LaurentianUniversity
UniversitéLaurentienne

BHARTI SCHOOL OF ENGINEERING
AND COMPUTER SCIENCE

PROJECT -6

Course: INTRODUCTION TO COMPUTATIONNAL SCIENCE (CPSC-5506)

Semester: WINTER-2024

Title: Butterfly Optimization Algorithm (BOA) for solving the Travel Salesman Problem.

1. Introduction

In this project, we are interested in implementing a simple version of the Butterfly optimization algorithm to solve the Travel Salesman problem. This project will be carried out as part of the master's course entitled "Introduction to computational science" (CPSC-5506). A literature review and some extensions of the problem will be presented. Some numerical experiments will be performed, commented and analyzed.

2. The Travel Salesman Problem:

The Travel Salesman Problem is a NP-hard problem extensively studied in the literature. It can be defined as follows: Given a list of cities and the distances between each pair of cities, what is the shortest possible route that visits each city exactly once and returns to the origin city?
The TSP has several applications even in its purest formulation, such as planning, logistics, and the manufacture of microchips. The vehicle routing problem is a generalization of TSP.
.

2.1 **Mathematical model of the Travel Salesman Problem**:
Given a set of n cities numbered from 1 up to n, and $c_{ij} > 0$ the distance from city i to city j, the TSP can be formulated as follow:

$$
\begin{cases}
\min \displaystyle\sum_{i=1}^{n} \sum_{\substack{j=1 \\ j\neq i}}^{n} c_{ij} x_{ij} \\
\text{Subject to} \\
\quad \displaystyle\sum_{\substack{i=1 \\ i\neq j}}^{n} x_{ij} = 1, \ \ j = 1, \cdots, n \\
\quad \displaystyle\sum_{\substack{j=1 \\ j\neq i}}^{n} x_{ij} = 1, i = 1, \cdots, n \\
\displaystyle\sum_{i\in I}^{n} \sum_{\substack{j\in I \\ j\neq i}}^{n} x_{ij} \leq |I| - 1, \qquad for\ all\ I \subset \{1, 2, \cdots, n\} \quad |I| \geq 2
\end{cases}
$$

Where $x_{ij} = 1$ if the path goes from city i to city j and 0 otherwise.

2.2 **Hypotheses:**

a) The number of cities is given
b) The distance between each pair of cities is also given.

2.3 **Objectives of the study:**

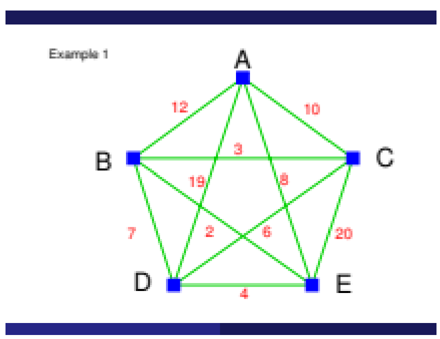
a) Review of the literature, the extensions the TSP and the main algorithms to solve it.
b) Implement the GWO algorithm in C++, in Python or in Java to solve the TSP.
c) Perform numerical experiments on the instances presented below in Section 4.

3. Butterfly Optimization Algorithm

The Butterfly Optimization Algorithm (BOA) is a metaheuristic for solving optimization problems. The algorithm replicates the behavior of the natural butterfly which can be described to perform a cooperative movement while navigating towards its food source and position. The butterflies are well known for having the power to analyze and receive smell in the air thereby tracing and discovering the direction of their partner and food source.

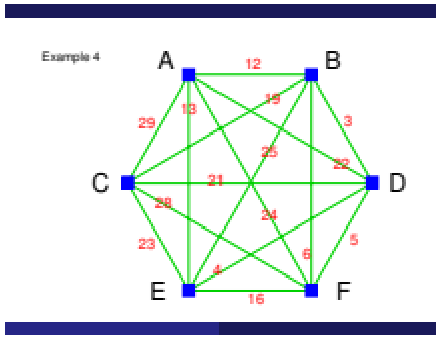
4. Test Problems

Problem 1: 5 cities



	A	B	C	D	E
A	∞	12	10	19	8
B	12	∞	3	7	2
C	10	3	∞	6	20
D	19	7	6	∞	4
E	8	3	20	2	∞

Problem 2: 6 cities



	A	B	C	D	E	F
A	∞	12	29	22	13	24
B	12	∞	19	3	25	6
C	29	19	∞	21	23	28
D	22	3	21	∞	4	5
E	13	25	23	4	∞	16
F	24	6	28	5	16	∞

Problem 3: 6 cities

∞	64	378	519	434	200
64	∞	318	455	375	164
378	318	∞	170	265	344
519	455	170	∞	223	428
434	375	265	223	∞	273
200	164	344	428	273	∞

