Studying Heuristics Adaptation to a Specific Degree of Fuzziness

Gloria Cerasela Crişan
Faculty of Sciences
Vasile Alecsandri University
Bacău, Romania
ceraselacrisan@ub.ro

Camelia–M. Pintea
Faculty of Sciences
Technical University Cluj-Napoca
Baia-Mare, Romania
dr.camelia.pintea@ieee.org

Petrica C. Pop
Faculty of Sciences
Technical University Cluj-Napoca
Baia-Mare, Romania
petrica.pop@cunbm.utcluj.ro

***Abstract*—Nowadays, in the world are produced at high speed tremendous volumes of data which is sometimes incomplete, inconsistent, and/or uncertain. Therefore, operating and analyzing such data must be error-tolerant and should maintain a specific degree of result reliability. The behavior of several greedy and adaptive heuristics, including a natural-inspired optimization approach, the Discrete Fruit Fly Optimization Algorithm (DFOA), when applied to a set of fuzzified Traveling Salesman Problem (TSP) instances is studied. Several symmetric TSP instances were systematically transformed using two information imperfection parameters: the volume and the amplitude. The specified heuristics adaptation to a systematic degree of fuzziness is analyzed.**

***Index Terms*—Traveling Salesman Problem; Discrete Fruit Fly Optimization Algorithm; Optimization; Uncertainty Model**

I. INTRODUCTION

Information produced, stored and analyzed in our days is not fully reliable, as it is imperfect. Most problems we face today do not embed clear-cut facts and must be solved when the data is incomplete, inconsistent or uncertain. Computational applications are therefore requested to efficiently operate with inexact information, like the human mind does.

When coping with low-quality data, two strategies could be used: either using pre-processing procedures, to clean the data [1], or designing more reliable applications, able to deal with inexact data [2].

One of the first approaches to decision under risk [3] defined the expected utility of an agent competing for resources in an Economic environment.

The decision under uncertainty evolved later on in [4], by combining the expected utility with subjective probabilities into the subjective expected utility. Mathematical models based on rationality and axioms helped in defining the Game Theory domain and also in addressing the inherent complexity of social, natural or scientific phenomena.
The sources of uncertainty are multiple and a specific process could compound its uncertainties [5]:

- model parameters which are not exactly known;
- input data characteristics which can vary;
- model bias, as a result from the insufficient model accuracy;
- numerical approximations from computations;
- observation errors, due to imperfect measuring devices or human errors;
- interpolation errors, due to insufficient experimental data.

Quantifying the uncertainty relies mostly on computer simulations [6]. The goal of uncertainty quantification is to find how likely a process will produce a specific outcome if some characteristics of the process are not exactly known.

Approximate computing is a new computing paradigm that produces a bounded approximate result instead of an exact result, with considerable gain in computational cost [7]. This relaxed approach to traditional computing is fitted to error-tolerant applications which are expected to run with high speed; for example, search engines response to queries or multimedia processing.

Trans-precision computing is a step forward from the approximate computing [8]. It proposes dynamic adjustments during computations:

- floating-point data are stored on less bits than the common IEEE 754 Standard;
- applications are stopped when a specific precision of the result is reached.

The current work uses the uncertainty quantification paradigm in order to investigate the behavior of a nature-inspired algorithm when the data fed are uncertain. Following the classification from [5], here we use variability of the input data in order to quantify the reliability of *Discrete Fruit Fly Optimization Algorithm (DFOA)*, when applied to a set of fuzzified *Traveling Salesman Problem (TSP)* instances.

The investigated nature-inspired algorithm (*DFOA*) belongs to the *Swarm Intelligence* category of solving methods [9], which uses natural or artificial systems composed of coordinated individuals that complete a difficult common task using decentralized control and self-organization. *DFOA* is a population-based meta-heuristics, which is designed to approach a Combinatorial Optimization Problem [10].

The framework starts by creating a population of solutions to the proposed problem; each solution models a food location; for a number of generations, a cycle of smelling, evaluating and flocking is performed by the fruit flies; at the end, the global best solution is returned.

978-1-7281-6932-3/20/$31.00 ©2020 IEEE

DFOA was later improved in several ways. For example, *Elimination-based Fruit Fly Optimization Algorithm* (*EFOA*) reinforces the smelling process, eliminates the weak solutions and perform gradual flocking, instead of direct mass move towards the best fly [11].

An improved FOA with immune response was proposed in [12]. The basic idea of this version is to use a decision mechanism for interrupting the FOA and to use a method for generating antigens which enforce the space exploration. The literature reports good results for DFOA and its versions: basic version is comparable to other metaheuristic approaches, and improved versions obtain best results on benchmark datasets [13].

The *Traveling Salesman Problem (TSP)* [14] was chosen here as optimization benchmark problem for the *DFOA*. The classical variant of TSP (with perfect data), and a fuzzified TSP variant, including a controlled degree of uncertainty, were approached by DFOA.

This paper-work focuses on the behavior of several heuristic methods when applied to uncertain data. In particular, the flexibility of a *Swarm Intelligence* algorithm, a local search method and several construction heuristics is studied when fuzzy data are fed. The work explores the *Discrete Fruit Fly Optimization Algorithm (DFOA), Lin-Kernighan (LK), Greedy, Nearest Neighbor,* $Bor\mathring{u}vka$, and *Quick-*$Bor\mathring{u}vka$ approaches and studies their sensitivity to fuzzy data, continuing the practical experiments from [15], [16].

