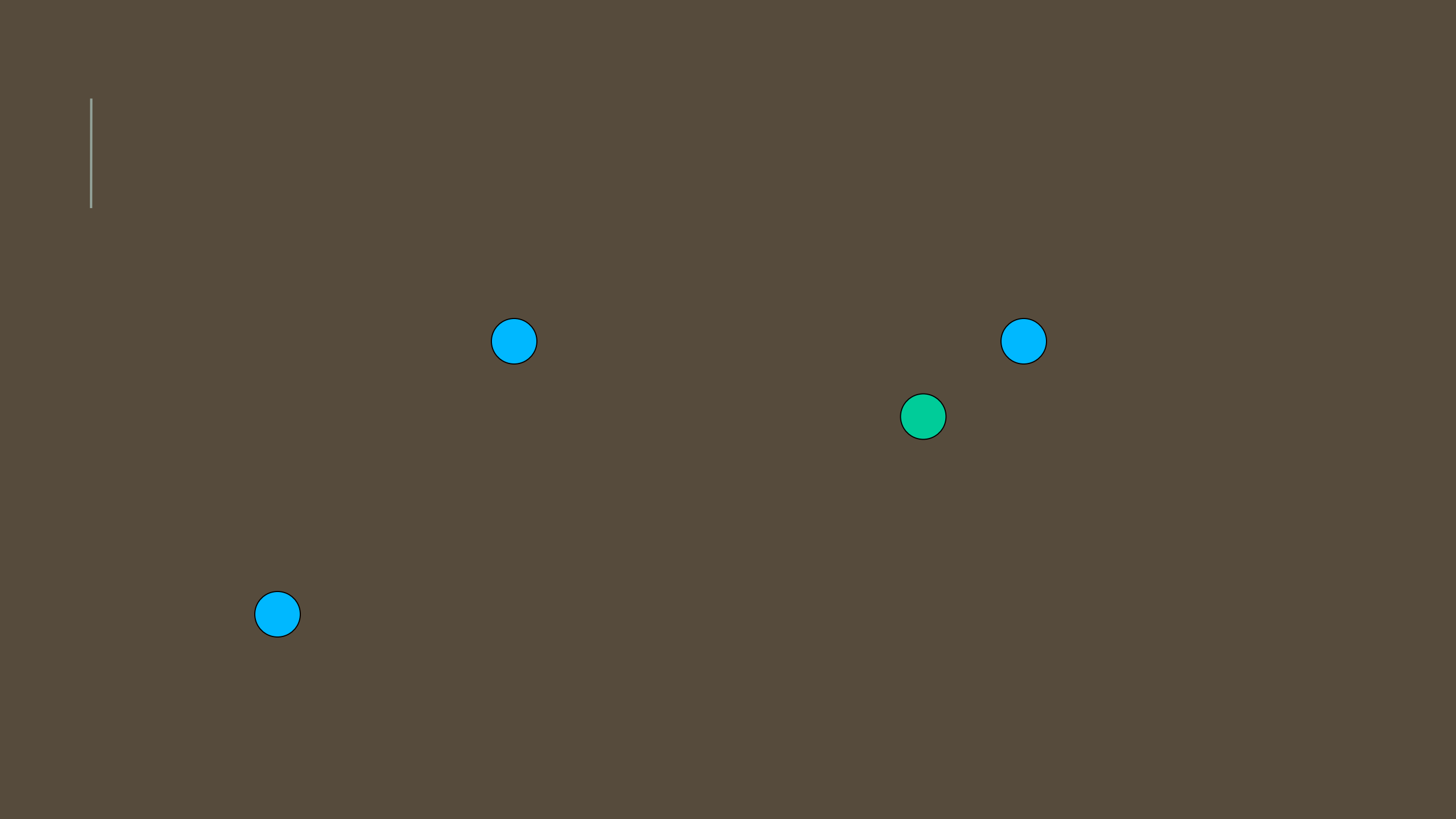
Exploitation (work in progress++)

