DEFAULT REASONING BY ANT COLONY OPTIMIZATION

MIHAIELA LUPEA

Abstract. Drawing conclusions from incomplete information by making default assumptions represents default reasoning. Default logics, a class of nonmonotonic logical systems, formalize this type of reasoning using special inference rules, the defaults. During the inferential process, a default theory is extended with plausible conclusions (beliefs) obtaining default extensions. The very high theoretical complexity of the extension computation problem suggests the use of non-deterministic techniques for an efficient computation. In this paper we propose a uniform theoretical approach of the extension computation problem for all default logics (classical, justified, constrained, rational) applying Ant Colony Optimization metaheuristic.

1. Introduction

A lot of applications from Artificial Intellingence domain suppose reasoning with incomplete information. The specificity of this reasoning process, the *nonmonotonicity*, imposes that in the light of new information, some already derived conclusions (which are only consistent, not necessarily true) to be invalidated.

A special case of nonmonotonig reasoning, *default reasoning*, uses reasoning patterns of the form: "in the absence of information to the contrary of... it is consistent to assume that...". In the deductive process, default assumptions are applied in order to derive conclusions (called *beliefs*).

A class of nonmonotonic logical systems, *default logics* was introduced to formalize the default reasoning. Based on first-order logic, default logics use special inference rules, called *defaults*, to model the above nonmonotonic reasoning patterns. The differences among the versions (classical [10], justified [5], constrained [11], rational [8]) of default logic are caused by the semantics of the defaults.

Received by the editors: November 1, 2009.

2010 *Mathematics Subject Classification.* 03B79, 68T15, 68T27, 68T20.

1998 *CR Categories and Descriptors.* I.2[**Artificial Intelligence**]: Logics in artificial intelligence – *default logics, nonmonotonic reasoning*.

Key words and phrases. default logics, nonmonotonic reasoning, ant colony optimization.

Default logics provide a very expressive representation of an incomplete knowledge base (default theory), ruled by laws that are true with a few exceptions, using a simple syntactic formalism (first-order formulas and the defaults). The default reasoning process consists of combining the classical deduction with the defaults in order to derive new facts (beliefs), and obtain the *default extensions*, even if some information are not available.

This great power of the inferential process causes a high level of theoretical complexity. The problem of computing the extensions (classical, justified, constrained, rational) of a default theory is NP^{NP}-complete, this class belonging to the second level of the polynomial hierarchy of complexity classes based on calculus with oracles. For an efficient computation non-deterministic approaches must be used.

Automated proof systems for default logics proposed in the literature [1, 2, 6, 12] and based on the well known classical theorem proving methods as resolution, connection method, semantic tableaux method, have good performances only for particular classes of default theories and are not efficient for general non-trivial default theories.

In the paper [9] new generation systems for default reasoning were introduced. These are based on heuristics such as Genetic Algorithms, Ant Colony Optimization and Local Search, in order to overcome the high complexity and to obtain efficient reasoning systems. ANTDEL [9] is a system which uses Ant Colony Optimization to compute the classical default extension of a default theory that is equivalent to a logic program.

Inspired from the good performances of ANTDEL, in this paper we propose a uniform theoretical approach of the *extension computation problem* (ECP) for all versions (classical, justified, constrained, rational) of default logic using Ant Colony Optimization metaheuristic.

The paper is structured as follows. In Section 2 the main theoretical aspects of default logics are presented. A heuristic approach of the extension computation problem for default logics (classical, justified, constrained, rational) is introduced in Section 3. In section 4 an Ant Colony Optimization based procedure to compute all types of extensions for a default theory is proposed. Conclusions and future work are outlined in Section 5.

2. Default logics

Definition 1. [10] A *default theory* $\Delta = (D, W)$ consists of a set D of *default rules* and W, a set of consistent first-order logic formulas (the *facts*). A *default* has the form $d = \frac{\alpha:\beta_1,\ldots,\beta_m}{\gamma}$, where: α is called *prerequisite*, $\beta_1, \ldots, \beta_m$ are called *justifications* and γ is called *consequent*.

