Rendu TP

Résolution du problème de sélection d'attributs avec JMetalPy.

Prof

Julie JACQUES

Membres de l'équipe

- Arnaud KADERI
- Mouctar Amadou FOFANA

Master 2 MIAGE, Université de Lille

1. Introduction

La sélection d'attributs est un processus dans le domaine de l'apprentissage automatique et de l'exploration de données qui vise à identifier et à choisir un sous-ensemble optimal d'attributs (variables) à partir d'un ensemble plus vaste d'attributs disponibles. L'objectif de la sélection d'attributs est de réduire la dimensionnalité des données en conservant uniquement les caractéristiques les plus importantes et informatives, tout en éliminant celles qui peuvent être redondantes, bruyantes ou moins significatives.

```
%pip install scikit-learn
%pip install matplotlib
```

```
Requirement already satisfied: scikit-learn in /usr/local/lib/python3.11/site-packages (1.4
Requirement already satisfied: numpy<2.0,>=1.19.5 in /usr/local/lib/python3.11/site-packages
Requirement already satisfied: scipy>=1.6.0 in /usr/local/lib/python3.11/site-packages (from
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.11/site-packages (fro
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.11/site-packag

[notice] A new release of pip is available: 23.3.2 -> 24.0
[notice] To update, run: python3.11 -m pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
Requirement already satisfied: matplotlib in /usr/local/lib/python3.11/site-packages (3.8.2
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.11/site-packages (
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.11/site-packages (from
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.11/site-packages
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.11/site-packages
Requirement already satisfied: numpy<2,>=1.21 in /usr/local/lib/python3.11/site-packages (fr
Requirement already satisfied: packaging>=20.0 in /Users/girubuntu/Library/Python/3.11/lib/p
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.11/site-packages (from ma
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.11/site-packages (
Requirement already satisfied: python-dateutil>=2.7 in /Users/girubuntu/Library/Python/3.11/
Requirement already satisfied: six>=1.5 in /Users/girubuntu/Library/Python/3.11/lib/python/s

[notice] A new release of pip is available: 23.3.2 -> 24.0
[notice] To update, run: python3.11 -m pip install --upgrade pip
Note: you may need to restart the kernel to use updated packages.
```

```
# importation des datasets
from sklearn.datasets import load_iris
from sklearn.datasets import load_breast_cancer

# à enlever car la target de diabetes n'est pas bien définie
#from sklearn.datasets import load_diabetes

from jmetal.core.problem import BinaryProblem
from jmetal.core.solution import BinarySolution
# import des librairie metals
from jmetal.algorithm.singleobjective.local_search \
import LocalSearch
from jmetal.operator import BitFlipMutation

#from jmetal.util.observer import ProgressBarObserver
from jmetal.util.observer import PrintObjectivesObserver
from jmetal.util.termination_criterion \
import StoppingByEvaluations
from jmetal.util.observer import ProgressBarObserver

# Importez le problème OneMax inclu
from jmetal.problem import OneMax
import pandas as pd
import time as tm
import random
import matplotlib.pyplot as plt

from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import accuracy_score, recall_score, f1_score
```

```
class SA(OneMax):
    def __init__(self, dataset: pd.DataFrame, classification_algorithm: str, select_features
        super(OneMax, self).__init__()
        self.number_of_bits = dataset.data.shape[1]
        self.number_of_objectives = 4
        self.number_of_variables = 1
        self.number_of_constraints = 0
        self.dataset = dataset
        self.obj_directions = [self.MINIMIZE]
        self.obj_labels = ['SA']
        self.classification_algorithm = classification_algorithm
        self.select_features = select_features

    def create_solution(self) -> BinarySolution:
      """
      Crée une solution binaire aléatoire
      On va prendre un vecteur de taille number_of_bits
      et chaque élément du vecteur sera soit 0 soit 1
      à 0 ça veut dire que l'attribut correspondant n'est pas sélectionné
      et à 1 ça veut dire que l'attribut correspondant est sélectionné
      """
      new_solution = BinarySolution(number_of_variables=self.number_of_variables,
                                    number_of_objectives=self.number_of_objectives,
                                    number_of_constraints=self.number_of_constraints)
      if self.select_features:
        new_solution.variables[0] = [random.randint(0, 1) for _ in range(self.number_of_bits
        random_idx = random.randint(0, self.number_of_bits-1)
        new_solution.variables[0][random_idx] = 1
      else:
        new_solution.variables[0] = [1 for _ in range(self.number_of_bits)]
      return new_solution


    def evaluate(self, solution: BinarySolution) -> BinarySolution:
        """
        Évalue la solution avec les algos knn et renvoie un score par rapport
        au score de classification de l'algo knn
        """
        # On récupère les attributs sélectionnés
        selected_features = [i for i in range(self.number_of_bits) if solution.variables[0]
        if not selected_features:
            random_bit = random.randint(0, self.number_of_bits-1)
            solution.variables[0][random_bit] = 1
            selected_features = [i for i in range(self.number_of_bits) if solution.variables
        # On récupère les colonnes correspondantes dans le dataset
        X = self.dataset.data
        y = self.dataset.target
        # On divise le dataset en ensemble d'entrainement et ensemble de test
        # 70% pour l'entrainement et 30% pour le test
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3)
        # On entraine le modèle
        if self.classification_algorithm == 'KNN':
            model = KNeighborsClassifier(n_neighbors=self.number_of_bits+1)
        elif self.classification_algorithm == 'SVM':
            model = SVC()
        model.fit(X_train, y_train)
        # On prédit les résultats
        y_pred = model.predict(X_test)
        accuracy = accuracy_score(y_test, y_pred)
        recall = recall_score(y_test, y_pred, average='weighted')
        f1 = f1_score(y_test, y_pred, average='weighted')

        # On renvoie le score de classification
        solution.objectives[0] = model.score(X_test, y_test)
        solution.objectives[1] = recall
        solution.objectives[2] = f1
        solution.objectives[3] = accuracy

        return solution


    def get_name(self) -> str:
        return 'SA'


from jmetal.algorithm.singleobjective.genetic_algorithm \
import GeneticAlgorithm
from jmetal.operator import BitFlipMutation, SPXCrossover
from jmetal.problem import OneMax
from jmetal.util.termination_criterion import StoppingByEvaluations


def run_genetic_algorithm(population_size, offspring_population_size,
                          mutation_rate, crossover_rate, dataset, classification_algorithm,
                          select_features,
                          max_evaluations=5000):
    """
    Runs a genetic algorithm to solve the SA problem.

    Args:
        population_size (int): The size of the initial population.
        offspring_population_size (int): The size of the offspring
        population.
        mutation_rate (float): The probability of mutation for each
        bit in the population.
        crossover_rate (float): The probability of crossover between two
        individuals in the population.
        max_evaluations (int, optional): The maximum number of evaluations allowed.
        Defaults to 5000.
        dataset (pd.DataFrame): The dataset to use for the SA problem.
    Returns:
        int: The fitness value of the best solution found by the genetic algorithm.
    """

    problem = SA(dataset, classification_algorithm, select_features)
    algorithm = GeneticAlgorithm(
        problem=problem,
        population_size=population_size,
        offspring_population_size=offspring_population_size,
        mutation=BitFlipMutation(mutation_rate / problem.number_of_bits),
        crossover=SPXCrossover(crossover_rate),
        termination_criterion=StoppingByEvaluations(max_evaluations=max_evaluations),
    )


    progress_bar = ProgressBarObserver(max=max_evaluations)
    algorithm.observable.register(progress_bar)

    # Exécutez l'algorithme de recherche locale
    algorithm.run()
    result = algorithm.get_result()
    print('binary solution: ', result.get_binary_string())
    print('fitness: ', result.objectives[0])

    return result.objectives[0], result.objectives[1], result.objectives[2], result.objectiv




def train(problem_function, problem_function_params, nb_runs=20):
  """
  Trains a problem function by running it multiple times
  and collecting fitness results and running times.

  Parameters:
  problem_function (callable): The problem function to be trained.
  problem_function_params (dict): The parameters to be passed
  to the problem function.
  nb_runs (int, optional): The number of times to run the problem function.
  Defaults to 20.

  Returns:
  pd.Series: A series containing the fitness results of each run.
  pd.Series: A series containing the running times of each run.
  """
```

```
    fitnesses = []
    running_times = []
    rappels = []
    f1s = []
    precisions = []

    for _ in range(nb_runs):
        start_time = tm.time()
        fitness_result, rappel, f1, precision = problem_function(**problem_function_params)
        execution_time = tm.time() - start_time

        fitnesses.append(fitness_result)
        rappels.append(rappel)
        f1s.append(f1)
        precisions.append(precision)

        running_times.append(execution_time)

    return pd.Series(fitnesses), pd.Series(running_times), pd.Series(rappels), pd.Series(f1s),
```

