A logo with a symbol and text

Description automatically generated with medium confidence A logo with blue text

Description automatically generated

Helwan University | Faculty of Computers & Artificial Intelligence AI Dept. Module: Evolutionary Algorithms [AI420] SPRING 2024

* Project Idea: Particle Swarm Optimization (PSO) and Simulated Annealing for the Traveling Salesperson Problem (TSP)
* Team Discussion Time: (15:20 PM)
* Team ID: 19

|  |  |
| --- | --- |
| **Name** | **ID** |
| **محمد احمد حنفي حسين** | **20210729** |
| **عبدالله حمدي احمد مصطفى** | **20210542** |
| **عمر عادل احمد عبداللطيف** | **20210607** |
| **محمد هيثم عبدالمنعم السيد** | **20210846** |
| **تسنيم احمد طه محمد** | **20210237** |
| **ياسمين السيد سعد احمد** | **20211045** |

# 1-Traveling Salesperson Problem:

The Traveling Salesman Problem (TSP) stands as one of the most famous and studied conundrums in the realm of computer science and optimization. At its core, the TSP seeks the most efficient route fora salesman to visit a set of cities exactly once and return to the starting point, with the objective of minimizing the total distance traveled. Despite its seemingly simple premise, the TSP is a quintessential example of an NP-hard problem, meaning its solution's complexity grows exponentially with the number of cities. This challenge has sparked intense interest across various disciplines, from mathematics to logistics and beyond. The quest to develop efficient algorithms to solve the TSP has led to innovations in optimization techniques, ranging from exact algorithms like branch and bound to heuristic approaches such as genetic algorithms and simulated annealing. Beyond its theoretical intrigue, the TSP finds practical applications in diverse fields, including transportation planning, circuit design, and even DNA sequencing, making it a cornerstone problem in the intersection of mathematics and real-world problem-solving.

# 2- Main functionalities:

a- at PSO algorithm :

\* This code iteratively improves the solution by updating the positions of particles based on their personal best and global best solutions, aiming to find the shortest route that visits each city exactly once and returns to the starting point\*

1- City Class: Represents a city with its coordinates. It also has a method to calculate the distance to another city.

2- Particle Class: Represents a particle in the PSO algorithm. Each particle has a route (sequence of cities), velocity, cost (total distance of the route), and personal best (pbest) route and cost.

3- PSO Class: Implements the PSO algorithm. It initializes a population of particles, updates their positions (routes), and velocities, and finds the best solution (global best or gbest) among all particles.

4- Random Route Generation: Generates random routes for particles.

5- Greedy Route Generation: Generates a greedy route starting from a specified city index.

6- Visualization: Visualizes the best solution found at each iteration using Matplotlib.

7- Read Cities Function: Reads city coordinates from a CSV file and creates City objects.

8- Main Execution: Reads cities from a file, initializes and runs the PSO algorithm, and prints the best solution found.

9- Shifting Route: After finding the best solution, it shifts the route so that it starts from the takeoff point (assuming the first city is the takeoff point).

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

b- simulated annealing:

\* These functionalities together form a solution to the Traveling Salesman Problem (TSP) using the simulated annealing optimization technique\*

1- Loading Dataset: The code loads a dataset of cities with their coordinates from a CSV file.

2- Distance Calculation: It defines a function to calculate the Euclidean distance between two cities using their coordinates.

3- Nearest Neighbor Initial Route: This function implements the nearest neighbor algorithm to generate an initial route for the traveling salesman problem. It starts from a random city and iteratively selects the nearest unvisited city until all cities are visited, returning to the starting city.

4- Total Distance Calculation: This function calculates the total distance of a given route, including returning to the starting city.

5- Simulated Annealing Algorithm: The code implements the simulated annealing algorithm to find an optimized route for the traveling salesman problem. It iteratively swaps two cities randomly and accepts the new route based on the Metropolis criterion. The algorithm aims to minimize the total distance traveled.

6- Plotting the Best Route: After finding the best route using simulated annealing, the code plots the path of the best route on a 2D graph.

7- Setting Initial Parameters: The code sets initial parameters such as initial temperature, cooling rate, and the number of iterations for the simulated annealing algorithm.

8- Printing Results: Finally, the code prints the best route found, its corresponding distance, and the length of the route.

# 3-Similar applications in the market:

There are many optimization areas where we can use PSO algorithm. Some of them are:

1. Energy storage optimization.
2. Scheduling electrical loads.
3. Flood control and routing.
4. Disease detection and classification.
5. Medical image segmentation.
6. Water quality monitoring.
7. Agriculture monitoring

Simulated annealing has also been applied to many combinatorial problems coming from the industry and real-world operations. To mention just a few:

* 1. Airline Crew Scheduling
  2. Railway Crew Scheduling
  3. Traveling Salesman Problem
  4. Vehicle Routing Problem
  5. Layout-Routing of Electronic Circuits
  6. Large Scale Aircraft Trajectory Planing
  7. Complex portfolio problem
  8. Graph coloring problem
  9. High-dimensionality minimization problems

# 4- A literature review of Academic publications (papers) relevant to the idea( particle Swarm Optimization (PSO) and Simulated Annealing for TSP problem)

1-

Simple summary: Traveling salesman problem (TSP) is a well-established NP-complete problem and many evolutionary techniques like particle swarm optimization (PSO) are used to optimize existing solutions for that. PSO is a method inspired by the social behavior of birds. In PSO, each member will change its position in the search space, according to personal or social experience of the whole society. In this paper, we combine the principles of PSO and crossover operator of genetic algorithm to propose a heuristic algorithm for solving the TSP more efficiently. Finally, some experimental results on our algorithm are applied in some instances in TSPLIB to demonstrate the effectiveness of our methods which also show that our algorithm can achieve better results than other approaches.

