MATEC Web of Conferences **76**, 04039 (2016) DOI: 10.1051/matecconf/20167604039
CSCC 2016

Priority Derivation from Pairwise Comparison Matrices by Multi-Objective Evolutionary Computing

Ludmil Mikhailov[a]

Manchester Business School, University of Manchester, Manchester, United Kingdom

Abstract. The paper investigates the application of evolutionary algorithms (EA) for solving prioritisation problems, in the framework of the Analytical Hierarchy Process. A new two-objective prioritization (TOP) method was proposed recently by the author, where the prioritisation problem is formulated as an optimization task for minimization of the Euclidean norm and the number of rank violations. The TOP method derives Pareto optimal solutions, which requires the application of efficient computational algorithms. We propose two evolutionary computing approaches, based on single-objective and multi-objective evolutionary algorithms. Our preliminary results from a Monte-Carlo simulation show that the multi-objective EA outperforms the single-objective solution approach with respect to accuracy and computational efficiency.

1 Introduction

The assessment of weights of criteria and scores of alternatives is one of the most important tasks in the multicriteria decision-making. In the Analytical Hierarchy Process (AHP), proposed by Saaty [1], the values of weights and scores are assessed indirectly from comparison judgments. The elicitation process for both weights and scores is the same, so they are often called *priorities*.

The pairwise comparison process in the AHP assumes that the decision-maker can compare any two elements at a given hierarchical level and to provide a numerical value of the ratio of their importance. Comparing any two elements E_i and E_j, the decision-maker assigns a ratio a_{ij}, which represents a judgment concerning the relative importance of preference of the decision element E_i over E_j. If E_i is preferred to E_j then $a_{ij}>1$, otherwise $0<a_{ij}\le 1$.

A full set of ratio-scale judgments for a level with n elements requires *n(n-1)/2* comparisons. In order to derive a priority vector $w=(w_1,w_2,...,w_n)^T$ from a given set of judgments, Saaty constructs a positive reciprocal matrix $A=\{a_{ij}\}$.

With the exception of the traditional Eigenvector prioritization method, all other methods for deriving priorities in the AHP are based on some optimization approach. The optimal prioritization methods, as the Goal programming, the Direct Least Squares, the Logarithmic Least Squares and the Fuzzy Preference Programming introduce an objective function, which measures the degree of approximation or the distance between the initial judgments and the solution ratios [2]. Thus the problem of priority derivation is formulated as an optimization task of minimizing the objective function, subject to normalization and some additional constraints.

Despite the multicriteria nature of the requirements, regarding the properties of their solutions, all optimal prioritization methods optimize a *single* objective function. However, a single objective function cannot encompass and satisfy all requirements about the quality of solutions.

A new two-objective prioritization (TOP) method was proposed recently by the author [3], where the prioritisation problem is formulated as an optimization task for minimization of the Euclidean norm and the number of rank violations. The TOP method derives Pareto optimal solutions, which requires the application of efficient computational algorithms.

The paper investigates the application of evolutionary computing for solving the TOP problem. In order to eliminate the drawbacks of the numerical methods, we propose two evolutionary computing approaches. In the first one, the TOP problem is transformed into a single-objective one, which is then solved by a standard single-objective EA. The second approach applies a multi-objective EA for solving the TOP problem without such transformation.

[a] Email address: ludmil.mikhailov@manchester.ac.uk

In order to compare the solution approaches, we perform Monte-Carlo simulation experiments, by randomly generating a large number of pairwise comparison matrices. The paper presents some initial results from this simulation. Both computational approaches are also illustrated also by a numerical example.

2 The two objective prioritisation problem

Let $S = \{a_{ij} | j > i\}$ be a set of pairwise comparison judgments. The feasible set Q is defined as the set of all priority vectors $w = (w_1,..., w_n)^T$, which satisfy the normalization and non-negativity constraints:

$$Q = \left\{ (w_1,..., w_n) \middle| w_i > 0, \sum_{i=1}^{n} w_i = 1 \right\} \qquad (1)$$

The accuracy of the each priority vector $w \in Q$, approximately satisfying the comparison judgments can be measured by the Total deviation criterion:

$$T(w) = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} (a_{ij} - \frac{w_i}{w_j})^2 \qquad (2)$$

This criterion is equivalent to the squared Euclidean distance for the upper triangular part of a Saaty's reciprocal comparison matrix.

