Chapter 12

OFNant Method Based on TSP Ant Colony Optimization

Jacek M. Czerniak

Abstract This chapter presents a hybrid method of swarm intelligence current. Intelligence represented by ant colonies has been enriched with fuzzy logic arithmetics. In this case Kosiński's Ordered Fuzzy Numbers were specifically used. Apart from a fuzzy decision model of a single ant used earlier by other researchers, the author used the order as a trend support. By associating the direction of a number in Ordered Fuzzy Numbers (OFNs) with the trend observed in the ant colony it is possible to provide a unique description of a fuzzy observation of a colony behavior. The experiments were carried out in the area of searching for the optimal connecting route in the field. The experiment covered 10 complex issues of searching for the optimal route. All are benchmarks from the TSPlib repository which are well known among researchers. They represent the actual problems of route selection such as transport connections depending on geographic conditions and optimizing the machining process or the layout of the power networks. The complexity level of optimal solutions for problems to be solved amounted from several hundred to several thousand connections. Each of them was solved using six swarm intelligence methods and five well-known classical methods dedicated to the traveling salesman problem (TSP). The results were presented in the form of tables and graphs, and some of the routes were shown in graphical form. Final conclusions of the experiment indicate the superiority of methods based on ant colony optimization as regards closeness to optimal solutions. The results achieved by the OFNant method are generally better (in 92% of cases) than those achieved by classic methods and are in the forefront of solutions from the swarm intelligence group.

12.1 Introduction

The observation of living organisms is an interesting research field not only for biologists. A new current within artificial intelligence called swarm intelligence acquired significance in the 1990s [15]. Those studies were inspired by observation of animals

J.M. Czerniak (✉)

Kazimierz Wielki University in Bydgoszcz, Institute of Technology,
ul. Chodkiewicza 30, 85-064 Bydgoszcz, Poland
e-mail: jczerniak@ukw.edu.pl

© The Author(s) 2017

P. Prokopowicz et al. (eds.), *Theory and Applications of Ordered Fuzzy Numbers*,
Studies in Fuzziness and Soft Computing 356, DOI 10.1007/978-3-319-59614-3_12

207

and insects living in colonies [44]. We have finally got successful experiments and methods based on ant or termite colony observation [13, 20, 43, 45]. Observations of birds in V-formation inspired many researchers to create and to develop the concept of particle swarm optimization. [24]. Those studies in the field of AI were also inspired by information obtained from marine biologists on the collective intelligence of a shoal of fish or plankton. Other sources of inspiration stemmed from the development of industry, in particular the automotive industry in that case. Particle swarm optimization was created thanks to studies on, among others, sandblasting of a car body or other corroded metal parts. Hence, generally, this branch of AI has been called swarm intelligence [11, 14, 25, 38]. Conversion of those intelligence mechanisms prevailing among simple individuals into the field of computer systems resulted in creation of the current sometimes called computational swarm intelligence. It exists parallel to the branch of science called multiagent systems and those two fields often overlap one another. Although they are often not directly based on associations with colonies of living organisms, they are often similar in their rules of operation. They enable creation of interesting implementations in the domain of parallel computing. The development of swarm intelligence was preceded by the development of multiple-valued logic, in particular, fuzzy logic. The author of fuzzy logic is an American professor at Columbia University in New York City and Berkeley University in California, Lotfi A. Zadeh, who published the paper entitled "Fuzzy Sets" in the journal, *Information and Control*, in 1965 [5]. He defined the term of fuzzy set there, thanks to which imprecise data could be described using values from the interval $(0,1)$. The number assigned to them represents their degree of membership in this set. It is worth mentioning that in his theory Zadeh used the article on three-valued logic published 45 years before by a Pole, A. Janukasiewicz [6]. That is why many scientists in the world regard this Pole as the "father" of fuzzy logic. The next decades saw the rapid development of fuzzy logic. As the next milestones in the history of that discipline one should necessarily mention L-R representation of fuzzy numbers proposed by D. Dubois and H. Prade [7, 8], which enjoys great success today. Coming back to the original analogy, an observer can see a trend, that is, a general increase during a rising tide or decrease during low tide, regardless of momentary fluctuations of the water surface level. This resembles a number of macro- and micro-economic mechanisms where trends and time series can be observed. The most obvious example of that seems to be the bull and bear markets on stock exchanges, which indicate the general trend, while shares of individual companies may temporarily fall or rise. The aim is to capture the environmental context of changes in the economy or another limited part of reality. Changes in an object described using fuzzy logic [30, 32] seem to be thoroughly studied in many papers. But it is not necessarily the case as regards linking those changes with a trend [39, 41, 42]. This might be the opportunity to apply generalizations of fuzzy logic which are, in the opinion of authors of that concept, W. Kosiński [9–11] and his team [12, 13], Ordered Fuzzy Number (OFN) [28, 33, 40]. There are already interesting studies available published by well-known scientists [1, 18] that present successful implementation of fuzzy logic to swarm intelligence methods, including methods inspired by ant and termite colonies. However, according to the best knowledge of