Algorithme génétique

Ci-dessous, l'algorithme génétique est évaluée sur 3000 itérations (évaluations), appliquée à un problème SA (Sélection d'attributs) sur deux datasets load_iris et load_breast_cancer . Pour garantir des résultats fiables, chaque expérimentation a été répétée 10 fois avec des initialisations différentes.

L'algorithme générique a été évalué en faisant varier la taille de la population initiale (`population_size`), la probabilité de mutation (`mutation`), et le taux de crossover(`crossover_rate`). Pour une question de temps de calcul, nous considérerons ici que `offspring_population_size` = `population_size`.

La population_size a comme valeur 3O, la mutation_rate est à O.2 et le crossover_rate est à 1

On va lancer le modèle SVM sur load_iris avec sélection d'attributs avec les paramètres optimaux

```
dataset = load_iris()
classification_algorithm = 'SVM'
select_features = True
SAProblem_params_optimal_ag = {
         "population_size" : 30
        ,"offspring_population_size" : 30
        ,"mutation_rate" : 0.2
        ,"crossover_rate" : 1
        ,"dataset" : dataset
        ,"classification_algorithm" : classification_algorithm
        ,"select_features" : select_features
        , "max_evaluations" : 3000
    }

fitnesses, running_times, rappels, f1s, precisions = train(problem_function=run_genetic_algo
                                  problem_function_params=SAProblem_params_optimal_ag, nb_run
```

```
[2024-03-10 23:01:49,061] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:01:49,063] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:01:49,177] [jmetal.core.algorithm] [DEBUG] Initializing progress...
[2024-03-10 23:01:49,178] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:08<00:00, 343.91it/s]
[2024-03-10 23:01:57,902] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:01:57,902] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:01:57,903] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:01:57,991] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1111
fitness:  0.8444444444444444

[2024-03-10 23:01:57,993] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:08<00:00, 346.24it/s]
[2024-03-10 23:02:06,658] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:02:06,658] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:02:06,659] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:02:06,748] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1100
fitness:  0.8444444444444444

[2024-03-10 23:02:06,749] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:08<00:00, 335.23it/s]
[2024-03-10 23:02:15,699] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:02:15,699] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:02:15,700] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:02:15,787] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  0101
fitness:  0.7777777777777778

[2024-03-10 23:02:15,789] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:08<00:00, 344.51it/s]
[2024-03-10 23:02:24,497] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:02:24,497] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:02:24,498] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:02:24,583] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  0111
```

```
fitness:  0.8444444444444444

[2024-03-10 23:02:24,584] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:08<00:00, 338.81it/s]
[2024-03-10 23:02:33,439] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:02:33,439] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:02:33,440] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:02:33,526] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  0111
fitness:  0.8444444444444444

[2024-03-10 23:02:33,528] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:08<00:00, 344.93it/s]
[2024-03-10 23:02:42,225] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:02:42,226] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:02:42,226] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:02:42,312] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1011
fitness:  0.8444444444444444

[2024-03-10 23:02:42,313] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:08<00:00, 345.36it/s]
[2024-03-10 23:02:51,000] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:02:51,001] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:02:51,001] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:02:51,090] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1101
fitness:  0.8

[2024-03-10 23:02:51,091] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:08<00:00, 344.39it/s]
[2024-03-10 23:02:59,803] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:02:59,803] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:02:59,803] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:02:59,890] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1010
fitness:  0.8444444444444444

[2024-03-10 23:02:59,892] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:08<00:00, 336.99it/s]
[2024-03-10 23:03:08,794] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:03:08,795] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:03:08,795] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:03:08,881] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  0101
fitness:  0.8222222222222222
```

```
[2024-03-10 23:03:08,883] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:08<00:00, 338.71it/s]
[2024-03-10 23:03:17,740] [jmetal.core.algorithm] [DEBUG] Finished!
```

```
binary solution:  1111
fitness:  0.8
```

```python
print(" La moyenne des fitnesses est :", fitnesses.mean())
print("la moyenne des rappels est :", rappels.mean())
print("la moyenne des f1 est :", f1s.mean())
print("la moyenne  des précisions est :", precisions.mean())

print(" L'écart type des fitnesses est :", fitnesses.std(axis=0))
print(" Le meilleur fitness est :", fitnesses.max(axis=0))
print(" Le pire fitness est :", fitnesses.min(axis=0))


print(" La moyenne des temps (en seconde) d'exécution est :",
      running_times.mean(axis=0))
print(" L'écart type des temps d'exécution (en seconde) est :",
      running_times.std(axis=0))
```

```
 La moyenne des fitnesses est : 0.8266666666666665
la moyenne des rappels est : 0.8266666666666665
la moyenne des f1 est : 0.8285270507087759
la moyenne  des précisions est : 0.8266666666666665
 L'écart type des fitnesses est : 0.025228720542113173
 Le meilleur fitness est : 0.8444444444444444
 Le pire fitness est : 0.7777777777777778
 La moyenne des temps (en seconde) d'exécution est : 8.867935633659362
 L'écart type des temps d'exécution (en seconde) est : 0.1024537028975631
```

```python
def select_features(dataset, selected_features):
    data = dataset.data
    df = pd.DataFrame(data, columns=dataset.feature_names)

    selected_columns = df.iloc[:, [i for i, selected in enumerate(selected_features) if sele

    return selected_columns
```

```python
print (select_features(load_iris(), [1, 0, 1, 0]))
```

```
     sepal length (cm)  petal length (cm)
0                  5.1                1.4
1                  4.9                1.4
2                  4.7                1.3
3                  4.6                1.5
4                  5.0                1.4
..                 ...                ...
```

```
145                6.7                5.2
146                6.3                5.0
147                6.5                5.2
148                6.2                5.4
149                5.9                5.1

[150 rows x 2 columns]
```