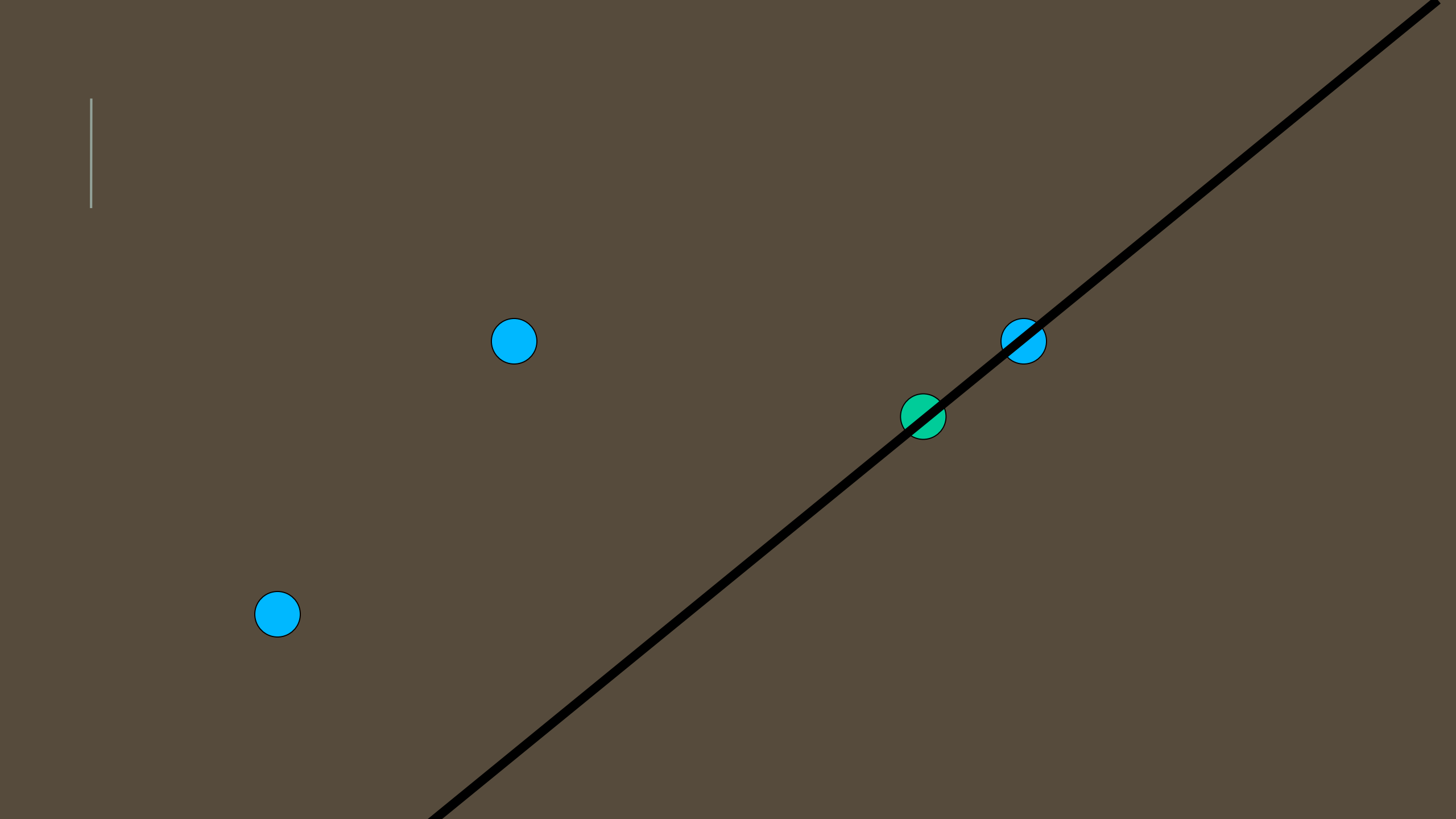
Calcul aire de recherche:

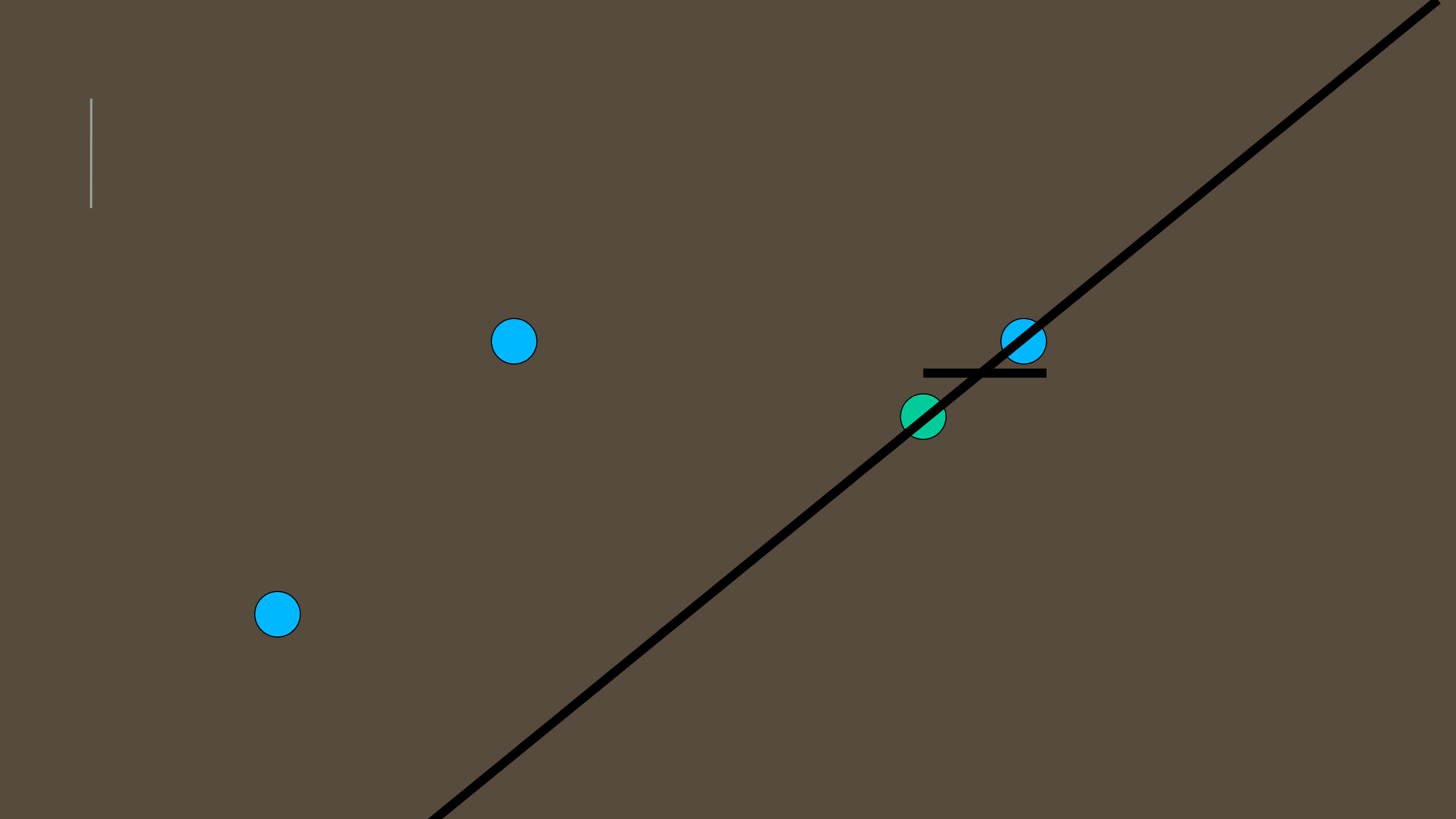
1. Fixe une variance maximal.
2. Trouver le point réellement évalué le plus proche de notre point de départ (qui est lui aussi un point réellement évalué).
3. Tracer un segment L entre les deux points.
4. Utiliser le régresseur pour calculer la variance à la moitié entre les deux points.
5. Si variance < variance maximal: éloigner le point

