Multiple Criteria Decision Making Problems and Neural Networks

Leoneed M. Kirilov[1], Stefan L. Koynov[1] and Hristo I. Toshev[1]

Abstract – **In this paper a formulation and classification of multiple criteria decision making problems (MCDMP) is given. Some aspects of their solving are discussed. A number of different methods are previewed. Attention is given to the neural network approach.**

Keywords – **multiple criteria, neural networks, optimization.**

I. Multiple Criteria Decision Making Problems (MCDMP)

MCDMP have been studied intensively since the 70s. Note that Kuhn and Tucker first proved mathematical properties of multiple objective mathematical programming problems (see below) in the 50s.

Depending on the type of decision variables discrete (in particular integer) and continuous MCDMP are known. Depending on the type of functions used in a model we consider linear and nonlinear MCDMP. And finally, depending on the number of alternatives we consider MCDMP with finite or infinite number of alternatives. The problems with finite number of alternatives are known as Multiple Attribute Decision Making Problems (MADMP) – [1]. These problems can be formulated in a matrix form, where the rows represent the alternatives and the columns represent the objective values over the set of alternatives.

The other classes of MCDMP (integer, linear etc.) can be summarized as Multiple Objective Mathematical Programming Problems (MOMPP). Note that discrete MCDMP have a finite (but unknown) number of alternatives unlike continuous MCDMP. On the other hand the alternatives in discrete MCDMP are not explicitly given unlike MADMP.

MOMPP are presented in [2]-[4].

The general formulation of MOMPP is in the form:

$$\begin{aligned}&\text{“max”}\ (f_1(x), \dots , f_k(x))\\ &\text{s.t.}\\ &x \in S: g_j(x) \le 0, \text{ for } j = 1,2, \dots, m\end{aligned} \tag{1}$$

where

$x = ((x_1, \dots , x_n) \in R^n$ is the vector of decision variables

$fi(.), gj(.) : R^n \to R^1$ are real valued functions for i and j resp.

The symbol “max” means that all the objectives have to be maximized simultaneously.

[1] The authors are with the Inst. of Information Technologies – Bulgarian Acad. of Sciences, Acad. G. Bonchev Str., Bl.29A, 1113 Sofia, Bulgaria, E-mails: Lkirilov@iinf.bas.bg, Toshev@iinf.bas.bg, Baruch@iinf.bas.bg

The main difficulties in solving MCDMP can be viewed as follows.

By considering conflicting objectives and excluding the fact of existing a solution optimizing all the objectives we receive a set of equally good optimal solutions. They are known as efficient solutions. This is connected with using partial orders in a k-dimensional objective space for comparing objective vectors. The most common used orders are pareto and weak pareto orders. Consequently pareto and weak pareto efficient points are defined

Definition 1.1 A solution x is (pareto) efficient if and only if there does not exist another solution y such that

$$\begin{aligned}&f_i(x) \le f_i(y) \text{ for } i = 1, 2, \dots, k\\ &\text{and}\\ &f_j(x) < f_j(y) \text{ for at least one index } j \in I = \{1, 2, \dots , k\}\end{aligned} \tag{2}$$

Definition 1.2 A solution x is weak (pareto) efficient if and only if there does not exist another solution y such that

$$f_i(x) < f_i(y) \text{ for } i = 1, 2, \dots, k \tag{3}$$

In other words the set of (weak) efficient solutions is the set of maximal points according to the chosen order.

The corresponding vector f(x) is called nondominated vector.

The above definitions are true for MADMP also.

The MCDMP is solved when one solution is chosen – *final, best compromise solution*. Due to the nature of efficient set we need additional information to narrow it. Usually such information is given by the person – the expert/decision maker (DM).

Let us assume that the DM can express his/her preferences in an analytical form, i.e. there exists a real-valued function u(.): $R^k \rightarrow R^1$ such that if the solution x is preferred to the solution y by the DM then u(x) > u(y).

Then the above problem is simply converted into the usual single objective problem

$$\begin{aligned}&\max u(x)\\ &x \in S\end{aligned} \tag{4}$$

Two questions arise about the utility function. How to define it in an analytical form and when it is possible, i.e. when does the utility function exist?

It is proven that there are some cases in which such function does not exist. Further, in case of existing of such function, it is difficult to write it in an analytical form – [5]-[7].