The article is organized as follows. In Section II, the solving methods and the fuzzification scheme for the Traveling Salesman Problem (*TSP*) are described. Section III describes and analyses the computational results obtained, and Section IV concludes this work.

II. Problem and Solving Methods

In this Section, several discrete Optimization methods are presented, and also the *TSP* benchmark with perfect and uncertain/fuzzy data.

A. Traveling Salesman Problem (TSP)

The *Traveling Salesman Problem* (*TSP*) is one of the most-studied Combinatorial Optimization Problems [17]. It has a simple specification and broad applications [14] in transportation, logistics, industry, communications, etc.

One TSP specification, based on Graph Theory follows. *Given a complete graph with weights on the edges (arcs), find a Hamiltonian cycle with minimum total weight.*

The mathematical formulation of TSP considers a complete graph $G = (V, E)$ with vertex set $V = \{1, 2, \ldots, n\}$ and weights on edges stored in the distance matrix. For each edge from E, a corresponding binary variable is set to 1 if and only if the edge belongs to the chosen Hamiltonian cycle. The goal of TSP is *to minimize the objective function Z*:

$$Z = \sum_{i=1}^{n} \sum_{j=1}^{n} d_{ij} y_{ij}, \tag{1}$$

s.t.

$$\sum_{j=1}^{n} y_{ij} = 1, \forall i \in V, \tag{2}$$

$$\sum_{i=1}^{n} y_{ij} = 1, \forall j \in V, \tag{3}$$

$$\sum_{i \in S} \sum_{j \in S} y_{ij} \leq |S| - 1, \forall S \subset V, |S| > 2. \tag{4}$$

The objective function (1) describes the total length of a set of edges. The constraints (2) and (3) ensure that for each vertex from V, one and only one edge from the set of selected edges starts from that vertex, and respectively one and only one edge from the set of selected edges ends in that vertex. The constraint (4) eliminates the cycles that are not Hamiltonian.

There are special cases of TSP; for example, if the distance matrix is symmetric, then we have *Symmetric TSP*. If the vertices are points in a plane and the distance matrix is computed using the Euclidean norm, then we have *2D Euclidean TSP*.

There are multiple collections of classical TSP instances [18], [19]; several real-life complex problems are available at [20], [21].

The first *uncertain TSP* is described in [22]. The *Probabilistic TSP* searches for a most efficient *fixed* Hamiltonian cycle when each city is to be visited with a specific probability. Fuzzy *TSP* is defined as *TSP* with fuzzy numbers as distances on the edges [23]. The solution in this case is a classic TSP solution with a fuzzy number as length. This *TSP* variant could consider solutions that are comparable, so the solving method needs to take appropriate decisions to orient the search. The *TSP with interval data* specifies the distance between any two cities as interval of real numbers [24]. The solution to this *TSP* variant is the tour that minimizes the maximum deviations over all possible distances. *Dynamic TSP* models are various; one example is in [25]; the distance function is dynamic, and so is the number of cities. A dynamic, clustered *TSP* is investigated in [26]. An uncertain multi-objective *TSP* is defined and solved in [27].

TSP is a NP-hard Optimization problem [17]; therefore, exact methods are unlikely to solve large instances in an affordable time. Currently, the best exact solver is Concorde [28] which found the optimal value to all the 110 *TSPLIB* instances from [18]. Efficient parallel branch-and-bound methods are described in [29]. Heuristic methods represent a class of solving approaches which balance the quality of the results with the computational resources needed.

Heuristics provide in most cases “good-enough” feasible solutions to difficult problems in a fraction of time needed by exact methods, but no guarantee exists: it is possible to return bad solutions or no solution.

Several biologically-inspired heuristic methods for approaching difficult TSP instances are *Ant Colony Optimization* (*ACO*) [30], *Particle Swarm Optimization* (*PSO*) [31] or *Discrete Fruit Fly Optimization algorithm* (*DFOA*) [10].

The class of greedy heuristic methods uses a myopic strategy: they construct a feasible solution "step by step, in a local optimum way". Several such methods for TSP are: Nearest Neighbor [32], Greedy [33], $Bor\mathring{u}vka$, Quick-$Bor\mathring{u}vka$ [34].

Local search heuristics improve an already existing solution. For TSP, the most simple and known such method is Lin-Kernighan [35]. A eugenic bacterial memetic algorithm (EBMA) was adapted for fuzzy TSP in [36]. It is one early paper which identified and investigated the potential of using swarm intelligence in approaching fuzzy versions of Combinatorial Optimization Problems.

B. Discrete Fruit Fly Optimization Algorithm

The *Fruit Fly Optimization Algorithm* (FOA) is a recent continuous optimization algorithm inspired by the foraging behavior of fruit flies [37]. Their senses of smelling, seeing and tasting are very accurate and support the search.

When a fruit fly searches for food, it randomly flies until it discovers by smelling a favorable location or it sees a concentration of other fruit flies. The next step is to taste the location, and therefore to evaluate its quality. If the quality is high, then the fruit fly remains near the promising region, explores the neighborhood and exchanges information with peers.