the authors of this chapter, nobody thus far has published studies on implementation of ordered fuzzy logic into ant colony optimization. This fact was one of the reasons for execution of the research described in this chapter. The main emphasis here is on application of a new hybrid method of ant colony optimization (ACO) with implemented decision logic of an ant calculated in the OFN domain in order to solve the optimum route selection problem. To make a comparison, the authors selected several well-known ant methods and several heuristic methods dedicated to solving the same problem, the methodology of which does not use either swarm intelligence or, in particular, ACO.

12.2 Application of Ant Colony Algorithms in Searching for the Optimal Route

Ant colony optimization is currently one of the best known ant colony algorithms. It was first defined by Dorigo, Di Caro, and Gambardell in 1999 [16] as a method for discrete optimization problems. ACO was presented as the algorithm that can find a good route using a graph. It was inspired by foraging theory [14] both for ant colonies and for discrete optimization problems. This algorithm is designed for solving two kinds of static and dynamic optimization problems. In the general case, ant colony optimization is performed according to the diagram shown in Fig. 12.1). Studies of ant colony algorithms commenced based on observation of ant colony environments. The scientists noticed the interesting fact that ants communicate mainly using chemical substances which they produce. As has already been mentioned, the key matter in this

Fig. 12.1 ACO block diagram

algorithm is indirect foraging communication represented by pheromone trace. The advantage of the evaporation of that pheromone is that it can prevent convergence for local optimum solutions. Assuming there is no evaporation issue, each time each path selected by the first artificial agents would be treated in the same way and would be equally attractive, which would make it inapplicable to optimization problems. Thus, when one ant finds a good path from the colony to the food source, this path becomes preferable for other ants. The idea behind the ACO [13] algorithm is to follow that behavior using artificial agents moving within the frame of a graph in order to solve a given problem. The ACO algorithm has been used for solving the traveling salesman problem. This algorithm has an advantage over genetic algorithms or the simulated annealing algorithm. Its important feature is that for a dynamically changing graph, the ACO algorithm can work continuously and it can adapt to the changes in real-time. Thanks to such properties, it has been applied to the method of solving the problem of network routing and urban transportation systems.

Route selection: An ant shall travel the distance from point i to point j with the probability of:

$$p_{ij} = \frac{(\tau_{ij}^\alpha)(\eta_{ij}^\beta)}{\sum (\tau_{ij}^\alpha)(\eta_{ij}^\beta)} \quad (12.1)$$

where τ_{ij} is the quantity of pheromone on the route i,j , η_{ij} defines attraction of the route i,j , α is the parameter used for effect control τ_{ij} , and β is the parameter used for effect control η_{ij} .

Pheromone update: This issue is represented by the following formula.

$$\tau_{ij} = p\tau_{ij} + \Delta\tau_{ij} \quad (12.2)$$

where τ_{ij} is the quantity of pheromone on the route i,j , $\Delta\tau_{ij}$ represents the quantity of remaining pheromone, and P is the pheromone evaporation scale.

Below, we present a more detailed pseudocode of one of the numerous ant colony algorithms, called the ACS (ant colony system), that is, ant colony optimization. Tables 12.1 and 12.2 present the most important ant colony optimization algorithms dedicated to TSP in the chronological order of their publishing. In the methodological sense, all the algorithms listed below and described in the following section are direct successors of the ant system. This is due to an obvious reason, the ant system method, which has become the foundation for the entire new branch of knowledge, was the first worldwide success of then young scientist, Marco Dorigo. Now Professor M. Dorigo [14] is a world-class expert in the field of swarm intelligence. The set of methods presented below is in chronological order.

Algorithm 1 Pseudocode of the ACS Ant Colony Algorithm [10]

```

1: Initialize
2: Repeat {
3:   Place each ant in a randomly chosen city;
4:   For each ant
5:     Repeat {
6:       Choose NextCity (each ant);
7:       Update pheromone levels using a local rule;
8:     } Until (No more cities to visit);
9:   Return to the initial cities;
10:  Compute the length of the tour found by each ant;
11:  End For;
12:  Update pheromone level using a global rule;
13: }
14: Print Best Path;

```

Table 12.1 ACO algorithms that have already been applied to the TSP

ACO method	Authors
Ant System (AS)	Dorigo 1992; Dorigo, Manizzo, Colorni 1996;
Elitist AS (EAS)	Dorigo 1992; Dorigo, Manizzo, Colorni 1996;
Ant-Q (AQ)	Gambardella, Dorigo 1995–96;
Ant Colony System (ACS)	Dorigo, Gambardella 1997;
Max-Min AS (MMAS)	Sttzle 1999; Sttzle, Hoos 2000;
Rank-base AS (ASrank)	Bullnheimer, Hartl, Strauss 1997–99;

12.3 OFNAnt, a New Ant Colony Algorithm

Implementation of OFN to the ant colony system consists mainly in determination of the trend and in establishing relationship to the order of the OFN. This order is used in OFNAnt in two ways. It is related to pheromone evaporation on the route and its mathematical description, and it also concerns the decision-making process of a single ant.