Therefore, methods for direct solving MODMP were developed mainly.

II. Basic Approaches for Solving MCDMP

MCDM methods have two phases: a phase of computation and a phase of dialog. A number of solutions are generated in the computation phase. The solutions are evaluated by the DM in the phase of dialog. There he/she sets his/hers aspirations, too.

If the phase of dialog is before the computation phase then the method is called an *apriori* method. Otherwise it is called an *aposteriori* method. If the two phases follow each other in a cyclic mode then the method is called *interactive*.

The practice and the theory showed as most promising the group of interactive methods and they are the biggest part of proposed methods.

The DM can set his/her aspirations in different ways – as levels or goals in the objective space, by comparing two or more solutions, by giving trade-offs between objectives etc.

Usually a single criterion problem is solved in the computation phase. It is constructed on the base of primary MCDMP and the preferences given by the DM.

One MCDMP method realizes a strategy for sampling the efficient set. A short list of interactive methods according to the used strategy follows.

For example in STEM (Step Method, method of constraints) the objectives are divided in two classes: to be improved and to be relaxed to the fixed values. Then a solution is found optimizing the first group of objectives at each iteration – [8]. Other well known methods are VIG of Korhonen-Laakso [10] and Pareto Race – Korhonen and Wallenius [9] for linear MOMPP. They generate a number of solutions at each iteration using a linear parametric method. A reference point approach of Wierzbicki [11] generates solutions according to the preferences of the DM in terms of reference points in the objective space. The reference direction method for nonlinear MOMPP [12], [13] projects a reference direction onto the nondominated surface. A cone contracting method of Steuer [15] and an augmented Tchebysheff procedure of Steuer and Choo [14] reduce the cone of objectives and the set of weighting coefficients resp. The latter approach is also used in Kirilov - Vassilev [16] and Zionts - Walenius [17].

As it is seen the proposed methods differ in their approach for solving MCDMP. But what kind of final solution did they find? Is the solution optimal for the utility function? In other words can we talk about a convergence of a particular method like in a single objective optimization? The latter problem is not solved up to now.

Another view of approaches for solving MCDMP is – one can find a solution in an apriori given number of iterations. The question is does the method give the DM a possibility of a full understanding of the solved problem (learning effect) for the fixed number of steps? If yes, then the final solution is an optimal solution of its utility function. The other approach gives the DM the possibility to move free in efficient set. Then the DM decides when to stop the method. In this case we talk about behavioral convergence. The question here is can the method generate all efficient solutions or its subset in a reasonably few number of iterations. This task is studied more extensively and most methods from the second group can generate the whole efficient set. But the problem when to stop still remains.

Note that in both cases some kind of surrogate (single objective) problem is solved at each (multi-objective) iteration. Thus to the complexity of a chosen single objective problem is added the multiple-objective complexity.

Therefore in spite of existing a variety of methods for solving MCDMP new techniques for solving are looked for.

One of them is the neural network approach.

III. Neural Networks as a Tool for Solving Problems

The **ANN features** include [26]: *advantages* (convenient for ill-defined tasks with implicit algorithm formulations, iterative modeling - structural and algorithmic, adaptivity, stability for a fuzzy and noisy input, robustness); *disadvantages* (curse of dimensionality; design problems; choice of the energy function; time for learning; small input changes lead to a similar, but new training; unpredictable behavior; *asymptotic* convergence with classification problems); *ANN classification* (feedforward ANN and recurrent ANN) and *learning* (learning paradigm, learning rule, ANN architecture, learning algorithm). The learning rule and the ANN architecture are united around the learning paradigm which can be supervised, unsupervised and hybrid. The learning algorithm is a realization of the mathematical description for the concrete learning rule with respect to the ANN architecture. Respectively the **ANN tasks** can be: *basic* (function approximation, coding and decoding with ANN, data and image compression, modeling), *tasks for dynamic systems* (identification, modeling, optimization, control), *tasks for recognition* (image and speech recognition). The ANN tasks for dynamic systems can be modeled with time series (generation, prediction). It is evident that the set of properties including the wide range of the applications is a source of principal troubles during the processes of the design, the choice of parameters, the mathematical description, the optimization and the initialization of ANN.

New techniques are continuously attracted from other AI domains for improving the properties of the artificial neural networks (ANN). One of them is for example the evolutionary computation (EC).