The FOA is a population-based algorithm, where a constant number of solutions are developed during a fixed number of cycles. At the beginning, the solutions are randomly generated, or a construction heuristic is used. Each cycle consists of:

- *smelling*: each agent searches near the swarm location;
- *tasting*: each agent evaluates its current position;
- *moving*: each agent moves toward the best position in this cycle.

At the end, FOA returns the *Global Best* position. Due to its simple and adaptable structure, adjustable parameters and generous solution representation, FOA was successfully used for solving many continuous optimization problems [37].

The discrete version of FOA (DFOA) was published in [10]. This framework simplified the iterations to only two steps presented in Fig. 1 ([10]):

- *smelling*: each agent randomly borrows a part from the information stored in the *Global Best* position and mixes it with its current position. It moves to the new position and evaluates what it found.
- *tasting*: a local search around the promising position, in order to find new, better position.

The computational time complexity of DFOA is given by formula (5):

$$popsize \cdot max\{fit, cross, upd\} \cdot MaxGen \quad (5)$$

where *popsize* is the constant number of fruit flies, *fit*, *cross* and *upd* are the complexities of the fitness function, crossover operator and update operators, and *MaxGen* is the maximum number of generations allowed. If we denote the problem size by *n*, usually the *popsize* is *n*, the fitness function has O(*n*) complexity, the crossover is $O(n)$, and update is $O(n)$ or $O(n^2)$.

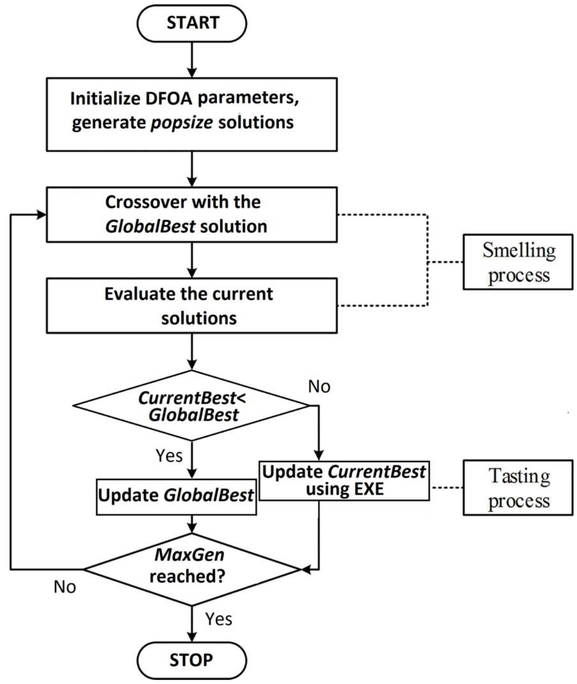

Fig. 1. Discrete Fruit Fly Algorithm (DFOA) [10]

.

Therefore, the maximum time complexity of DFOA is $O(n^3)$ [10].

When applied to 2D Euclidean TSP, each solution represents a Hamiltonian cycle. At the beginning, the solutions are generated randomly and the *Global Best* is computed.

The *smelling* phase is the exploitation part of the iteration, where each agent randomly generates two position of the *Global Best* solution, extracts the corresponding subtour and includes it in its solution, by reconstructing in the best way the Hamiltonian cycle. Therefore, it is a crossover operator applied always with the *Global Best* solution. The first phase ends after all new positions are evaluated. The second phase is the exploration phase, when each agent tries to improve the *Current Best* solution, by eliminating the edge intersections (not considering a vertex as edge intersection). If some solutions are better than the *Global Best*, the *Global Best* solution is updated. The algorithm returns the *Global Best* solution, when the *MaxGen* value was reached.

The balance between exploitation and exploration (information exchange and local search) provided encouraging results when compared against Parallel Hybrid Genetic Algorithm [38] and Particle Swarm Optimization [39].

C. Greedy methods for Traveling Salesman Problem

The group of *solution construction* methods makes use of a specific strategy to iteratively add components to a partial solution, until a complete solution is obtained. Among the solution construction strategies, the *greedy* ones make myopic decisions: at each step, is chosen the best component among the available ones, without long-term evaluation.

These methods usually deliver good solutions, but not (local) optima ones. One of the first greedy methods is *Nearest Neighbor* (NN) [32]. Given a TSP instance, NN uses a binary flag for marking each vertex: unmark vertices, randomly choose and mark a vertex i as *current vertex,**find an unmarked vertex j closest to i; if no unmarked vertices, STOP; set and mark j as *current vertex* and go to step *. In the end, the Hamiltonian cycle found is printed, and its length is computed.

Greedy method constructs a feasible solution to TSP [40] as it follows: unmark all edges, select the edge with minimum cost, mark it and print it (in case of multiple such edges, choose one at random); *find the unmarked edge with minimum cost, which does not close the tour and does not create a degree-3 vertex; if no unmarked edges, STOP. Mark the selected edge, print it and go to step *.

The idea is to collect the shortest edges as long as they could belong to a Hamiltonian cycle.