<https://www.tandfonline.com/doi/full/10.1080/23311835.2015.1048581>

2-

Simple summary:

The Probabilistic Traveling Salesman Problem (PTSP) is a variation of the well known Traveling Salesman

Problem (TSP). This problem arises when the information about customers demand is not available at the moment of the

tour generation and/or the tour re-calculating cost is too elevated. In this article, a Hybrid Algorithm combining Particle

Swarm Optimization (PSO) and Simulated Annealing (SA) is proposed, in order to solve the PTSP. The PSO heuristic

offers a simple structured algorithm which supplies a high level of exploration and fast convergence, compared with other

evolutionary algorithms. The SA algorithm is used to improve the particle diversity and to avoid the algorithm being

trapped into local optimum. Two well-known benchmarks of the literature are used and the proposed PSO-SA algorithm

obtains acceptable results. In fact, the hybrid algorithm improves the performance of simple PSO algorithm for all instances

<https://sic.ici.ro/wp-content/uploads/2012/03/SIC_2012-1-Art6.pdf>

3- Simple summary:

To solve travelling salesman problems (TSPs), most existing evolutionary algorithms search for optimal solutions from zero initial information without taking advantage of the historical information of solving similar problems. This paper studies a transfer learning-based particle swarm optimization (PSO) algorithm, where the optimal information of historical problems is used to guide the swarm to find optimal paths quickly. To begin with, all cities in the new and historical TSP problems are clustered into multiple city subsets, respectively, and a city topology matching strategy based on geometric similarity is proposed to match each new city subset to a historical city subset. Then, on the basis of the above-matched results, a hierarchical generation strategy of the feasible path (HGT) is proposed to initialize the swarm to improve the performance of PSO. Moreover, a problem-specific update strategy, i.e. the particle update strategy with adaptive crossover and clustering-guided mutation, is introduced to enhance the search capability of the proposed algorithm. Finally, the proposed algorithm is applied to 20 typical TSP problems and compared with 12 state-of-the-art algorithms. Experimental results show that the transfer learning mechanism can accelerate the search efficiency of PSO and make the proposed algorithm achieve better optimal paths.

<https://academic.oup.com/jcde/article/9/3/933/6590609>

4-

Simple summary: Although metaheuristic algorithms, such as PSO, have excellent performance than heuristic algorithms, they fall in premature convergence when used for complex problems in daily life. In other words, this concept causes particles, which show the role of feasible solutions to the problem, to be stuck in local optimal points, and spend a long time without improvement. Therefore, the quality of the obtained answers cannot grow continuously and achieve quality solutions. For this reason and to overcome premature convergence, several modifications are presented to improve the PSO algorithm in the literature [41–45]. In the PSO algorithm, pbest and gbest are considered, while in the proposed algorithm, called MPSO, the best current answer is also considered as gcbest. The method is that the solutions gbest and gcbest have two coefficients of a1 and a2, respectively, in which the sum of these two variables is equal to 100%. The values of these two coefficients change over time as the algorithm runs, as described in the following. Finally, three local search algorithms are considered for intensification to further improve the answers if gbest or pbest is updated. The steps of the proposed algorithm are described as follows.

<https://www.hindawi.com/journals/complexity/2021/6668345/>

5-

Simple summary:

Nowadays, the systems that are inspired by biological structures have gained importance and attracted the attention of researchers.

The Multiple Travelling Salesman Problem (MTSP) is an extended version of the TSP. The aim in the MTSP is to find the tours for m

salesmen, who all start and end at the depot, such that each intermediate node is visited exactly once and the total cost of visiting nodes is

minimized. The Particle Swarm Optimization (PSO) algorithm which is a meta-heuristic algorithm based on the social behaviour of birds.

In this article, 2 algorithms based on PSO, called APSO and HAPSO, were proposed to solve the MTSP. The APSO algorithm is based on

the PSO and 2-opt algorithms, the path-relink and swap operators. While the HAPSO algorithm is based on the GRASP, PSO and 2-opt

algorithms, the path-relink and swap operators. In the experiments, 5 TSP instances are used and the algorithms are compared with the GA

and ACO algorithms. According to the results, the HAPSO algorithm has the better performance than the other algorithms on the most

instances. Moreover the HAPSO algorithm produces more stable results than the APSO algorithm and the performance of the HAPSO

algorithm is better in all the MTSP instances. Therefore, the HAPSO algorithm is more robust than the APSO algorithm.

https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://www.ijisae.org/IJISAE/article/download/1012/583&ved=2ahUKEwiRxbWKiYGGAxWd\_7sIHZHTBV84ChAWegQIBRAB&usg=AOvVaw0JBVg0186RuuhC0I2afj0h

6-

Simple summary: The Travelling Salesman Problem, briefly TSP is a description of a large class of prob-

lems known as combinatorial optimization problems. The concept of the problem is as

follows: A salesman has a number of cities to visit with a distance or time between two

cities, he wants to take the shortest possible route from all these cities so that he does

not pass from the same city twice and eventually return to where he left off [1]. TSP

is usually simple to explain but very hard to solve, if the number of cities is small, the

answer can be easily found by looking at all possible routes and choosing the shorter

route, but with the increasing the number of cities, this method will become inefficient

and the complexity of solving the problem also increases [2].

Computational techniques stimulated by natural phenomenon were of great interest

in the recent years. The natural phenomenon of insects or large animals is studied to

develop different computing techniques in last few decades [2]. Various combinatorial

optimization problems such as Traveling Salesman Problem (TSP), Job-shop Scheduling

Problem ( JSP), and Vehicle Routing Problem (VRP) were also approached by various

modern heuristic methods, like ACO, IPSO and SFLA [3].

In this paper we give an overview of some meta-heuristic (ACO, PSO, IPSO and SFLA)

algorithms to get the shortest route for the TSP. These algorithms are introduced in

Section 2. Section 3 gives the experimental results of ACO, PSO, modified PSO, SFLA

and modified SFLA for TSP under the MATLAB tool. Consequently, in Section 4, we

conclude the paper with a summarization of results by emphasizing on the importance

of this study.

https://www.google.com/url?sa=t&source=web&rct=j&opi=89978449&url=https://knepublishing.com/index.php/KnE-Social/article/view/1394/3196&ved=2ahUKEwiOsoPzioGGAxWK87sIHbWLCEs4ChAWegQIChAB&usg=AOvVaw2BwbAR3XTr4jo6NPlLCMjc

# 5- the Dataset employed:

We'll assume the salesperson lives in a boring, flat, 2D Cartesian plane, and that each city can be described simply as existing at a single (x, y) location . Each datafile describes some number of cities, one city per line in the file, where each city has a index (first city is index-0) and a location.

We used the large dataset.

Link: https://www.kaggle.com/datasets/mexwell/traveling-salesman-problem

# 6- Details of the algorithm(s)/approach(es) used and the results of the experiments:

Particle Swarm Optimization (PSO)

* Algorithm Description: PSO is a population-based optimization algorithm inspired by the social behavior of birds flocking or fish schooling. In PSO, particles (candidate solutions) move through the search space looking for the optimal solution by adjusting their positions based on their own experience (pbest) and the collective experience of the swarm (gbest).
* Approach: The PSO algorithm in the provided code initializes a population of particles, where each particle represents a potential solution to the TSP. The particles update their positions (routes) iteratively based on their velocities, which are influenced by their personal best solutions and the global best solution found by the swarm.
* Results: The PSO algorithm's output is the best-found route and its corresponding cost. Additionally, the algorithm visualizes the convergence of the cost function over iterations and plots the best-found route for visualization purposes.

Simulated Annealing

* Algorithm Description: Simulated Annealing is a probabilistic optimization algorithm inspired by the annealing process in metallurgy. It starts with an initial solution and iteratively explores neighboring solutions, accepting worse solutions with a certain probability determined by a temperature parameter. Over time, the temperature decreases, reducing the probability of accepting worse solutions, allowing the algorithm to converge towards an optimal solution.
* Approach: The provided code initializes the Simulated Annealing algorithm with a nearest neighbor initial route and applies a cooling schedule to decrease the temperature over iterations. It then iteratively explores neighboring solutions, accepting worse solutions based on the acceptance probability determined by the current temperature.
* Results: The Simulated Annealing algorithm outputs the best-found route and its corresponding distance. It visualizes the best-found route for visualization purposes.

Nearest Neighbor Algorithm

* Algorithm Description: The Nearest Neighbor Algorithm is a simple heuristic algorithm that starts from an arbitrary city and repeatedly selects the closest unvisited city as the next stop until all cities are visited, returning to the starting city.
* Approach: The provided code initializes the Nearest Neighbor Algorithm with a random starting city and iteratively selects the nearest unvisited city until all cities are visited, forming a complete tour.
* Results: The Nearest Neighbor Algorithm outputs the best-found route and its corresponding distance. It visualizes the best-found route for visualization purposes.

Summary of Results

* PSO: The PSO algorithm's best-found route has a cost of approximately 53.41.
* Simulated Annealing: The Simulated Annealing algorithm's best-found route has a distance of approximately 29.40.
* Nearest Neighbor Algorithm: The Nearest Neighbor Algorithm's best-found route has a distance of approximately 28.1.

# 7-Development platform:

-tools: jupyter , pycharm,..

- programming languages: python.

- libraries: numpy , random , math , matplotlib , spicy .