



جامعة وهران للعلوم والتكنولوجيا محمد بوضياف

Université des Sciences et de la Technologie d'Oran Mohamed-Boudiaf USTOMB

Département : Informatique

# **Conception d'un outil de comparaison de métaheuristiques sur des problèmes d'optimisation connus**

Présenté par : HADRI-KHOUSSA Salma

Encadrante : Dr. DEKHICI Latifa

Présidente : Dr.KIES Karima

Examinatrice : Dr.FERHANE Samia

Date : 06 Juin 2024

# PLAN

- 01 Problématique
- 02 Introduction à l'optimisation
- 03 Techniques d'optimisation
- 04 Les métaheuristiques
- 05 Comparateur de métaheuristiques
- 06 Fonctions de comparaison
- 07 Résultats et Discussion
- 08 Conclusion et Perspectives

# Problématique



## Objectif

- Développer un outil de comparaison de métaheuristiques.
- Analyser leurs performances en termes de précision, de temps de calcul, et de robustesse.



## Importance de l'étude

- Permettre d'identifier la méthode la plus efficace pour chaque type de problème.



## Contributions

- Aider à choisir la méthode d'optimisation la plus convenable pour une prise de décision éclairée.

# Introduction à l'optimisation

## L'optimisation

- Branche des mathématiques appliquées qui détermine les meilleures solutions possibles (ou optimales) pour des problèmes complexes dans des domaines tels que la finance et les transports.

### 1.Problème du Sac à Dos (domaine de la finance)

- Quels projets choisir pour que le tout rapporte le plus d'argent possible ?

### 2.Problème du Voyageur de Commerce (secteur des transports)

- Trouver le chemin le plus court passant par plusieurs points.