The pheromone quantity on the route is updated in accordance with OFN arithmetic. If the pheromone trace (quantity) on the route increases, then this trend is marked as a positive order trend, whereas if this quantity decreases it is marked as a negative order trend. Each pass of the k th ant, which is associated with placing pheromone trace results in the update of the pheromone trace on the route by the amount left by the ant resulting in positive order on the route with increasing trend and with negative order for decreasing trend of the route. The above relationship is pursuant to the formula:

$$\tau_{ij}[l_A, 1_A^-, 1_A^+, p_A] \leftarrow \tau_{ij}[l_A, 1_A^-, 1_A^+, p_A] + \sum_{k=1}^m \Delta \tau_{ij}^k[l_k, 1_k^-, 1_k^+, p_k] \quad (12.3)$$

Table 12.2 List of analyzed problems including their optimum values

No	Problem designation	Optimum	Description	Author
1	Eil51	426	Problem for 51 towns	Christofides / Eilon
2	D198	15780	Represents the Dribling Problem. Size of the problem: 198 holes	Reinelt
3	Gil262	2378	Problem for 262 towns	Gillet/Johnson
4	Lin318	42029	Problem for 318 towns	Lin/Kernighan
5	Pcb442	50778	Represents the Dribling Problem. Size of the problem: 442 holes	Groetschel/Juenger/Reinelt
6	Rat783	8806	Problem of 783 points connected to the power network	Pulleyblank
7	Pcb1173	56892	Represents the Dribling Problem. Size of the problem: 1173 holes	Juenger/Reinelt
8	D1291	50801	Represents the Dribling Problem. Size of the problem: 1291 holes	Reinelt
9	Nrw1379	56638	The problem for 1379 towns/villages in North Rhine-Westphalia	Bachem/Wottawa
10	Pr2392	378032	Problem for 2392 towns	Padberg/Rinaldi

Every ant constructs a complete route, and the ants make a decision at each stage of the route construction. This creates a multistage process of fuzzy control. When talking about route construction, we usually refer to the situation when an ant located in town i wants to go to town j and makes a decision based on the following information.

1. Parameters defining the effect of the pheromone trace τ_{ij}^α .
2. (b) Parameters defining the effect of heuristic information η_{ij}^β , used to estimate attraction of the route.
3. N_i^k Parameter representing the list of k available neighbors of an ant. The "available" neighbors mean the towns that have not been visited yet.

The decision-making process taken by an ant at each node of the route is associated with calculation of fuzzy probability in the OFN sense. The probability is calculated pursuant to the following redefined formula of the route selection probability (Fig. 12.2).

$$P_{ij}^k[l_A, 1_A^-, 1_A^+, p_A] = \frac{|\tau_{ij}[l_A, 1_A^-, 1_A^+, p_A]|^\alpha |\eta_{ij}[l_A, 1_A^-, 1_A^+, p_A]|^\beta}{\sum_{l \in N_i^k} |\tau_{ij}[l_k, 1_k^-, 1_k^+, p_k]|^\alpha |\eta_{ij}[l_k, 1_k^-, 1_k^+, p_k]|^\beta} \quad (12.4)$$

Fig. 12.2 An ant located in the town i selects next town j

12.4 Experiment

12.4.1 Experiment Execution Method

In this section, the author compares the effectiveness of heuristic methods, meta-heuristic methods, and the new hybrid method OFNAnt. All those methods are tested using 10 benchmarks for their performance in solving an NP-hard problem such as TSP. Thus, it is a comparison of well-known algorithms with a completely new approach represented by OFN arithmetics implemented to control an ant colony in order to solve optimizing problems [21, 22, 27]. They are tested according to the following principles.