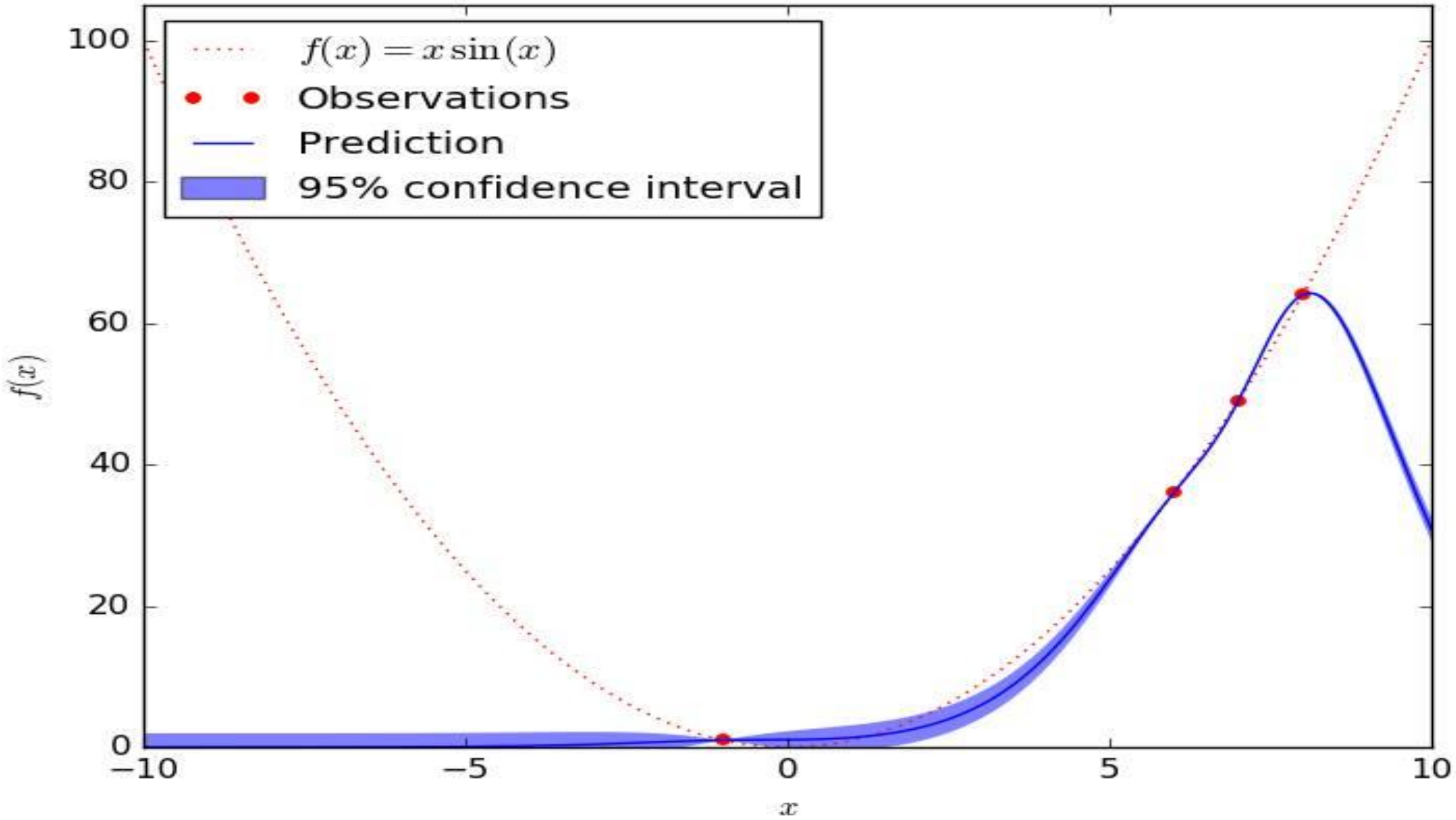
   Sinon: rapprocher le point.