on remarque que les attributs séléctionnés pour le meilleur fitness sont sepal length et petal length et le temps moyen d'entrainement du modèle svm est 8.87 sec

on va lancer le modèle svm sur load_iris sans sélection d'attributs avec les paramètres optimaux

```
dataset = load_iris()
classification_algorithm = 'SVM'
select_features = False
SAProblem_params_optimal_ag = {
         "population_size" : 30
        ,"offspring_population_size" : 30
        ,"mutation_rate" : 0.2
        ,"crossover_rate" : 1
        ,"dataset" : dataset
        ,"classification_algorithm" : classification_algorithm
        ,"select_features" : select_features
        , "max_evaluations": 3000
    }

fitnesses, running_times, rappels, f1s, precisions = train(problem_function=run_genetic_algo
                                 problem_function_params=SAProblem_params_optimal_ag, nb_run
```

```
[2024-03-10 23:08:29,473] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:08:29,473] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:08:29,587] [jmetal.core.algorithm] [DEBUG] Initializing progress...
[2024-03-10 23:08:29,588] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:08<00:00, 334.87it/s]
[2024-03-10 23:08:38,548] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:08:38,548] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:08:38,548] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:08:38,641] [jmetal.core.algorithm] [DEBUG] Initializing progress...
```

```
binary solution:  1111
fitness:  0.8
```

```
[2024-03-10 23:08:38,642] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:09<00:00, 329.13it/s]
[2024-03-10 23:08:47,757] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:08:47,758] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
```

```
[2024-03-10 23:08:47,758] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:08:47,851] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1111
fitness:  0.8444444444444444

[2024-03-10 23:08:47,852] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:08<00:00, 339.13it/s]
[2024-03-10 23:08:56,699] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:08:56,699] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:08:56,699] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:08:56,792] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1111
fitness:  0.8

[2024-03-10 23:08:56,794] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:08<00:00, 334.56it/s]
[2024-03-10 23:09:05,761] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:09:05,762] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:09:05,762] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:09:05,852] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1111
fitness:  0.8222222222222222

[2024-03-10 23:09:05,854] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:08<00:00, 333.46it/s]
[2024-03-10 23:09:14,851] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:09:14,851] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:09:14,851] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:09:14,940] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1110
fitness:  0.7333333333333333

[2024-03-10 23:09:14,941] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:09<00:00, 327.54it/s]
[2024-03-10 23:09:24,101] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:09:24,101] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:09:24,101] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:09:24,188] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1010
fitness:  0.8222222222222222

[2024-03-10 23:09:24,190] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:08<00:00, 342.77it/s]
[2024-03-10 23:09:32,942] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:09:32,943] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:09:32,943] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
```

```
[2024-03-10 23:09:33,030] [jmetal.core.algorithm] [DEBUG] Initializing progress...
```

```
binary solution:  0011
fitness:  0.8444444444444444
```

```
[2024-03-10 23:09:33,031] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:09<00:00, 328.62it/s]
[2024-03-10 23:09:42,161] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:09:42,161] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:09:42,161] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:09:42,250] [jmetal.core.algorithm] [DEBUG] Initializing progress...
```

```
binary solution:  0111
fitness:  0.8222222222222222
```

```
[2024-03-10 23:09:42,251] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:08<00:00, 334.60it/s]
[2024-03-10 23:09:51,217] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:09:51,218] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:09:51,218] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:09:51,305] [jmetal.core.algorithm] [DEBUG] Initializing progress...
```

```
binary solution:  1111
fitness:  0.8222222222222222
```

```
[2024-03-10 23:09:51,307] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress: 100%|##########| 3000/3000 [00:09<00:00, 332.61it/s]
[2024-03-10 23:10:00,326] [jmetal.core.algorithm] [DEBUG] Finished!
```

```
binary solution:  1110
fitness:  0.8444444444444444
```

```python
print(" La moyenne des fitnesses est :", fitnesses.mean())
print("la moyenne des rappels est :", rappels.mean())
print("la moyenne des f1 est :", f1s.mean())
print("la moyenne  des précisions est :", precisions.mean())

print(" L'écart type des fitnesses est :", fitnesses.std(axis=0))
print(" Le meilleur fitness est :", fitnesses.max(axis=0))
print(" Le pire fitness est :", fitnesses.min(axis=0))


print(" La moyenne des temps (en seconde) d'exécution est :",
      running_times.mean(axis=0))
print(" L'écart type des temps d'exécution (en seconde) est :",
      running_times.std(axis=0))
```

```
 La moyenne des fitnesses est : 0.8155555555555555
la moyenne des rappels est : 0.8155555555555555
la moyenne des f1 est : 0.8193446311619473
la moyenne  des précisions est : 0.8155555555555555
 L'écart type des fitnesses est : 0.03320964706882948
 Le meilleur fitness est : 0.8444444444444444
 Le pire fitness est : 0.7333333333333333
 La moyenne des temps (en seconde) d'exécution est : 9.085421323776245
 L'écart type des temps d'exécution (en seconde) est : 0.1258007575544716
```

on remarque que sans sélectionner les attributs, le meilleur fitness est 0.844 le temps moyen d'entrainement du modèle svm est 9.085 sec