<u>Problem 4</u>: 15 cities

∞	141	134	152	173	289	326	329	285	401	388	366	343	305	276
141	∞	152	150	153	312	354	313	249	324	300	272	247	201	176
134	152	∞	24	48	168	210	197	153	280	272	257	237	210	181
152	150	24	∞	24	163	206	182	133	257	248	233	214	187	158
173	153	48	24	∞	160	203	167	114	234	225	210	190	165	137
289	312	168	163	160	∞	43	90	124	250	264	270	264	267	249
326	354	210	206	203	43	∞	108	157	271	290	299	295	303	287
329	313	197	182	167	90	108	∞	70	164	183	195	194	210	201
285	249	153	133	114	124	157	70	∞	141	147	148	140	147	134
401	324	280	257	234	250	271	164	141	∞	36	67	88	134	150
388	300	272	248	225	264	290	183	147	36	∞	33	57	104	124
366	272	257	233	210	270	299	195	148	67	33	∞	26	73	96
343	247	237	214	190	264	295	194	140	88	57	26	∞	48	71
305	201	210	187	165	267	303	210	147	134	104	73	48	∞	30
276	176	181	158	137	249	287	201	134	150	124	96	71	30	∞

<u>Problem 5</u>: 29 cities

∞ 74 4110 3048 2267 974 4190 3302 4758 3044 3095 3986 5093 6407 5904 8436 6963
6694 6576 8009 7399 7267 7425 9639 9230 8320 9300 8103 7799

74 ∞ 4070 3000 2214 901 4138 3240 4702 2971 3021 3915 5025 6338 5830 8369 6891
6620 6502 7939 7326 7193 7351 9571 9160 8249 9231 8030 7725

4110 4070 ∞ 1173 1973 3496 892 1816 1417 3674 3778 2997 2877 3905 5057 5442 4991
5151 5316 5596 5728 5811 5857 6675 6466 6061 6523 6165 6164

3048 3000 1173 ∞ 817 2350 1172 996 1797 2649 2756 2317 2721 3974 4548 5802 4884
4887 4960 5696 5537 5546 5634 7045 6741 6111 6805 6091 5977

2267 2214 1973 817 ∞ 1533 1924 1189 2498 2209 2312 2325 3089 4401 4558 6342 5175
5072 5075 6094 5755 5712 5828 7573 7222 6471 7289 6374 6187

974 901 3496 2350 1533 ∞ 3417 2411 3936 2114 2175 3014 4142 5450 4956 7491 5990
5725 5615 7040 6430 6304 6459 8685 8268 7348 8338 7131 6832

4190 4138 892 1172 1924 3417 ∞ 1233 652 3086 3185 2203 1987 3064 4180 4734 4117
4261 4425 4776 4844 4922 4971 5977 5719 5228 5780 5302 5281

3302 3240 1816 996 1189 2411 1233 ∞ 1587 1877 1979 1321 1900 3214 3556 5175 4006
3947 3992 4906 4615 4599 4700 6400 6037 5288 6105 5209 5052

4758 4702 1417 1797 2498 3936 652 1587 ∞ 3286 3374 2178 1576 2491 3884 4088 3601
3818 4029 4180 4356 4469 4497 5331 5084 4645 5143 4761 4787

3044 2971 3674 2649 2209 2114 3086 1877 3286 ∞ 107 1360 2675 3822 2865 5890 4090
3723 3560 5217 4422 4257 4428 7000 6514 5455 6587 5157 4802
3095 3021 3778 2756 2312 2175 3185 1979 3374 107 ∞ 1413 2725 3852 2826 5916 4088
3705 3531 5222 4402 4229 4403 7017 6525 5451 6598 5142 4776

3986 3915 2997 2317 2325 3014 2203 1321 2178 1360 1413 ∞ 1315 2511 2251 4584 2981
2778 2753 4031 3475 3402 3531 5734 5283 4335 5355 4143 3897