$Bor\mathring{u}vka$ approach to TSP is a modification following the Christofides idea [41] of the algorithm for Minimum Spanning Tree proposed in [34].

Quick-$Bor\mathring{u}vka$ is proposed by Concorde implementation, as a rapid variant of the $Bor\mathring{u}vka$ method [42]: set the solution C to empty set; put all the vertices from V in a list; *explore the list from the beginning; for each vertex I with degree less than 2 in C, consider the set of its incident edges not in C; among these edges, put in C the shortest one which neither prematurely close the cycle, nor create a vertex of degree greater than 2; if C is not an Hamiltonian cycle, go to step *, else print C, STOP.

D. Lin-Kernighan method for Traveling Salesman Problem

The group of local search methods is based on an initial solution which is improved using a specific strategy. A common idea is to delete a set of edges from a TSP solution and to better reconnect the parts. Such attempts are known as local moves, as they transform a solution into a neighbor one (in the solution space). Lin-Kernighan technique is an extended local search approach to TSP. Based on the instance, it decides how far to go when testing iteratively 2-opt, 3-opt, etc. An effective implementation is in [35].

E. Fuzzified Traveling Salesman Problem

The theory of fuzzy sets was developed by Lotfi A. Zadeh in [43]. In this seminal paper, the *fuzzy sets* were defined as collections of elements with degrees of membership belonging to the interval [0, 1]. Therefore, a fuzzy set generalizes the notion of set: the membership function that characterizes a set has only two values. Fuzzy numbers are fuzzy sets with numbers in the same range and are used to represent the vagueness, the same way the random values express the probability. Further details on fuzzy concepts and applications are given in [44].

The idea of TSP fuzzification is to introduce a perturbation in data which models the real-life situations when the distance between two vertices is changed due to road closures, bottlenecks or bad weather.

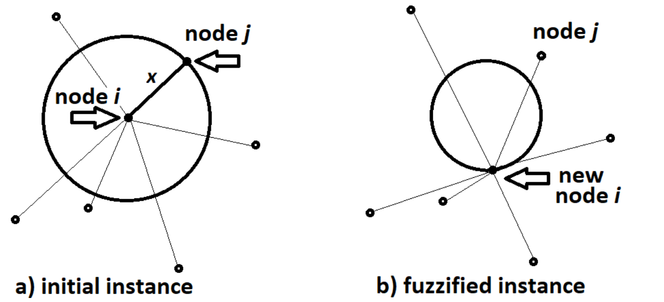

Fig. 2. a) Representation of an initial instance. b) The node i is moved in its neighborhood.

An example of fuzzification method is described in [45]. A different fuzzification approach used here follows. Each *instance* chosen for experiments was fuzzified by slightly moving several nodes within their neighborhood.

We define the *neighborhood* of a vertex i, as the disk with center i and radius the minimum distance from i to any other vertex. For example, in Fig. 2 a) the *neighborhood* of i is the closed disk in thick line.

The fuzzification procedure applied to an *instance* determines *how many* vertices are moved and *how far* in its neighborhood is each one moved.

Two parameters $a \in (0, 100)$ and $b \in (0, 100)$ are used to express the degree of uncertainty, i.e., the fuzziness for each *instance*. The parameter a controls the number of moved nodes. The parameter b controls how far the nodes are moved.

Fig. 2 b) shows the modified position of node i, randomly chosen on the circle *C(i, r)*, the circle with the center in the old node i and the radius $r = x \cdot \frac{b}{100}$, where x is the distance from i to the nearest node.

The pseudocode of the procedure *fuzzifyInstance* follows.

procedure fuzzifyInstance
read $instance, a, b$
$k = instance.Nodes \cdot \frac{a}{100}$
unmark all nodes from $instance$
for $j = 1$ to k **do**
  randomly choose an unmarked node i
  $x = min\{d(i, u),\, u \in instance, u \neq i\}$
  $r = x \cdot \frac{b}{100}$
  randomly choose $new_i \in C(i, r)$
  mark new_i
end for
return instance

The function returns the modified *instance* after applying the moves controlled by a and b. The function *min* returns the minimum of the distances between the current node i and any other node from *instance*, and *C(i, r)* is the circle with the center in the current node i and radius r.

III. EXPERIMENTAL RESULTS

We continue experimental work on studying the natural computing resilience in fuzzy environments; an ant-based algorithm and a version of particle swarm algorithm were previously analyzed [15], [16]. A Discrete Fruit Fly Optimization Algorithm and another adaptive heuristic, Lin Kernighan are tested; extended experiments are made also on several greedy techniques using an AMD computation device (2.8GHz, 3GB RAM). An analysis and comparison of the results follows.

The experimental process makes use of several TSPLIB large instances [18] namely: *rd400, pr439, pr1002, rl1304, rl1323, vm1084, vm1748 and rl1889*. We choose higher dimensions data when compared with [16] as *DFOA* obtains the best solution for all included instances. The specified group of instances was successively modified based on the *volume* and *the amplitude* parameters, denoted by *a* and *b*.

Continuing on the same basis as in [15], [16] for each instance from the group were created fuzzified instances corresponding to the *(a,b)* pair of parameters: $(10, 25), (10, 50), (25, 25)$ and $(25, 50)$.