The rank preservation properties of the solutions can be measured by the Number of Violations criterion [2]:

$$NV = \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} v_{ij} \qquad (3)$$

where

$$v_{ij} = \begin{cases} 1, & if \ w_i > w_j \ and \ a_{ij} < 1, \ or \ w_i < w_j \ and \ a_{ij} > 1, \\ 1/2, & if \ w_i = w_j \ and \ a_{ij} \neq 1, \ or \ w_i \neq w_j \ and \ a_{ij} = 1, \\ 0, & otherwise. \end{cases}$$

The Number of Violations criterion (3) can be represented in the following compact form:

$$V(w) = \frac{1}{2} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} \left| signum(a_{ij} - 1) - signum(\frac{w_j}{w_j} - 1) \right|$$

where the *signum* function is defined as:

$$signum(b) = \begin{cases} 1, & if \ b > 1 \\ 0, & if \ b = 0 \\ -1, & if \ b < 1. \end{cases}$$

The two-objective prioritization (TOP) problem [3] is to find a feasible priority vector that 'simultaneously' minimizes the Total deviation and the Number of violations:

$$\begin{array}{l} minimize \ T(w) \text{ and } V(w) \\ subject \ to \ w \in Q , \end{array} \qquad (4)$$

where $T : R^n \to R^1$ and $V : R^n \to R^1$ are real-valued objective functions.

Each feasible vector $w \in Q$ determines a unique value of the objective function vector $y = (T(w), V(w))$. Therefore, the feasible set Q in the space of decision variables can be transformed into a *payoff set* Y in the two-dimensional objective space. The payoff set represents a feasible region of the admissible values of $T(w)$ and $V(w)$, and can be considered as the image of the feasible set Q in the objective space.

The payoff set Y of the TOP problem consists in parallel line segments, as the function $V(w)$ takes non-negative discrete values in some range, and the function $T(w)$ is bounded.

3 Computational approaches for solving the TOP problem

Some classical multi-objective optimization (MOO) methods, which can be applied for finding Pareto optimal solutions to the TOP problem are the Weighting Method, the ε-Constraint Method, the Goal Programming Method or the Proper Equality Constraints method [4]. Generally, the main strength of classical MOO methods is their efficiency and ability to generate strong Pareto optimal solutions. However, these methods have some weaknesses in generating the Pareto optimal solutions, when specific problem knowledge is not available. Additionally, they cannot generate all Pareto optimal solutions with non-convex surfaces. From computational point of view, many optimization runs are required to obtain an approximation set of the Pareto optimal solutions [5].

Recently, the evolutionary algorithms have become an alternative to the classical methods for generating Pareto optimal solutions; since they can eliminate some of the drawbacks of the classical MOO methods.

Taking into account the specific properties of the TOP problem, we can transform it into a single-objective optimisation problem, which is easily solved by standard single-objective EA.

The TOP problem can be transformed into a single-objective problem by associating weights k and *(1-k)* to both objective functions, where $k \in (0,1)$. The modified objective function

$$J(w) = kT(w) + (1-k)V(w) \qquad (5)$$

is used as a fitness function of a single-objective EA, where the value of the weight coefficient is given by the user. This value represents his preferences with respect to the relative importance of those two objectives.

Some multi-objective EA, as the Vector Evaluated Genetic Algorithm (VEGA), the Non-dominated Sorting Genetic Algorithm (NSGA), the Niched Pareto GA (NPGA), the Multi-objective Genetic Algorithm (MOGA) [5] and the Pareto Envelope-based Selection Algorithm (PESA II) [6].

However, it is well known that the presence of constraints scientifically affects the performance of multi-objective EA. Additionally, as opposed to the single objective case, the ranking of a population in the multi-objective case is not unique [7].

In order to assess the applicability of EA for solving the TOP problem, we perform a series of computational experiments by Monte-Carlo simulation. In our study we use the PESA-II, which has some advantages compared to other EA. PESA-II follows the standard procedures of an EA, but with the difference that two populations of solutions are maintained: an internal population (IP) of fixed size, and an external population (EP) of non-fixed but limited size. The internal population's job is to explore new solutions, and it achieves this by the standard EA processes of reproduction and variation (i.e., recombination and mutation). The purpose of the external population is to store and exploit good solutions; it does this by maintaining a large and diverse set of the non-dominated solutions discovered during search [8].

4 Monte-Carlo simulation

Monte-Carlo simulation experiments have been carried out, consisting in generation of comparison matrices with different dimensions and applying the single-objective and multi-objective evolutionary approaches. Initially random consistent pairwise matrices are generated; then they are perturbed by a user-driven parameter, denoted as p, which determines the degree of inconsistency.

The matrices for this comparison study are of dimensions n=3, 4, 5, 6, 7, 8 and 9. For every value of n the parameter p takes 9 values, p=10, 20,…, 90 and each combination of $\{n, p\}$ is replicated 30 times, which gives a total number of 810 n-dimensional matrices with different degrees of inconsistency. The overall number of generated random matrices is 5670.

The single-objective EA (a standard Genetic Algorithm) and PESA-II have been applied for solving the TOP problem for each pairwise comparison matrix, using the jMetal toolkit [9]. In the single-objective EA each chromosome is represented by a string of n components, associated with the n-dimensional priority vector w. The EA performs the basic genetic operators, which are roulette wheel selection, crossover with random mating and simple mutation.

Elitism has also been applied as an additional selection strategy, to make sure that the best performing chromosome always survives. The elitism has been realized by comparing the fitness of chromosomes from the current population and the fitness of the corresponding offspring. The fittest chromosome from the initial population survives for the next generation.

At the beginning of each cycle, all chromosomes are normalised, so that the values of their genes sum up to one. The stopping condition is the number of generations, which is selected to be equal to 100. The experimental results show that the single-objective EA converges to the optimal solution for less than 50 generation cycles.

The high-level pseudocode, showing the main steps in the PESA-II algorithm, is given below [8]:

PESA-II pseudocode

- initially set IP and EP to be empty
- randomly generate ipsize solutions and evaluate them
- store the nondominated solutions in EP
- do (until stopping criterion is met) {
- for 1 to ipsize do{
- select a niche uniformly at random in EP and select a solution
- uniformly at random from within it; place the solution in IP}
- for each pair of solutions in IP do{
- with probability pc, crossover the pair and replace them with the two children created
- mutate both solutions}
- for each solution s in IP do{
- evaluate s
- update EP with s{
- if s dominates one or more solutions in EP; add s to EP and remove the dominated ones
- else if s is dominated by one or more solutions in EP; do nothing
- else {
- if EP has less than epsize solutions; add s to EP
- else if s is in a less crowded niche than a solution in EP; add s to EP and discard a randomly chosen solution from a most crowded niche}

}

}

- return EP, a set of nondominated solutions

END

5 Numerical results

Consider a problem with 5 comparison elements [8], where the DM provides the following pairwise comparison matrix:

$$A = \begin{array}{c} \\ \end{array}\begin{array}{cc} \begin{array}{ccccc} E_1 & E_2 & E_3 & E_4 & E_5 \end{array} & \\ \begin{bmatrix} 1 & 5 & 1/3 & 7 & 2/3 \\ 1/5 & 1 & 2 & 1/2 & 4 \\ 3 & 1/2 & 1 & 1/2 & 1/4 \\ 1/7 & 2 & 2 & 1 & 1/3 \\ 3/2 & 1/4 & 4 & 3 & 1 \end{bmatrix} & \begin{array}{c} E_1 \\ E_2 \\ E_3 \\ E_4 \\ E_5 \end{array} \end{array}$$

The TOP problem is formulated as a single-objective one (5) by the Weighting method and solved by the single-objective EA. The parameters of the EA are selected as follows:

- Population size = 100;- Crossover probability=0.9;
- Mutation probability=0.01.

The results obtained are given in Table 1.

Table 1. Pareto optimal solutions obtained by a single-objective evolutionary algorithm

w_1	w_2	w_3	w_4	w_5	T	V
0.309	0.113	0.096	0.135	0.347	50.048	2
0.344	0.075	0.109	0.111	0.361	39.806	3
0.345	0.058	0.140	0.071	0.386	29.717	4
0.363	0.067	0.146	0.058	0.367	26.962	5
0.381	0.076	0.139	0.062	0.342	26.720	6

PESA-II was applied for solving the same problem. A 30-bit Gray code was used to represent each of the five weights, giving a 150-bit binary chromosome. Uniform crossover was applied with probability 0.2 and a bit-flip per-gene mutation rate of 0.01 was used. It was found that PESA-II is not sensitive to these parameters, and other values give similar performance.