on va lancer le modèle knn sur load_iris sans sélection d'attributs avec les paramètres optimaux

```
dataset = load_iris()
classification_algorithm = 'KNN'
select_features = False
SAProblem_params_optimal_ag = {
         "population_size" : 30
        ,"offspring_population_size" : 30
        ,"mutation_rate" : 0.2
        ,"crossover_rate" : 1
        ,"dataset" : dataset
        ,"classification_algorithm" : classification_algorithm
        ,"select_features" : select_features
        , "max_evaluations": 3000
    }

fitnesses, running_times, rappels, f1s, precisions = train(problem_function=run_genetic_algo
                                 problem_function_params=SAProblem_params_optimal_ag, nb_run
```

```
[2024-03-10 23:12:40,417] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:12:40,418] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:12:40,616] [jmetal.core.algorithm] [DEBUG] Initializing progress...
[2024-03-10 23:12:40,618] [jmetal.core.algorithm] [DEBUG] Running main loop until terminatio
Progress:  83%|########2 | 4140/5000 [11:42<02:25,  5.89it/s]
Progress: 100%|##########| 3000/3000 [00:16<00:00, 178.95it/s]
[2024-03-10 23:12:57,383] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:12:57,383] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:12:57,383] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:12:57,550] [jmetal.core.algorithm] [DEBUG] Initializing progress...
```

```
binary solution:  1101
fitness:  0.8222222222222222
```

```
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:12:57,551] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:16<00:00, 178.68it/s]
[2024-03-10 23:13:14,341] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:13:14,341] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:13:14,342] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
```

```
[2024-03-10 23:13:14,510] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1111
fitness:  0.8666666666666667

Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:13:14,511] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:16<00:00, 177.69it/s]
[2024-03-10 23:13:31,395] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:13:31,395] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:13:31,395] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:13:31,564] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1111
fitness:  0.8444444444444444

Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:13:31,565] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:16<00:00, 179.81it/s]
[2024-03-10 23:13:48,249] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:13:48,250] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:13:48,250] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:13:48,413] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1111
fitness:  0.8666666666666667

Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:13:48,414] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:16<00:00, 178.75it/s]
[2024-03-10 23:14:05,198] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:14:05,198] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:14:05,198] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:14:05,391] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1110
fitness:  0.8444444444444444

Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:14:05,392] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:17<00:00, 172.33it/s]
[2024-03-10 23:14:22,801] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:14:22,801] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:14:22,802] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:14:22,991] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1111
fitness:  0.8666666666666667

Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:14:22,992] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:16<00:00, 179.37it/s]
[2024-03-10 23:14:39,718] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:14:39,718] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:14:39,719] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:14:39,884] [jmetal.core.algorithm] [DEBUG] Initializing progress...
```

```
binary solution:  0111
fitness:  0.8444444444444444
```

```
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:14:39,885] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:16<00:00, 177.60it/s]
[2024-03-10 23:14:56,778] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:14:56,778] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:14:56,778] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:14:56,950] [jmetal.core.algorithm] [DEBUG] Initializing progress...
```

```
binary solution:  1111
fitness:  0.8666666666666667
```

```
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:14:56,952] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:16<00:00, 178.87it/s]
[2024-03-10 23:15:13,724] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:15:13,724] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:15:13,724] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:15:13,895] [jmetal.core.algorithm] [DEBUG] Initializing progress...
```

```
binary solution:  1110
fitness:  0.8444444444444444
```

```
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:15:13,896] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:16<00:00, 181.27it/s]
[2024-03-10 23:15:30,446] [jmetal.core.algorithm] [DEBUG] Finished!
```

```
binary solution:  1111
fitness:  0.8444444444444444
```

```python
print(" La moyenne des fitnesses est :", fitnesses.mean())
print("la moyenne des rappels est :", rappels.mean())
print("la moyenne des f1 est :", f1s.mean())
print("la moyenne  des précisions est :", precisions.mean())

print(" L'écart type des fitnesses est :", fitnesses.std(axis=0))
print(" Le meilleur fitness est :", fitnesses.max(axis=0))
print(" Le pire fitness est :", fitnesses.min(axis=0))


print(" La moyenne des temps (en seconde) d'exécution est :",
      running_times.mean(axis=0))
print(" L'écart type des temps d'exécution (en seconde) est :",
      running_times.std(axis=0))
```

```
 La moyenne des fitnesses est : 0.8511111111111112
la moyenne des rappels est : 0.8511111111111112
la moyenne des f1 est : 0.8525917251929375
la moyenne  des précisions est : 0.8511111111111112
 L'écart type des fitnesses est : 0.014998856838012308
 Le meilleur fitness est : 0.8666666666666667
 Le pire fitness est : 0.8222222222222222
 La moyenne des temps (en seconde) d'exécution est : 17.002938914299012
 L'écart type des temps d'exécution (en seconde) est : 0.23182295272913248
```

on remarque que sans sélectionner les attributs, le meilleur fitness est 0.866 le temps moyen d'entrainement du modèle knn est 17 sec

on va lancer le modèle knn sur load_iris avec sélection d'attributs avec les paramètres optimaux

```
dataset = load_iris()
classification_algorithm = 'KNN'
select_features = True
SAProblem_params_optimal_ag = {
         "population_size" : 30
        ,"offspring_population_size" : 30
        ,"mutation_rate" : 0.2
        ,"crossover_rate" : 1
        ,"dataset" : dataset
        , "select_features" : select_features
        ,"classification_algorithm" : classification_algorithm
        , "max_evaluations": 3000
    }

fitnesses, running_times, rappels, f1s, precisions = train(problem_function=run_genetic_algo
                                  problem_function_params=SAProblem_params_optimal_ag, nb_run
```

```
[2024-03-10 23:17:36,375] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:17:36,376] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:17:36,560] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:17:36,561] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:16<00:00, 176.84it/s]
[2024-03-10 23:17:53,526] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:17:53,527] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:17:53,527] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:17:53,712] [jmetal.core.algorithm] [DEBUG] Initializing progress...
```

```
binary solution:  1111
fitness:  0.8444444444444444
```

```
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:17:53,713] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:17<00:00, 175.72it/s]
[2024-03-10 23:18:10,786] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:18:10,786] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:18:10,787] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:18:10,973] [jmetal.core.algorithm] [DEBUG] Initializing progress...
```

```
binary solution:  1101
```

```
fitness:  0.8666666666666667

Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:18:10,974] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:16<00:00, 178.92it/s]
[2024-03-10 23:18:27,742] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:18:27,742] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:18:27,742] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:18:27,923] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  0001
fitness:  0.8222222222222222

Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:18:27,924] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:16<00:00, 177.30it/s]
[2024-03-10 23:18:44,845] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:18:44,845] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:18:44,846] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:18:45,008] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1110
fitness:  0.8666666666666667

Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:18:45,009] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:16<00:00, 177.95it/s]
[2024-03-10 23:19:01,868] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:19:01,869] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:19:01,869] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:19:02,031] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  0101
fitness:  0.8444444444444444

Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:19:02,032] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:16<00:00, 177.81it/s]
[2024-03-10 23:19:18,904] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:19:18,905] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:19:18,905] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:19:19,085] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1011
fitness:  0.8666666666666667

Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:19:19,086] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:16<00:00, 180.16it/s]
[2024-03-10 23:19:35,738] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:19:35,738] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:19:35,738] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:19:35,906] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  1001
fitness:  0.8444444444444444
```

```
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:19:35,907] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:17<00:00, 174.82it/s]
[2024-03-10 23:19:53,067] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:19:53,068] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:19:53,068] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:19:53,246] [jmetal.core.algorithm] [DEBUG] Initializing progress...
```

```
binary solution:  1100
fitness:  0.8444444444444444
```

```
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:19:53,247] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:16<00:00, 179.05it/s]
[2024-03-10 23:20:10,003] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:20:10,003] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:20:10,004] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:20:10,171] [jmetal.core.algorithm] [DEBUG] Initializing progress...
```

```
binary solution:  0100
fitness:  0.8444444444444444
```

```
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:20:10,172] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:16<00:00, 178.55it/s]
[2024-03-10 23:20:26,975] [jmetal.core.algorithm] [DEBUG] Finished!
```

```
binary solution:  1101
fitness:  0.8444444444444444
```

```python
print(" La moyenne des fitnesses est :", fitnesses.mean())
print("la moyenne des rappels est :", rappels.mean())
print("la moyenne des f1 est :", f1s.mean())
print("la moyenne  des précisions est :", precisions.mean())

print(" L'écart type des fitnesses est :", fitnesses.std(axis=0))
print(" Le meilleur fitness est :", fitnesses.max(axis=0))
print(" Le pire fitness est :", fitnesses.min(axis=0))


print(" La moyenne des temps (en seconde) d'exécution est :",
      running_times.mean(axis=0))
print(" L'écart type des temps d'exécution (en seconde) est :",
      running_times.std(axis=0))
```

```
 La moyenne des fitnesses est : 0.8488888888888889
la moyenne des rappels est : 0.8488888888888889
la moyenne des f1 est : 0.8526642161409708
la moyenne  des précisions est : 0.8488888888888889
 L'écart type des fitnesses est : 0.01405456737852615
 Le meilleur fitness est : 0.8666666666666667
 Le pire fitness est : 0.8222222222222222
 La moyenne des temps (en seconde) d'exécution est : 17.060037302970887
 L'écart type des temps d'exécution (en seconde) est : 0.15276687494858962
```

```
print(select_features(load_iris(), [1, 1, 1, 0]))
```

```
     sepal length (cm)  sepal width (cm)  petal length (cm)
0                  5.1               3.5                1.4
1                  4.9               3.0                1.4
2                  4.7               3.2                1.3
3                  4.6               3.1                1.5
4                  5.0               3.6                1.4
..                 ...               ...                ...
145                6.7               3.0                5.2
146                6.3               2.5                5.0
147                6.5               3.0                5.2
148                6.2               3.4                5.4
149                5.9               3.0                5.1

[150 rows x 3 columns]
```