Concorde software [28] was used for the exact solver CPLEX, Lin Kernigan and the greedy techniques.

The *Discrete Fruit Fly Optimization software* from [10] was used for the *DFOA* experiments. For each type of techniques, the predefined parameters were included.

Two cases are presented for all considered fuzzy data sets:

A) Comparing the results of fuzzy data with the same algorithm optimal of the initial data set.

B) Comparing the results of fuzzy data with the CPLEX optimal of the initial data set.

A) The same algorithm is run with both the initial and fuzzy data set. The obtained results, the optimal of the initial data set and the best of fuzzy data, are afterwards compared.

Firstly, on focus on the *Discrete Fruit Fly Optimization (DFOA) fuzzy data*: (10,25), (10,50), (25,25) and (25,50) *results when compared with initial data set group results based on each optimal solution.* The values of Table 1 are using the measure of the fuzzy impact with the percentage change, a difference between obtained solution and the best solution in proportion with the best solution. The average, minimal and maximal values after ten executions for each *DFOA* instance are shown in Table 1.

Follows the exact solver CPLEX and the *Lin Kernighan (LK)* heuristic are tested on initial and fuzzy data: *(10,25), (10,50), (25,25)* and *(25,50)*. The fuzzy data results are compared with initial data set group results based on each optimal solution. Table 2 shows the fuzzy impact values, the average, minimal and maximal percentage change values.

In Table 3 the fuzzy impact values, the average, minimal and maximal percentage change values for the greedy techniques are included. The values are based on the *fuzzy data*: (10,25), (10,50), (25,25) and (25,50) and *compared with optimum of each initial data set.*

TABLE I
COMPARISON BETWEEN INITIAL DATA SET AND FUZZY DATA, OPTIMAL SOLUTIONS OF THE *Discrete Fruit Fly Optimization Algorithm (DFOA).*

DFOA	***(10, 25)***	***(10, 50)***	***(25,25)***	***(25, 50)***
Average	0.021	-0.065	0.005	0.103
Min	-0.244	-0.563	-0.347	-0.416
Max	0.327	0.329	0.360	0.884

TABLE II
COMPARISON BETWEEN INITIAL DATA SET AND FUZZY DATA OPTIMAL SOLUTION OF *CPLEX and Lin Kernighan (LK).*

	(10, 25)	***(10, 50)***	***(25,25)***	***(25, 50)***
CPLEX				
Average	0.021	-0.065	0.005	0.103
Min	-0.244	-0.563	-0.347	-0.416
Max	0.327	0.329	0.360	0.884
Lin Kernighan (LK)				
Average	1.172	0.808	3.028	-0.230
Min	-3.570	-3.671	-0.736	-1.920
Max	6.288	4.798	9.615	0.879

TABLE III
COMPARISON BETWEEN INITIAL DATA SET AND FUZZY DATA OPTIMAL SOLUTION OF *Greedy techniques Greedy, Borůvka, q- Borůvka* AND *NN.*

	(10, 25)	***(10, 50)***	***(25,25)***	***(25, 50)***
Greedy				
Average	0.298	1.944	-0.457	1.142
Min	-3.840	-1.764	-4.171	-2.453
Max	7.094	5.404	7.401	5.497
Borůvka				
Average	-0.560	-0.220	-1.648	-1.135
Min	-5.725	-6.273	-5.370	-4.322
Max	4.198	7.063	2.315	2.703
q- *Borůvka*				
Average	0.842	0.668	0.270	0.490
Min	-1.378	-0.237	-3.185	-3.649
Max	4.021	2.400	6.390	4.021
NN				
Average	1.128	1.328	0.697	2.195
Min	-5.121	-3.570	-3.671	-0.736
Max	4.949	6.288	4.798	9.615

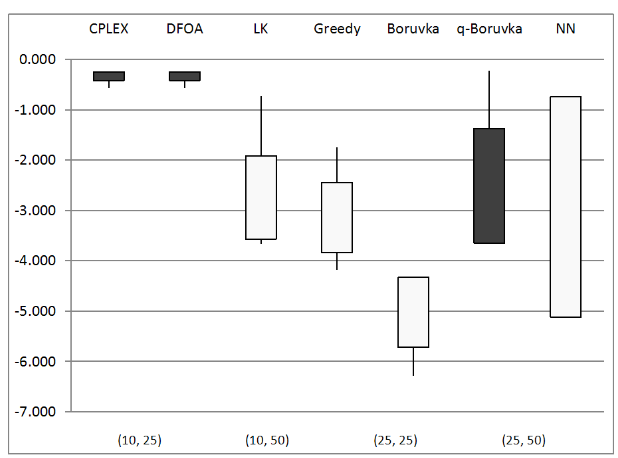

Fig. 3. Comparison of the best values from Table 1-3 heuristics

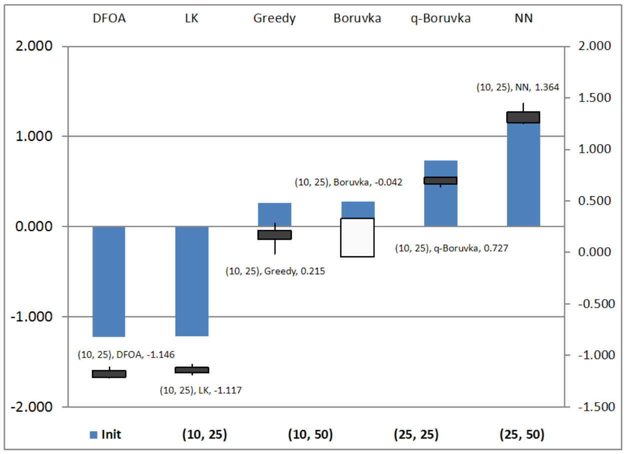

Fig. 4. Comparison of the normalized (z-scores) best values from Table 4-5 heuristics.