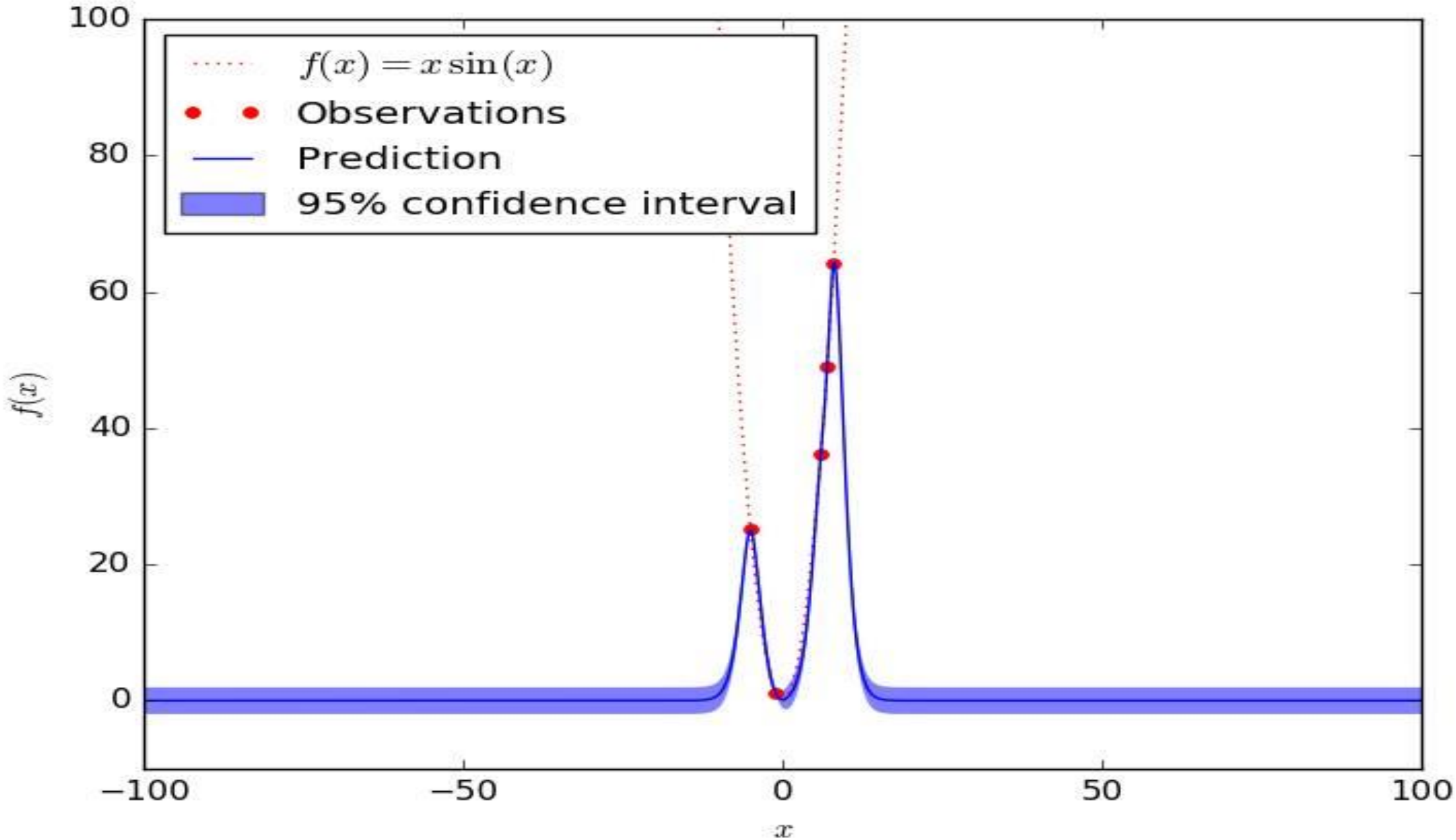









$f(x) = x\sin(x)$
Observations
Prediction
95% confidence interval
100
80
60
40
20
0
$f(x)$
−10
−5
0
5
10
x