on remarque que les attributs séléctionnés pour le meilleur fitness sont sepal length, sepal width et petal length et le temps moyen d'entrainement du modèle svm est 17 sec

on va lancer le modèle knn sur load_breast_cancer avec sélection d'attributs

```
dataset = load_breast_cancer()
classification_algorithm = 'KNN'
select_features = True
SAProblem_params_optimal_ag = {
         "population_size" : 30
        ,"offspring_population_size" : 30
        ,"mutation_rate" : 0.2
        ,"crossover_rate" : 1
        ,"dataset" : dataset
        , "select_features" : select_features
        ,"classification_algorithm" : classification_algorithm
        , "max_evaluations": 3000
    }

fitnesses, running_times, rappels, f1s, precisions = train(problem_function=run_genetic_algo
                                 problem_function_params=SAProblem_params_optimal_ag, nb_run
```

```
[2024-03-10 23:29:42,594] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:29:42,595] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

[2024-03-10 23:29:42,822] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:29:42,823] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:18<00:00, 160.92it/s]
```

```
[2024-03-10 23:30:01,467] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:30:01,467] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:30:01,468] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:30:01,654] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  101001110011101100111000000111
fitness:  0.8362573099415205

Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:30:01,655] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:20<00:00, 147.82it/s]
[2024-03-10 23:30:21,950] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:30:21,951] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:30:21,952] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  100010110110111101111010101110
fitness:  0.847953216374269

[2024-03-10 23:30:22,179] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:30:22,180] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:20<00:00, 144.75it/s]
[2024-03-10 23:30:42,906] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:30:42,907] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:30:42,907] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  110111101010001101111111100001
fitness:  0.8304093567251462

[2024-03-10 23:30:43,107] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:30:43,108] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:20<00:00, 144.40it/s]
[2024-03-10 23:31:03,884] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:31:03,884] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:31:03,885] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  101000101101011111100000101010
fitness:  0.847953216374269

[2024-03-10 23:31:04,093] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:31:04,094] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:21<00:00, 140.89it/s]
[2024-03-10 23:31:25,388] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:31:25,388] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:31:25,389] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  000000011101010111010110110001
fitness:  0.8421052631578947

[2024-03-10 23:31:25,603] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:31:25,605] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:22<00:00, 135.28it/s]
[2024-03-10 23:31:47,781] [jmetal.core.algorithm] [DEBUG] Finished!
```

```
[2024-03-10 23:31:47,782] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:31:47,784] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  101100001111110111111001100111
fitness:  0.8362573099415205

[2024-03-10 23:31:48,004] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:31:48,006] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:21<00:00, 139.41it/s]
[2024-03-10 23:32:09,525] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:32:09,526] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:32:09,527] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  010000011001011111001100101001
fitness:  0.8362573099415205

[2024-03-10 23:32:09,809] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:32:09,835] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:21<00:00, 136.47it/s]
[2024-03-10 23:32:31,819] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:32:31,819] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:32:31,820] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  000001011101011110011011001011
fitness:  0.8362573099415205

[2024-03-10 23:32:32,020] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:32:32,021] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:21<00:00, 139.75it/s]
[2024-03-10 23:32:53,489] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:32:53,489] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:32:53,490] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  001010100010011101110001111011
fitness:  0.8538011695906432

[2024-03-10 23:32:53,720] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:32:53,721] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:20<00:00, 147.57it/s]
[2024-03-10 23:33:14,051] [jmetal.core.algorithm] [DEBUG] Finished!

binary solution:  110011111101101011010100101101
fitness:  0.8421052631578947
```

```python
print(" La moyenne des fitnesses est :", fitnesses.mean())
print("la moyenne des rappels est :", rappels.mean())
print("la moyenne des f1 est :", f1s.mean())
print("la moyenne  des précisions est :", precisions.mean())

print(" L'écart type des fitnesses est :", fitnesses.std(axis=0))
print(" Le meilleur fitness est :", fitnesses.max(axis=0))
print(" Le pire fitness est :", fitnesses.min(axis=0))


print(" La moyenne des temps (en seconde) d'exécution est :",
      running_times.mean(axis=0))
print(" L'écart type des temps d'exécution (en seconde) est :",
      running_times.std(axis=0))
```

```
 La moyenne des fitnesses est : 0.84093567251462
la moyenne des rappels est : 0.84093567251462
la moyenne des f1 est : 0.8343538916856341
la moyenne  des précisions est : 0.84093567251462
 L'écart type des fitnesses est : 0.007188728621670863
 Le meilleur fitness est : 0.8538011695906432
 Le pire fitness est : 0.8304093567251462
 La moyenne des temps (en seconde) d'exécution est : 21.145801615715026
 L'écart type des temps d'exécution (en seconde) est : 1.0350304929504979
```

```
print(select_features(load_breast_cancer(), [0,0,1,0,1,0,1,0,0,0,1,0,0,1,1,1,0,1,1,1,0,0,0,1
```

```
     mean perimeter  mean smoothness  mean concavity  radius error  \
0            122.80          0.11840         0.30010        1.0950
1            132.90          0.08474         0.08690        0.5435
2            130.00          0.10960         0.19740        0.7456
3             77.58          0.14250         0.24140        0.4956
4            135.10          0.10030         0.19800        0.7572
..              ...              ...             ...           ...
564          142.00          0.11100         0.24390        1.1760
565          131.20          0.09780         0.14400        0.7655
566          108.30          0.08455         0.09251        0.4564
567          140.10          0.11780         0.35140        0.7260
568           47.92          0.05263         0.00000        0.3857

     area error  smoothness error  compactness error  concave points error  \
0        153.40          0.006399            0.04904               0.01587
1         74.08          0.005225            0.01308               0.01340
2         94.03          0.006150            0.04006               0.02058
3         27.23          0.009110            0.07458               0.01867
4         94.44          0.011490            0.02461               0.01885
..          ...               ...                ...                   ...
564      158.70          0.010300            0.02891               0.02454
565       99.04          0.005769            0.02423               0.01678
566       48.55          0.005903            0.03731               0.01557
567       86.22          0.006522            0.06158               0.01664
568       19.15          0.007189            0.00466               0.00000

     symmetry error  fractal dimension error  worst area  worst smoothness  \
0           0.03003                 0.006193      2019.0           0.16220
1           0.01389                 0.003532      1956.0           0.12380
2           0.02250                 0.004571      1709.0           0.14440
3           0.05963                 0.009208       567.7           0.20980
4           0.01756                 0.005115      1575.0           0.13740
..              ...                      ...         ...               ...
564         0.01114                 0.004239      2027.0           0.14100
565         0.01898                 0.002498      1731.0           0.11660
566         0.01318                 0.003892      1124.0           0.11390
567         0.02324                 0.006185      1821.0           0.16500
568         0.02676                 0.002783       268.6           0.08996

     worst compactness  worst concavity  worst symmetry  \
0              0.66560           0.7119          0.4601
1              0.18660           0.2416          0.2750
2              0.42450           0.4504          0.3613
3              0.86630           0.6869          0.6638
4              0.20500           0.4000          0.2364
..                 ...              ...             ...
564            0.21130           0.4107          0.2060
565            0.19220           0.3215          0.2572
566            0.30940           0.3403          0.2218
567            0.86810           0.9387          0.4087
568            0.06444           0.0000          0.2871

     worst fractal dimension
0                    0.11890
1                    0.08902
2                    0.08758
3                    0.17300
4                    0.07678
..                       ...
564                  0.07115
565                  0.06637
566                  0.07820
567                  0.12400
568                  0.07039

[569 rows x 16 columns]
```