The values of the PESA-II parameter settings are:

IPsize=10; EPsize=100; Generations=50; pm=0.01; Pc=0.2; #grid-cells (niches)=100; representation=30-bits per weight.

The best values of the priority vectors obtained from 20 runs of PESA-II are shown in Table 2.

Table 2. Pareto optimal solutions obtained by PESA II

w_1	w_2	w_3	w_4	w_5	T	V
0.356	0.096	0.096	0.096	0.356	39.469	2
0.362	0.078	0.099	0.100	0.362	38.631	3
0.358	0.065	0.149	0.065	0.363	27.438	4
0.361	0.074	0.144	0.060	0.361	26.622	5
0.398	0.083	0.158	0.065	0.296	26.479	6

By comparing the values of T for each value of V, it is seen that PESA-II solutions outperform those obtained by the single-objective EA, with respect to the accuracy.

Regarding the computational efficiency, the average processing time of the single-objective EA for this example is 1325 milliseconds, while the PESA II algorithm requires 622 milliseconds to find the optimal solution.

The performance comparison was obtained using an Intel-based PC with a Core2Duo T5500 CPU running at 1.66GHz and 2GB of physical memory. The tests were executed on Windows 7 with Java NetBeans IDE running in parallel.

The preliminary results from the Monte-Carlo simulation show that the multi-objective EA gives better accuracy than the single-objective EA, especially for high-dimensional and rather inconsistent pairwise comparison matrices.

Regarding the computation time, both approaches have rather similar performance for pairwise comparison matrices of lower dimension, n=3 and n=4. When the size of the matrices increases, PESA–II strongly outperforms the single-objective EA. The single-objective EA is particularly slower in inconsistent and non-transitive problems with many Pareto optimal solutions.

Conclusions

The paper investigates the application of evolutionary algorithms for solving the TOP problem and sows that they are very good alternatives to the numerical multi-objective optimization methods. The numerical example and the preliminary results from a Monte-Carlo simulation experiment show that the multi-objective EA outperforms the single-objective EA with respect to the computational efficiency and accuracy of solutions.

References

1. T. Saaty, A caling method for priorities in hierarchical structures, Journal of Mathematical Psychology, **15**, 234-281 (1977)
2. B. Golany, M. Kress, A multicriteria evaluation of methods for obtaining weights from ratio-scale matrices, European Journal of Operational Research, **69**, 210-220 (1993)
3. L. Mikhailov, Deriving priorities from ratio-scale comparison judgements: A multiple objective approach, *Proceedings of the 17th International Conference on Multiple Criteria Decision Making MCDM'2004*, August 2004, Whistler, Canada
4. Y. Sawaragi, H. Nakyama, T. Tanino, *Theory of multiobjective optimization*. Academic Press, Orlando (1985)
5. K. Deb, *Evolutionary algorithms for multi-criterion optimization in engineering design*. John Wiley & Sons, Chichester, UK (1999)
6. D. Corne, N. Jerram, J. Knowles, M. Oates, PESA-II: Region-based Selection in Evolutionary Multiobjective Optimization, *Proceedings of the Genetic and Evolutionary Computation Conference*, 283-290 (2001)
7. C. Fonseca, P. Fleming, Multiobjective optimization and multiple constraints handling with evolutionary algorithms, part 1: A unified formulation, IEEE Transactions on SMC, **28**, 26-37 (1998)
8. L. Mikhailov, J. Knowles, Priority Elicitation in the AHP by a Pareto Envelope-based Selection Algorithm. *Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems* (M Ehrgott, B Naujoks, T Stewart, and J Wallenius eds.), Springer's Lecture Notes in Economics and Mathematical Systems, **634**, 249-258, Springer-Verlag, London (2010)
9. J. Durillo, A. Nebro, F. Luna, B. Dorronsoro, E. Alba, jMetal: *A Java Framework for Developing Multi-Objective Optimization Metaheuristics*, Departamento de Lenguajes y Ciencias de la Computación, University of Málaga (2006)