1. As regards ant colony algorithms, a program with implemented method is run three times at $t=10$ for each problem, and for implementation of heuristic algorithms a program is also run three times, but without additional parameters.
2. Effectiveness of a given algorithm is assessed as follows:
 - By specification of the obtained result (route length)
 - As a percentage, that is, optimum achieved in $x\%$, as presented in the table including the set of benchmarks
3. A graph showing the effectiveness of individual algorithms is presented for each of the 10 problems.
4. Each such graph is provided with a short summary where the obtained results are discussed.
5. An overall graph showing the effectiveness of all algorithms is presented at the end. The value optimum achieved in $x\%$ is totaled for each algorithm, and thus the overall score per 1,000 available points is calculated. Such a data presentation

allows easy assessment of the hierarchy of all the algorithms on the basis of the 10 benchmarks used for tests.

12.4.2 Software Used for Experiment

The author's own implementation of ant colony methods developed in JAVA language was used in the experiments and the results obtained by the implementation were verified on the basis of ACOTSP [14]. The author's OFNAnt method was added to the implementation. The CONCORDE application was developed to solve symmetric TSP-type problems and other problems of network optimization [3, 17]. The application is supported by the Office of Naval Research, National Science Foundation, and by the School of Industrial and System Engineering at the Georgia Institute of Technology, United States. This program uses the cutting planes algorithm. The interface of the program shows the optimum solution searching process displayed at the end of each main iteration. The edges are colored according to currently calculated LP value (linear programming relaxation). At the moment when a new, better solution is found, the color of edges is changed to red. The program includes several algorithms designed to create edges used by the program to search for the optimum solution. Those algorithms include:

1. Delaunay triangulation
2. Minimum spanning tree
3. Different variations of nearest neighbors

The program also includes several heuristic [2] algorithms for the TSP problem. Those algorithms include:

1. Greedy algorithm (GR)
2. Boruvka algorithm (BOR)
3. Quick Boruvka algorithm (QBOR)
4. Nearest neighbor algorithm (NN)
5. Lin-Kernighan algorithm (LK)

12.4.3 Experimental Data

Table 12.3 shows 10 benchmarks selected from the TSPLib library of TSP problems, including the expected optimum value for each. They were applied in a way described in the previous paragraph as a set of benchmarks for testing well-known algorithms and a new OFNAnt method.

Table 12.3 List of analyzed problems including their optimum values

Data sets		ACOTSP				Concorde TSP		
*.tsp file	optimum	AS	ASRK	OFNAnt	ACS	GR	QBOR	LK
eil51	426	426	426	426	426	521	480	426
%		100,00	100,00	100,00	100,00	81,77	88,75	100,00
d198	15780	15781	15780	15780	15780	18399	18140	15828
%		99,99	100,00	100,00	100,00	85,77	86,99	99,70
gil262	2378	2380	2378	2378	2378	2846	2818	2380
%		99,92	100,00	100,00	100,00	83,56	84,39	99,92
lin318	42029	42091	42029	42029	42029	49744	54090	42272
%		99,85	100,00	100,00	100,00	84,49	77,70	99,43
pcb442	50778	50964	50883	50778	50778	61891	58695	51071
%		99,64	99,79	100,00	100,00	82,04	86,51	99,43
rat783	8806	8833	8812	8808	8806	10294	10402	8831
%		99,69	99,93	99,98	100,00	85,54	84,66	99,72
pcb1173	56892	57612	56950	57040	56897	65829	66493	57063
%		98,75	99,90	99,74	99,99	86,42	85,56	99,70
d1291	50801	51020	50824	50870	50820	59293	57228	52729
%		99,57	99,95	99,86	99,96	85,68	88,77	96,34
nwr1379	56638	57281	56859	56917	56770	66371	66110	56756
%		98,88	99,61	99,51	99,77	85,34	85,67	99,79
pr2392	378032	386541	382089	381077	379602	444853	448641	383277
%		97,80	98,94	99,20	99,59	84,98	84,26	98,63

12.5 Results of Experiment

A number of tests were performed according to the above-specified rules, using 10 selected problems. Results of individual tests are presented below assisted by the diagram and a brief note for each.

Eil51 The authors presented an experiment for 51 towns, the optimum value for which amounts to 426. As a result of the calculations 7 out of 10 algorithms generated an optimum result. It is worth noting that only one classical algorithm (ALK) generated the best result. Other ALK algorithms showed low effectiveness for a relatively small problem. All ant colony algorithms showed excellent performance when solving the above problem (Fig. 12.3).

D198 Another test was performed for 198 towns and it showed the advantage of ant colony algorithms (ALM) over ALK algorithms again. This time 5 out of 10 available ALM algorithms have found the optimum, but AS achieved a result only 0.01% worse than the optimum. The best ALK, namely Lin-Kernighan achieved 99.70% of the optimum, which is quite a good result. Other ALK algorithms achieved only 88% of the optimum value (Fig. 12.4).