5093 5025 2877 2721 3089 4142 1987 1900 1576 2675 2725 1315 ∞ 1323 2331 3350 2172
2275 2458 3007 2867 2935 2988 4547 4153 3400 4222 3376 3307

6407 6338 3905 3974 4401 5450 3064 3214 2491 3822 3852 2511 1323 ∞ 2350 2074 1203
1671 2041 1725 1999 2213 2173 3238 2831 2164 2901 2285 2397

5904 5830 5057 4548 4558 4956 4180 3556 3884 2865 2826 2251 2331 2350 ∞ 3951 1740
1108 772 2880 1702 1450 1650 4779 4197 2931 4270 2470 2010

8436 8369 5442 5802 6342 7491 4734 5175 4088 5890 5916 4584 3350 2074 3951 ∞ 2222
2898 3325 1276 2652 3019 2838 1244 1089 1643 1130 2252 2774

6963 6891 4991 4884 5175 5990 4117 4006 3601 4090 4088 2981 2172 1203 1740 2222 ∞
684 1116 1173 796 1041 974 3064 2505 1368 2578 1208 1201

6694 6620 5151 4887 5072 5725 4261 3947 3818 3723 3705 2778 2275 1671 1108 2898 684
∞ 432 1776 706 664 756 3674 3090 1834 3162 1439 1120

6576 6502 5316 4960 5075 5615 4425 3992 4029 3560 3531 2753 2458 2041 772 3325 1116
432 ∞ 2174 930 699 885 4064 3469 2177 3540 1699 1253

8009 7939 5596 5696 6094 7040 4776 4906 4180 5217 5222 4031 3007 1725 2880 1276 1173
1776 2174 ∞ 1400 1770 1577 1900 1332 510 1406 1002 1499

7399 7326 5728 5537 5755 6430 4844 4615 4356 4422 4402 3475 2867 1999 1702 2652 796
706 930 1400 ∞ 371 199 3222 2611 1285 2679 769 440

7267 7193 5811 5546 5712 6304 4922 4599 4469 4257 4229 3402 2935 2213 1450 3019 1041
664 699 1770 371 ∞ 220 3583 2970 1638 3037 1071 560

7425 7351 5857 5634 5828 6459 4971 4700 4497 4428 4403 3531 2988 2173 1650 2838 974
756 885 1577 199 220 ∞ 3371 2756 1423 2823 852 375

9639 9571 6675 7045 7573 8685 5977 6400 5331 7000 7017 5734 4547 3238 4779 1244 3064
3674 4064 1900 3222 3583 3371 ∞ 620 1952 560 2580 3173

9230 9160 6466 6741 7222 8268 5719 6037 5084 6514 6525 5283 4153 2831 4197 1089 2505
3090 3469 1332 2611 2970 2756 620 ∞ 1334 74 1961 2554

8320 8249 6061 6111 6471 7348 5228 5288 4645 5455 5451 4335 3400 2164 2931 1643 1368
1834 2177 510 1285 1638 1423 1952 1334 ∞ 1401 648 1231

LaurentianUniversity
UniversitéLaurentienne

9300 9231 6523 6805 7289 8338 5780 6105 5143 6587 6598 5355 4222 2901 4270 1130 2578
3162 3540 1406 2679 3037 2823 560 74 1401 ∞ 2023 2617

8103 8030 6165 6091 6374 7131 5302 5209 4761 5157 5142 4143 3376 2285 2470 2252 1208
1439 1699 1002 769 1071 852 2580 1961 648 2023 ∞ 594

7799 7725 6164 5977 6187 6832 5281 5052 4787 4802 4776 3897 3307 2397 2010 2774 1201
1120 1253 1499 440 560 375 3173 2554 1231 2617 594 ∞

PLAN OF THE REPORT

Table of content
Introduction
Chapter 1: THE TRAVEL SALESMAN PROBLEM
1.1 Introduction
1.2 The Travel Salesman Problem
1.3 Extension of the Travel Salesman Problem
1.4 Conclusion

Chapter 2: LITERATURE REVIEW
2.1 Introduction
2.2 Solution methods for the Travel Salesman Problem
2.3 Conclusion

Chapter 3: NUMERICAL EXPERIMENTS

4.1 Introduction
4.2 Grey Wolf Optimization Algorithm for solving the TSP
4.3 Numerical Results
4.4 Conclusion

Conclusion

Bibliography