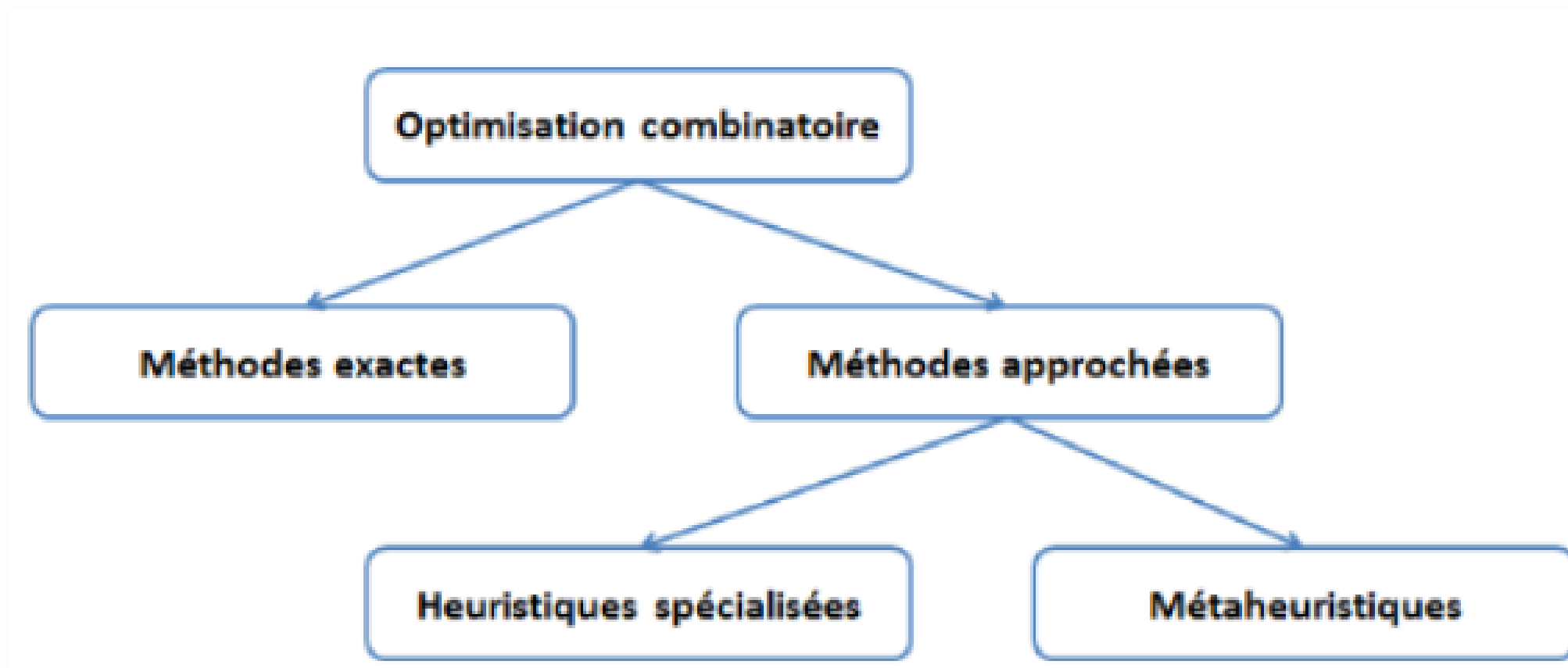
## Fonction objectifs

- But d'un problème d'optimisation ; minimisation ou maximisation.

# Techniques d'optimisation

Nous distinguons

1. Les méthodes exactes : donnent la solution optimale, pour un espace de taille raisonnable
2. Les méthodes approchées : donnent la solution approchée, adaptées aux problèmes où l'espace de recherche est large.



Classification des méthodes approchées

Heuristiques : signifiant "découvrir" approches classiques, dédiées à un problème spécifique. Ex : Branch & Bound.

Métaheuristiques : signifie "au-delà" des approches classiques, Pour une large gamme de problèmes. Ex : PSO.

# Les métaheuristiques

- Les problèmes deviennent de plus en plus complexes.
- Les méthodes classiques atteignent leurs limites en raison de la taille de l'espace des solutions.
- Trouvent une solution proche de l'optimale en un temps raisonnable où l'espace de recherche est immense.
- Car il s'agit du thème proposé.

# Métaheuristiques

```
graph TD; A[Métaheuristiques] --> B[Métaheuristiques à solution unique]; A --> C[Métaheuristiques à population de solution]; B --> D[Recherche locale]; B --> E[Recuit simulé]; B --> F[Recherche tabou]; C --> G[Algorithmes génétiques]; C --> H[Colonies de fourmis]; C --> I[Essaims particuliers];
```

## Métaheuristiques à solution unique

Recherche locale

Recuit simulé

Recherche tabou

## Métaheuristiques à population de solution

Algorithmes génétiques

Colonies de fourmis

Essaims particuliers

# COMPARATEUR DE MÉTAHEURISTIQUES

Comment choisir la métaheuristique la plus adaptée parmi une diversité de méthodes pour résoudre efficacement les problèmes d'optimisation?

## Hypothèse

Concevoir un comparateur qui analyse les performances des différentes métaheuristiques et les résultats en termes de temps d'exécution et de convergence.

Un comparateur se base sur deux critères:

1. La performance (temps passé pour accomplir la tâche souhaitée)
2. La convergence : taux d'erreur de l'algorithme (fiabilité des résultats).



# IMPLÉMENTATION

Dans le développement de la plateforme de comparaison, nous avons utilisé langage de programmation **Python 3.10** et le système d'exploitation **Ubuntu 22.04**

- **VecMetaPy** : Fichier contenant une sélection de métaheuristiques.
- **EvoPy** : Bibliothèque Python pour implémenter des métaheuristiques.
- **Tkinter** : Interface graphique (GUI) pour la sélection des benchmarks, méthodes et les paramètres.

# IMPLÉMENTATION

Accueil

Fonctions

Algorithmes

Paramètres

Graphes

Résultats

☐ F1:  $\sum_{i=1}^n x_i^2$

☐ F2:  $\sum_{i=1}^n |x_i| + \prod_{i=1}^n |x_i|$

☐ F3:  $\sum_{i=1}^{dim} (\sum_{j=1}^i x_j)^2$

☐ F4:  $\max(|x|)$

☐ F5:  $\sum_{i=2}^{dim} [100(x_i - x_{i-1})^2 + (x_{i-1} - 1)^2]$

☐ F6:  $\sum |x + 0.5|^2$

☐ F7:  $\sum_{i=1}^{dim} |x_i|^4 + \text{Uniform}(0,1)$

☐ F8:  $\sum -x_i \sin(\sqrt{|x_i|})$

☐ F9:  $\sum [x^2 - 10\cos(2\pi x)] + 10\dim$

☐ F10:  $-20\exp(-0.2\sqrt{(\sum x_i^2/dim)}) - \exp(\sum \cos(2\pi x_i)/dim) + 20 + \exp(1)$

☐ F11:  $(\sum x_i^2/4000) - \prod \cos(x_i/\sqrt{i}) + 1$

☐ F12:  $(\pi/dim) [10(\sin(\pi(1+(x_1+1)/4)))^2 + \dots]$

☐ F13:  $0.1 [\sin(3\pi x_1)^2 + \sum_{i=1}^{dim} (x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1}))]$

☐ F14:  $((1/500) + \sum(1/(1 + \sum(x_i - a_i)^0)))^{-1}$

☐ F15:  $\sum[a_i - (L_1(b_i^2 + L_2b_i)/(b_i^2 + L_3b_i + L_4))]$

☐ F16:  $4L_1^2 - 2.1L_1^4 + L_1^5/3 + L_1L_2 - 4L_2^2 + 4L_2^4$

☐ F17:  $(L_2 - (L_1^2) * 5.1 / (4\pi^2) + 5 / \pi * L_1 - 6)^2 + 10 (1 - 1/(8\pi))\cos(L_1) + 10$

☐ F18:  $[1 + (L_1 + L_2 + 1)^2 * (19 - 14L_1 + 3L_1^2)]$

☐ F19:  $\sum [-c_1 \exp(-(\sum(a_i(L - p_i)^2)))]$

☐ F20:  $\sum [-c_2 \exp(-(\sum(a_i(L - p_i)^2)))]$

☐ F21:  $\sum[-1/((L - a_1)^{\gamma}(L - a) + c_i)]$

☐ F22:  $\sum[-1/((L - a_2)^{\gamma}(L - a) + c_i)]$

☐ F23:  $\sum[-1/((L - a_3)^{\gamma}(L - a) + c_i)]$

Quitter ✕

Accueil

Fonctions

Algorithmes

Paramètres

Graphes

Résultats

☐ Essaim de salpes (SSA)

☐ Essaim particules (PSO)

☐ Algorithme génétique (GA)

☐ Chauve-souris (BAT)

☐ Lucioles (FFA)

☐ Loup gris (GWO)

☐ Baleine (WOA)

☐ Multi-Verse Optimiseur (MVO)

☐ Flamme des papillons (MFO)

☐ Recherche du coucou (CS)

☐ Faucon-pèlerin (HHO)

☐ Chenille spirale (SCA)

☐ Meilleure amélioration(JAYA)

☐ Différenciation évolutionnaire DE

Quitter ✕

Accueil

Fonctions

Algorithmes

Paramètres

Graphes

Résultats

Taille de la population

-

10

+

Nombre d'itérations

-

3

+

Nombre de tours

-

2

+

Lancer la comparaison

Quitter ✕

Accueil

Fonctions

Algorithmes

Paramètres

Graphes

Résultats

Accueil

Fonctions

Algorithmes

Paramètres

Graphes

Résultats

Aide ?

Quitter ✕

# IMPLÉMENTATION

Accueil

Fonctions

Algorithmes

Paramètres

Graphes

Résultats

☐ F1:  $\sum_{i=1}^n x_i^2$

☐ F2:  $\sum_{i=1}^n |x_i| + \prod_{i=1}^n |x_i|$

☐ F3:  $\sum_{i=1}^{\text{dim}} (\sum_{j=1}^i x_j)^2$

☐ F4:  $\max(|x|)$

☐ F5:  $\sum_{i=2}^{\text{dim}} [100(x_i - x_{i-1}^2)^2 + (x_{i-1} - 1)^2]$

☐ F6:  $\sum |x + 0.5|^2$

☐ F7:  $\sum_{i=1}^{\text{dim}} |x_i|^4 + \text{Uniform}(0,1)$

☐ F8:  $\sum -x_i \sin(\sqrt{(|x_i|)})$

☐ F9:  $\sum [x_i^2 - 10\cos(2\pi x_i)] + 10\text{dim}$

☐ F10:  $-20\exp(-0.2\sqrt{(\sum x_i^2/\text{dim})}) - \exp(\sum \cos(2\pi x_i)/\text{dim}) + 20 + \exp(1)$

☐ F11:  $(\sum x_i^2/4000) - \prod \cos(x_i/\sqrt{i}) + 1$

☐ F12:  $(\pi/\text{dim}) [10(\sin(\pi(1+(x_1+1)/4)))^2 + \sum_{i=1}^{\text{dim}} [(x_i+1)/4]^2]$

☐ F13:  $0.1 [\sin(3\pi x_1)^2 + \sum_{i=1}^{\text{dim}} (x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1}))]$

☐ F14:  $((1/500) + \sum (1/(i + \sum (x_i - a_0)^6))))^{-1}$

☐ F15:  $\sum [a_i - (L_1(b_i^2 + L_2 b_i)/(b_i^2 + L_3 b_i + L_4))]^2$

☐ F16:  $4L_1^2 - 2.1L_1^4 + L_1^6/3 + L_1L_2 - 4L_2^2 + 4L_2^4$

☐ F17:  $(L_2 - (L_1^2) * 5.1 / (4\pi^2) + 5 / \pi * L_1 - 6)^2 + 10 (1 - 1/(8\pi))\cos(L_1) + 10$

☐ F18:  $[1 + (L_1 + L_2 + 1)^2 * (19 - 14L_1 + 3L_1^2 - 14L_2 + 6L_1L_2]$

☐ F19:  $\sum [-c1_i \exp(-(\sum (a_{0i}(L - p_{0i})^2)))]$

☐ F20:  $\sum [-c2_i \exp(-(\sum (a_{0i}(L - p_{0i})^2)))]$

☐ F21:  $\sum [-1/((L - a1_i)^T(L - a_i) + c_i)]$

☐ F22:  $\sum [-1/((L - a2_i)^T(L - a_i) + c_i)]$

☐ F23:  $\sum [-1/((L - a3_i)^T(L - a_i) + c_i)]$

Quitter ✕

## Segment 1 : Les fonctions

Les fonctions de test sont des fonctions mathématiques que nous voulons minimiser, c'est-à-dire trouver les valeurs qui donnent le plus petit résultat possible.

Pour tester un algorithme, nous n'utilisons pas des cas réels, mais des benchmarks, qui sont des fonctions dont nous connaissons déjà la solution. Nous donnons ces fonctions à l'algorithme pour qu'il cherche la solution et la compare avec celle que nous connaissons.

Si l'algorithme s'approche bien de la solution connue, alors c'est un bon algorithme.

# IMPLÉMENTATION

The screenshot shows a web application interface with a dark header bar containing navigation tabs: 'Accueil', 'Fonctions', 'Algorithmes' (highlighted in green), 'Paramètres', 'Graphes', and 'Résultats'. The main content area has a light green background and displays a grid of 10 algorithm names, each preceded by an unchecked checkbox. The algorithms are: Essaim de salpes (SSA), Essaim particules (PSO), Algorithme génétique (GA), Chauve-souris (BAT), Lucioles (FFA), Loup gris (GWO), Baleine (WOA), Multi-Verse Optimiseur (MVO), Flamme des papillons (MFO), Recherche du coucou (CS), Faucon-pèlerin (HHO), and Chenille spirale (SCA). At the bottom center, there is a red button labeled 'Quitter ✕'.

<input type="checkbox"/> Essaim de salpes (SSA)	<input type="checkbox"/> Essaim particules (PSO)	<input type="checkbox"/> Algorithme génétique (GA)	<input type="checkbox"/> Chauve-souris (BAT)
<input type="checkbox"/> Lucioles (FFA)	<input type="checkbox"/> Loup gris (GWO)	<input type="checkbox"/> Baleine (WOA)	<input type="checkbox"/> Multi-Verse Optimiseur (MVO)
<input type="checkbox"/> Flamme des papillons (MFO)	<input type="checkbox"/> Recherche du coucou (CS)	<input type="checkbox"/> Faucon-pèlerin (HHO)	<input type="checkbox"/> Chenille spirale (SCA)
<input type="checkbox"/> Meilleure amélioration(JAYA)	<input type="checkbox"/> Différenciation évolutionnaire DE		

Quitter ✕

## Segment 2 : Les algorithmes pour chercher le minimum

Permet de faire 315 combinaison possibles

Contient les implémentations des différentes métaheuristiques

# IMPLÉMENTATION

Accueil Fonctions Algorithmes Paramètres Graphes Résultats

Taille de la population - 10 +

Nombre d'itérations - 3 +

Nombre de tours - 2 +

Lancer la comparaison

## Segment 3 : Les paramètres d'optimisation

- TAILLE DE LA POPULATION : NOMBRE D'INDIVIDUS DANS LES ESSAIS.
- NOMBRE D'ITÉRATIONS : COMBIEN DE FOIS L'ALGORITHME FAIT DES ESSAIS.
- VARIABLES INITIALES ALÉATOIRES : POUR TESTER LA STABILITÉ DE L'ALGORITHME.
- IMPORTANCE DE LA VARIANCE : CERTAINS ALGORITHMES SONT CONSTANTS ET TROUVENT LA SOLUTION OPTIMALE À CHAQUE ESSAI, PEU IMPORTE LES VALEURS INITIALES. D'AUTRES SONT TRÈS SENSIBLES AUX VALEURS DE DÉPART ET PEUVENT DONNER DES RÉSULTATS VARIABLES.

# VALEURS OPTIMALES DES FONCTIONS

## TESTS POUR LA COMPARAISON

Fonction	Valeur optimale
F1	0
F2	0
F3	0
F4	0
F5	0
F6	0
F7	0 (sans variable aléatoire)
F8	$-418.9829 \times \text{dim}$ (approx.)
F9	0

F10	0
F11	0
F12	0
F13	0
F14 à F15	Dépendent des paramètres spécifiques
F16	-1.0316 (approx.)
F17	0.398 (approx.)
F18	3 (approx.)
F19 à F23	Dépendent des paramètres spécifiques

# LES FONCTIONS CHOISIES

Accueil

Fonctions

Algorithmes

Paramètres

Graphes

Résultats

☒ F1:  $\sum_{i=1}^n x_i^2$

☒ F4:  $\max(|x|)$

☐ F7:  $\sum_{i=1}^{\dim} ix_i^4 + \text{Uniform}(0,1)$

☐ F10:  $-20\exp(-0.2\sqrt{(\sum x_i^2/\dim)}) - \exp(\sum \cos(2\pi x_i)/\dim) + 20 + \exp(1)$

☐ F13:  $0.1 [\sin(3\pi x_1)^2 + \sum_{i=1}^{\dim} (x_i - 1)^2 (1 + \sin^2(3\pi x_{i+1}))]$

☐ F16:  $4L_1^2 - 2.1L_1^4 + L_1^5/3 + L_1L_2 - 4L_2^2 + 4L_2^4$

☐ F19:  $\sum [-c1_i \exp(-(\sum(a_{ij}(L - p_{ij})^2)))]$

☐ F22:  $\sum [-1/((L - a2)^T(L - a) + c_i)]$

☐ F2:  $\sum_{i=1}^n |x_i| + \prod_{i=1}^n |x_i|$

☐ F5:  $\sum_{i=2}^{\dim} [100(x_i - x_{i-1})^2 + (x_{i-1} - 1)^2]$

☐ F8:  $\sum -x_i \sin(\sqrt{(|x_i|)})$

☐ F11:  $(\sum x_i^2/4000) - \prod \cos(x_i/\sqrt{i}) + 1$

☐ F14:  $((1/500) + \sum(1/(1 + \sum(x_i - a_{ij})^6)))^{-1}$

☐ F17:  $(L_2 - (L_1^2) * 5.1 / (4\pi^2) + 5 / \pi * L_1 - 6)^2 + 10 (1 - 1/(8\pi))\cos(L_1) + 10$

☐ F20:  $\sum [-c2_i \exp(-(\sum(a_{ij}(L - p_{ij})^2)))]$

☐ F23:  $\sum [-1/((L - a3)^T(L - a_i) + c_i)]$

☐ F3:  $\sum_{i=1}^{\dim} (\sum_{j=1}^i x_j)^2$

☐ F6:  $\sum |x + 0.5|^2$

☐ F9:  $\sum [x_i^2 - 10\cos(2\pi x_i)] + 10\dim$

☐ F12:  $(\pi/\dim) [10(\sin(\pi(1+(x_1+1)/4)))^2 + \sum_{i=1}^{\dim} [(x_i+1)/4]^2]$

☐ F15:  $\sum [a_i - (L_1(b_i^2 + L_2b_i)/(b_i^2 + L_3b_i + L_4))]^2$

☐ F18:  $[1 + (L_1 + L_2 + 1)^2 * (19 - 14L_1 + 3L_1^2 - 14L_2 + 6L_1L_2$

☐ F21:  $\sum [-1/((L - a1_i)^T(L - a_i) + c_i)]$

Quitter ✕

# LES MÉTHODES CHOISIES

Accueil

Fonctions

Algorithmes

Paramètres

Graphes

Résultats

☐ Essaim de salpes (SSA)

☐ Essaim particules (PSO)

☐ Algorithme génétique (GA)

☐ Chauve-souris (BAT)

☒ Lucioles (FFA)

☒ Loup gris (GWO)

☐ Baleine (WOA)

☐ Multi-Verse Optimiseur (MVO)

☐ Flamme des papillons (MFO)

☐ Recherche du coucou (CS)

☐ Faucon-pèlerin (HHO)

☐ Chenille spirale (SCA)