Some of the first efforts at applying EC to optimizing ANN can be found in [27]-[28]. More recent research has involved simultaneously evolving both the structure and weights of feed forward and recurrent ANN [29]-[31]. Some attention has also being given to evolving fuzzy ANN in which classification of input patterns are made with respect to their fuzzy membership in evolved clusters [32].

The speculation that ANN could be optimized using simulated evolution goes back at least to [33]. The aspects of the cooperation between the EC and ANN include: ANN design [34]-[37], ANN training [38], [34]-[37], [39] and genetic synthesis and ANN optimization [40].

[41]-[43], [25] comprise a review of the modern ANN applications. The **ANN modern applications** include: *language learning* (inborn abilities or 'phylogeny', acquired abilities or 'epigenesis', [41]) and *ANN populations* (evolution on the global level or 'phylogeny', learning on an individual level or 'epigenesis', [42-43], [25]).

IV. NEURAL NETWORKS FOR SOLVING MCDMP

Quite a few approaches for solving the MCDM basic problems on the base of the ANN are developed up to now.

In [18] a number of methods for solving MOMPP are presented. The general idea is to convert the current MOMPP model into a single criterion one and after that this problem to be solved by a suitable ANN. Authors give a list of such surrogate problems: mini-max problem and shifted mini-max problem resp., weighting problem, Lp-problem and goal programming problem.

These methods belong to the group of the apriori methods, according to the classification in section II. They need hard dialog with the DM. And the main disadvantage of such approach is how precise is the dialog, is it user-friendly.

Naturally if the DM does not accept the computed solution, the methods need restarting.

Another view of the above scalarising problems is that they can be interpreted as an approximation of the DM' utility function. And this is another restriction of these methods because the form of the function is assumed in advance.

The next two ANN approaches are more considered with the multiobjectivity. They use the following results. The FFANN with three layers can represent any continuous mapping from R^n to R^k [21] and FFANN with two hidden layers can represent any set in R^n [22].

Sun, Stam and Steuer [19], [20] developed Interactive FFANN Procedure, using a Feed-Forward Artificial Neural Network (FFANN) for solving linear MOMPP.

The idea of this method is to use Feed Forward ANN for presenting the DM's preference structure, i.e. utility function. As input patterns for FFANN efficient points are used. They are generated by Interactive Weighted Tchebycheff Method [2]. The training algorithms based on error-back propagation are used to train FFANN. The result is a nonlinear real valued function. Its maximization (solving a single nonlinear objective problem) yields to the best compromise solution of the linear MOMPP.

The authors use two ways for comparing the objective nondominated vectors to be used later for training and retraining FFANN. First, by using an interval-scale preference value. It is given by the DM. The scale is defined with an ideal vector and 'nadir' vector – *absolute preference information*. The other is by using pair-wise comparison like in AHP process [23] – *relative preference information*.

Malakooti and Zhou [24] developed Adaptive FFANN (AFFANN) approach for solving MCDM problems with finite number of alternatives.

Again the role of FFANN is to describe the DM's preference structure (utility function). The ANN topology is not fixed in advance. The method consists of two parts:

1) Construct AFFANN according to the given training set, starting with a rudimentary AFFANN at the beginning.
2) Use the AFFANN from step 1 to present DM's utility function, rank the set of alternatives and ask the DM if he/she is satisfied with the solution. If yes – stop; otherwise go to step1.

During the process of solving the model adapts to the changes in the training patterns by adding new nodes and links. To assure consistent solving the input patterns are divided into two subsets: the training one and the test one resp.

The DM has the possibility to test the AFFANN until he is satisfied with the results.

The approach of Malakooti-Zhou has the advantage that it can approximate utility functions with a complex nonlinearity. This is done by adding new nodes to the FFANN (adaptivity). The problem arising in this case is that the ANN can lose its predictability and generalization properties. Therefore the approach could be used carefully.

V. CONCLUSION

On the base of the above presentation following conclusions can be drawn.

ANN approach was not applied in full its potential for solving MCDM problems.

As it is seen from the currently developed ANN-MCDM methods they successfully solve mathematically ill-defined multicriteria models.

REFERENCES

[1] Ph. Vincke, *Multicriteria Decision Aid*, New York, J. Willey&Sons, 1992.

[2] R. Steuer, *Multiple Criteria Optimization: Theory, Computation and Application*, New York, John Wiley & Sons, 1986.