Fig. 12.3 Graphical representation of the results for the problem Eil51

Fig. 12.4 Graphical representation of the results for the problem D198

Pcb442 represents the problem covering 442 towns. It is the hardest problem thus far, because only 3 out of 10 tested algorithms, BWAS, EAS, and ACS have found the optimum. MMAS missed the best solution by only 0.01%. Leading algorithms as regards this problem also include, respectively: ASRK with the result of 99.79%, AS 99.64%, and LK 99.43%. As can be noted, 3 ALM algorithms achieved optimum, one missed the optimum by the skin of its teeth, then, two further ALM algorithms achieved very good results and, again, the best of the ALK algorithms, that is, LK was the last on the list. The remaining ALK algorithms performed even worse than for the problem with 318 towns and achieved from 80 to 87% of the optimum (Fig. 12.5).

Rat783 represents the problem covering 783 towns. In this case 3 algorithms achieved optimum solution, namely EAS, MMAS, and ACS; BWAS missed the optimum solution by 0.02%, and ASRK missed it by 0.07%. For the first time we have the situation where the ALK algorithm, that is, LK with the score of 99.72% outdistanced a representative of ALM algorithms, the AS algorithm with the score of 99.69%. Other ALK algorithms were unrivaled (Fig. 12.6).

Pr2392 The last of the 10 presented problems was also the biggest one as it included as many as 2,392 towns. There is no doubt that the bigger the problem is, the worse the solutions. The first four places on the list were taken by ALM algorithms, where ACS was the best with the score of 99.59%. One of the ALK algorithms,

Fig. 12.5 Graphical representation of the results for the problem Pcb442

Fig. 12.6 Graphical representation of the results for the problem Rat783

namely LK, was fifth on the list with the score of 98.63%. Two subsequent places on the list were taken by ALM algorithms, that is, by EAS with the score of 98.39% and by AS with the score of 97.80%. The remaining ALK algorithms followed the trend of worse solutions and with the increased problem complexity they achieved from 79 to 85% of the optimum.

12.6 Summary and Conclusions

Having performed a number of experiments according to the rules specified above, one can be certain about the superiority of ant colony algorithms over classical algorithms (Fig. 12.7). There was only one case out of 10 studied samples, where the Lin-Kernighan (LK) algorithm achieved better results than all other known methods, including ant colony methods. This could have resulted from the nature of the problem, that is, nrw1379. In that case the results obtained by the LK algorithm were only slightly worse than the results of the OFNAnt algorithm. For the remaining files, the LK algorithm outpaced, at best, only older ant colony methods, AS and EAS.

Fig. 12.7 Graphical representation of the results for the problem Pr2392

The remaining algorithms from the group of heuristic methods performed definitely much worse than the leading algorithms. They fulfilled the optimum solution within the range from 79 to 89%, which is far from the results of the leading algorithms. The noticeable feature of the studied group of algorithms is their tendency for worse results with the increase of problem magnitude. A clear example of that tendency is the seventh tested problem, pcb1173. This statement is confirmed by problem d1291 and further large datasets. The diagram presented above, which summarizes all performed tests, shows the hierarchy of all algorithms and their respective scores. The maximum available score is 1,000 points. The scores closest to the maximum were achieved by representatives of ant colony algorithms, including OFNAnt with the score of 999.31 points. It is worth noting that the first four places on the list of optimum solution searching efficiency are taken by ant colony algorithms. Subsequent places on the list are taken by representatives of heuristic methods with their definite leader, the LK algorithm, which is widely regarded as one of the best methods for solving the traveling salesman problem. Ant colony algorithms represent a new generation of optimizing algorithms using a metaheuristic approach to NP-hard problems, the approach that gave excellent results. Ant colony algorithms find many more applications other than TSP. Those applications include many real-life fields. Based on the results of experiments with the new method using trend and fuzzy logic, one can also expect obtaining interesting solutions for problems other than those where ant colonies have already been successfully applied. The new method, OFNAnt, which is a hybrid combination of ACO and OFN, and introduces fuzzy decision of an ant, is the first known attempt to implement the arithmetic of Ordered Fuzzy Numbers to ant colony optimization. Performed experiments confirmed efficiency of that method in solving TSP problems. Currently, there are ongoing works on application of the modification of that method for solving problems of other classes.

Acknowledgements The authors would like to express their thanks to employees of AIRlab - Artificial Intelligence and Robotics Laboratory at Casimir the Great University in Bydgoszcz for their commitment and help during research and tests performed within this study.