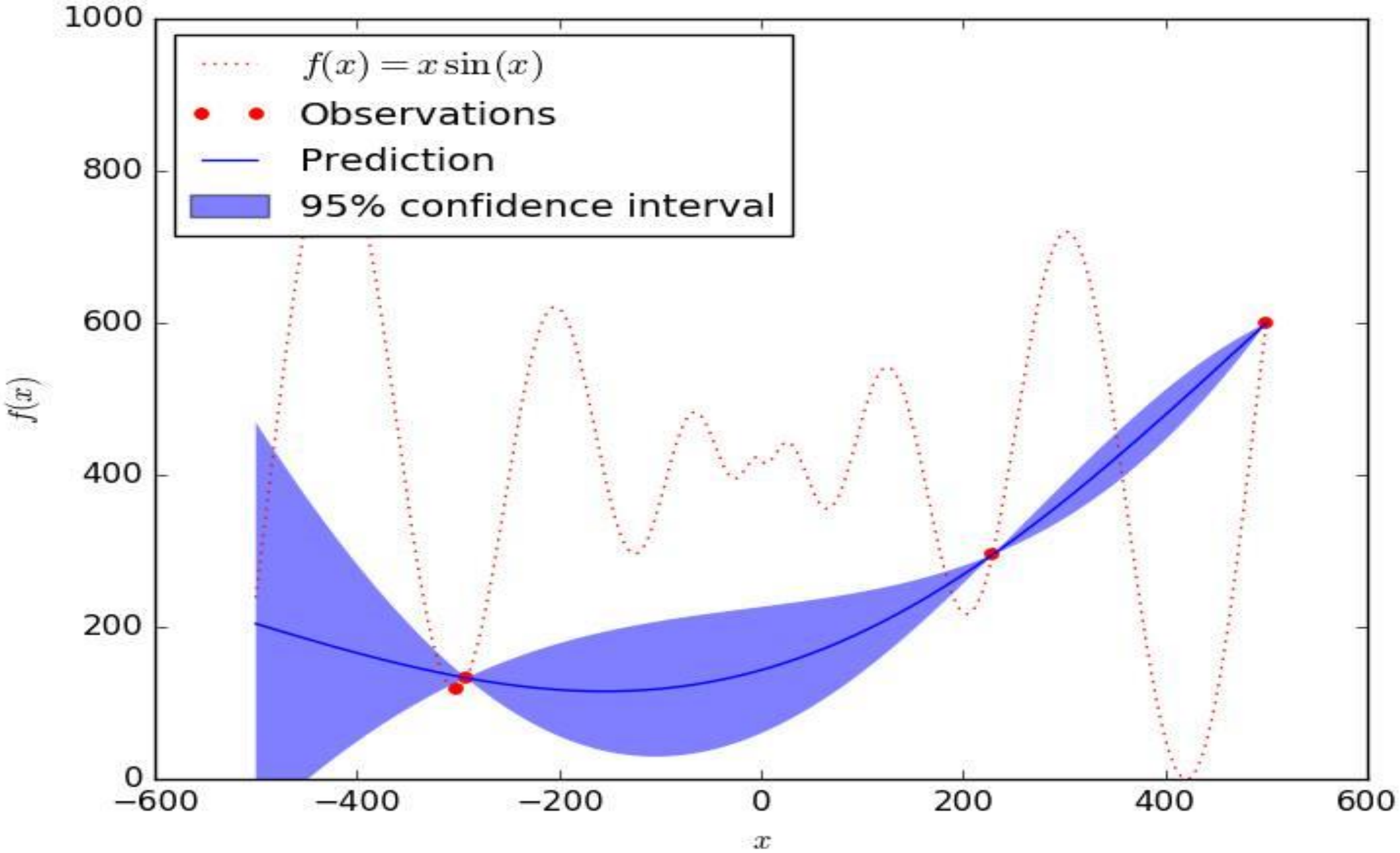

$f(x) = x\sin(x)$
Observations
Prediction
95% confidence interval
100
80
60
40
20
0
$f(x)$
−100
−50
0
50
100
x

Exploration

Aire de recherche: L'aire complete de la function à optimiser.

Augmenter le nombre d'évaluations sur la fonction d'approximation.

Utiliser un régresseur beaucoup plus « smooth » (changer noyau/algorithme régression).


$f(x) = x \sin(x)$
Observations
Prediction
95% confidence interval
1000
800
600
400
200
0
−600
−400
−200
0
200
400
600
$f(x)$
x

Questions ?