[3] K. Miettinen, *Nonlinear Multiobjective Optimization*, Boston, Kluwer, 1999.

[4] J. Karaivanova and S. Narula, "The current state of the multiple objective integer programming methods", *Int. J. of Management and Systems*, vol.14, no.1, pp. 17-46, 1998.

[5] P. Fishbern, *Utility theory for decision making*, New York, Wiley, 1970.

[6] Y. Sawaragi, H. Nakayama and T. Tanino, *Theory for Multiobjective Optimization*, Toronto, Academic Press, 1985

[7] L. Kirilov, *Interactive Methods for Solving Continuous Multiobjective Mathematical Programming Problems*, Ph.D. thesis, IIT – BAS, Sofia, 1992.

[8] R. Benayoun, J. de Montgolfier, J. Tergny and O. Laritchev, "Linear Programming with Multiple Objective Functions: Step Method (STEM)", *Mathem. Programming*, vol. 1, No3, pp. 366-371, 1971.

[9] P. Korhonen and J, Wallenius, "A Multiple Objective Linear Programming Decision Support System", *Decision Support Systems*, vol. 6, pp. 243-251, 1990.

[10] P. Korhonen and J. Laakso, "A Visual Interactive Method for Solving the Multiple Criteria Problem", *Europ. Journal of Oper. Research*, vol. 24, pp. 277-287, 1986.