References

1. Angryk, R.A., Czerniak, J.: Heuristic algorithm for interpretation of multi-valued attributes in similarity-based fuzzy relational databases. *Int. J. Approx. Reason.* **51**(8), 895–911 (2010)
2. Apiecionek, L., Czerniak, J.M.: Qos solution for network resource protection. In: *Informatics 2013: Proceedings of the Twelfth International Conference on Informatics*, pp. 73–76 (2013)
3. Apiecionek, L., Czerniak, J.M., Dobrosielski, W.T.: Quality of services method as a DDOS protection tool. In: *Intelligent Systems*, pp. 225–234 (2014), (vol. 2: Tools, Architectures, Systems, Applications 323 (2015))
4. Apiecionek, L., Czerniak, J.M., Zarzycki, H.: Protection tool for distributed denial of services attack. Beyond databases, architectures and structures. *BDAS* **424**, 405–414 (2014)
5. Czabanski, R., Jezewski, J., Horoba, K., Jezewski, M.: Fetal state assessment using fuzzy analysis of fetal heart rate signals – agreement with the neonatal outcome. *Biocyber. Biomed. Eng.* **33**(3), 145–155 (2013)
6. Czabanski, R., Jezewski, M., Horoba, K., Jezewski, J., Leski, J.: Fuzzy analysis of delivery outcome attributes for improving the automated fetal state assessment. *Appl. Artif. Intel.* **30**(6), 556–571 (2016)
7. Czerniak, J., Zarzycki, H.: Application of rough sets in the presumptive diagnosis of urinary system diseases. *Artif. Intel. Secur. Comput. Syst.* **752**, 41–51 (2003)
8. Czerniak, J.: Evolutionary approach to data discretization for rough sets theory. *Fundam. Inform.* **92**(1–2), 43–61 (2009)
9. Czerniak, J.M., Apiecionek, L., Zarzycki, H.: Application of ordered fuzzy numbers in a new ofnant algorithm based on ant colony optimization. Beyond databases, architectures and structures. *BDAS* **424**, 259–270 (2014)
10. Czerniak, J.M., Dobrosielski, W., Zarzycki, H., Apiecionek, L.: A proposal of the new owlant method for determining the distance between terms in ontology. In: *Intelligent Systems*, pp. 235–246 (2014). (vol 2: Tools, Architectures, Systems, Applications 323 (2015))
11. Czerniak, J., Apiecionek, L., Zarzycki, H., Ewald, D.: Proposed caeva simulation method for evacuation of people from a buildings on fire. *Adv. Intel. Syst. Comput.* **401**, 315–326 (2016)
12. Czerniak, J., Dobrosielski, W., Apiecionek, L.: Representation of a trend in ofn during fuzzy observance of the water level from the crisis control center. In: *Proceedings of the Federated Conference on Computer Science and Information Systems*, IEEE Digital Library, ACSIS **5**, 443–447 (2015)
13. Czerniak, J., Macko, M., Ewald, D.: The cutmag as a new hybrid method for multi-edge grinder design optimization. *Adv. Intel. Syst. Comput.* **401**, 327–337 (2016)
14. Czerniak, J., Smigielski, G., Ewald, D., Paprzycki, M.: New proposed implementation of abc method to optimization of water capsule flight. In: *Proceedings of the Federated Conference on Computer Science and Information Systems*, IEEE Digital Library, ACSIS **5**(489–493) (2015)
15. Dorigo, M., Stutzle, T.: *Ant Colony Optimization*. MIT Press (2003)
16. Dubois, D., Prade, H., Richard, G.: Multiple-valued extensions of analogical proportions. *Fuzzy Sets Syst.* **292**, 193–202 (2016). <http://www.sciencedirect.com/science/article/pii/S0165011415001682>. (Special Issue in Honor of Francesc Esteva on the Occasion of his 70th Birthday)
17. Ewald, D., Czerniak, J.M., Zarzycki, H.: Approach to solve a criteria problem of the ABC algorithm used to the WBDP multicriteria optimization. In: *Intelligent Systems*, 129–137 (2014). (vol. 1: Mathematical Foundations, Theory, Analyses 322 (2015))