on remarque que les attributs séléctionnés pour le meilleur fitness sont mean perimeter mean smoothness mean concavity radius error area error smoothness error compactness error concave points error symmetry error fractal dimension error worst area worst smoothness worst compactness worst concavity worst symmetry worst fractal dimension le temps moyen d'entrainement du modèle knn est 21 sec on va lancer le modèle knn sur load_breast_cancer sans sélection d'attributs avec les paramètres optimaux

```
dataset = load_breast_cancer()
classification_algorithm = 'KNN'
select_features = False
SAProblem_params_optimal_ag = {
         "population_size" : 30
        ,"offspring_population_size" : 30
        ,"mutation_rate" : 0.2
        ,"crossover_rate" : 1
        ,"dataset" : dataset
        , "select_features" : select_features
        ,"classification_algorithm" : classification_algorithm
        , "max_evaluations": 3000
    }

fitnesses, running_times, rappels, f1s, precisions = train(problem_function=run_genetic_algo
                                 problem_function_params=SAProblem_params_optimal_ag, nb_run
```

```
[2024-03-10 23:37:38,872] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:37:38,873] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:37:39,065] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:37:39,066] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:18<00:00, 158.51it/s]
[2024-03-10 23:37:57,993] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:37:57,993] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:37:57,994] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
```

```
binary solution:  111111111111101111111111111111
fitness:  0.8304093567251462
```

```
[2024-03-10 23:37:58,195] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:37:58,196] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:18<00:00, 165.67it/s]
[2024-03-10 23:38:16,305] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:38:16,305] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:38:16,306] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:38:16,476] [jmetal.core.algorithm] [DEBUG] Initializing progress...
```

```
binary solution:  111111101111111011111111111111
fitness:  0.8421052631578947
```

```
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:38:16,478] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:18<00:00, 163.07it/s]
[2024-03-10 23:38:34,875] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:38:34,875] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:38:34,875] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
```

```
binary solution:  111111111111111111111111111111
fitness:  0.8421052631578947

[2024-03-10 23:38:35,079] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:38:35,080] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:20<00:00, 148.08it/s]
[2024-03-10 23:38:55,340] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:38:55,340] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:38:55,341] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  111111111111101111111111111111
fitness:  0.8421052631578947

[2024-03-10 23:38:55,544] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:38:55,546] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:21<00:00, 141.02it/s]
[2024-03-10 23:39:16,820] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:39:16,821] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:39:16,821] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:39:17,017] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  111111111111111011111111111111
fitness:  0.8421052631578947

Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:39:17,020] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:20<00:00, 148.91it/s]
[2024-03-10 23:39:37,167] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:39:37,167] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:39:37,168] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  111101111111111111111111111111
fitness:  0.847953216374269

[2024-03-10 23:39:37,437] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:39:37,439] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:20<00:00, 144.88it/s]
[2024-03-10 23:39:58,146] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:39:58,146] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:39:58,147] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:39:58,338] [jmetal.core.algorithm] [DEBUG] Initializing progress...

binary solution:  111111111111111111111110110111
fitness:  0.8421052631578947

Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:39:58,340] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:20<00:00, 144.44it/s]
[2024-03-10 23:40:19,111] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:40:19,111] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:40:19,112] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  111111111111111111111111111111
```

```
fitness:  0.8245614035087719
```

```
[2024-03-10 23:40:19,312] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:40:19,315] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:20<00:00, 147.28it/s]
[2024-03-10 23:40:39,685] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:40:39,685] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:40:39,685] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:40:39,883] [jmetal.core.algorithm] [DEBUG] Initializing progress...
```

```
binary solution:  111111111111111111111111101111
fitness:  0.8362573099415205
```

```
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:40:39,885] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:20<00:00, 145.71it/s]
[2024-03-10 23:41:00,474] [jmetal.core.algorithm] [DEBUG] Finished!
```

```
binary solution:  111111111111111111111111111111
fitness:  0.8362573099415205
```

```
print(" La moyenne des fitnesses est :", fitnesses.mean())
print("la moyenne des rappels est :", rappels.mean())
print("la moyenne des f1 est :", f1s.mean())
print("la moyenne  des précisions est :", precisions.mean())

print(" L'écart type des fitnesses est :", fitnesses.std(axis=0))
print(" Le meilleur fitness est :", fitnesses.max(axis=0))
print(" Le pire fitness est :", fitnesses.min(axis=0))


print(" La moyenne des temps (en seconde) d'exécution est :",
      running_times.mean(axis=0))
print(" L'écart type des temps d'exécution (en seconde) est :",
      running_times.std(axis=0))
```

```
 La moyenne des fitnesses est : 0.8385964912280702
la moyenne des rappels est : 0.8385964912280702
la moyenne des f1 est : 0.8326240973781092
la moyenne  des précisions est : 0.8385964912280702
 L'écart type des fitnesses est : 0.006864256086416758
 Le meilleur fitness est : 0.847953216374269
 Le pire fitness est : 0.8245614035087719
 La moyenne des temps (en seconde) d'exécution est : 20.16024582386017
 L'écart type des temps d'exécution (en seconde) est : 1.0943733832844424
```

