Evaluation of MO-ACO Algorithms Using a New Fast Inter-Criteria Analysis Method



53

Jean Dezert, Stefka Fidanova, and Albena Tchamova

Abstract In this paper, we present a fast Belief Function based Inter-Criteria Analysis (BF-ICrA) method based on the canonical decomposition of basic belief assignments defined on a dichotomous frame of discernment. This new method is then applied for evaluating the Multiple-Objective Ant Colony Optimization (MO-ACO) algorithm for Wireless Sensor Networks (WSN) deployment and for Workforce Planning Problem (WPP).

Keywords Inter-Criteria Analysis • Ant Colony Optimization • Belief functions • Canonical decomposition • Proportional Conflict Redistribution • Fusion rules

1 Introduction

In our previous work [1] we propose a new and improved version of classical Atanassov's InterCriteria Analysis (ICrA) [2–4] approach based on Belief Functions (BF-ICrA). This method proposes a better construction of Inter-Criteria Matrix that fully exploits all the information of the score matrix, and the closeness measure of agreement between criteria based on belief interval distance. In [6], we show how the fusion of many sources of evidences represented by Basic Belief Assignments (BBAs) defined on a same dichotomous frame of discernment can be fast and easily done thanks to the Proportional Conflict Redistribution rule no. 5 based canonical decomposition of the BBAs, proposed recently in [7]. In [8] we did show how to use this fast fusion method for decision-making support. In this paper we propose

S. Fidanova (⊠) · A. Tchamova Institute of I&C Technology, Bulgarian Academy of Sciences, Sofia, Bulgaria e-mail: stefka@parallel.bas.bg

A. Tchamova e-mail: tchamova@bas.bg

J. Dezert

The Frenche Aerospace Lab/ONERA, Palaiseau, France e-mail: jean.dezert@onera.fr

[©] The Author(s), under exclusive license to Springer Nature Switzerland AG 2022 S. Fidanova (ed.), *Recent Advances in Computational Optimization*, Studies in Computational Intelligence 986, https://doi.org/10.1007/978-3-030-82397-9_3

a new fast BF-ICrA method based on this canonical decomposition. Then we show how to apply it for the evaluation of the Multiple-Objective Ant Colony Optimization (MO-ACO) algorithm for Wireless Sensor Networks (WSN) deployment, and for workforce planning problem. After a condensed presentation of basics of belief functions in Sect. 2, including the short description of canonical decomposition of dichotomous BBAs approach (in Sect. 2.2), and the main steps of fast fusion method of dichotomous BBAs (in Sect. 2.3), the BF-ICrA method is described and analyzed in Sect. 3, and the fast BF-ICrA method in Sect. 4. In the Sect. 5 we present the Multiple-Objective Ant Colony Optimization (MO-ACO) algorithm. In Sect. 6 the results of the fast BF-ICrA method with the MO-ACO algorithm for WSN layout deployment [22] are presented and discussed. The fast BF-ICrA method has also been applied to the workforce planning problem [24], and the results are presented in Sect. 7. Conclusion is given in Sect. 8.

2 Basics of Belief Functions

2.1 Basic Definitions

Belief functions (BF) have been introduced by Shafer in [9] to model epistemic uncertainty and to combine distinct sources of evidence thanks to Dempster's rule of combination. In Shafer's framework, we assume that the answer¹ of the problem under concern belongs to a known finite discrete frame of discernment (FoD) $\Theta =$ $\{\theta_1, \theta_2, \ldots, \theta_n\}$, with n > 1, and where all elements of Θ are mutually exclusive and exhaustive. The set of all subsets of Θ (including empty set \emptyset and Θ) is the power-set of Θ denoted by 2^{Θ} . A proper Basic Belief Assignment (BBA) associated with a given source of evidence is defined [9] as a mapping $m(\cdot) : 2^{\Theta} \rightarrow [0, 1]$ satisfying $m(\emptyset) = 0$ and $\sum_{A \in 2^{\Theta}} m(A) = 1$. The quantity m(A) is called the mass of A committed by the source of evidence. Belief and plausibility functions are respectively defined from a proper BBA $m(\cdot)$ by

$$Bel(A) = \sum_{B \in 2^{\Theta} | B \subseteq A} m(B)$$
⁽¹⁾

and

$$Pl(A) = \sum_{B \in 2^{\Theta} | A \cap B \neq \emptyset} m(B) = 1 - \operatorname{Bel}(\bar{A}).$$
⁽²⁾

where \overline{A} is the complement of A in Θ .

Bel(A) and Pl(A) are usually interpreted respectively as lower and upper bounds of an unknown (subjective) probability measure P(A). The quantities $m(\cdot)$ and $Bel(\cdot)$

¹ I.e. the solution, or the decision to take.

are one-to-one and linked by the Möbius inverse formula (see [9], p. 39). A is called a Focal Element (FE) of $m(\cdot)$ if m(A) > 0. When all focal elements are singletons, $m(\cdot)$ is called a *Bayesian BBA* [9] and its corresponding *Bel*(·) function is equal to *Pl*(·) and they are homogeneous to a (subjective) probability measure $P(\cdot)$. The vacuous BBA, representing a totally ignorant source, is defined as $m_v(\Theta) = 1$. A dichotomous BBA is a BBA defined on a FoD which has only two proper subsets, for instance $\Theta = \{A, \overline{A}\}$ with $A \neq \Theta$ and $A \neq \emptyset$. A dogmatic BBA is a BBA such that $m(\Theta) = 0$. If $m(\Theta) > 0$ the BBA $m(\cdot)$ is nondogmatic. A simple BBA is a BBA that has at most two focal sets and one of them is Θ . A dichotomous non dogmatic mass of belief is a BBA having three focal elements A, \overline{A} and $A \cup \overline{A}$ with A and \overline{A} subsets of Θ .

In his Mathematical Theory of Evidence [9], Shafer proposed to combine $s \ge 2$ distinct sources of evidence represented by BBAs with Dempster's rule (i.e. the normalized conjunctive rule), which unfortunately behaves counterintuitively both in high and low conflicting situations as reported in [10–13]. In our previous works (see [14], Vols. 2 and 3 for full justification and examples) we did propose new rules of combination based on different Proportional Conflict Redistribution (PCR) principles, and we have shown the interest of the PCR rule No 5 (PCR5) for combining two BBAs, and PCR rule No 6 (PCR6) for combining more than two BBAs altogether [14], Vol. 2. PCR6 coincides with PCR5 when one combines two sources. The difference between PCR5 and PCR6 lies in the way the proportional conflict redistribution is done as soon as three (or more) sources are involved in the fusion. PCR5 transfers the conflicting mass only to the elements involved in the conflict and proportionally to their individual masses, so that the specificity of the information is entirely preserved in this fusion process.

The general (complicate) formulas for PCR5 and PCR6 rules are given in [14], Vol. 2. The fusion of two BBAs based on PCR5 (or PCR6) rule which will be use for canonical decomposition of a dichotomous BBA is obtained by the formula

$$m_{PCR5}(X) = \sum_{\substack{X_1, X_2 \in 2^{\Theta} \\ X_1 \cap X_2 = X}} m_1(X_1) m_2(X_2) + \sum_{\substack{X_2 \in 2^{\Theta} \\ X_2 \cap X = \emptyset}} \left[\frac{m_1(X)^2 m_2(X_2)}{m_1(X) + m_2(X_2)} + \frac{m_2(X)^2 m_1(X_2)}{m_2(X) + m_1(X_2)} \right]$$
(3)

where all denominators in (3) are different from zero. If a denominator is zero, that fraction is discarded.

From the implementation point of view, PCR6 is simpler to implement than PCR5. For convenience, very basic (not optimized) MatlabTM codes of PCR5 and PCR6 fusion rules can be found in [14, 15] and from the toolboxes repository on the web [16]. The main drawback of PCR5 and PCR6 rules is their very high combinatorial complexity when the number of source is big, as well as the cardinality of the FoD. In this case, PCR5 or PCR6 rules cannot be used directly because of memory overflow.

Even for combining BBAs defined on a simple dichotomous FoD as those involved in the Inter-Criteria Analysis (ICrA), the computational time for combining more than 10 sources can take several hours.² That is why a fast fusion method to combine dichotomous BBAs is necessary, and we present it in the next subsections.

2.2 Canonical Decomposition of Dichotomous BBA

A FoD $\Theta = \{A, \overline{A}\}$ is called dichotomous if it consists of only two proper subsets A and \overline{A} with $A \cup \overline{A} = \Theta$ and $A \cap \overline{A} = \emptyset$, where \overline{A} is the complement of A in Θ and A is different from Θ and from Empty-Set. We consider a given proper BBA $m(\cdot) : 2^{\Theta} \rightarrow [0, 1]$ of the general form

$$m(A) = a, \quad m(\bar{A}) = b, \quad m(A \cup \bar{A}) = 1 - a - b$$
 (4)

The canonical decomposition problem consists in finding the two following simple proper BBAs m_p and m_c of the form

$$m_p(A) = x, \quad m_p(A \cup A) = 1 - x$$
 (5)

$$m_c(\bar{A}) = y, \quad m_c(A \cup \bar{A}) = 1 - y \tag{6}$$

with $(x, y) \in [0, 1] \times [0, 1]$, such that $m = Fusion(m_p, m_c)$, for a chosen rule of combination denoted by $Fusion(\cdot, \cdot)$. The simple BBA $m_p(\cdot)$ is called the *pro-BBA* (or pro-evidence) of A, and the simple BBA $m_c(\cdot)$ the *contra-BBA* (or contra-evidence) of A. The BBA $m_p(\cdot)$ is interpreted as a source of evidence providing an uncertain evidence in favor of A, whereas $m_c(\cdot)$ is interpreted as a source of evidence providing an uncertain contrary evidence about A.

In [7], we have shown that this decomposition is possible with Dempster's rule only if 0 < a < 1, 0 < b < 1 and a + b < 1, and we have $x = \frac{a}{1-b}$ and $y = \frac{b}{1-a}$. However, any dogmatic BBA $m(A) = a, m(\bar{A}) = b$ with a + b = 1 is not decomposable from Dempster's rule for the case when $(a, b) \neq (1, 0)$ and $(a, b) \neq (0, 1)$, and the dogmatic BBAs m(A) = 1, $m(\bar{A}) = 0$, or m(A) = 0, $m(\bar{A}) = 1$ have infinitely many decompositions based on Dempster's rule of combination. We have also proved that this canonical decomposition cannot be done from conjunctive, disjunctive, Yager's [17] or Dubois-Prade [18] rules of combination, neither from the averaging rule. The main result of [7] is that this canonical decomposition is unique and is always possible in all cases using the PCR5 rule of combination. This is very useful to implement a fast efficient approximating fusion method of dichotomous BBAs as presented in details in [6]. We recall the following two important theorems proved in [7].

² With a MacBook Pro 2.8 GHz Intel Core i7 with 16 Go 1600 MHz DDR3 memory running Matlab[™] R2018a.

Theorem 1 Consider a dichotomous $FoD \Theta = \{A, \bar{A}\}$ with $A \neq \Theta$ and $A \neq \emptyset$ and a nondogmatic BBA $m(\cdot) : 2^{\Theta} \rightarrow [0, 1]$ defined on Θ by m(A) = a, $m(\bar{A}) = b$, and $m(A \cup \bar{A}) = 1 - a - b$, where $a, b \in [0, 1]$ and a + b < 1. Then the BBA $m(\cdot)$ has a unique canonical decomposition using PCR5 rule of combination of the form m = $PCR5(m_p, m_c)$ with pro-evidence $m_p(A) = x$, $m_p(A \cup \bar{A}) = 1 - x$ and contraevidence $m_c(\bar{A}) = y$, $m_c(A \cup \bar{A}) = 1 - y$ where $x, y \in [0, 1]$.

Theorem 2 Any dogmatic BBA defined by m(A) = a and $m(\bar{A}) = b$, where $a, b \in [0, 1]$ and a + b = 1, has a canonical decomposition using PCR5 rule of combination of the form $m = PCR5(m_p, m_c)$ with $m_p(A) = x$, $m_p(A \cup \bar{A}) = 1 - x$ and $m_c(\bar{A}) = y$, $m_c(A \cup \bar{A}) = 1 - y$ where $x, y \in [0, 1]$.

Theorems 1 and 2 prove that the decomposition based on PCR5 always exists and it is unique for any dichotomous (nondogmatic, or dogmatic) BBA.

For the case of dichotomous nondogmatic BBA considered in Theorem 1, one has to find x and y solutions of the system

$$a = x(1 - y) + \frac{x^2 y}{x + y} = \frac{x^2 + xy - xy^2}{x + y}$$
(7)

$$b = (1-x)y + \frac{xy^2}{x+y} = \frac{y^2 + xy - x^2y}{x+y}$$
(8)

under the constraints $(a, b) \in [0, 1]^2$, and 0 < a + b < 1. The explicit expression of x and y are difficult to obtain analytically (even with modern symbolic computing systems like MathematicaTM, or MapleTM) because one has a quartic equation to solve whose general analytical expression of its solutions is very complicate. Fortunately, the solutions can be easily calculated numerically by these computing systems, and even with MatlabTM system (thanks to the *fsolve* function) as soon as the numerical values are committed to a and to b, and this is what we use in our simulations.

2.3 Fast Fusion of Dichotomous BBAs

The main idea for making the fast fusion of dichotomous BBAs $m_s(.)$, for s = 1, 2, ..., S defined on the same FoD Θ is based on the three following main steps:

- 1. In the first step, one decomposes canonically each dichotomous BBA $m_s(\cdot)$ into its pro and contra evidences $m_{p,s} = (m_{p,s}(A), m_{p,s}(\bar{A}), m_{p,s}(A \cup \bar{A})) = (x_s, 0, 1 x_s)$ and $m_{c,s} = (m_{c,s}(A), m_{c,s}(\bar{A}), m_{c,s}(A \cup \bar{A})) = (0, y_s, 1 y_s)$,
- 2. In the second step, one combines the pro-evidences $m_{p,s}$ for s = 1, 2, ..., S altogether to get a global pro-evidence m_p , and in parallel one combines all the contra-evidences $m_{c,s}$ for s = 1, 2, ..., S altogether to get a global contra-evidence m_c . The fusion step of pro and contra evidences is based on conjunctive rule of combination.

3. Once m_p and m_c are calculated, then one combines them with PCR5 fusion rule to get the final result.

Because the PCR5 rule of combination is not associative, the fusion of the canonical BBAs followed by their PCR5 fusion will not provide in general the same result as the direct fusion of the dichotomous BBAs altogether but only an approximate result, which is normal. However, this new fusion approach is interesting because the fusion of the pro-evidence $m_{p,s}$ (resp. contra-evidences $m_{c,s}$) is very simple because there is non conflict between $m_{p,s}$ (resp. between $m_{c,s}$), so that their fusion can be done quite easily and a large number of sources can be combined without a high computational burden. In fact, with this fusion approach, only one PCR5 fusion step of simple (combined) canonical BBAs is needed at the very end of the fusion process. In [6], we have proved with a Monte-Carlo simulation analysis that the approximation obtained by this new fusion method based on the fusion of pro-evidences and contra-evidences with respect to the direct fusion of the BBAs with PCR5 (or PCR6 when considering more than two sources to combine) is effective because the agreement between the decision taken from the direct fusion method, and the indirect (canonical decomposition based) method is very good. This new fusion method based on this canonical decomposition does not suffer of combinatorial complexity limitation which is of great interest in some applications because many (hundreds or even thousands) of dichotomous BBAs could be easily combined very quickly. Actually with this method what takes a bit time is only the canonical decomposition done by the numerical solver. Our analysis [6] has shown that complexity of this fast approach is quasi-linear with the number of sources to combine.

3 The BF-ICrA Method

In [1], we did present an improved version of Atanassov's Inter-Criteria Analysis (ICrA) method [2–4] based on belief functions. This new method has been named BF-ICrA (Belief Function based Inter-Criteria Analysis) for short. It has already been applied to GPS surveying problems in [19]. We present briefly in this section the principles of BF-ICrA.

BF-ICrA starts with the construction of an $M \times N$ BBA matrix $\mathbf{M} = [m_{ij}(\cdot)]$ from the score matrix $\mathbf{S} = [S_{ij}]$. The BBA matrix \mathbf{M} is obtained as follows—see [20] for details and justification.

$$m_{ij}(A_i) = Bel_{ij}(A_i) \tag{9}$$

$$m_{ij}(\bar{A}_i) = Bel_{ij}(\bar{A}_i) = 1 - Pl_{ij}(A_i)$$
 (10)

$$m_{ij}(A_i \cup A_i) = Pl_{ij}(A_i) - Bel_{ij}(A_i)$$
(11)

Evaluation of MO-ACO Algorithms Using a New Fast ...

where³

$$Bel_{ij}(A_i) \triangleq Sup_j(A_i)/A_{\max}^j$$
 (12)

$$Bel_{ij}(\bar{A}_i) \triangleq Inf_j(A_i)/A_{\min}^j$$
 (13)

with

$$Sup_{j}(A_{i}) \triangleq \sum_{k \in \{1, \dots, M\} | S_{kj} \le S_{ij}} |S_{ij} - S_{kj}|$$

$$(14)$$

$$Inf_j(A_i) \triangleq -\sum_{k \in \{1,\dots,M\} | S_{kj} \ge S_{ij}} |S_{ij} - S_{kj}|$$
(15)

and

$$A_{\max}^{j} \triangleq \max Sup_{j}(A_{i}) \tag{16}$$

$$A_{\min}^{j} \triangleq \min_{i} Inf_{j}(A_{i}) \tag{17}$$

For another criterion $C_{j'}$ and the j'th column of the score matrix we will obtain another set of BBA values $m_{ij'}(\cdot)$. Applying this method for each column of the score matrix we are able to compute the BBA matrix $\mathbf{M} = [m_{ij}(\cdot)]$ whose each component is in fact a triplet $(m_{ij}(A_i), m_{ij}(\bar{A}_i), m_{ij}(A_i \cup \bar{A}_i))$ of BBA values in [0, 1] such that $m_{ij}(A_i) + m_{ij}(\bar{A}_i) + m_{ij}(A_i \cup \bar{A}_i)) = 1$ for all i = 1, ..., M and j = 1, ..., N.

The next step of BF-ICrA approach is the construction of the $N \times N$ Inter-Criteria Matrix $\mathbf{K} = [K_{jj'}]$ from $M \times N$ BBA matrix $\mathbf{M} = [m_{ij}(\cdot)]$ where elements $K_{jj'}$ corresponds to the BBA $(m_{jj'}(\theta), m_{jj'}(\bar{\theta}), m_{jj'}(\theta \cup \bar{\theta}))$ about positive consonance θ , negative consonance $\bar{\theta}$ and uncertainty between criteria C_j and $C_{j'}$ respectively. The construction of the triplet $K_{jj'} = (m_{jj'}(\theta), m_{jj'}(\bar{\theta}), m_{jj'}(\theta \cup \bar{\theta}))$ is based on two steps:

• Step 1 (BBA construction): Getting $m_{jj'}^i(.)$.

For each alternative A_i for i = 1, ..., M, we first compute the belief assignment $(m_{jj'}^i(\theta), m_{jj'}^i(\bar{\theta}), m_{jj'}^i(\theta \cup \bar{\theta}))$ for any two criteria $j, j' \in \{1, 2, ..., N\}$. For this, we consider two sources of evidences (SoE) indexed by j and j' providing the BBA m_{ij} and $m_{ij'}$ defined on the simple FoD $\{A_i, \bar{A}_i\}$ and denoted $m_{ij} = [m_{ij}(A_i), m_{ij}(\bar{A}_i), m_{ij}(A_i \cup \bar{A}_i)]$ and $m_{ij'} = [m_{ij'}(A_i), m_{ij'}(\bar{A}_i), m_{ij'}(A_i \cup \bar{A}_i)]$. We also denote $\Theta = \{\theta, \bar{\theta}\}$ the FoD about the relative state of the two SoE, where θ means that the two SoE agree, $\bar{\theta}$ means that they disagree and $\theta \cup \bar{\theta}$ means that we don't know. Hence, two SoE are in total agreement if both commit their maximum belief mass to the same element A_i or to the same element \bar{A}_i . Similarly, two SoE are in total disagreement if each one commits its maximum mass of belief to one element and the other to its opposite, that is if one has $m_{ij}(A_i) = 1$ and

³ Assuming that $A_{\text{max}}^j \neq 0$ and $A_{\text{min}}^j \neq 0$. If $A_{\text{max}}^j = 0$ then $Bel_{ij}(A_i) = 0$, and if $A_{\text{min}}^j = 0$ then $Pl_{ij}(A_i) = 1$.

 $m_{ij'}(\bar{A}_i) = 1$, or if $m_{ij}(\bar{A}_i) = 1$ and $m_{ij'}(A_i) = 1$. Based on this very simple and natural principle, one can now compute the belief masses as follows:

$$m_{jj'}^{i}(\theta) = m_{ij}(A_i)m_{ij'}(A_i) + m_{ij}(\bar{A})m_{ij'}(\bar{A})$$
(18)

$$m_{jj'}^{i}(\bar{\theta}) = m_{ij}(A_i)m_{ij'}(\bar{A}_i) + m_{ij}(\bar{A}_i)m_{ij'}(A_i)$$
(19)

$$m_{jj'}^{i}(\theta \cup \bar{\theta}) = 1 - m_{jj'}^{i}(\theta) - m_{jj'}^{i}(\bar{\theta})$$
(20)

 $m_{jj'}^{i}(\theta)$ represents the degree of agreement between the BBA $m_{ij}(\cdot)$ and $m_{ij'}(\cdot)$ for the alternative $A_i, m_{jj'}^{i}(\bar{\theta})$ represents the degree of disagreement of the two BBAs and $m_{jj'}^{i}(\theta \cup \bar{\theta})$ the level of uncertainty (i.e. how much we don't know if they agree or disagree). By construction $m_{jj'}^{i}(\cdot) = m_{j'j}^{i}(\cdot), m_{jj'}^{i}(\theta), m_{jj'}^{i}(\bar{\theta}), m_{jj'}^{i}(\theta \cup \bar{\theta}) \in$ [0, 1] and $m_{jj'}^{i}(\theta) + m_{jj'}^{i}(\bar{\theta}) + m_{jj'}^{i}(\theta \cup \bar{\theta}) = 1$. This BBA modeling permits to build a set of M symmetrical Inter-Criteria Belief Matrices (ICBM) $\mathbf{K}^{i} = [K_{jj'}^{i}]$ of dimension $N \times N$ relative to each alternative A_i whose components $K_{jj'}^{i}$ correspond to the triplet of BBA values $m_{jj'}^{i} = (m_{jj'}^{i}(\theta), m_{jj'}^{i}(\bar{\theta}), m_{jj'}^{i}(\theta \cup \bar{\theta}))$ modeling the belief of agreement and of disagreement between C_j and $C_{j'}$ based on A_i .

• Step 2 (fusion): Getting m_{ii}/(.).

In this step, one needs to combine the BBAs $\mathbf{m}_{\mathbf{jj}'}^{\mathbf{i}}(.)$ for i = 1, ..., M altogether to get the component $K_{jj'} = (m_{jj'}(\theta), m_{jj'}(\bar{\theta}), m_{jj'}(\theta \cup \bar{\theta}))$ of the Inter-Criteria Belief matrix⁴ (ICBM) $\mathbf{K} = [K_{jj'}]$. For this and from the theoretical standpoint, we recommend to use the PCR6 fusion rule [14] (Vol. 3) because of known deficiencies of Dempster's rule.

Once the global Inter-Criteria Belief Matrix (ICBM) is calculated as the matrix $\mathbf{K} = [K_{jj'} = (m_{jj'}(\theta), m_{jj'}(\bar{\theta}), m_{jj'}(\theta \cup \bar{\theta}))]$, we can identify the criteria that are in strong agreement, in strong disagreement, and those on which we are uncertain. For identifying the criteria that are in strong agreement, we evaluate the distance of each component of $K_{jj'}$ with the BBA representing the best agreement state and characterized by the specific BBA⁵ $m_T(\theta) = 1$. From a similar approach we can also identify, if we want, the criteria that are in very strong disagreement using the distance of $m_{jj'}(\cdot)$ with respect to the BBA representing the best disagreement state characterized by the specific BBA $m_F(\bar{\theta}) = 1$. We use the belief interval distance $d_{BI}(m_1, m_2)$ presented in [21] for measuring the distance between the two BBAs.

⁴ For presentation convenience, the ICBM $\mathbf{K} = [K_{jj'} = (m_{jj'}(\theta), m_{jj'}(\bar{\theta}), m_{jj'}(\theta \cup \bar{\theta}))]$ is decomposed into three matrices $\mathbf{K}(\theta) = [K_{jj'}^{\theta} = m_{jj'}(\theta)]$, $\mathbf{K}(\bar{\theta}) = [K_{jj'}^{\bar{\theta}} = m_{jj'}(\bar{\theta})]$ and $\mathbf{K}(\theta \cup \bar{\theta}) = [K_{jj'}^{\theta \cup \bar{\theta}} = 1 - m_{jj'}(\theta) - m_{jj'}(\bar{\theta})]$.

⁵ We use the index T in the notation $m_T(\cdot)$ to refer that the agreement is true, and F in $m_F(\cdot)$ to specify that the agreement is false.

4 Fast BF-ICrA Method

The computational complexity of BF-ICrA is of course higher than the complexity of ICrA because it makes a more precise evaluation of local and global inter-criteria belief matrices with respect to inter-criteria matrices calculated by Atanassov's ICrA. The overall reduction of the computational burden of the original MCDM problem thanks to BF-ICrA depends highly on the problem under concern, the complexity and cost to evaluate each criteria involved in it, as well as the number of redundant criteria identified by BF-ICrA method.

The main drawback of BF-ICrA method is the PCR6 combination required in its step 2 for combining altogether the dichotomous BBAs $m_{jj'}^i(.)$. Because of combinatorial complexity of PCR6 rule, it cannot work in reasonable computational time as soon as the number of sources to combine altogether is greater than 10, which prevents its use for solving ICrA problems involving more than 10 alternatives (as in the Examples 2 and 3 presented in Sect. 6). That is why it is necessary to adapt the original BF-ICrA method for working with a large number of alternatives and criteria. For this, we can in step 2 of BF-ICrA exploit the method for the fast fusion of dichotomous BBAs presented in Sect. 2.3. More precisely, each dichotomous BBA $m_{jj'}^i(.)$ will be canonically decomposed in its pro-evidence $m_{jj',p}^i(.)$ and its contra-evidence $m_{jj',p}^i(.)$ and the global contra-evidence $m_{jj',c}(.)$. Then, the BBAs $m_{jj',p}(.)$ and $m_{jj',c}(.)$ are combined with PCR5 rule to get the BBAs $m_{jj'}(.)$ and, finally, the global Inter-



Fig. 1 Principle of fast fusion of $m_{ii'}^i(.)$ of Step 2 of BF-ICrA

Criteria Belief Matrix $\mathbf{K} = [K_{jj'} = (m_{jj'}(\theta), m_{jj'}(\bar{\theta}), m_{jj'}(\theta \cup \bar{\theta}))]$. The principle of this modified step 2 of BF-ICrA is summarized in the Fig. 1 for convenience.

Another simpler fusion method to combine the dichotomous BBAs $m_{jj'}^i$ (.) would just consist to average them. In Sect. 6, we will show how these two methods behave in the examples chosen for the evaluation of MO-ACO Algorithm for optimal WSN deployment.

5 Multi-objective ACO Algorithm

Recently Wireless Sensor Networks (WSNs) have attracted the attention of the research scientists community, conditioned by a set of challenges: theoretical and practical. WSNs consists of distributed sensor nodes and their main purpose is to monitor the real-time environmental status, based on gathering available sensor information, processing and transmitting the collected data to the specified remote base station. It is a promising technology that is used in a coverage of application requiring minimum human contribution, ranging from civil and military to healthcare and environmental monitoring. One of the key mission of WSN is the full surveillance of the monitoring region with a minimal number of sensors and minimized energy consumption of the network. The lifetime of the sensors is strongly related to the amount of the power loaded in the battery, that is why the control of the energy consumption of sensors is an important active research problem. The small energy storage capacity of sensor nodes intrudes the possibility to gather the information directly to the main base. Because of this they transfer their data to the so called High Energy Communication Node (HECN), which is able to collect the information from across the network and to transmit it to the base computer for processing. The sensors transmit their data to the HECN, either directly or via hops, using closest sensors as communication relays. The WSN can have large numbers of nodes and the problem can be very complex.

In order to solve successfully the key mission of WSNs, in [22], we did apply multiobjective Ant Colony Optimization (ACO) to solve this hard, from the computational point of view, telecommunication problem. The number of ants is one of the key algorithm parameters in the ACO and it is important to find the optimal number of ants needed to achieve good solutions with minimal computational resources. In [22], the optimal solution was obtained by applying the classical Atanassov's ICrA method. In the next section we will present the results obtained by the fast BF-ICrA approach and compare their results.

The problem of designing a WSN is multi-objective, with two objective functions: (1) one wants to minimize the energy consumption of the nodes in the network, and (2) one wants to minimize the number of nodes. The full coverage of the network and connectivity are considered as constraints. For solving this problem, we have proposed to use a Multi-Objective Ant Colony Optimization (MO-ACO) algorithm in [22] and we have studied the influence of the number of ants on the algorithm performance and quality of the achieved solutions. The computational resources,

which the algorithm needs, are not negligible. The computational resources depends on the size of the solved problem and on the number of ants. The aim is to find a minimal number of ants which allow the algorithm to find good solution for WSN deployment.

The ACO algorithm uses a colony of artificial ants that behave as cooperating agents. With the help of the pheromone and the heuristic information they try to construct better solutions and to find the optimal ones. The pheromone corresponds to the global memory of the ants and the heuristic information is a some preliminary knowledge of the problem. The problem is represented by a graph and the solution is represented by a path in the graph or by tree in the graph. Ants start from random nodes and construct feasible solutions. When all ants construct their solution the pheromone is updated. The new, added, pheromone depends to the quality of the solution. The elements of the graph, which belong to better solutions will receive more pheromone and will be more desirable in the next iteration. In our implementation, we use the MAX-MIN Ant System (MMAS) which is one of the most successful ant approaches originally presented in [23]. In our case, the graph of the problem is represented by a square grid. The nodes of the graph are enumerated. The ants will deposit their pheromone on the nodes of the grid. We will deposit the sensors on the nodes of the grid too. The solution is represented by tree. An ant starts to create a solution starting from random node, which communicates with the HECN. After it includes next nodes in the solution applying probabilistic rule, called transition probability which is a product of the heuristic information and quantity of the pheromone, corresponding to this new node. Construction of the heuristic information is a crucial point in the ant algorithms. It is problem dependent and helps us to manage the search process. Our heuristic information represented by (21) is a product of three values.

$$\eta_{ij}(t) = s_{ij} l_{ij} (1 - b_{ij}) \tag{21}$$

where s_{ij} is the number of the new points (nodes of the graph) which the new sensor will cover, and which are not covered by other sensors, and

$$l_{ij} = \begin{cases} 1 & \text{if communication exists ;} \\ 0 & \text{if there is no communication.} \end{cases}$$
(22)

and where b_{ij} is the solution matrix. The matrix element b_{ij} equals 1 when there is sensor on this position, otherwise $b_{ij} = 0$. With s_{ij} , we try to increase the number of points covered by one sensor and thus to decrease the number of sensors we need. With l_{ij} , we guarantee that all sensors will be connected. With b_{ij} we guarantee that maximum one sensor will be mapped on the same point. The search stops when transition probability $p_{ij} = 0$ for all values of *i* and *j*. It means that there are no more free positions, or that all area is fully covered. At the end of every iteration the quantity of the pheromone is updated according to the rule:

$$\tau_{ij} \leftarrow \rho \tau_{ij} + \Delta \tau_{ij}, \tag{23}$$

with the increment $\Delta \tau_{ij} = 1/F(k)$ if (i, j) belongs to the non-dominated solution constructed by ant k, or $\Delta \tau_{ij} = 0$ otherwise. The parameter ρ is a pheromone decreasing parameter chosen in [0, 1]. This parameter ρ models evaporation in the nature and decreases the influence of old information on the search process. After that, we add the new pheromone, which is proportional to the value of the fitness function constructed as:

$$F(k) = \frac{f_1(k)}{\max_i(f_1(i))} + \frac{f_2(k)}{\max_i(f_2(i))},$$
(24)

where $f_1(k)$ is the number of sensors proposed by the *k*th ant, and $f_2(k)$ is the energy of the solution of the *k*th ant. These are also the objective functions of the WSN layout problem. We normalize the values of two objective functions with their maximal achieved values from the first iteration.

6 Application to WSN Layout Deployment

In this section we present the results of the fast BF-ICrA method with the MO-ACO algorithm for WSN layout deployment. Fidanova and Roeva have developed a software, which realizes the MO-ACO algorithm. This software can solve the problem at any rectangular area, the communication and the coverage radius can be different and can have any positive value. We can have regions in the area. The program was written in C language, and the tests were run on computer with an Intel Pentium 2.8 GHz processor. In their tests, they use an example where the area is square. The coverage and communication radii cover 30 points. The HECN is fixed in the centre of the area. In the sequel we consider three examples of areas with three sizes: 350×350 points, 500×500 points, and 700×700 points. The MO-ACO algorithm is based on 30 runs for each number of ants. We extract the Pareto front from the solutions of these 30 runs, and we show the achieved non dominated solutions (approximate Pareto fronts) for each case on which the BF-ICrA will be applied. The score matrices for each case is given in Tables 1, 2 and 3 [22].

Table 1	The 6 \times	10 score	matrix	S for	350	\times 350	case	(Example	1)
---------	----------------	----------	--------	--------------	-----	--------------	------	----------	----

		ACO_1	ACO_2	ACO ₃	ACO ₄	ACO ₅	ACO ₆	ACO ₇	ACO ₈	ACO ₉	ACO_{10}
	111 [- 30	36	30	30	30	30	30	30	30	30 -
	112	30	36	30	30	30	30	30	30	30	30
C _	113	28	35	28	30	30	30	28	28	28	28
3 =	114	26	26	26	26	26	26	26	26	26	26
	115	26	26	26	26	26	26	26	26	26	26
	116	26	26	26	26	26	26	25	25	26	25

		ACO_1	ACO_2	ACO ₃	ACO_4	ACO_5	ACO_6	ACO ₇	ACO ₈	ACO ₉	ACO_{10}
	223 [- 90	96	90	90	89	81	90	90	90	90
	224	61	96	89	89	88	65	61	59	57	71
	225	61	96	74	58	60	58	57	58	57	57
	226	59	95	73	57	59	57	56	58	57	57
	227	60	57	57	57	57	56	56	57	57	57
	228	60	57	57	57	57	56	56	57	54	57
	229	58	57	57	55	57	56	56	56	54	56
	230	57	57	57	55	57	52	56	54	54	56
	231	57	55	57	55	55	52	56	54	54	56
	232	57	55	55	51	54	50	52	51	54	48
s –	233	57	55	55	51	54	50	51	51	54	48
5 –	234	57	55	55	51	53	50	51	48	53	48
	235	57	55	54	51	53	50	51	48	50	48
	236	57	55	54	51	53	50	51	48	50	48
	237	57	55	54	51	53	50	51	48	50	48
	238	57	55	53	51	53	50	51	48	50	48
	239	56	55	53	50	53	50	51	48	50	48
	240	53	53	53	50	53	50	51	48	50	48
	241	53	53	53	50	53	50	51	48	50	48
	242	53	53	53	50	53	50	51	48	50	48
	243	53	53	53	50	53	50	51	48	50	48
	244	53	53	53	50	52	50	51	48	50	48

Table 2 The 22 \times 10 score matrix S for 500 \times 500 case (Example 2)

Table 3 The 19 \times 10 score matrix **S** for 700 \times 700 case (Example 3)

		ACO_1	ACO_2	ACO ₃	ACO_4	ACO_5	ACO ₆	ACO ₇	ACO_8	ACO ₉	ACO_{10}
	437 [173	173	173	173	173	118	168	172	261	172]
	438	173	173	173	173	173	118	112	117	260	172
	439	172	173	173	173	140	93	110	115	131	172
	440	172	173	173	173	115	93	110	114	111	162
	441	172	173	173	122	111	93	110	114	111	110
	442	172	173	173	114	111	93	110	112	111	110
	443	172	150	123	114	111	93	110	112	111	110
	444	124	112	112	106	107	93	110	102	111	105
	445	117	112	112	106	107	93	110	102	108	105
$\mathbf{S} =$	446	117	112	105	105	105	93	107	102	104	105
	447	117	112	105	105	105	93	105	102	102	105
	448	115	111	105	105	105	93	105	102	102	105
	449	115	111	105	105	105	93	102	99	102	105
	450	113	111	105	105	105	93	102	99	102	105
	451	113	109	105	105	105	93	102	99	97	105
	452	113	109	105	105	105	93	99	99	97	104
	453	113	109	105	105	105	93	99	99	97	104
	454	113	109	105	105	96	93	96	96	96	104
	455	106	106	105	105	96	93	96	96	96	97

```
Table 4 Matrix \mathbf{K}_{\approx PCR6}(\theta) for Example 1
```

 0.865
 0.821
 0.865
 0.790
 0.790
 0.790
 0.806
 0.806
 0.865
 0.806

 0.821
 0.928
 0.821
 0.950
 0.950
 0.950
 0.805
 0.805
 0.821
 0.805

 0.865
 0.821
 0.865
 0.790
 0.790
 0.790
 0.806
 0.806
 0.865
 0.806

 0.865
 0.821
 0.865
 0.790
 0.790
 0.790
 0.806
 0.806
 0.865
 0.806

 0.790
 0.950
 0.790
 1.000
 1.000
 0.795
 0.795
 0.790
 0.795

 0.790
 0.950
 0.790
 1.000
 1.000
 1.000
 0.795
 0.795
 0.790
 0.795

 0.790
 0.950
 0.790
 1.000
 1.000
 0.795
 0.795
 0.790
 0.795

 0.790
 0.950
 0.790
 1.000
 1.000
 0.795
 0.795
 0.790
 0.795

 0.806
 0.805
 0.806
 0.795
 0.795
 0.795
 0.843
 0.843
 0.806
 0.843

 0.806

Each row of **S** corresponds to the number of sensors used in WSN to cover the area as indicated in the first column at the left side of the score matrix. Each column of **S** corresponds to ACO_j algorithm used with j ants (j = 1, 2, ..., 10). Each element S_{ij} of **S** corresponds to the energy corresponding to this number of sensors and with the number of ants used for Multiple Objective ACO algorithm.

6.1 Application of Fast BF-ICrA in Example 1 (350 × 350 Points)

In this example, one sees from the score matrix of the Table 1 that ACO_1 , ACO_3 and ACO_9 algorithms perform equally for all alternatives (i.e. all rows) and they define a first group/cluster of methods providing exactly the same performances. Similarly, ACO_4 , ACO_5 and ACO_6 constitute a second group of algorithms. The third group is made of ACO_7 , ACO_8 and ACO_{10} algorithms. It is worth noting that these three groups { ACO_1 , ACO_3 , ACO_9 }, { ACO_4 , ACO_5 , ACO_6 }, and { ACO_7 , ACO_8 , ACO_{10} } differ only very slightly, whereas the ACO_2 algorithm (i.e the 2nd column of the score matrix **S**) differs a bit more from all the three aforementioned groups.

Example 1 with fast PCR6: If we apply the fast BF-ICrA method using approximate PCR6 fusion rule based on the canonical decomposition of the M = 6 dichotomous BBAs $(m_{jj'}^i(\theta), m_{jj'}^i(\bar{\theta}), m_{jj'}^i(\theta \cup \bar{\theta}))$, we get the matrix of mass of belief of agreement between criteria given in Table 4.⁶

The matrix of distances to full agreement based on fast BF-ICrA method, denoted by $\mathbf{D}_{\approx PCR6}(\theta)$, is given in Table 5.

In examining the Table 5, one sees that ACO1, ACO3 and ACO9 are at a small distance 0.134, with respect to other algorithms, so that they belong to the same

⁶All the numerical values presented in the matrices have been truncated at their 3rd digit for typesetting convenience.

Evaluation of MO-ACO Algorithms Using a New Fast ...

Table 5 Matrix $\mathbf{D}_{\approx PCR6}(\theta)$ with fast BF-ICrA for Example 1

	0.134	0.178	0.134	0.209	0.209	0.209	0.193	0.193	0.134	0.193
	0.178	0.071	0.178	0.049	0.049	0.049	0.194	0.194	0.178	0.194
	0.134	. 0.178	0.134	0.209	0.209	0.209	0.193	0.193	0.134	0.193
	0.209	0.049	0.209	0	0	0	0.204	0.204	0.209	0.204
	0.209	0.049	0.209	0	0	0	0.204	0.204	0.209	0.204
	0.209	0.049	0.209	0	0	0	0.204	0.204	0.209	0.204
	0.193	0.194	0.193	0.204	0.204	0.204	0.156	0.156	0.193	0.156
	0.193	0.194	0.193	0.204	0.204	0.204	0.156	0.156	0.193	0.156
	0.134	0.178	0.134	0.209	0.209	0.209	0.193	0.193	0.134	0.193
	0.193	0.194	0.193	0.204	0.204	0.204	0.156	0.156	0.193	0.156_
ļ	_0.193	0.194	0.193	0.204	0.204	0.204	0.150	0.150	0.195	0.150

Table 6 Matrix $\mathbf{D}_{Aver.}(\theta)$ with BF-ICrA using averaging rule for Example 1

0.084	0.082	0.084	0.081	0.081	0.081	0 1 5 6	0 1 5 6	0.084	0 156
0.001	0.002	0.001	0.001	0.001	0.001	0.150	0.150	0.001	0.150
0.082	0.030	0.082	0.016	0.016	0.016	0.142	0.142	0.082	0.142
0.084	0.082	0.084	0.081	0.081	0.081	0.156	0.156	0.084	0.156
0.081	0.016	0.081	0	0	0	0.138	0.138	0.081	0.138
0.081	0.016	0.081	0	0	0	0.138	0.138	0.081	0.138
0.081	0.016	0.081	0	0	0	0.138	0.138	0.081	0.138
0.156	0.142	0.156	0.138	0.138	0.138	0.198	0.198	0.156	0.198
0.156	0.142	0.156	0.138	0.138	0.138	0.198	0.198	0.156	0.198
0.084	0.082	0.084	0.081	0.081	0.081	0.156	0.156	0.084	0.156
0.156	0.142	0.156	0.138	0.138	0.138	0.198	0.198	0.156	0.198_

group and behave similarly. Same remarks holds for the group $\{ACO_4, ACO_5, ACO_6\}$ because its inter-distance is zero, and for the group $\{ACO_7, ACO_8, ACO_{10}\}$ because its inter-distance is 0.156. In a relative manner ACO₂ appears closer to $\{ACO_4, ACO_5, ACO_6\}$, than $\{ACO_1, ACO_3, ACO_9\}$ or $\{ACO_7, ACO_8, ACO_{10}\}$, which intuitively makes sense when comparing directly the columns of the matrix of Table 1.

Example 1 with averaging fusion: The matrix of distances to full agreement based on BF-ICrA method using average fusion rule, denoted by $\mathbf{D}_{\text{Aver.}}(\theta)$, is given in Table 6.

One sees that only the group $\{ACO_4, ACO_5, ACO_6\}$ can be clearly identified based on the averaging fusion rule. The other groups ACO_2 appears also close to $\{ACO_4, ACO_5, ACO_6\}$. But ACO_1 , ACO_3 and ACO_9 are closer to $\{ACO_4, ACO_5, ACO_6\}$ also than in-between. Same remarks holds for ACO_7 , ACO_8 , and ACO_{10} . So one sees that the averaging fusion rule is not recommended for making the BF-ICrA in this example.

6.2 Application of Fast BF-ICrA in Example 2 (500 × 500 Points)

Example 2 with fast PCR6: If we apply the fast BF-ICrA method using approximate PCR6 fusion rule based on the canonical decomposition of the M = 22 dichotomous BBAs $(m_{jj'}^i(\theta), m_{jj'}^i(\bar{\theta}), m_{jj'}^i(\theta \cup \bar{\theta}))$, we get the following matrix of distances to full agreement, denoted by $\mathbf{D}_{\approx PCR6}(\theta)$, given in Table 7.

Based on these results, one sees that no clear group can be identified but we emphasize in boldface in Table 7 the minimal value for each row of the distance matrix $\mathbf{D}_{\approx PCR6}(\theta)$ (diagonal elements excluded). We see that ACO₂ is at the farthest distance of ACO₁ because $D_{12}(\theta) = 0.376$, but in the mean time ACO₂ is at closest distance to ACO₁ because $D_{2j}(\theta) > 0.376$ (for j > 2) as shown in second line of Table 7. So we can conclude that ACO₂ is not close to any other algorithm in fact. If we choose a ad-hoc distance threshold, say for instance 0.28, then we can identify the group {ACO₁, ACO₇, ACO₈, ACO₉}.

Example 2 with averaging fusion: The matrix of distances to full agreement based on BF-ICrA method using average fusion rule, denoted by $\mathbf{D}_{\text{Aver.}}(\theta)$, is given in Table 8.

Based on the average fusion rule there is no clear clustering of algorithms. However based on shortest inter-distance we could make the following distinct pairwise groupings { ACO_2 , ACO_3 }, { ACO_6 , ACO_7 }, { ACO_4 , ACO_{10} }, { ACO_8 , ACO_9 } and { ACO_1 , ACO_5 } if necessary, but remember that average fusion rule cannot provide the best result as shown in Example 1.

Table 7 Matrix $\mathbf{D}_{\approx PCR6}(\theta)$ with fast BF-ICrA for Example 2

```
      0.158
      0.376
      0.338
      0.300
      0.286
      0.279
      0.247
      0.251
      0.225
      0.280

      0.376
      0.324
      0.426
      0.456
      0.437
      0.453
      0.457
      0.433
      0.435
      0.449

      0.338
      0.426
      0.407
      0.411
      0.382
      0.423
      0.418
      0.402
      0.393
      0.414

      0.300
      0.456
      0.411
      0.382
      0.423
      0.418
      0.402
      0.393
      0.414

      0.300
      0.456
      0.411
      0.382
      0.423
      0.418
      0.402
      0.393
      0.414

      0.300
      0.456
      0.411
      0.382
      0.423
      0.418
      0.402
      0.393
      0.414

      0.300
      0.456
      0.411
      0.349
      0.323
      0.381
      0.368
      0.370
      0.362
      0.363

      0.279
      0.453
      0.423
      0.381
      0.348
      0.316
      0.298
      0.317
      0.308
      0.308

      0.247
      0.457
      0.418
      0.368
      0.334
      0.298
      0.235
      0.265
      0.260
      0.303

      0.251
```

Evaluation of MO-ACO Algorithms Using a New Fast ...

Table 8 Matrix $\mathbf{D}_{Aver.}(\theta)$ with BF-ICrA using averaging rule for Example 2

0.3610.3160.3100.3110.3360.3000.3060.3160.3200.3090.3160.1250.1580.1980.2250.1870.2160.2250.2400.2060.3100.1580.1650.1850.2150.1780.2000.2150.2270.1930.3110.1980.1850.1830.2160.1810.1970.2170.2310.1920.3360.2250.2150.2160.2430.2140.2310.2490.2610.2260.3000.1870.1780.1810.2140.1590.1750.1940.2100.1760.3060.2160.2000.1970.2310.1750.1810.2020.2160.1860.3160.2250.2150.2170.2490.1940.2020.2150.2290.2040.3200.2400.2270.2310.2610.2160.2290.2330.2220.3090.2060.1930.1920.2260.1760.1860.2040.2220.183

Table 9 Matrix $\mathbf{D}_{\approx PCR6}(\theta)$ with fast BF-ICrA for Example 3

0.3130.3880.4650.4980.4690.5000.4260.4510.4980.4770.3880.3390.4030.4960.4610.5000.4210.4400.4970.4640.4650.4030.3480.4930.4560.5000.4160.4370.4950.4570.4980.4960.4930.3620.3850.5000.3760.3910.4700.3030.4690.4610.4560.3850.2300.3800.2560.2880.3000.3240.5000.5000.5000.5000.38000.3120.3560.3080.5000.4260.4210.4160.3760.2560.3120.1370.1850.2720.3300.4510.4400.4370.3910.2880.3060.1850.2050.3140.3510.4980.4970.4950.4700.3000.3080.2720.3140.2830.4380.4770.4640.4570.3030.3240.5000.3300.3510.4380.228

6.3 Application of Fast BF-ICrA in Example 3 (700 × 700 Points)

Example 3 with fast PCR6: If we apply the fast BF-ICrA method using approximate PCR6 fusion rule based on the canonical decomposition of the M = 19 dichotomous BBAs $(m_{jj'}^i(\theta), m_{jj'}^i(\bar{\theta}), m_{jj'}^i(\theta \cup \bar{\theta}))$, we get the matrix of distances to full agreement, denoted by $\mathbf{D}_{\approx PCR6}(\theta)$, given in Table 9.

We observe that the average distance between ACO algorithms is much higher than in Tables 5 and 7 of Examples 1 and 2. This shows clearly the difficulty to precisely identify the clusters of similar algorithms because only few ACO algorithms perform actually very well for this third example. Eventually, and based on shortest interdistance we could make the first pairwise group {ACO₇, ACO₈} because $D_{78}(\theta) =$ 0.185 is the minimal inter-distance we have between the ACO algorithms. Once the rows and columns of Table 9 corresponding to ACO₇ and ACO₈ are eliminated, then the second best group will be {ACO₅, ACO₉} because $D_{59}(\theta) = 0.300$. Similarly, we will get the group {ACO₄, ACO₁₀} because $D_{4,10}(\theta) = 0.303$, and then the group **Table 10** Matrix $\mathbf{D}_{\text{Aver.}}(\theta)$ with BF-ICrA using averaging rule for Example 3

0.1700.1540.1420.2210.3510.3500.3920.3450.3320.2980.1540.1200.0920.1670.3210.2950.3690.3130.2900.2610.1420.0920.0420.1140.2890.2370.3420.2790.2420.2240.2210.1670.1140.0540.2550.1390.3270.2600.1840.1770.3510.3210.2890.2550.3390.2450.3910.3550.2870.3240.3500.2950.2370.1390.24500.3040.2420.1150.2470.3920.3690.3420.3270.3910.3040.3900.3680.3360.3870.3450.3130.2790.2600.3550.2420.3680.3280.2880.3410.3320.2900.2420.1840.2870.1150.3360.2880.1900.2790.2980.2610.2240.1770.3240.2470.3870.3410.2790.261

{ACO₁, ACO₂} because $D_{12}(\theta) = 0.388$. Finally we could also cluster ACO₃ with ACO₆ because $D_{36}(\theta) = 0.500$, although this distance of agreement is quite large to be considered as a trustable cluster.

Example 3 with averaging fusion: The matrix of distances to full agreement based on BF-ICrA method using average fusion rule, denoted by $\mathbf{D}_{\text{Aver.}}(\theta)$, is given in Table 10.

Surprisingly, the use of averaging rule provides in this example lower distance values on average with respect to values given in Table 9. However no clear clustering of algorithms can be made because only few ACO algorithms perform actually very well for this third example. If we adopt the pairwise strategy to cluster algorithms, we will obtain now as first group {ACO₂, ACO₃} because $D_{23}(\theta) = 0.092$, as second group {ACO₆, ACO₉} because $D_{69}(\theta) = 0.115$, as third group {ACO₄, ACO₁₀} because $D_{4,10}(\theta) = 0.177$, as fourth group {ACO₁, ACO₈} because $D_{18}(\theta) = 0.345$, and finally we could also cluster ACO₅ with ACO₇ because $D_{57}(\theta) = 0.391$. One sees that there is no strong correlation between results obtained from BF-ICrA based on fast PCR6 and those based on averaging rule, which is not surprising because the rules are totally different. Nevertheless the group {ACO₄, ACO₁₀} is agreed by both methods here.

7 Application to Workforce Planning Problem (WPP)

In this section we present a new application of our new Fast BF-ICrA method for solving the workforce planning problem (WPP). This problem has been addressed recently by Fidanova et al. in [24] using the classical Atanassov's ICrA method [2–5]. Before presenting our new results, it is necessary to present briefly the WPP.

7.1 The Workforce Planning Problem (WPP)

The workforce planning is a part of the human resource management. It includes multiple level of complexity, therefore it is a hard optimization problem (NP-hard). This problem consists of two decision sets: selection and assignment. The first set shows selected employees from available workers. The assignment set shows which worker which job will perform. The aim is to fulfill the work requirements with minimal assignment cost. Such hard optimization problem with strong constraints is usually impossible to solve with exact methods or traditional numerical methods for instances with realistic size and that is why these methods (exact or numerical) can be applied only on some simplified variants of the original problem (see [24] for a detailed bibliography of the existing methods). One must emphasize that the convex optimization methods are not applicable complex non-linear workforce planning problems. Nowadays, nature-inspired metaheuristic methods receive great attention [25–29]. In the WPP considered heresome heuristic method including genetic algorithm [31, 32], memetic algorithm [30], scatter search [31] etc., have been already applied. So far the Ant Colony Optimization (ACO) algorithm is proved to be very effective solving various complex optimization problems [33, 34]. In our previous work [35] we did propose ACO algorithm for workforce planning. We have considered the variant of the workforce planning problem proposed in [31]. More recently in [24] we proposed a hybrid ACO algorithm which is a combination of ACO with a local search procedure and the classical InterCriteria Analysis (ICrA) was used to analyze the algorithm performance according the local search procedures and to study the correlations between the different variants in order to improve the algorithm performance for solving efficiently the WPP.

In this section we solve the WPP proposed in [31, 36]. The set of jobs $J = \{1, ..., m\}$ must be completed during a fixed period of time. The job *j* requires d_j hours to be completed. $I = \{1, ..., n\}$ is the set of workers, candidates to be assigned. The minimal number of hours that every job must require by every assigned worker is h_{min} . Availability of the worker *i* is s_i hours. One worker can be assigned to maximum j_{max} jobs. The set A_i shows the jobs, that worker *i* is qualified. Maximum *t* workers can be assigned during the planed period, or at most *t* workers may be selected from the set *I* of workers. The selected workers need to be capable to complete all the jobs. The aim is to find feasible solution, that optimizes a given objective function.

Let c_{ij} is the cost of assigning the worker *i* to the job *j*. The mathematical model of the WPP can be described by the following variables

$$x_{ij} \triangleq \begin{cases} 1, & \text{if worker } i \text{ is assigned to job } j, \\ 0, & \text{otherwise.} \end{cases} \text{ and } y_i \triangleq \begin{cases} 1, & \text{if worker } i \text{ is selected,} \\ 0, & \text{otherwise.} \end{cases}$$

 $z_{ij} \triangleq$ number of hours that worker *i* is assigned to perform job *j*

 $Q_j \triangleq$ set of workers qualified to perform job j

The WPP consists to minimize the total assignment cost under some constraints, more precisely we want to

$$\text{Minimize} \sum_{i \in I} \sum_{j \in A_i} c_{ij} . x_{ij}$$
(25)

subject to the following constraints

$$\sum_{i \in A_i} z_{ij} \le s_i . y_i, \qquad \text{with } i \in I \qquad (26)$$

$$\sum_{i \in O_j} z_{ij} \ge d_j \qquad \text{with } j \in J \qquad (27)$$

$$\sum_{j \in A_i} x_{ij} \le j_{max}.y_j \qquad \text{with } i \in I \qquad (28)$$

$$h_{min}.x_{ij} \le z_{ij} \le s_i.x_{ij} \qquad \text{with } i \in I, \ j \in A_i$$
(29)

$$\sum_{i \in I} y_i \le t \tag{30}$$

where $x_{ij} \in \{0, 1\}$, $y_i \in \{0, 1\}$ and $z_{ij} \ge 0$ with $i \in I$ and $j \in A_i$.

The constraint (26) stipulates that number of hours for each selected worker is limited, and the constraint (27) stipulates that the work must be totally achieved. The number of jobs that every worker can perform is limited according to the constraint (28). The inequality (29) stipulates the minimal number of hours that every job must require by every assigned worker. The number of assigned workers is limited by the constraint (30). WPP is difficult to solve because of very restrictive constraints especially the relation between the parameters h_{min} and d_j . It is easier to solve (to find feasible solution) when the problem is structured (when d_j is a multiple of h_{min}) than for unstructured problems (when d_j and h_{min} are not related).

Based on our previous works [24, 35], we will apply the ACO algorithm for workforce planning problem coupled with different local search procedures and based on our new fast BF-ICrA approach. One of the main points of the ant algorithm is the proper representation of the problem by graph. In our case the graph of the problem is 3 dimensional and the node (i, j, z) corresponds worker with number *i* to be assigned to the job *j* for time *z*. The graph of the problem is asymmetric, because the maximal value of *z* depends of the value of *j*, different jobs needs different time to be completed. At the beginning of every iteration every ant starts to construct their solution, from random node of the graph of the problem. For every ant, three random numbers are generated. The first random number corresponds to a worker we randomly select in the interval [0, ..., n]. The second random number in the interval [0, ..., m] corresponds to the job that must be done by the worker. We check if the worker is qualified to perform the job, if not we randomly choose for him/her another compatible job. The third random number in $[h_{min}, ..., min\{d_j, s_i\}]$ corresponds to the number of hours allocated to worker *i* to performs the job *j*. After, the ant applies the transition probability rule to include next nodes in the partial solution, until a feasible solution is completed, or there is no possibility to include new node. The heuristic information η_{ijl} is problem dependent and is used for better management of the search process. In order to assign the most cheapest worker as longer as possible, we define the η_{ijl} parameter of the ACO algorithm by:

$$\eta_{ijl} = \begin{cases} l/c_{ij} & \text{if } l = z_{ij} \\ 0 & \text{otherwise} \end{cases}$$
(31)

The node with a highest probability is chosen to be the next node, included in the solution. When there are several candidate nodes with a same probability, the next node is randomly drawn among these candidates. When some move of the ant do not meets the problem constraints, then the probability of this move is set to be 0. If for all possible nodes the value of the transition probability is 0, it is impossible to include new node in the solution and the solution construction stops. When the constructed solution is feasible, the value of the objective function is the sum of the assignment cost of the assigned workers. If the constructed solution is not feasible, the value of the objective function is not feasible solutions and depose a new pheromone on the elements of their solutions. More precisely, the main pheromone trail update rule is given by

$$\tau_{i,j} \leftarrow \rho \tau_{i,j} + \Delta \tau_{i,j}, \tag{32}$$

where ρ decreases the value of the pheromone (like the evaporation in a nature), and where the new added pheromone $\Delta \tau_{i,j}$ is equal to the reciprocal value of the objective function given by

$$\Delta \tau_{i,j} = \frac{\rho - 1}{\min \sum_{i \in I} \sum_{j \in A_i} c_{ij} . x_{ij}}$$
(33)

The nodes of the graph belonging to solutions with less value of the objective function, receive more pheromone than others and become more desirable in the next iteration. At the end of every iteration we compare the iteration best solution with the best solution obtained so far. If the best solution from the current iteration is better than the best so far solution (global best solution), we update the global best solution with the current iteration best solution. The end condition used in our algorithm is the number of iterations.

In order to decrease the time to find the best solution and eventually to improve the achieved solutions, we use the local search proposed in [24] because it increases the possibility to find feasible solution and thus the chance to improve current solution. If the solution is not feasible we remove part of the assigned workers and after that we assign in their place new workers. The workers which will be removed are chosen randomly. On this partial solution we assign new workers applying the rules of ant

algorithm. The ACO algorithm (denoted ACO_1) is a stochastic algorithm, therefore the new constructed solution is different from previous one with a high probability. We have proposed three variants for the local search procedure:

- ACO₂: with removed workers are quarter of all assigned workers (ACO quarter);
- ACO₃: with removed workers are half of all assigned workers; (ACO half)
- ACO₄: with all assigned workers are removed and the solution is constructed from the beginning (ACO restart).

7.2 WPP Addressed in This Paper

We use the artificially generated problem instances considered in [31]. The characteristics of this WPP are: $m = 20, n = 20, t = 10, s_i \in [50, 70], j_{max} \in [3, 5]$, and $h_{\min} \in [10, 15]$. The set of test problems consists of ten structured problems, and ten unstructured problems. For structured problems d_i is proportional to h_{\min} . In our previous work [35] we have shown that our ACO algorithm outperforms the genetic and scatter search algorithms presented in [31]. The number of iterations is a stopping criteria for our hybrid ACO algorithm. The ACO parameter settings are as follows: $\rho = 0.5$, $\tau_0 = 0.5$, a = 1, b = 1, the number of ants is 20, and the maximum number of iterations is fixed to 100. Further, the problem instances are enumerated as $S20_{01}$ to $S20_{10}$ for the ten structured problems using 20 ants, and as $U20_{01}$ to $U20_{10}$ for the ten unstructured problems using 20 ants. The WPP has very restrictive constraints. Therefore only 2–3 of the ants, per iteration, find feasible solution. Sometimes some iterations do not generate a feasible solution. Its complicates the search process. Our aim is to decrease the number of unfeasible solutions, in order to increase the possibility for ants to find good solutions, and therefore to decrease the needed number of iterations to get a good solution. We observe that after the local search procedure applied on the first iteration, the number of unfeasible solutions in a next iterations decreases. It is another reason why the computation time does not increase significantly. We are analyzing four cases: (1) without local search procedure (ACO_1) , and with the three aforementioned variant for the local search procedure (ACO₂, ACO₃ and ACO₄). We did perform 30 independent runs for each of the four cases (because the algorithm is stochastic) to guarantee the robustness of the average results. We apply ANOVA test for statistical analysis to guarantee the significance of the achieved results. The obtained results are presented in Tables 11 and 12. Tables 11 presents the minimal number of iterations to achieve the best solution and Table 12—the computation time needed to achieve the best solution.

Evaluation of MO-ACO Algorithms Using a New Fast ...

Algo type	ACO1	ACO ₂	ACO ₃	ACO ₄
S20 ₀₁	13	10	15	16
S20 ₀₂	17	28	28	35
S20 ₀₃	29	27	37	33
S20 ₀₄	77	66	41	23
S20 ₀₅	21	21	4	14
S20 ₀₆	21	13	20	1
S20 ₀₇	43	34	29	40
S20 ₀₈	57	15	50	33
S20 ₀₉	36	28	22	48
S20 ₁₀	26	19	16	35
U20 ₀₁	17	23	11	21
U20 ₀₂	17	16	12	15
U20 ₀₃	28	22	20	48
U20 ₀₄	41	56	28	28
U20 ₀₅	14	20	15	4
U20 ₀₆	46	46	45	20
U20 ₀₇	29	44	37	39
U20 ₀₈	11	14	16	26
U20 ₀₉	46	68	41	42
U20 ₁₀	30	30	30	30

 Table 11 Minimal number of iterations to achieve the best solution

7.3 Results of WPP Obtained with Fast BF-ICrA

The test problems $S20_{01}$ to $S20_{10}$ and $U20_{01}$ to $U20_{10}$ are considered as objects, and algorithms ACO₁, ACO₂, ACO₃ and ACO₄ as criteria. We did apply the fast BF-ICrA approach to identify the relation between the proposed ACO hybrid algorithms. The hybrid algorithms are compared based on the obtained results according to the number of iterations (Table 11), and according to the computation time (Table 12).

Based on the values of Table 11, we get the distances between ACO algorithms reported in matrix $\mathbf{D}_{\approx PCR6}^{it}(\theta)$ with fast PCR6 rule, and the matrix $\mathbf{D}_{Aver.}^{it}(\theta)$ when using the simple averaging rule of combination.

$$\mathbf{D}_{\approx PCR6}^{it}(\theta) = \begin{bmatrix} 0.3181 \ 0.4419 \ 0.4218 \ 0.4867 \\ 0.4419 \ 0.3743 \ 0.4812 \ 0.4925 \\ 0.4218 \ 0.4812 \ 0.3316 \ 0.4802 \\ 0.4867 \ 0.4925 \ 0.4802 \ 0.3575 \end{bmatrix}$$
$$\mathbf{D}_{Aver.}^{it}(\theta) = \begin{bmatrix} 0.3736 \ 0.4018 \ 0.4253 \ 0.4885 \\ 0.4018 \ 0.3450 \ 0.4372 \ 0.4834 \\ 0.4253 \ 0.4372 \ 0.3933 \ 0.4791 \\ 0.4885 \ 0.4834 \ 0.4791 \ 0.3792 \end{bmatrix}$$

Algo type	ACO1	ACO ₂	ACO ₃	ACO ₄
S20 ₀₁	1.20	0.94	0.96	2.29
S20 ₀₂	3.94	8.62	6.22	14.75
S20 ₀₃	5.19	5.79	11.93	3.06
S20 ₀₄	3.06	16.66	7.00	6.11
S20 ₀₅	0.63	1.312	0.396	0.90
S20 ₀₆	2.48	2.12	2.64	0.59
S20 ₀₇	6.78	4.82	6.78	6.60
S20 ₀₈	6.38	1.87	10.42	8.59
S20 ₀₉	4.68	5.31	4.48	5.70
S2010	1.45	1.25	1.28	10.43
U20 ₀₁	3.10	4.48	2.00	2.50
U20 ₀₂	1.98	1.18	0.92	0.93
U20 ₀₃	2.14	2.41	1.54	1.88
U20 ₀₄	3.08	3.35	3.12	3.47
U20 ₀₅	1.55	2.76	2.06	1.056
U20 ₀₆	10.92	11.8	4.36	7.05
U20 ₀₇	4.22	6.55	3.54	3.27
U20 ₀₈	0.89	1.48	1.19	1.77
U20 ₀₉	6.48	8.72	7.10	7.21
U20 ₁₀	3.74	3.88	3.69	10.00

 Table 12
 Computation time needed to achieve the best solution

The analysis of values of $\mathbf{D}_{\approx PCR6}^{it}(\theta)$ matrix shows clearly that none of these algorithms are close of each others because their distances are much bigger than zero. This is because the BBAs of the Inter-Criteria matrix **K** are in fact quite ambiguous (i.e. the focal elements θ and $\bar{\theta}$ have comparable mass values), even if there is only a little mass committed to uncertainty $\theta \cup \bar{\theta}$. However, ACO₄ is more distant of other algorithms which indicates a different behavior compared to ACO₁, ACO₂ and ACO₃, as already mentioned in [24]. The averaging fusion rule makes the distances values more close which makes the separability of criteria even more difficult to identify.

Based on the numerical values of Table 12, we get the distances between ACO algorithms reported in matrix $\mathbf{D}_{\approx PCR6}^{sec}(\theta)$ with fast PCR6 rule, and the matrix $\mathbf{D}_{Aver}^{sec}(\theta)$ when using the simple averaging rule of combination.

$$\mathbf{D}_{\approx PCR6}^{sec}(\theta) = \begin{bmatrix} 0.3021 & 0.4452 & 0.4186 & 0.4603 \\ 0.4452 & 0.3509 & 0.4747 & 0.4807 \\ 0.4186 & 0.4747 & 0.3600 & 0.4798 \\ 0.4603 & 0.4807 & 0.4798 & 0.3595 \end{bmatrix}$$

$$\mathbf{D}_{Aver.}^{sec}(\theta) = \begin{bmatrix} 0.3841 & 0.4030 & 0.4034 & 0.4276 \\ 0.4030 & 0.3215 & 0.3976 & 0.4108 \\ 0.4034 & 0.3976 & 0.3475 & 0.4156 \\ 0.4276 & 0.4108 & 0.4156 & 0.3378 \end{bmatrix}$$

Based on the values of $\mathbf{D}_{\approx PCR6}^{sec}(\theta)$ matrix, one sees also that there is no clear clustering of the different ACO algorithms because the distances values are much bigger than zero, however one can also reasonably infer that ACO₄ shows a different behavior compared to ACO₁, ACO₂ and ACO₃, as inferred in the previous analysis based on input values of Table 11. The difference comes from deleting of infeasible solutions and constructing new solutions from beginning. Thus the new constructed solutions can be very different from previous ones and as a consequence the change of the pheromone can be significant. In summary, even there are some numerical differences when the hybrid algorithms are compared based on minimal number of iterations and based on computation time, our conclusions of (fast) BF-ICrA are consistent.

8 Conclusions

The fast Belief Function based Inter-Criteria Analysis method, using the canonical decomposition of basic belief assignments defined on a dichotomous frame of discernment was applied, tested and analysed in this paper for two applications: (1) for evaluating the Multiple-Objective Ant Colony Optimization (MO-ACO) algorithm for Wireless Sensor Networks (WSN) deployment, and (2) for evaluating the Multiple-Objective Ant Colony Optimization (MO-ACO) algorithm for the Workforce Planning Problem (WPP).

For our first application (WSN deployment), based on the BF-ICrA outcomes we have shown a very high correlation with fast PCR6 rule for the ACO₁, ACO₃ and ACO₉ group, for the ACO₄, ACO₅ and ACO₆ group, and for the ACO₇, ACO₈ and ACO₁₀ group of algorithms in Example 1 (case of size 350×350) as intuitively expected. This is because the considered ACO algorithms can solve the problem with good solution quality in Example 1. These high correlations were not observed in the other two cases for Example 2 (case of size 500×500) and 3 (case of size 700×700) because only few ACO algorithms perform actually very well for these examples. So, if we considered results in case of larger problem sizes, the BF-ICrA results show that the number of ants has the significant influence on the obtained results, as already pointed out in [22].

For our second application (WPP), based on the fast BF-ICrA results we have shown that the third variant of ACO approach, i.e. ACO_4 (ACO restart) has a quite distinct behavior with respect to the methods ACO_1 , ACO_2 and ACO_3 .

Acknowledgements This work is partially supported by the grant No BG05M20P001-1.001-0003, financed by the Science and Education for Smart Growth Operational Program (2014-2020) and co-financed by the European Union through the European structural and Investment funds and the Bulgarian scientific fund by the grant DN 12/5.

References

- Dezert, J., Tchamova, A., Han, D., Tacnet, J.-M.: Simplification of multi-criteria decisionmaking using inter-criteria analysis and belief functions. In: Proceedings of Fusion 2019 International Conference on Information Fusion, Ottawa, Canada, July 2–5 (2019)
- Atanassov, K., Mavrov, D., Atanassova, V.: Intercriteria decision making: a new approach for multicriteria decision making, based on index matrices and intuitionistic fuzzy sets. Issues IFSs GNs 11, 1–8 (2014)
- Atanassov, K., Atanassova, V., Gluhchev, G.: InterCriteria analysis: ideas and problems. Notes IFS 21(1), 81–88 (2015)
- 4. Atanassov, K., et al.: An approach to a constructive simplification of multiagent multicriteria decision making problems via intercriteria analysis. C.R. de l'Acad. Bulgare des Sci. **70**(8) (2017)
- Atanassov, K.: Intuitionistic Fuzzy Logics. Studies in Fuzziness and Soft Computing, vol. 351. Springer, Berlin (2017) ISBN 978-3-319-48952-0
- Dezert, J., Smarandache, F., Tchamova, A., Han, D.: Fast fusion of basic belief assignments defined on a dichotomous frame of discernment. In: Proceedings of Fusion 2020 (Online) Conference, Pretoria, South Africa (2020)
- Dezert, J., Smarandache, F.: Canonical decomposition of dichotomous basic belief assignment. Int. J. Intell. Syst. 1–21 (2020)
- Dezert, J., Smarandache, F.: Canonical decomposition of basic belief assignment for decisionmaking support. In: Proceedings of MDIS 2020 (7th international conference on Modelling and Development of Intelligent Systems), Lucian Blaga University of Sibiu, Sibiu, Romania, Oct. 22–24 (2020)
- 9. Shafer, G.: A Mathematical Theory of Evidence. Princeton University Press, Princeton (1976)
- Dezert, J., Wang, P., Tchamova, A.: On the validity of Dempster-Shafer theory. In: Proceedings of Fusion 2012, Singapore, July 9–12 (2012)
- 11. Tchamova, A., Dezert, J.: On the behavior of dempster's rule of combination and the foundations of dempster-shafer theory. In: IEEE IS-2012, Sofia, Bulgaria, Sept. 6–8 (2012)
- 12. Dezert, J., Tchamova, A.: On the validity of Dempster's fusion rule and its interpretation as a generalization of Bayesian fusion rule. Int. J. Intell. Syst. **29**(3), 223–252 (2014)
- 13. Smarandache, F., Dezert, J.: On the consistency of PCR6 with the averaging rule and its application to probability estimation. In: Proceedings of Fusion 2013, Istanbul, Turkey (2013)
- Smarandache, F., Dezert, J. (eds.): Advances and Applications of DSmT for Information Fusion, vol. 1–4. American Research Press, Santa Fe, (2004–2015). http://www.onera.fr/staff/jeandezert?page=2
- 15. Smarandache, F., Dezert, J., Tacnet, J.-M.: Fusion of sources of evidence with different importances and reliabilities. In: Proceedings of Fusion 2010 Conference, Edinburgh, UK (2010)
- 16. https://bfasociety.org/
- 17. Yager, R.: On the Dempster-Shafer framework and new combination rules. Inf. Sci. **41**, 93–138 (1987)
- 18. Dubois, D., Prade, H.: Representation and combination of uncertainty with belief functions and possibility measures. Comput. Intell. 4 (1988)
- Fidanova, S., Dezert, J., Tchamova, A.: Inter-criteria analysis based on belief functions for GPS surveying problems. In: Proceedings of IEEE International Conference on INnovations in Intelligent SysTems and Applications (INISTA 2019), Sofia, Bulgaria, July 3–5 (2019)

- 20. Dezert, J., Han, D., Yin, H.: A new belief function based approach for multi-criteria decisionmaking support. In: Proceedings of Fusion 2016 Conference
- Han, D., Dezert, J., Yang, Y.: Measures, new distance, of evidence based on belief intervals. In: Proceedings of Belief, Oxford, p. 2014 (2014)
- 22. Fidanova, S., Roeva, O.: Multi-objective ACO algorithm for WSN layout: intercriteria analysis. In: Large-Scale Scientific Computing. Springer, Berlin (2020)
- 23. Dorigo, M., Stutzle, T.: Ant Colony Optimization. MIT Press, Cambridge (2004)
- Fidanova, S., Roeva, O., Luque, G., Paprzycki, M.: InterCriteria analysis of different hybrid ant colony optimization algorithms for workforce planning. Recent Advances in Computational Optimization, In: Fidanova S. (eds) Recent Advances in Computational Optimization. Studies in Computational Intelligence, vol. 838, pp. 61–81 (2020)
- Albayrak, G., Özdemir, I.: A state of art review on metaheuristic methods in time-cost trade-off problems. Int. J. Struct. Civil Eng. Res. 6(1), 30–34 (2017)
- Mucherino, A., Fidanova, S., Ganzha, M.: Introducing the environment in ant colony optimization, recent advances in computational optimization, studies in computational. Intelligence 655, 147–158 (2016)
- 27. Roeva, O., Atanassova, V.: Cuckoo search algorithm for model parameter identification. Int. J. Bioautom. **20**(4), 483–492 (2016)
- Tilahun, S.L., Ngnotchouye, J.M.T.: Firefly algorithm for discrete optimization problems: a survey. J. Civil Eng. 21(2), 535–545 (2017)
- 29. Toimil, D., Gómes, A.: Review of metaheuristics applied to heat exchanger network design. Int. Trans. Oper. Res. **24**(1–2), 7–26 (2017)
- 30. Soukour, A., Devendeville, L., Lucet, C., Moukrim, A.: A Memetic algorithm for staff scheduling problem in airport security service. Expert Syst. Appl. **40**(18), 7504–7512 (2013)
- Alba, E., Luque, G., Luna, F.: Parallel metaheuristics for workforce planning. J. Math. Modell. Algor. 6(3), 509–528 (2007). Springer
- 32. Li ,G., Jiang, H., He, T.: A genetic algorithm-based decomposition approach to solve an integrated equipment-workforce-service planning problem. Omega **50**, 1–17 (2015). Elsevier
- Grzybowska, K., Kovács, G. (2014) Sustainable supply chain supporting tools. In: Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, vol. 2, pp. 1321–1329 (2014)
- Fidanova, S., Roeva, O., Paprzycki, M., Gepner, P.: InterCriteria analysis of ACO start strategies. In: Proceedings of the 2016 Federated Conference on Computer Science and Information Systems, pp. 547–550 (2016)
- Fidanova, S., Luquq, G., Roeva, O., Paprzycki, M., Gepner, P.: Ant colony optimization algorithm for workforce planning. In: FedCSIS'2017, IEEE Xplorer, IEEE catalog number CFP1585N-ART, pp. 415–419 (2017)
- Glover, F., Kochenberger, G., Laguna, M., Wubbena, T.: Selection and assignment of a skilled workforce to meet job requirements in a fixed planning period. In: MAEB'04, pp. 636–641 (2004)