[11] A. Wierzbicki, “On the Completeness and Constructiveness of Parametric Characterization to Vector Optimization Problems”, *OR Spektrum*, vol. 8, pp. 73-87, 1986.
[12] V. Vassilev, L. Kirilov and A. Atanassov, "Reference Direction Approach for Solving Multiple Objective Programming Problems", *Problems of Engineering Cybernetics and Robotics*, No38, pp. 17-23, 1992.
[13] S. C. Narula, L. Kirilov and V. Vassilev, “Reference Direction Approach for Solving Multiple Objective Nonlinear Programming Problems”, *IEEE Transactions on Systems, Man, and Cybernetics*, vol.24, No5, pp.804-806, 1994.
[14] R. Steuer and E. Choo, “An Interactive Weighted Tchebycheff Procedure for Multiple Objective Programming”, *Mathematical Programming*, vol. 26, pp. 326-344, 1983.
[15] R. Steuer, “An Interactive Multiple Objective Linear Programming Procedure”, *TIMS Studies in the Management Scinces*, vol. 6, pp. 225-239, 1977.
[16] L. Kirilov and V. Vassilev, “A Method for Solving Multiple Objective Linear Programming Problems”, *Multicriteria Analysis (J. Climaco - Ed.)*, Springer-Verlag, Berlin, pp. 302-309, 1997.
[17] S. Zionts and J. Wallenius “An interactive multiple objective linear programming method for a class of underlying nonlinear utility functions”, *Management science*, vol. 29, pp. 519-529, 1983.
[18] A. Cichocki and R. Unbehauen, *Neural Networks for Optimization and Signal Processing*, New York, Willey, 1992
[19] M. Sun, Ant. Stam and R. Steuer “Solving Multiple Objective Programming Problems Using Feed-Forward Artificial Neural Networks: The Interactive FFANN Procedure”, *Management Science*, vol. 42, no.6, 1996
[20] M. Sun, Ant. Stam and R. Steuer “Interactive Multiple Objective Programming Using Tchebycheff Programs and Artificial Neural Networks’, *Computers and Operations Research*, vol.27, no7-8, pp. 601-620, 2000.
[21] R. Hecht-Nielsen “Kolmogorov’s Mapping Neural Network Existence Theorem”, *IEEE Proceedings of the First Int. Conf. on Neural Networks*, vol.1, pp.11-14, 1987.
[22] P. Zwietering, E. Aarts and J. Wesels “ The classification capabilities of Exact Two-Layered Perceptrons”, *Memorandum COSOR 91-09*, Eindhoven University, 1991
[23] T. Saaty, *Multicriteria Decision Making: The Analytic Hierarchy Process*, Pittsburgh, RWS Publications, 1988.
[24] B. Malakooti and Y. Zhou, “Feedforward Artificial Neural Networks for Solving Discrete Multiple Criteria Decision Making Problems”, *Management Science*, vol.40, no11, pp.1542-1561, 1994
[25] Y. Liu, and X. Yao, “Evolutionary design of artificial neural networks with different nodes”, *Proc. IEEE Int. Conf. Evol. Comput. (ICEC'96)*, pp-670-675, 1996
[26] A. Jain, “Artificial neural networks: a tutorial”, *IEEE Computer*, pp-31-44, March 1996
[27] D. B. Fogel, L. J. Fogel and V. W. Porto, “Evolving Neural Networks”, *Biological Cybernetics*, V. 63:6, pp-487-493, 1990
[28] D. B. Fogel, “An Information Criterion for Optimal Neural Network Selection”, *IEEE Trans. Neural Networks*, V. 2, pp-490-497, 1991
[29] J. R. McDonell, and D. Waagen, “Neural Network Structure Design by Evolutionary Programming”, *Proc. of the Second Annual Conference on Evolutionary Programming*, D. B. Fogel and W. Atmar (eds.), Evolutionary Programming Society, La Jolla, CA, pp-79-89, 1993
[30] J. R. McDonell, and D. Waagen, “Evolving Recurrent Perceptrons for Time-Series Modeling”, *IEEE Trans. Neural Networks*, V. 5:1, pp-24-38, 1994
[31] P. J. Angeline, G. M. Saunders, and J. B. Pollack, “An Evolutionary Algorithm that Constructs Recurrent Neural Networks”, *IEEE Trans. Neural Networks*, V. 5:1, p-54-65, 1994
[32] T. W. Brotherton, and P. K. Simpson, “Dynamic Feature Set Training of Neural Nets for Classification”, *Evolutionary Programming IV: Proc. of the Fourth Annual Conference on Evolutionary Programming*, J. R. McDonell, R. G. Reynolds and D. B. Fogel (eds.), MIT Press, Cambridge, MA, pp-83-94, 1995
[33] H. J. Bremermann, “Numerical optimization procedures derived from biological evolution processes”, *Cybernetics Problems in Bionics*, H. L. Östereicher and D. R. Moore (eds.), Gordon and Breach, London, pp-543-562, 1966
[34] A. V. Sebald, and D. B. Fogel, “Design of Fault Tolerant Neural Networks for Pattern Classification”, *Proc. of the First Annual Conference on Evolutionary Programming*, D. B. Fogel and W. Atmar (eds.), Evolutionary Programming Society, La Jolla, CA, pp-90-99, 1992
[35] V. W. Porto, “Alternative Methods for Training Neural Networks”, *Proc. of the First Annual Conference on Evolutionary Programming*, D. B. Fogel and W. Atmar (eds.), Evolutionary Programming Society, La Jolla, CA, pp-100-110, 1992
[36] J. R. McDonell, “Training Neural Networks with Weight Constraints”, *Proc. of the First Annual Conference on Evolutionary Programming*, D. B. Fogel and W. Atmar (eds.), Evolutionary Programming Society, La Jolla, CA, pp-111-119, 1992
[37] A. V. Sebald, J. Schlenzig, and D. B. Fogel, “Minimax Design of CMAC Encoded Neural Controllers for Systems with Variable Time Delay”, *Proc. of the First Annual Conference on Evolutionary Programming*, D. B. Fogel and W. Atmar (eds.), Evolutionary Programming Society, La Jolla, CA, pp-120-126, 1992
[38] D. B. Fogel, L. J. Fogel, and V. W. Porto, “Evolving Neural Networks”, *Biological Cybernetics*, V. 63:6, pp-487-493, 1990
[39] D. B. Fogel, E. C. Wasson, and E. M. Boughton, “Evolving Neural Networks for Detecting Breast Cancer”, *Cancer Letters*, V. 96, pp-49-53, 1995
[40] Electric Power Research Institute, “Genetic Optimization of Neural Network Architecture TR-104074”, ERPI Dist. Ctr. Pleasant Hill, CA, 1994
[41] L. Steels, “A self-organizing spatial vocabulary”, *Artificial Life*, V. 2:3, pp-319-332, 1995
[42] X. Yao, “Evolutionary artificial neural networks”, *Int. J. Neural Syst.*, V. 4:3, pp-203-222, 1993
[43] S. Nolfi, D. Parisi, and J. L. Elman, “Learning and evolution in neural networks”, *Adaptive Behavior*, V. 3:1, pp-5-28, 1994