18. Jezewski, M., Czabanski, R., Horoba, K., Leski, J.: Clustering with pairs of prototypes to support automated assessment of the fetal state. *Appl. Artif. Intel.* **30**(6), 572–589 (2016)
19. Jezewski, M., Leski, J.M., Czabanski, R.: Classification Based on Incremental Fuzzy (1 + p)-Means Clustering, pp. 563–572. Springer International Publishing, Cham (2016)
20. Kacprzak, M., Kosiński, W.: On lattice structure and implications on ordered fuzzy numbers. In: *Proceedings of EUSFLAT. Artificial Intelligence and Soft Computing. LNCS*, vol. 7267, pp. 247–255 (2011)
21. Kacprzak, M., Starosta, B., Węgrzyn-Wolska, K.: Metasets and opinion mining in new decision support system. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L.A., Zurada, J.M. (eds.) *Artificial Intelligence and Soft Computing, Part II. Lecture Notes in Artificial Intelligence*, vol. 9120, pp. 625–636. Springer International Publishing, Berlin (2015)
22. Kacprzak, M., Starosta, B., Węgrzyn-Wolska, K.: New approach to decision making. In: Abraham, A., Węgrzyn-Wolska, K., Hassani, A.E., Snasel, V., Alimi, A.M. (eds.) *Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015. Advances in Intelligent Systems and Computing*, vol. 427, pp. 397–407. Springer International Publishing, Berlin (2015)
23. Kacprzyk, J., Wilbik, A.: Using fuzzy linguistic summaries for the comparison of time series: an application to the analysis of investment fund quotations. In: *IFSA/EUSFLAT Conference*, pp. 1321–1326 (2009)
24. Kacprzyk, J., Wilbik, A., Zadrożny, S.: Linguistic summarization of time series using a fuzzy quantifier driven aggregation. *Fuzzy Sets Syst.* **159**(12), 1485–1499 (2008)
25. Kacprzyk, J., Wilbik, A., Zadrożny, S.: On some types of linguistic summaries of time series. In: *2006 3rd International IEEE Conference on Intelligent Systems*, pp. 373–378. IEEE, New Jersey (2006)
26. Klir, G.J.: Chapter 2 fuzzy logic. In: M. Nikravesh, F.A., Zadeh, L. (eds.) *Soft Computing and Intelligent Data Analysis in Oil Exploration, Developments in Petroleum Science*, vol. 51, pp. 33–49. Elsevier, Amsterdam (2003). <http://www.sciencedirect.com/science/article/pii/S0376736103800067>
27. Kosinski, W., Chwastyk, A.: Ordered fuzzy numbers in financial stock and accounting problems. In: *Proceedings of the 2013 Joint IFSA World Congress And Nafips Annual Meeting (IFSA/NAFIPS)* pp. 546–551 (2013)
28. Kosiński, W., Prokopowicz, P., Rosa, A.: Defuzzification functionals of ordered fuzzy numbers. *IEEE Trans. Fuzzy Syst.* **21**(6), 1163–1169 (2013). doi:[10.1109/TFUZZ.2013.2243456](https://doi.org/10.1109/TFUZZ.2013.2243456)
29. Kosinski, W., Prokopowicz, P., Slezak, D.: Fuzzy reals with algebraic operations: algorithmic approach. In: *Proceedings of the Intelligent Information Systems 2002*, pp. 311–320 (2002)
30. Kosinski, W., Prokopowicz, P., Slezak, D.: Calculus with fuzzy numbers. In: Bolc, L., Michalewicz, Z., Nishida, T. (eds.) *Intelligent Media Technology For Communicative Intelligence. Lecture Notes in Artificial Intelligence*, vol. 3490, pp. 21–28 (2004). (2nd International Workshop on Intelligent Media Technology for Communicative Intelligence, Warsaw, POLAND, Sep 13–14, 2004)
31. Kosiński, W.: Evolutionary algorithm determining defuzzification operators. *Eng. Appl. Artif. Intell.* **20**(5), 619–627 (2007). <http://www.sciencedirect.com/science/article/pii/S0952197607000413>
32. Kosinski, W., Prokopowicz, P.: Fuzziness - representation of dynamic changes? In: Stepnicka, M., Novak, V., Bodenhofer, U. (eds.) *New Dimensions In Fuzzy Logic And Related Technologies, Proceedings*, vol. I, pp. 449–456. (European Soc Fuzzy Log and Technol, Univ Ostrava, Ostravska Univ & Ostrave, Dvorakova 7, Ostrava 1, 701 03, CZECH REPUBLIC (2007), 5th Conference of the European-Society-for-Fuzzy-Logic-and-Technology, Ostrava, Czech Republic, Sep 11–14, 2007)
33. Kosinski, W., Prokopowicz, P., Slezak, D.: Neural Networks and Soft Computing: *Proceedings of the Sixth International Conference on Neural Networks and Soft Computing, Zakopane, Poland, June 11–15, 2002. Chap. On Algebraic Operations on Fuzzy Reals*, pp. 54–61. Physica-Verlag HD, Heidelberg (2003). http://dx.doi.org/10.1007/978-3-7908-1902-1_8