Figure 3 illustrates an overall comparison of all considered greedy and adaptive heuristics from Tables 1, 2 and 3 related to the minimal best values influence on the data fuzziness for each (a, b) set of parameters; study case A).

Tables 1-2 and Figure 3 shows that Lin Kernighan (LK) is much sensitive to fuzzy data than CPLEX and DFOA, both last techniques with the same fuzzy impact in case A).

Based on Table 3, we measure the overall average, for all fuzzy data (10, 25), (10, 50), (25,25) and (25,50) of each algorithm average: Greedy is 0.7317, for $Bor\mathring{u}vka$ is -0.8907, for q-$Bor\mathring{u}vka$ is 0.5675 and for NN is 1.3368. NN detects the modifications in data, but is mainly used for small instances.

For larger data Greedy and $Bor\mathring{u}vka$ shows similar fuzzy impact and q-$Bor\mathring{u}vka$ seems to be less sensitive to data modifications. (Figure 3, case A).

B) We compare the CPLEX optimal of each initial data set with the best results of each algorithm on initial and fuzzy (10,25), (10,50), (25,25) and (25,50) data.

Table 4 presents the average, minimal and maximal fuzzy impact values for both adaptive heuristics the Discrete Fruit Fly Optimization (DFOA) and Lin Kernighan (LK).

Table 5 includes the average, minimal and maximal fuzzy impact values for best solutions of the Greedy techniques (Greedy, $Bor\mathring{u}vka$, q-$Bor\mathring{u}vka$ and NN) on initial and fuzzy data, when compared with the CPLEX optimal on initial data. Figure 4 illustrates an overall comparison of considered greedy and adaptive heuristics from Table 4-5 normalized values (z-scores), related to the influence on the data fuzziness for each (a,b) set of parameters; study case B).

For greedy techniques, in case B), based on Table 5 a similar conclusion with case A) can be made. We measure the overall average, for all fuzzy data (10, 25), (10, 50), (25, 25) and (25, 50) of each algorithm average: Greedy is 17.025, for $Bor\mathring{u}vka$ is 17.193, for q-$Bor\mathring{u}vka$ is 20.744 and for NN is 26.306. Therefore, from the studied greedy techniques, the Greedy algorithm and the $Bor\mathring{u}vka$ algorithm can detect more accurately the modifications in data.

TABLE IV
COMPARISON OF CPLEX OPTIMAL ON INITIAL DATA AND BEST SOLUTION OF THE *Discrete Fruit Fly Optimization Algorithm (DFOA) and Lin Kernighan (LK) on initial and fuzzy data.*

	Initial	*(10, 25)*	*(10, 50)*	*(25,25)*	*(25, 50)*
DFOA					
Average	0.000	0.021	-0.065	0.005	0.103
Min	0.000	-0.244	-0.563	-0.347	-0.416
Max	0.001	0.327	0.329	0.360	0.884
Lin Kernighan (LK)					
Average	1.533	1.433	1.581	1.147	1.296
Min	0.020	0.013	-0.229	-0.112	0.036
Max	3.802	4.257	7.561	4.151	4.295

TABLE V
COMPARISON OF CPLEX OPTIMAL ON INITIAL DATA AND BEST SOLUTION OF *Greedy techniques Greedy, $Bor\mathring{u}vka$, q- $Bor\mathring{u}vka$* AND *NN, on initial and fuzzy data.*

	Initial	*(10, 25)*	*(10, 50)*	*(25,25)*	*(25, 50)*
Greedy					
Average	16.383	16.675	18.606	15.788	17.670
Min	12.927	11.818	13.909	9.819	11.731
Max	20.494	20.938	21.173	21.285	20.809
$Bor\mathring{u}vka$					
Average	18.081	17.373	17.721	16.098	16.692
Min	13.040	9.538	13.673	10.567	13.544
Max	22.479	23.350	23.641	20.723	20.187
q-$Bor\mathring{u}vka$					
Average	20.214	21.216	21.006	20.495	20.788
Min	17.069	16.353	17.255	16.550	16.524
Max	24.893	26.893	24.597	24.549	26.893
NN					
Average	25.018	26.354	26.619	25.812	27.727
Min	20.860	22.005	23.134	23.464	21.941
Max	29.239	29.759	33.353	28.628	33.695

Figure 3 and 4 shows a direct influence of the fuzziness magnitude on the solution, therefore the *Discrete Fruit Fly Optimization Algorithm (DFAO)* proofs its adaptation to the fuzzy data. In [16] the fuzzy instances derived from different volume and amplitude parameters *(a,b)* manifest different behavior for the basic heuristics *ACO* and *PSO*; in the current study with local improvements, the difference is not significant.

IV. CONCLUSIONS

The current paper is a study of several heuristics including greedy techniques and adaptive heuristics facing the data fuzziness. The *Discrete Fruit Fly Optimization Algorithm (DFAO)* is the bio-inspired method included and tested.

The *DFAO* obtains significantly good solutions closer to the CPLEX optimal; the local search included leads to a positive influence on the fuzzy data experiments. This is also the case of Lin Kernighan (LK) technique where the variations of the results show that LK can detect the modifications in data.

From the greedy techniques analyzed, $Bor\mathring{u}vka$ and Greedy are able to correctly perceive the data modifications. The results are consistent as the data sets have large dimension (between 400 and 1889) when compared with [16] (dimension less than 150).

As an overall conclusion, $Bor\mathring{u}vka$, Greedy technique and adaptive heuristics as Lin Kernighan and DFAO with local improvements are feasible choices even when uncertain data is provided. The classical implementation of DFOA is usually slow. The Lin-Kernighan method has an empirical time complexity of O($n^{2.2}$) [46]. Other drawback of DFOA is, as in many population-based methods, the parameter set: there are multiple parameters which are difficult to tailor, as there is no procedure for finding their best values.

We intend to explore the idea of maintaining a pool of very good solutions and to randomly use the update operators. Another research direction is to apply DFOA to Rich Transportation problems, as multi-attributes Vehicle Routing Problems. Further work will be done on studying other algorithms including heuristics adaptation to other degrees of fuzziness as nowadays big data and cloud data could be damaged or uncertain.

ACKNOWLEDGMENT

The research performed by G.C.C. was funded by the Romanian Ministry of Education and Research, National Council for Higher Education Funding, through grant number CNFIS-FDI-2020-0461.

REFERENCES

[1] M.J. Eppler, "Managing Information Quality: Increasing the Value of Information in Knowledge-intensive Products and Processes", 2nd Ed., Springer, 2006.

[2] I. Finocchi, F. Grandoni, and G.F. Italiano, "Designing reliable algorithms in unreliable memories algorithms", Lect. Notes Comput. Sci., vol. 3669, pp.1–8, 2005.

[3] J. von Neumann, and O. Morgenstern, "Theory of Games and Economic Behavior", Princeton University Press, 1944.

[4] L.J. Savage, "The Foundations of Statistics". New York, Wiley, 1954.

[5] M.C. Kennedy, and A. O'Hagan, "Bayesian calibration of computer models", J. Royal Stat. Soc., Series B vol. 63(3), pp. 425–464, 2001.

[6] W.E. Walker, et al. "Defining Uncertainty: A Conceptual Basis for Uncertainty Management in Model-Based Decision Support", Integrated Assessment, vol. 4(1), pp.5–17, 2003.

[7] S. Mittal, "A Survey of Techniques for Approximate Computing", ACM Computing Surveys vol. 48(4), no. 62, 33 pages, 2016.

[8] C. Malossi, et al. "The Transprecision Computing Paradigm: Concept, Design and Applications", In Proceedings DATE, pp.1105–1110, 2018.

[9] G. Beni, and J. Wang, "Swarm Intelligence in cellular robotic systems", In Proceedings NATO Adv. W. Robots & Biol. Sys., pp. 703–712, 1989.

[10] Z.B. Jiang, and Qiong Yang. "A Discrete Fruit Fly Optimization Algorithm for the Traveling Salesman Problem." *PloS one* vol. 11(11), pp.1–15, 2016.

[11] L. Huang, *et al.* "An improved fruit fly optimization algorithm for solving traveling salesman problem." Front. Inf. Tech. & Electr. Eng., vol. 18, pp. 1525–1533, 2017.

[12] Y. Li, and M. Han, "Improved fruit fly algorithm on structural optimization", Brain Inf. vol. 7(1), 2020.

[13] X. Guo, J. Zhang, W. Li, and Y. Zhang, "A fruit fly optimization algorithm with a traction mechanism and its applications", Int. J. Distrib. Sensor Networks, vol. 13(11), pp.1–12, 2017.

[14] W.J. Cook, "In Pursuit of the Traveling Salesman: Mathematics at the Limits of Computation". Princeton University Press, 2012.

[15] G.C. Crişan, C. M. Pintea, and P. C. Pop, "On the resilience of an ant-based system in fuzzy environments. An empirical study", In Proceedings FUZZ-IEEE, pp. 2588–2593, 2014.

[16] C.-M. Pintea, S. A. Ludwig, and G. C. Crişan, "Adaptability of a Discrete PSO Algorithm applied to the Traveling Salesman Problem with Fuzzy Data", In Proceedings FUZZ-IEEE, pp.1–6, 2015.

[17] R.M. Karp, "Reducibility among Combinatorial problems", In Complexity of Computer Comput. The IBM Res. Symp., R.E. Miller, J.W. Thatcher (Eds.), pp.85–103, NY: Plenum. Press, 1972.

[18] Library of sample instances for the TSP. Available at: http://comopt.ifi.uni-heidelberg.de/software/TSPLIB95/

[19] 8th DIMACS Implementation challenge: The Traveler Salesman Problem. Available at:http://dimacs.rutgers.edu/Challenges/TSP/

[20] C-M. Pintea, C. Chira, D. Dumitrescu, and P.C. Pop, "A sensitive metaheuristic for solving a large optimization problem", Lect. Notes Comput. Sci., vol. 4910, pp.551–559, 2008.

[21] C-M. Pintea, G-C.Crisan, C.Chira, "Hybrid Ant Models with a Transition Policy for Solving a Complex Problem", Logic J. IGPL, vol. 20(3), pp. 560–569, 2012.

[22] P. Jaillet, "A priori solution of a Traveling Salesman Problem in which a random set of the customers are visited", Oper. Res., vol. 36(6), pp.929–936, 1988.

[23] G.C. Crişan, and E. Nechita, "Solving Fuzzy TSP with Ant Algorithms", Int. J. Comput. Comm. & Control, vol. III, s.i., pp.228–231, 2008.

[24] R. Montemanni, J. Barta, and L.M. Gambardella, "The robust traveling salesman problem with interval data", Tech. Rep. IDSIA-20-05, 2005.

[25] Z.C. Huang, X.L. Hu, and S.D. Chen, "Dynamic Traveling Salesman Problem based on Evolutionary Computation", Congress on Evolutionary Computation, IEEE Press, pp. 1283–1288, 2001.

[26] C-M. Pintea, P.C. Pop, and D. Dumitrescu, "An Ant-based technique for the Dynamic Generalized Traveling Salesman Problem", In Proceedings Int. Conf. Sys. Theory & Sci. Comp., pp. 257–261, 2007.

[27] Z. Wang, J. Guo, M. Zheng, and Y. Wang, "Uncertain multiobjective Traveling Salesman Problem", Eur.J. Oper. Res., vol. 241(2), pp.478–489, 2015.

[28] Concorde solver. Available at: http://www.math.uwaterloo.ca/tsp/concorde.html

[29] S. Tschoke, R. Lubling, and B. Monien, "Solving the traveling salesman problem with a distributed branch-and-bound algorithm on a 1024 processor network", In Proceedings Int. Parallel Proc.Symp., pp.182–189, 1995.

[30] M. Dorigo, and T. Sttzle, "Ant Colony Optimization". MIT Press, 2004.

[31] J. Kennedy, and R.C. Eberhart, "Swarm Intelligence". Morgan Kaufmann, 2001.

[32] K. Helsgaun, "An effective implementation of the Lin-Kernighan traveling salesman heuristic". Eur. J. Oper. Res., vol. 126, pp. 106–130, 2000.

[33] J. Bang-Jensen, G. Gutin, and A. Yeo, "When the greedy algorithm fails", Discrete Optim., vol. 1(2), pp.121–127, 2004.

[34] O. $Bor\mathring{u}vka$, "Contribution to the solution of a problem of economical construction of electrical networks". Elektr. Obzor vol. 15, pp.153–154, 1926.

[35] S. Lin,, B. W. Kernighan, "An Effective Heuristic Algorithm for the Traveling-Salesman Problem." Oper. Res., vol. 21(2), pp.498–516, 1973.

[36] P. Földesi, J. Botzheim, and L.T. Kóczy, Eugenic bacterial memetic algorithm for fuzzy road transport traveling salesman problem. Int. J. Innov. Comput., Inf. & Control, vol. 7(5B), pp. 2775–2798, 2011.

[37] W.T. Pan, "A new fruit fly optimization algorithm: taking the financial distress model as an example" Know.-Based Sys., vol. 26(2), 67–74, 2012.

[38] A.C. Spanos, et. al., "A new hybrid parallel genetic algorithm for the job-shop scheduling problem." Int. Trans. Oper. Res., vol. 39(1), pp.13–26, 2013.

[39] M. Clerc, and J. Kennedy, "The Particle Swarm - explosion, stability, and convergence in a multidimensional complex space", IEEE Trans. Evol. Comput., vol. 6(1), pp.58–73, 2002.

[40] G. Gutin, A. Yeo, and A Zverovich, "Traveling salesman should not be greedy: domination analysis of greedy-type heuristics for the TSP", Discrete Appl. Math., vol. 117(1–3), pp.81–86, 2002.

[41] N. Christofides, "Worst-case analysis of a new heuristic for the travelling salesman problem", Rep. 388, CMU, 1976.

[42] D. Applegate, et.al., "Finding tours in the TSP". Tech. Rep. 99885, University of Bonn, Germany, 1999.

[43] L.A. Zadeh, "Fuzzy sets". Inf. & Control, vol. 8, pp.338–353, 1965.

[44] C.V., Negoiţă and D.A., Ralescu, "Representation theorems for Fuzzy concepts", Kybernetes, vol. 4(3), pp.169-174, 1975.

[45] G.C. Crişan, "Ant Algorithms in Artificial Intelligence". PhD Thesis, A.I. Cuza University of Iaşi, Romania, 2008.

[46] C.H. Papadimitriou, "The Complexity of the Lin–Kernighan Heuristic for the Traveling Salesman Problem", SIAM J. Comput., vol. 21(3), pp. 450–465, 1992.