A default $d = \frac{\alpha:\beta_1,\ldots,\beta_m}{\gamma}$ can be applied and thus derive γ if α is believed and it is consistent to assume $\beta_1, \ldots, \beta_m$ (meaning that $\neg\beta_1, \ldots, \neg\beta_m$ are not believed).

Using the classical inference rules and the defaults, the set of facts, W, can be extended with new formulas, called *nonmonotonic theorems* (*beliefs*), obtaining *extensions*. The set of all the defaults used in the construction of an extension is called the *generating default set* for that extension.

Default theories can be represented by unitary theories (all the defaults have only one justification), in such a way that extensions (classical, justified, constrained, rational) are preserved. In the paper we will use only unitary default theories and the following notations:
$d = \frac{\alpha:\beta}{\gamma}$ - a default, $Prereq(d) = \alpha$, $Justif(d) = \beta$, $Conseq(d) = \gamma$,
$Prereq(D) = \bigcup_{d \in D} Prereq(d)$, $Justif(D) = \bigcup_{d \in D} Justif(d)$,
$Conseq(D) = \bigcup_{d \in D} Conseq(d)$,
$Th(U) = \{A | U \vdash A\}$, the classical deductive closure of the set U of formulas.

Definition 2. [11] A set X of defaults is *grounded* in the set of facts W if there is an enumeration $\langle d_i \rangle_{i \in I}$ of the defaults from X such that:
$\forall i \in I$, $W \cup Prereq(\{d_0, d_1, ..., d_{i-1}\}) \vdash Prereq(d_i)$.

The following theorems provide global characterizations for default extensions using the generating default sets.

Theorem 1. [11] Let (D, W) be a default theory, and let E be a set of formulas. E is a ***classical extension*** of (D, W) if and only if $E = Th(W \cup Conseq(D'))$ for a maximal set $D' \subseteq D$ such that D' is grounded in W and the conditions:

– $\forall d = \frac{\alpha:\beta}{\gamma} \in D'$: $W \cup Conseq(D') \cup \{\beta\}$ is consistent;

– $\forall d = \frac{\alpha:\beta}{\gamma} \notin D'$: $W \cup Conseq(D') \cup \{\beta\}$ is inconsistent **or**
$W \cup Conseq(D') \cup \{\neg\alpha\}$ is consistent

are satisfied.

Theorem 2. [6] Let (D, W) be a default theory, and let E, J be sets of formulas. (E, J) is a ***justified extension*** of (D, W) if and only if $E = Th(W \cup Conseq(D'))$ and $J = Justif(D')$ for a maximal set $D' \subseteq D$ such that D' is grounded in W and the conditions:

$\forall d = \frac{\alpha:\beta}{\gamma} \in D'$: $W \cup Conseq(D') \cup \{\beta\}$ is consistent;

are satisfied.

From the above theorems we remark that a classical/justified default extension is a consistent set and these two logics satisfy the *weak regularity property*, expressed as an individual consistency condition (stronger in justified logic than in classical default logic) for the justifications of the generating defaults.

Theorem 3. [11] Let (D, W) be a default theory, and let E, C be sets of formulas. (E, C) is a ***constrained extension*** of (D, W) if and only if $E = Th(W \cup Conseq(D'))$ and $C = Th(W \cup Conseq(D') \cup Justif(D'))$ for a maximal set $D' \subseteq D$ such that D' is grounded in W and the condition:
$W \cup Conseq(D') \cup Justif(D')$ is a consistent set;
is satisfied.

Theorem 4. [6] Let (D, W) be a default theory, and let E, C be sets of formulas. (E, C) is a ***rational extension*** of (D, W) if and only if $E = Th(W \cup Conseq(D'))$ and $C = Th(W \cup Conseq(D') \cup Justif(D'))$ for a maximal set $D' \subseteq D$ such that D' is grounded in W and the conditions:
– $W \cup Conseq(D') \cup Justif(D')$ is a consistent set;
– $\forall d \in D \backslash D'$ we have:
$W \cup Conseq(D') \cup \neg Prereq(d)$ is consistent **or**
$W \cup Conseq(D') \cup Justif(D' \cup d)$ is inconsistent;
are satisfied.

Theorems 3 and 4 show that the *strong regularity property* is common to these logics. According to this property, the reasoning process is guided by a consistent context C containing the actual extension E and the assumptions (justifications) of the applied defaults. For the rational default logic the set of generating defaults must be maximal-active [8] with respect to W and E.

Another important formal property is *semi-monotonicity* which expresses a "monotonicity" with respect to the set of defaults. Justified and constrained default logics have this desirable property (useful in the computation of an extension), with an important consequence: the existence of a justified/constrained extension for any default theory. The existence of a classical/rational extension for a default theory is not guaranteed and semi-monotonicity property is not satisfied by classical/rational default logics.

From theorems 1, 2, 3 and 4 we can conclude that all four types of extensions are deductive closures of the set W (explicit content) and the consequents of the generating default set D'(implicit content).

According to the initial fixed-point definitions of all variants of default logic, the generating default sets are defined as follows:

Definition 3. Let E_1 be a *clas*sical extension, (E_2, J) be a *just*ified extension, (E_3, C_3) be a *cons*trained extension and (E_4, C_4) be a *rat*ional extension of the default theory (D, W). The *generating default sets* are:
$GD^{E_1,clas}_{(D,W)} = \left\{ \frac{\alpha:\beta}{\gamma} \in D | if\ \alpha \in E_1\ and\ E_1 \cup \{\beta\}\ consistent,\ \ then\ \gamma \in E_1 \right\}$
for the classical extension E_1;
$GD^{(E_2,J),just}_{(D,W)} = \left\{ \frac{\alpha:\beta}{\gamma} \in D | if\ \alpha \in E_2\ and\ \forall \eta \in J \cup \{\beta\} : E2 \cup \{\gamma, \eta\}\ consistent\right.$
$\left. then\ \gamma \in E_2, \beta \in J \right\}$ for the justified extension (E_2, J);
$GD^{(E_3,C_3),cons}_{(D,W)} = \left\{ \frac{\alpha:\beta}{\gamma} \in D | if\ \alpha \in E_3\ and\ C_3 \cup \{\beta, \gamma\}\ consistent\right.$
$\left. then\ \gamma \in E_3,\ \beta, \gamma \in C_3 \right\}$ for the constrained extension (E_3, C_3);
$GD^{(E_4,C_4),rat}_{(D,W)} = \left\{ \frac{\alpha:\beta}{\gamma} \in D | if\ \alpha \in E_4\ and\ C_4 \cup \{\beta\}\ consistent, then\ \gamma \in E_4,\right.$
$\left. \beta, \gamma \in C_4 \right\}$ for the rational extension (E_4, C_4).

Example 1. The default theory (D, W), with $W = \{A\})$ and $D = \left\{ d1 = \frac{A:B}{F}, d2 = \frac{:\neg B}{G}, d3 = \frac{:\neg F \wedge \neg G}{H} \right\}$ has:
- one classical extension: $E_1 = Th(\{A, F, G\})$ and $GD^{E_1,clas}_{(D,W)} = D_1 = \{d1, d2\}$;
- two justified extensions:
(E_1, J_1)=$(Th(\{A, F, G\})$,$\{B, \neg B\})$, $GD^{(E_1,J_1),just}_{(D,W)} = D_1$;
(E_2, J_2)=$(Th(\{A, H\})$,$\{\neg F \wedge \neg G\})$, $GD^{(E_2,J_2),just}_{(D,W)} = D_2$=$\{d3\}$;
- three constrained extensions:
$(E_2, C_2) = (Th(\{A, H\}),Th(\{A, H, \neg F \wedge \neg G\}))$, $GD^{(E_2,C_2),cons}_{(D,W)} = D_2$;
$(E_3, C_3) = (Th(\{A, F\}),Th(\{A, F, B\}))$, $GD^{(E_3,C_3),cons}_{(D,W)} = D_3 = \{d1\}$;
$(E_4, C_4) = (Th(\{A, G\}),Th(\{A, G, \neg B\}))$, $GD^{(E_4,C_4),cons}_{(D,W)} = D_4 = \{d2\}$
- two rational extensions: (E_3, C_3) and (E_4, C_4) with $GD^{(E_3,C_3),rat}_{(D,W)} = D_3$ and $GD^{(E_4,C_4),rat}_{(D,W)} = D_4$.