34. Leski, J.M.: Neuro-fuzzy system with learning tolerant to imprecision. *Fuzzy Sets Syst.* **138**(2), 427–439 (2003)
35. Leski, J.M.: On support vector regression machines with linguistic interpretation of the kernel matrix. *Fuzzy Sets Syst.* **157**(8), 1092–1113 (2006)
36. Mikolajewska, E., Mikolajewski, D.: Exoskeletons in neurological diseases - current and potential future applications. *Adv. Clin. Exper. Med.* **20**(2), 227–233 (2011)
37. Piegat, A., Pluciński, M.: Computing with words with the use of inverse rdm models of membership functions. *Int. J. Appl. Math. Comput. Sci.* **25**(3), 675–688 (2015)
38. Prokopowicz, P.: Methods based on ordered fuzzy numbers used in fuzzy control. In: *Proceedings of the Fifth International Workshop on Robot Motion and Control*, 2005. RoMoCo '05. pp. 349–354 (2005). <http://dx.doi.org/10.1109/ROMOCO.2005.201448>
39. Prokopowicz, P.: Adaptation of rules in the fuzzy control system using the arithmetic of ordered fuzzy numbers. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L., Zurada, J. (eds.) *Artificial Intelligence and Soft Computing - ICAISC 2008, Lecture Notes in Computer Science*, vol. 5097, pp. 306–316. Springer, Berlin (2008). http://dx.doi.org/10.1007/978-3-540-69731-2_30
40. Prokopowicz, P.: Flexible and simple methods of calculations on fuzzy numbers with the ordered fuzzy numbers model. In: Rutkowski, L., Korytkowski, M., Scherer, R., Tadeusiewicz, R., Zadeh, L., Zurada, J. (eds.) *Artificial Intelligence and Soft Computing. Lecture Notes in Computer Science*, vol. 7894, pp. 365–375. Springer, Berlin (2013). http://dx.doi.org/10.1007/978-3-642-38658-9_33
41. Prokopowicz, P.: Analysis of the changes in processes using the kosinski's fuzzy numbers. In: Ganzha, M., Maciaszek, L., Paprzycki, M. (eds.) *Proceedings of the 2016 Federated Conference on Computer Science and Information Systems. Annals of Computer Science and Information Systems*, vol. 8, pp. 121–128. IEEE, New Jersey (2016). <http://dx.doi.org/10.15439/2016F140>
42. Prokopowicz, P.: *Proceedings of the Second International Afro-European Conference for Industrial Advancement AECIA 2015*, chap. The Directed Inference for the Kosinski's Fuzzy Number Model, pp. 493–503. Springer International Publishing, Cham (2016). http://dx.doi.org/10.1007/978-3-319-29504-6_46
43. Sobol, I., Kacprzak, D., Kosiński, W.: Optimizing of a company's cost under fuzzy data and optimal orders under dynamic conditions. *Optimum. Studia Ekonomiczne* **5**, 172–187 (2014)
44. Soleymani, T., Trianni, V., Bonani, M., Mondada, F., Dorigo, M.: Bio-inspired construction with mobile robots and compliant pockets. *Robot. Autonom. Syst.* **74**, 340–350 (2015)
45. Valentini, G., Brambilla, D., Hamann, H., Dorigo, M.: Collective perception of environmental features in a robot swarm. In: *Swarm Intelligence – Proceedings of ANTS 2016 – Tenth International Conference. Lecture Notes in Computer Science*, vol. 9882, pp. 65–76. Springer, Berlin (2016)
46. Wei, L., Keogh, E.: Semi-supervised time series classification. In: *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. pp. 748–753. ACM, New York (2006)
47. Xi, X., Keogh, E., Shelton, C., Wei, L., Ratanamahatana, C.A.: Fast time series classification using numerosity reduction. In: *Proceedings of the 23rd International Conference on Machine Learning*. pp. 1033–1040. ACM, New York (2006)
48. Zadeh, L.: Fuzzy sets. *Inf. Control* **8**(3), 338–353 (1965). <http://www.sciencedirect.com/science/article/pii/S001999586590241X>
49. Zadrozny, S., Kacprzyk, J.: On the use of linguistic summaries for text categorization. In: *Proceedings of IPMU*. pp. 1373–1380 (2004)
50. Zarzycki, H.: Computer system for the evaluation of options contracts with monte carlo approach [in polish]. *Stud. Proc. Pol. Assoc. Knowl. Manag.* **22**, 226–233 (2009)

51. Zarzycki, H.: Application of the finite difference method to value derivatives. Stud. Proc. Pol. Assoc. Knowl. Manag. **42**, 267–277 (2011)
52. Zarzycki, H.: Modern technologies as a chance for options markets development (in polish). In: National Scientific Conference Financial Markets in Electronic Space (Kulice 2003)
53. Zarzycki, H.: Index arbitrage on the Warsaw stock exchange [in polish]. VII Technical University of Szczecin Computer Science Department Symposium (Szczecin 2003)

Open Access This chapter is licensed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license and indicate if changes were made.

The images or other third party material in this chapter are included in the chapter's Creative Commons license, unless indicated otherwise in a credit line to the material. If material is not included in the chapter's Creative Commons license and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder.