



Module : *Algorithmes intelligents pour l'optimisation*

## TP n° 01

### Enoncé de l'exercice

- Trouver le maximum de la fonction  $f(x) = 15x - x^2$  sachant que  $x$  est un entier se varie entre 1 et 15
- Supposons que la taille de la population (nombre de chromosomes) est = 6 , probabilité de croisement est  $P_c = 0.7$  et de mutation  $P_m = 0.001$  et la fonction de fitness est représenté comme suit :  $15x - x^2$

### Solution

#### 1. Objectif

Notre objectif est de trouver la valeur  $x$  à condition que  $f(x)$  prend sa valeur maximale.

#### 2. Les étapes de l'algorithme génétique

##### 2.1.Représentation des données sous forme de gène et chromosome

Chaque valeur de  $x$  sera représentée en binaire (Tableau 01)

Entier	Code binaire	Entier	Code binaire	Entier	Code binaire
1	0001	6	0110	11	1011
2	0010	7	0111	12	1100
3	0011	8	1000	13	1101
4	0100	9	1001	14	1110
5	0101	10	1010	15	1111

Tableau 01

Le gène est représenté par 0 ou 1 tandis que le chromosome (solution) est représenté par 4 *genes* et la population aussi est représentée par 6 *chromosomes* voir (Figure 01)

Exemple  $x = 1$  est représenté comme suit :

0	1	0	1
---	---	---	---

Figure 01

#### 2.2.Initialisation des variables

##### 2.2.1. chromosomes

On va générer aléatoirement 6 chromosomes (solutions  $x_1 \dots x_6$ ) parce que la taille initiale de la population  $N = 6$  (Tableau 02)

##### 2.2.2. Fonction de fitness (optimum)

Calculer la fonction de fitness pour chaque chromosome (Tableau 02)

Chromosome	Codification	Entier	Fitness	Fitness %
$x_1$	1100	12	36	16.5
$x_2$	0100	4	44	20.2
$x_3$	0001	1	14	6.4
$x_4$	1110	14	14	6.4
$x_5$	0111	7	56	25.7
$x_6$	1001	9	54	24.8

Tableau 02

### 2.3. Représentation graphique

Fonction et solutions initiales (figure 02(a)) objectif final (figure 02 (b))

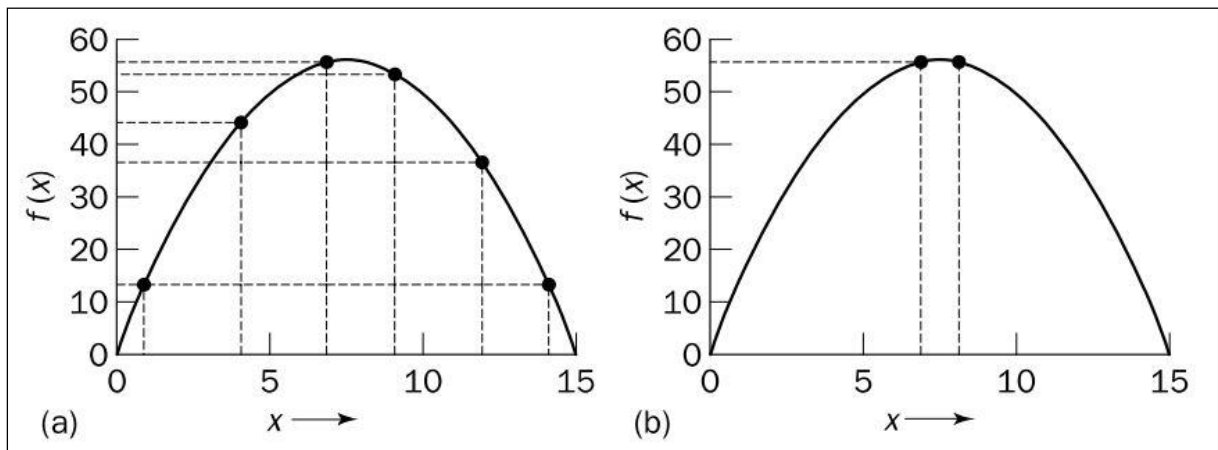


Figure 02

### 2.4. Sélection des chromosomes pour la reproduction

$P_c = 0.7$  et  $N = 6$  alors

*Selection* =  $0.7 * 6 \sim 4$  chromosomes seront sélectionnés aléatoirement de la population initiale (sélection par roulette Figure 03).

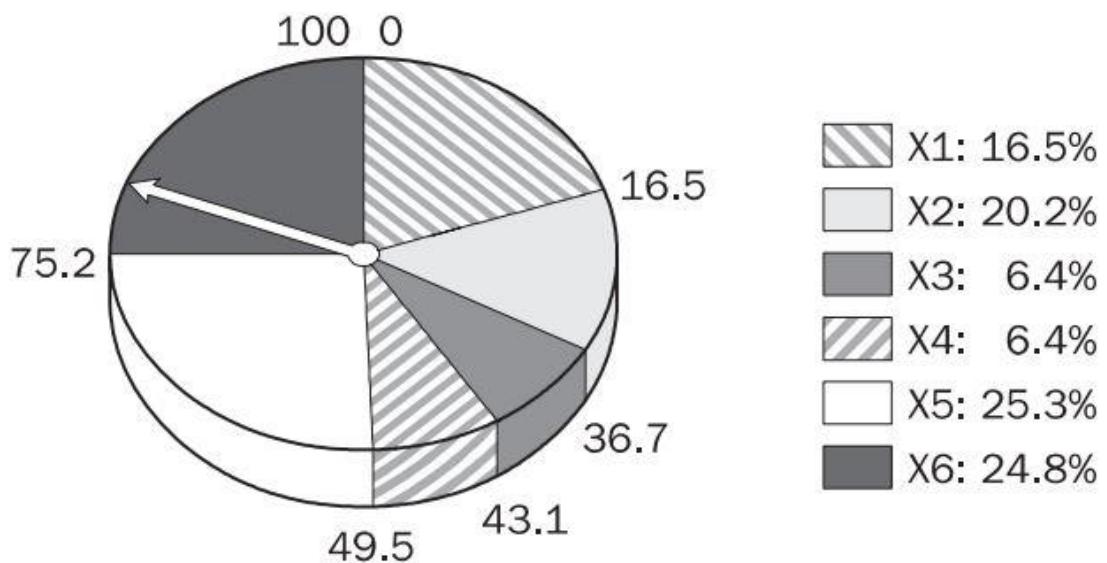


Figure 03

## 2.5. Croisement

Le croisement entre les parents a été appliqué sur un point (Figure 04)

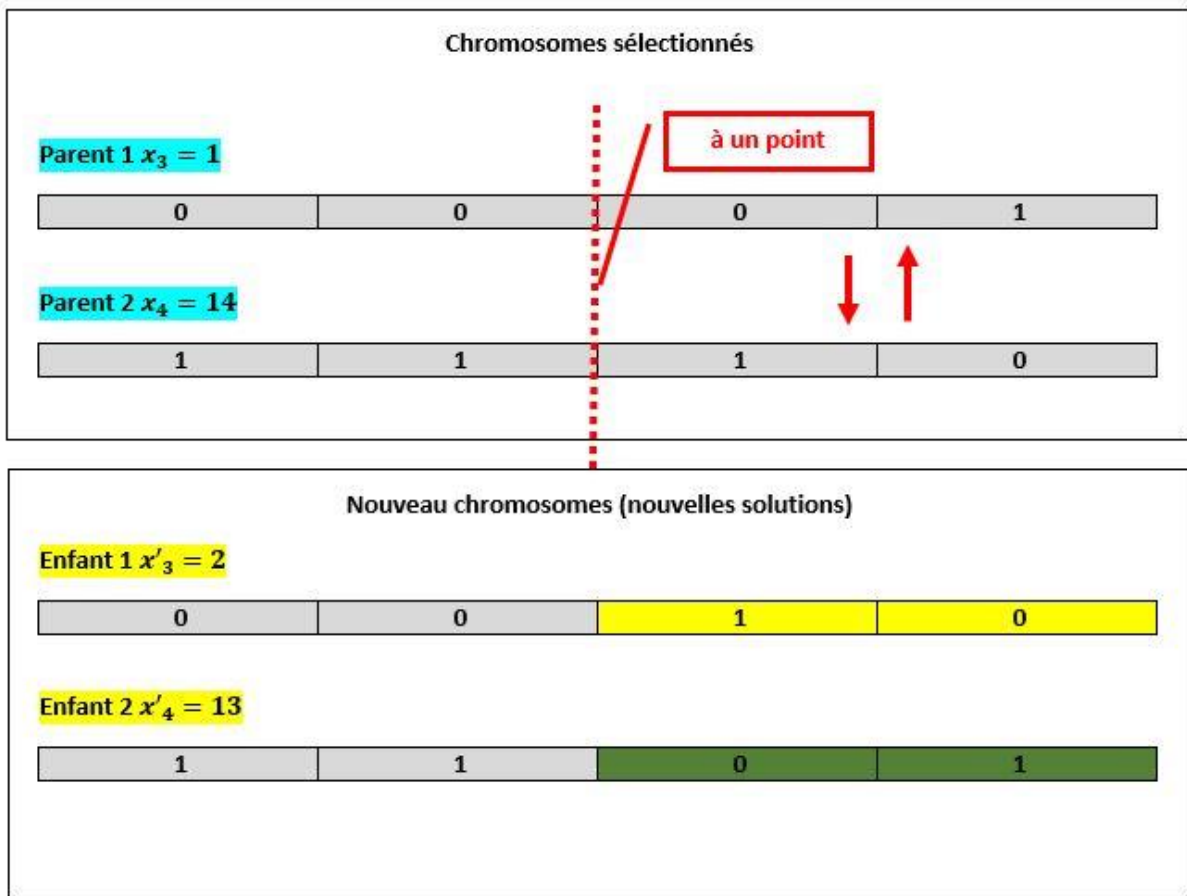


Figure 04

## 2.6. Croisement

Mutation est faite comme suit : un gène sera sélectionné aléatoirement et remplacé par 0 ou 1 (Figure 05).

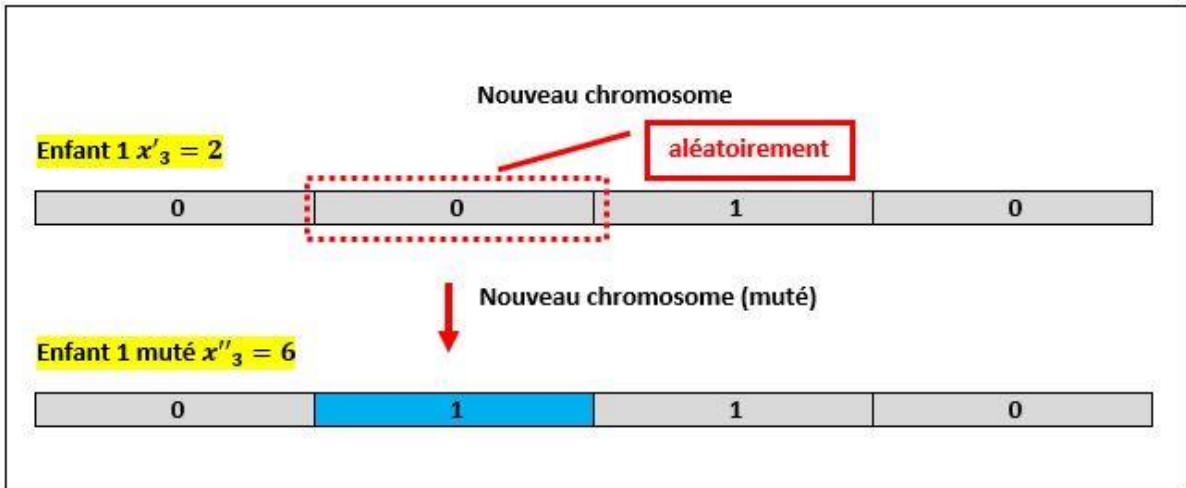


Figure 05

### 2.7. Création de la nouvelle population

Les nouveaux enfants obtenus et les chromosomes non sélectionnés seront ajoutés à la nouvelle population.

On répète les opérations de *croisement et mutation* de tous les parents et on ajoute à la fin les chromosomes non sélectionnés pour l'obtention d'une nouvelle génération composée de 6 chromosomes .

### 2.8. Nombre de génération

L'algorithme s'arrête après un nombre d'itération initialisé au début (*nombre de générations* )  
Figure 06.

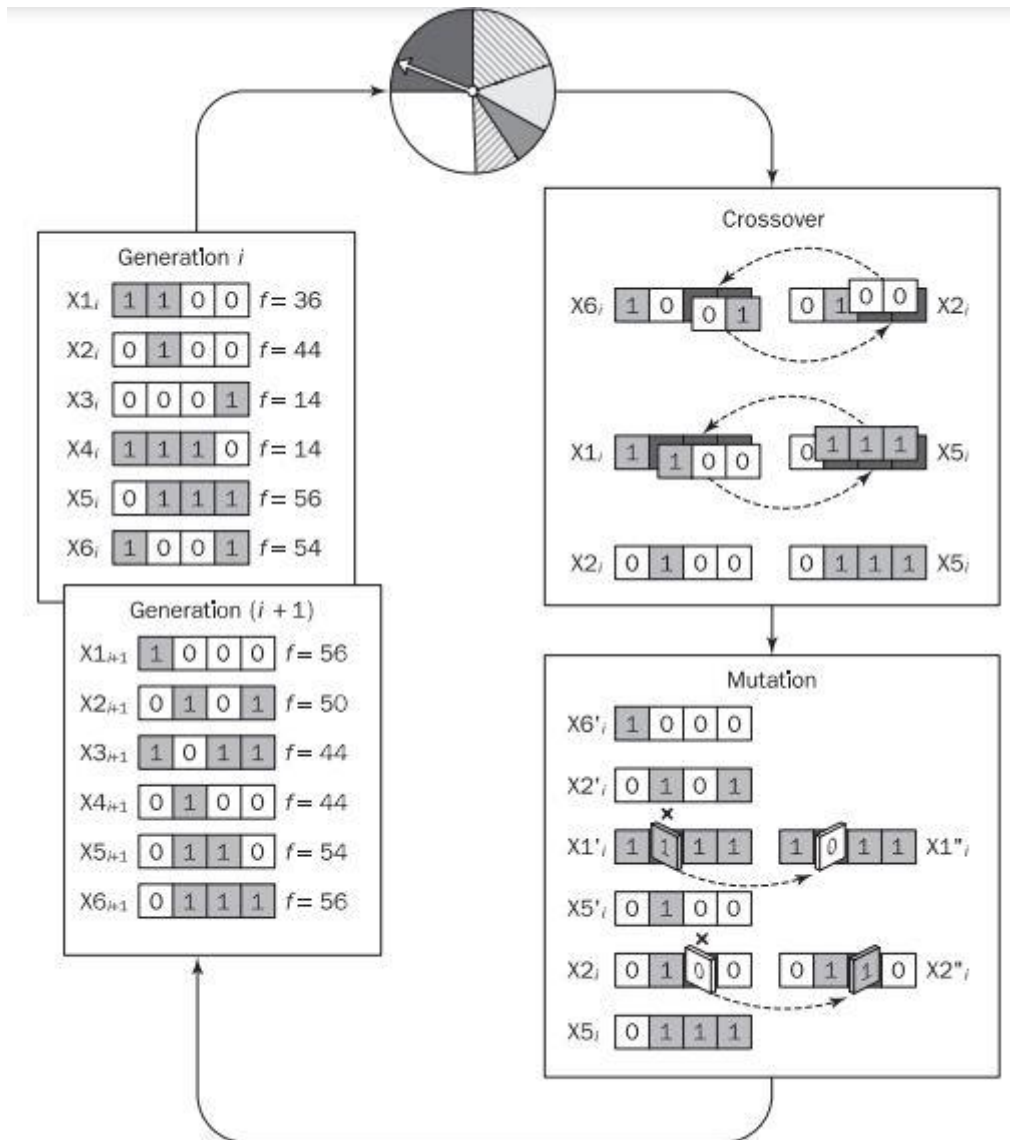


Figure 06

### 2.9. Critère d'arrêt

Le critère d'arrêt se base sur l'optimisation des résultats obtenus (chromosomes) c.-à-d. la valeur des fonctions de fitness sont presque les même dans les dernières générations ; dans ce cas l'algorithme s'arrête et on extrait la valeur de  $x$  (chromosome) qui représente la valeur maximale de la fonction de fitness de la dernière génération

$$\text{Max (fonction de fitness)} \gg \gg x = \text{binaire}$$

$$x = \text{binaire} \gg \gg x = \text{entier solution optimale}$$

### 3. Amélioration des résultats

On peut améliorer les résultats de l'algorithme génétique selon l'analyse de la courbe [fonction de fitness et les générations ( $x, y$ )] ainsi que les critères suivants :

- Augmenter la taille de la population initiale (nombre de chromosomes) et diminuer le nombre de génération (itération) Figure 7 et Figure 8.
- Améliorer la probabilité de mutation  $P_m = 0.01$  Figure 9 (pour ne pas tomber dans l'*optimum local* Figure 10).
- Si les résultats obtenus ne sont pas bons vous devez vérifier ou proposer une autre fonction de fitness.

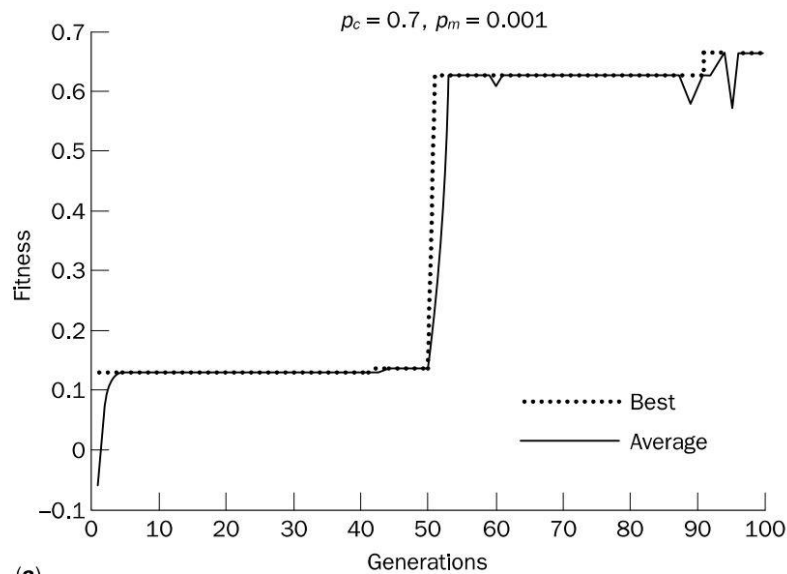


Figure 7

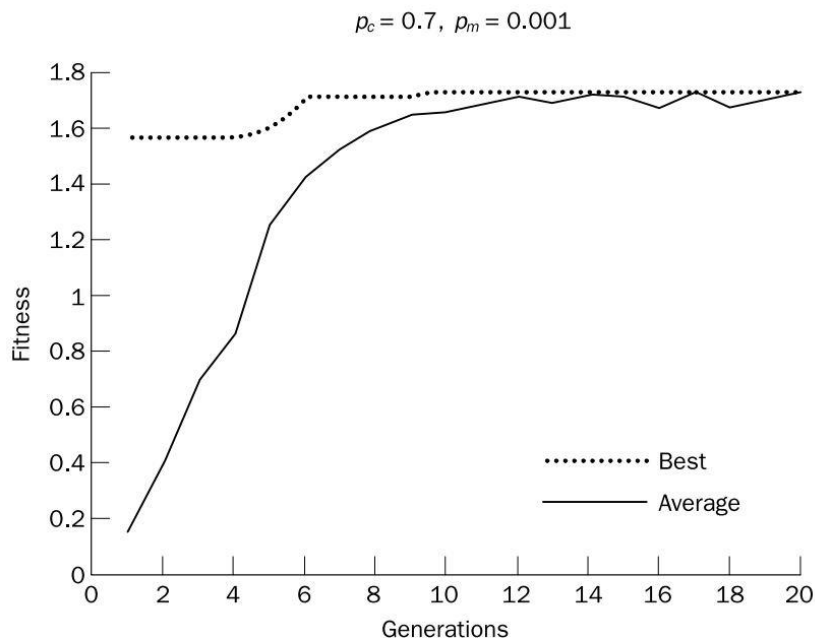
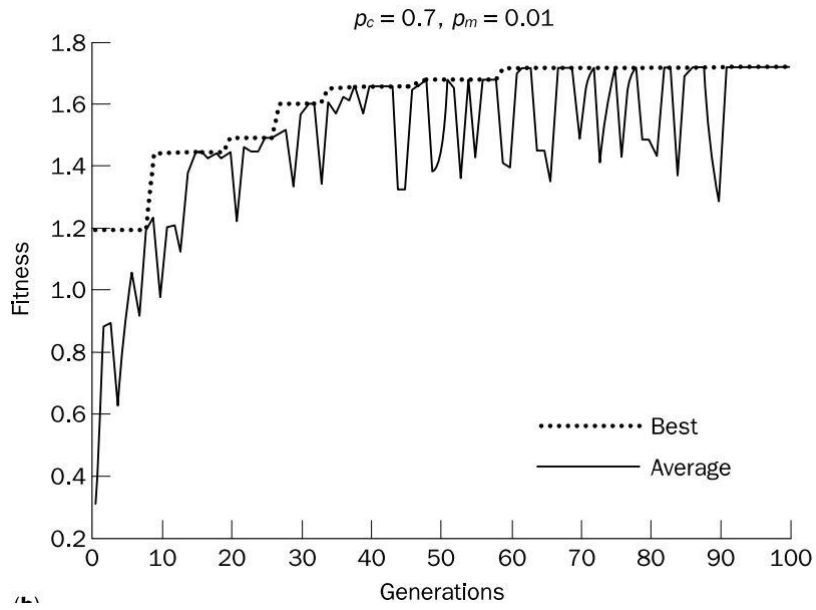
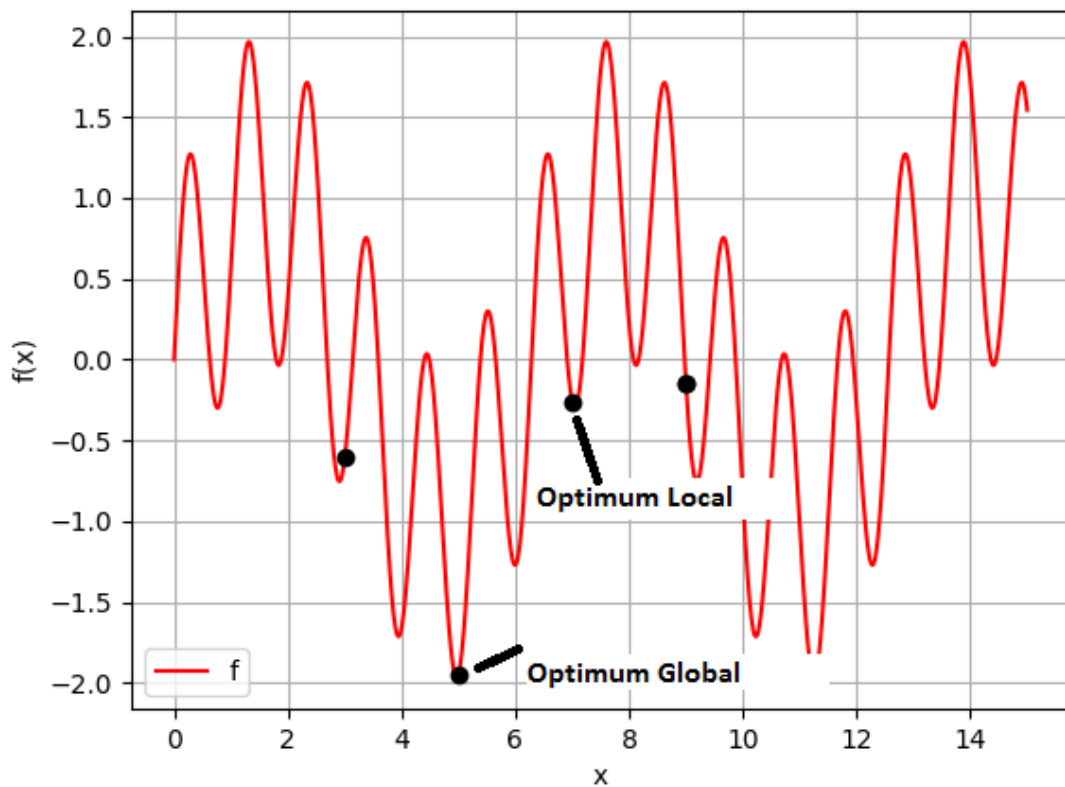


Figure 8



**Figure 9**



#### 4. Implémentation

L'algorithme génétique qui résout cette fonction  $15x - x^2$  est implémenté en langage python (*TP01.py*).