on remarque que sans sélectionner les attributs, le meilleur fitness est 0.84 le temps moyen d'entrainement du modèle knn sur load_breast_cancer est 20 sec

on va lancer le modèle svm sur load_breast_cancer sans sélection d'attributs avec les paramètres optimaux

```
dataset = load_breast_cancer()
classification_algorithm = 'SVM'
select_features = False
SAProblem_params_optimal_ag = {
         "population_size" : 30
        ,"offspring_population_size" : 30
        ,"mutation_rate" : 0.2
        ,"crossover_rate" : 1
        ,"dataset" : dataset
        , "select_features" : select_features
        ,"classification_algorithm" : classification_algorithm
        , "max_evaluations": 3000
    }

fitnesses, running_times, rappels, f1s , precisions = train(problem_function=run_genetic_alg
                                 problem_function_params=SAProblem_params_optimal_ag, nb_run
```

```
[2024-03-10 23:41:43,795] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:41:43,796] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:41:44,074] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:41:44,075] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:26<00:00, 114.29it/s]
[2024-03-10 23:42:10,323] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:42:10,323] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:42:10,324] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  111111111111011111111111111111
fitness:  0.8362573099415205

[2024-03-10 23:42:10,592] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:42:10,593] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:25<00:00, 115.96it/s]
[2024-03-10 23:42:36,463] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:42:36,464] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:42:36,464] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  111111111111011111111111111111
fitness:  0.8421052631578947

[2024-03-10 23:42:36,722] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:42:36,723] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:25<00:00, 116.44it/s]
[2024-03-10 23:43:02,488] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:43:02,488] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:43:02,488] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  111111111111111111111111011101
fitness:  0.8245614035087719
```

```
[2024-03-10 23:43:02,753] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:43:02,754] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:26<00:00, 112.80it/s]
[2024-03-10 23:43:29,349] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:43:29,349] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:43:29,350] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  110111111111111111111111111111
fitness:  0.8362573099415205

[2024-03-10 23:43:29,610] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:43:29,611] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:26<00:00, 115.23it/s]
[2024-03-10 23:43:55,645] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:43:55,646] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:43:55,646] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  111111111111111111111111101111
fitness:  0.847953216374269

[2024-03-10 23:43:55,901] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:43:55,902] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:26<00:00, 114.53it/s]
[2024-03-10 23:44:22,097] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:44:22,097] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:44:22,097] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  111111111111111111111101111111
fitness:  0.8362573099415205

[2024-03-10 23:44:22,349] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:44:22,350] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:25<00:00, 116.02it/s]
[2024-03-10 23:44:48,208] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:44:48,209] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:44:48,209] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  111111111111111111111111111111
fitness:  0.8362573099415205

[2024-03-10 23:44:48,469] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:44:48,470] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:25<00:00, 116.94it/s]
[2024-03-10 23:45:14,124] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:45:14,124] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:45:14,124] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  111111111111111111111111110111
fitness:  0.8362573099415205

[2024-03-10 23:45:14,380] [jmetal.core.algorithm] [DEBUG] Initializing progress...
```

```
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:45:14,381] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:26<00:00, 111.91it/s]
[2024-03-10 23:45:41,188] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:45:41,189] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:45:41,189] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
```

```
binary solution:  111111110111111111111111111111
fitness:  0.8304093567251462
```

```
[2024-03-10 23:45:41,448] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:45:41,449] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:26<00:00, 114.81it/s]
[2024-03-10 23:46:07,578] [jmetal.core.algorithm] [DEBUG] Finished!
```

```
binary solution:  111111111111111111111111111011
fitness:  0.8362573099415205
```

```
print(" La moyenne des fitnesses est :", fitnesses.mean())
print("la moyenne des rappels est :", rappels.mean())
print("la moyenne des f1 est :", f1s.mean())
print("la moyenne  des précisions est :", precisions.mean())

print(" L'écart type des fitnesses est :", fitnesses.std(axis=0))
print(" Le meilleur fitness est :", fitnesses.max(axis=0))
print(" Le pire fitness est :", fitnesses.min(axis=0))


print(" La moyenne des temps (en seconde) d'exécution est :",
      running_times.mean(axis=0))
print(" L'écart type des temps d'exécution (en seconde) est :",
      running_times.std(axis=0))
```

```
 La moyenne des fitnesses est : 0.8362573099415205
la moyenne des rappels est : 0.8362573099415205
la moyenne des f1 est : 0.8286259313251927
la moyenne  des précisions est : 0.8362573099415205
 L'écart type des fitnesses est : 0.0061642839379500575
 Le meilleur fitness est : 0.847953216374269
 Le pire fitness est : 0.8245614035087719
 La moyenne des temps (en seconde) d'exécution est : 26.378353571891786
 L'écart type des temps d'exécution (en seconde) est : 0.3662227503913703
```

on remarque que sans sélectionner les attributs, le meilleur fitness est 0.8479 le temps moyen d'entrainement du modèle svm est 26 sec

On va lancer le modèle svm sur load_breast_cancer avec sélection d'attributs avec les paramètres optimaux

```
dataset = load_breast_cancer()
classification_algorithm = 'SVM'
```

```
select_features = True
SAProblem_params_optimal_ag = {
         "population_size" : 30
        ,"offspring_population_size" : 30
        ,"mutation_rate" : 0.2
        ,"crossover_rate" : 1
        ,"dataset" : dataset
        , "select_features" : select_features
        ,"classification_algorithm" : classification_algorithm
        , "max_evaluations": 3000
    }

fitnesses, running_times, rappels, f1s, precisions = train(problem_function=run_genetic_algo
                                    problem_function_params=SAProblem_params_optimal_ag, nb_run
```

```
[2024-03-10 23:46:58,050] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:46:58,051] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
[2024-03-10 23:46:58,341] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:46:58,342] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:26<00:00, 112.81it/s]
[2024-03-10 23:47:24,935] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:47:24,936] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:47:24,936] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
```

```
binary solution:  111111000100000000011101000010
fitness:  0.8304093567251462
```

```
[2024-03-10 23:47:25,189] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:47:25,190] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:26<00:00, 114.27it/s]
[2024-03-10 23:47:51,444] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:47:51,444] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:47:51,445] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
```

```
binary solution:  001100101001110111011011000010
fitness:  0.8421052631578947
```

```
[2024-03-10 23:47:51,703] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:47:51,704] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:25<00:00, 115.94it/s]
[2024-03-10 23:48:17,579] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:48:17,579] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:48:17,580] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
```

```
binary solution:  111010010101111101110100110001
fitness:  0.8304093567251462
```

```
[2024-03-10 23:48:17,843] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:48:17,844] [jmetal.core.alg
```

```
Progress: 100%|##########| 3000/3000 [00:26<00:00, 112.78it/s]
[2024-03-10 23:48:44,444] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:48:44,444] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:48:44,445] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  101111011100110010100000100101
fitness:  0.8245614035087719

[2024-03-10 23:48:44,701] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:48:44,702] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:25<00:00, 116.22it/s]
[2024-03-10 23:49:10,515] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:49:10,515] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:49:10,516] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  101111101000100100111111110101
fitness:  0.8362573099415205

[2024-03-10 23:49:10,770] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:49:10,771] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:25<00:00, 116.84it/s]
[2024-03-10 23:49:36,447] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:49:36,448] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:49:36,448] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  101110101000100011001010110101
fitness:  0.8421052631578947

[2024-03-10 23:49:36,699] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:49:36,700] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:25<00:00, 116.41it/s]
[2024-03-10 23:50:02,471] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:50:02,471] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:50:02,472] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  110011100111111110001011111000
fitness:  0.8421052631578947

[2024-03-10 23:50:02,737] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:50:02,738] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:25<00:00, 117.79it/s]
[2024-03-10 23:50:28,207] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:50:28,208] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions
[2024-03-10 23:50:28,208] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...

binary solution:  101010110001000001101001110010
fitness:  0.8304093567251462

[2024-03-10 23:50:28,471] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:50:28,472] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:25<00:00, 115.83it/s]
```

```
[2024-03-10 23:50:54,373] [jmetal.core.algorithm] [DEBUG] Finished!
[2024-03-10 23:50:54,374] [jmetal.core.algorithm] [DEBUG] Creating initial set of solutions.
[2024-03-10 23:50:54,374] [jmetal.core.algorithm] [DEBUG] Evaluating solutions...
```

```
binary solution:  110011010011100011011011101000
fitness:  0.8421052631578947
```

```
[2024-03-10 23:50:54,638] [jmetal.core.algorithm] [DEBUG] Initializing progress...
Progress:   0%|          | 0/3000 [00:00<?, ?it/s][2024-03-10 23:50:54,639] [jmetal.core.alg
Progress: 100%|##########| 3000/3000 [00:26<00:00, 114.79it/s]
[2024-03-10 23:51:20,773] [jmetal.core.algorithm] [DEBUG] Finished!
```

```
binary solution:  011010001001001011001111011110
fitness:  0.847953216374269
```

```
print(" La moyenne des fitnesses est :", fitnesses.mean())
print("la moyenne des rappels est :", rappels.mean())
print("la moyenne des f1 est :", f1s.mean())
print("la moyenne  des précisions est :", precisions.mean())

print(" L'écart type des fitnesses est :", fitnesses.std(axis=0))
print(" Le meilleur fitness est :", fitnesses.max(axis=0))
print(" Le pire fitness est :", fitnesses.min(axis=0))


print(" La moyenne des temps (en seconde) d'exécution est :",
      running_times.mean(axis=0))
print(" L'écart type des temps d'exécution (en seconde) est :",
      running_times.std(axis=0))
```

```
 La moyenne des fitnesses est : 0.8368421052631578
la moyenne des rappels est : 0.8368421052631578
la moyenne des f1 est : 0.8297465850521755
la moyenne  des précisions est : 0.8368421052631578
 L'écart type des fitnesses est : 0.007524467471976132
 Le meilleur fitness est : 0.847953216374269
 Le pire fitness est : 0.8245614035087719
 La moyenne des temps (en seconde) d'exécution est : 26.272379779815672
 L'écart type des temps d'exécution (en seconde) est : 0.38512142773734076
```

```
print(select_features(load_breast_cancer(), [0,1,1,0,1,0,0,0,1,0,0,1,0,0,1,0,1,1,0,0,1,1,1,
```

```
    mean texture  mean perimeter  mean smoothness  mean symmetry  \
0          10.38          122.80          0.11840         0.2419
1          17.77          132.90          0.08474         0.1812
2          21.25          130.00          0.10960         0.2069
3          20.38           77.58          0.14250         0.2597
4          14.34          135.10          0.10030         0.1809
..           ...             ...              ...            ...
```

```
564         22.39         142.00          0.11100         0.1726
565         28.25         131.20          0.09780         0.1752
566         28.08         108.30          0.08455         0.1590
567         29.33         140.10          0.11780         0.2397
568         24.54          47.92          0.05263         0.1587

     texture error  smoothness error  concavity error  concave points error  \
0           0.9053          0.006399          0.05373               0.01587
1           0.7339          0.005225          0.01860               0.01340
2           0.7869          0.006150          0.03832               0.02058
3           1.1560          0.009110          0.05661               0.01867
4           0.7813          0.011490          0.05688               0.01885
..             ...               ...              ...                   ...
564         1.2560          0.010300          0.05198               0.02454
565         2.4630          0.005769          0.03950               0.01678
566         1.0750          0.005903          0.04730               0.01557
567         1.5950          0.006522          0.07117               0.01664
568         1.4280          0.007189          0.00000               0.00000

     worst radius  worst texture  worst perimeter  worst area  \
0          25.380          17.33           184.60      2019.0
1          24.990          23.41           158.80      1956.0
2          23.570          25.53           152.50      1709.0
3          14.910          26.50            98.87       567.7
4          22.540          16.67           152.20      1575.0
..            ...            ...              ...         ...
564        25.450          26.40           166.10      2027.0
565        23.690          38.25           155.00      1731.0
566        18.980          34.12           126.70      1124.0
567        25.740          39.42           184.60      1821.0
568         9.456          30.37            59.16       268.6

     worst compactness  worst concavity  worst concave points  worst symmetry
0              0.66560           0.7119                0.2654          0.4601
1              0.18660           0.2416                0.1860          0.2750
2              0.42450           0.4504                0.2430          0.3613
3              0.86630           0.6869                0.2575          0.6638
4              0.20500           0.4000                0.1625          0.2364
..                 ...              ...                   ...             ...
564            0.21130           0.4107                0.2216          0.2060
565            0.19220           0.3215                0.1628          0.2572
566            0.30940           0.3403                0.1418          0.2218
567            0.86810           0.9387                0.2650          0.4087
568            0.06444           0.0000                0.0000          0.2871

[569 rows x 16 columns]
```

Analyse des résultats

Évaluation des Résultats en termes de fouille de données :

Pour load_iris:

Sans sélection d'attributs : SVM - Précision: 0.815555, Rappel: 0.81555, F1-Mesure: 0.8193 KNN - Précision: 0.8511111, Rappel: 0.85111, F1-Mesure: 0.8525917

Avec sélection d'attributs : SVM avec sélection - Précision: 0.826667, Rappel: 0.82667, F1-Mesure: 0.8285 KNN avec sélection - Précision: 0.848888, Rappel: 0.848888, F1-Mesure: 0.8526642

Dans ce cas, la sélection d'attributs n'a pas vraiment amélioré les performances des modèles. Les performances (précision, rappel, F1-mesure) restent les mêmes que lorsque les modèles sont entraînés sans sélection d'attributs. Cependant, il est intéressant de noter que les deux modèles (SVM et KNN) ont des performances très similaires, que ce soit avec ou sans sélection d'attributs. la différence de temps entre les modèles avec et sans sélection d'attributs est négligeable dans ce cas.