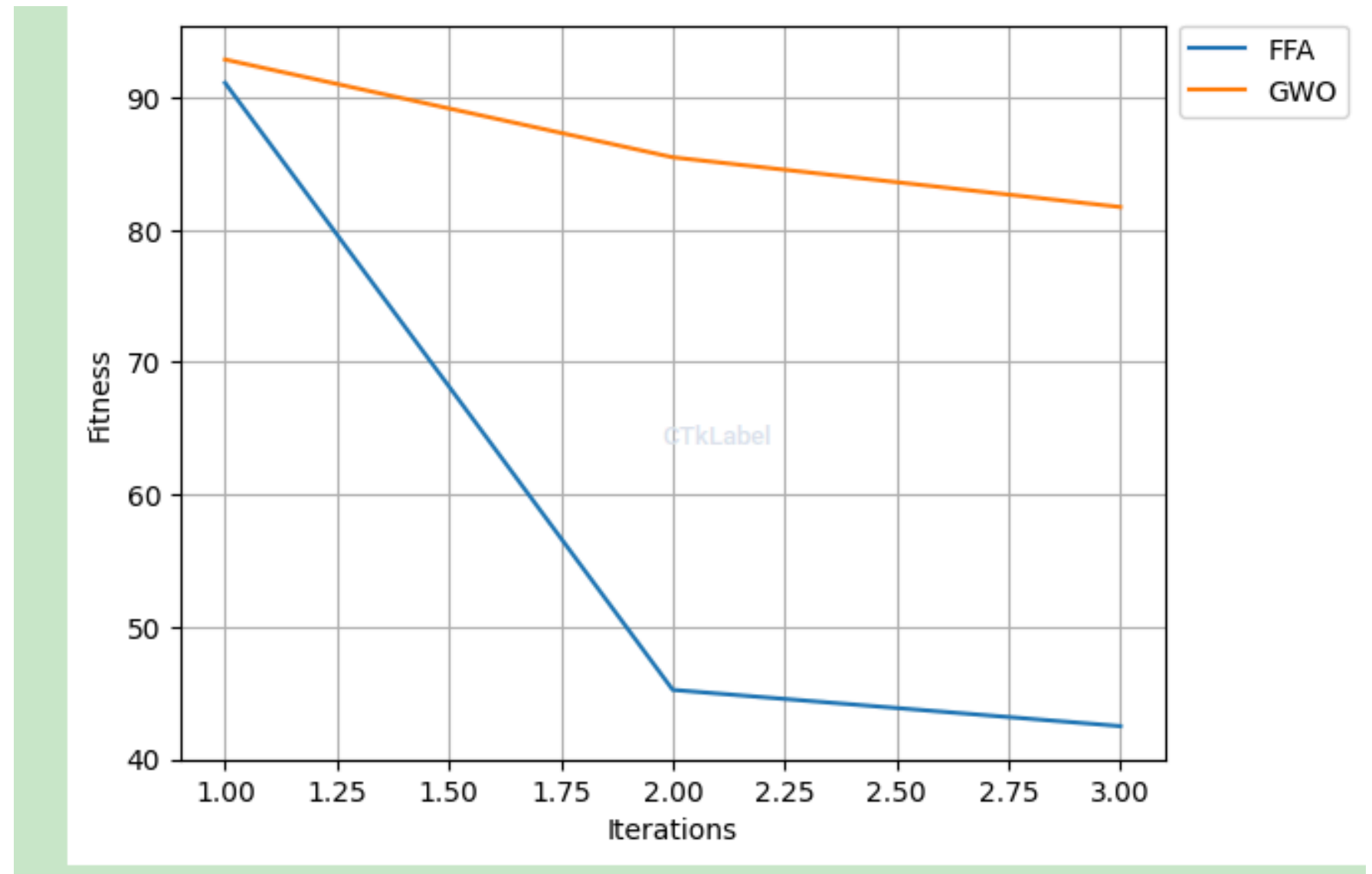
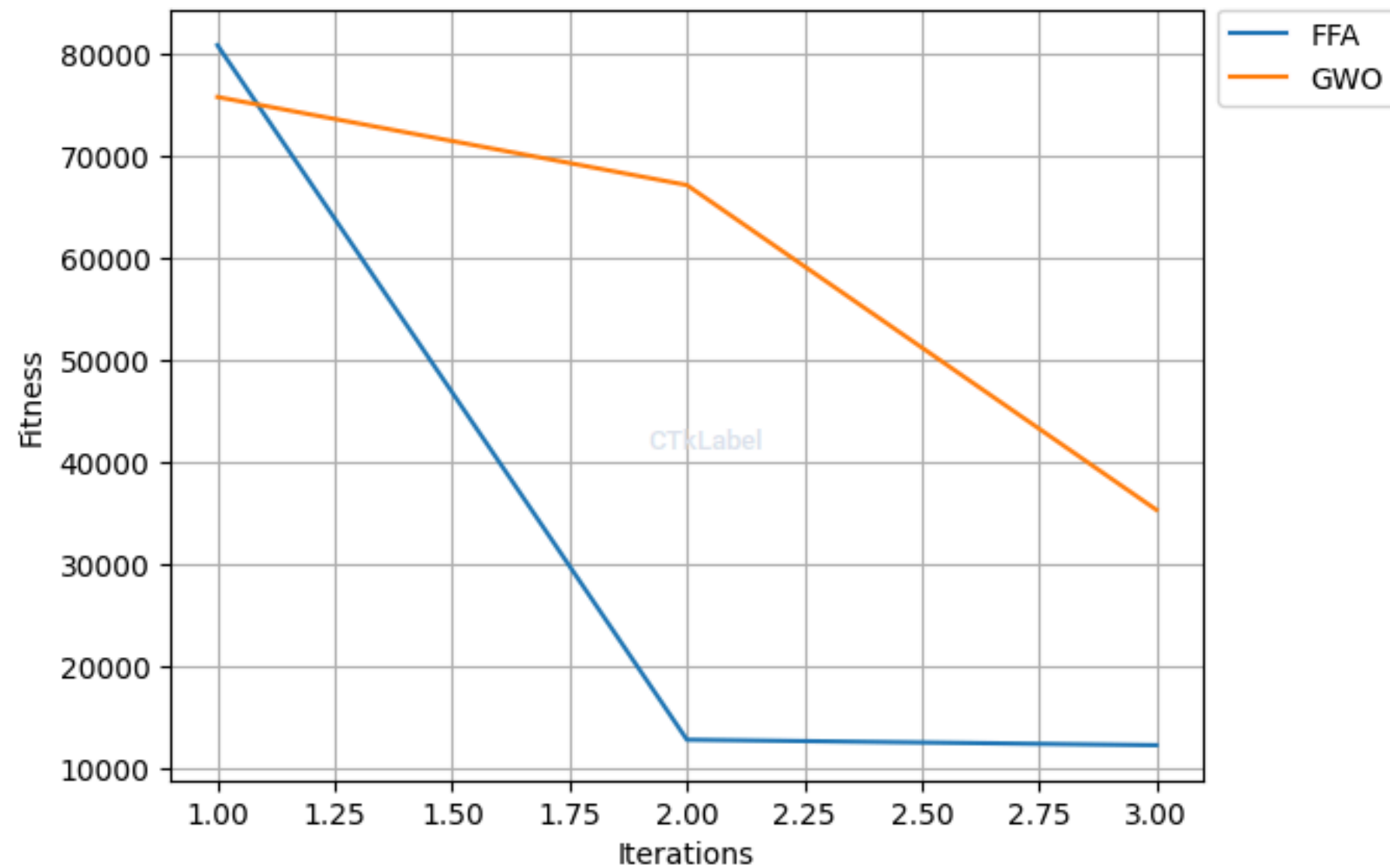
☐ Meilleure amélioration(JAYA)

☐ Différenciation évolutionnaire DE

Quitter ✕



# COMPARAISON



# RÉSULTATS ET DISCUSSION

Critères d'évaluation des métaheuristiques :

1. **Précision** : L'erreur doit être minimale.
2. **Rapidité** : Le temps d'exécution doit être raisonnable.

Analyse des méthodes :

Identification de la méthode la plus rapide et de la méthode la plus précise.

Détermination de la méthode offrant un **compromis entre précision et rapidité**.


# Résultats

Méthode	Fonction	Temps_Execution	Iter3
FFA	F1	0.02	12268
FFA	F4	0.01	42.49
GWO	F1	0.01	35357.1
GWO	F4	0.01	81.73



# CONCLUSION

L'OBJECTIF DE NOTRE PROJET DE FIN D'ÉTUDE EST ATTEINT :

- ✓ Développer un outil de comparaison de métaheuristiques.
  - ✓ Analyser leurs performances en termes de précision et de temps de calcul.
- 

# PERSPECTIVES

- **Le parallélisme** peut être une technique pour traiter les grandes instances de problèmes complexes. Décomposition de l'espace de recherche en sous-espaces. Plusieurs processeurs ou machines pour exécuter des algorithmes simultanément. Réduit le temps de calcul et améliore la qualité des solutions.
- **Généraliser à domaines plus complexes** en IA : optimisation des techniques d'apprentissage dans les réseaux de neurones...)



**MERCI DE VOTRE ATTENTION**