3. A heuristic approach of the extension computation problem

In this section we extend the heuristic approach of the classical ECP from [9] to all types of default extensions: justified, constrained, rational.

The theorems from the previous section show that the problem of finding extensions can be reduced to the problem of finding the generating default sets for those extensions.

In this heuristic approach we need to define a search space for the generating default sets and an evaluation function to compute the fitness of each element of this space according to the definitions of default extensions.

Definition 4. [7] For a default theory (D, W) we define the search space as the set $CGD = 2^D$, representing all possible configurations, called *candidate generating default sets.*

Definition 5. [7] Let (D, W) be a default theory and $X \in CGD$, a candidate generating default set. We define:
- the *candidate extension* associated to X: $CE(X) = Th(W \cup Conseq(X))$;
- the *candidate context* associated to X:
$$CC(X) = Th(W \cup Conseq(X) \cup Justif(X));$$
- the *candidate support set* associated to X: $CJ(X) = Justif(X)$.

For defining the evaluation function we need four intermediate functions: $f_0^{type}, f_1^{type}, f_2^{type}, f_3^{type}$, where *type=clas* for *clas*sical extensions, *type=just* for *just*ified extensions, *type=cons* for *cons*trained extensions and *type=rat* for *rat*ional extensions.

Using f_0^{type} we check if the candidate extension (for classical and justified default logics) or the candidate context (for constrained and rational default logics) is consistent or not, according to Theorems 1, 2, 3 and 4:

$$f_0^{clas}(X), f_0^{just}(X) = \begin{cases} 0 & \text{if } CE(X) \text{ is consistent} \\ 1 & \text{otherwise} \end{cases}$$

$$f_0^{cons}(X), f_0^{rat}(X) = \begin{cases} 0 & \text{if } CC(X) \text{ is consistent} \\ 1 & \text{otherwise} \end{cases}$$

f_1^{type} rates the correctness of the candidate generating default set according to the definitions of different types of default extensions.

$f_1^{type}(X) = \sum_{i=1}^{n} \pi(d_i)$, where $D = \{d_1, d_2, ..., d_n\}$

The table below defines $\pi(d_i) \in Z$, a penalty for each default $d_i \in D$, indicating if a default from X was correctly/wrongly applied and a default from $D - X$ was correctly/wrongly not applied, in order to generate the candidate extension $CE(X)$. k is an integer constant.

$d_i \in X$	$C_{pre}(X, d_i)$	$C_{justif}^{type}(X, d_i)$	$\pi(d_i)$	$d_i = \frac{\alpha_i : \beta_i}{\gamma_i}$
true	true	true	0	correctly applied
true	true	false	k	wrongly applied
true	false	true	k	wrongly applied
true	false	false	k	wrongly applied
false	true	true	k	wrongly not applied
false	true	false	0	correctly not applied
false	false	true	0	correctly not applied
false	false	false	0	correctly not applied

$C_{pre}(X, d_i) : CE(X) \vdash \alpha_i$ is the groundness condition for d_i.

The conditions C^{type}_{justif}, according to Definition 3, imply the *weak regularity property* for classical/justified default logics and the *strong regularity property* for constrained/rational default logics.

- $C^{clas}_{justif}(X, d_i)$: the set $CE(X) \cup \{\beta_i\}$ is consistent;
- $C^{just}_{justif}(X, d_i)$: $\forall \eta \in CJ(X) \cup \{\beta_i\}$, the set $CE(X) \cup \{\eta, \gamma_i\}$ is consistent;
- $C^{cons}_{justif}(X, d_i)$: the set $CC(X) \cup \{\beta_i, \gamma_i\}$ is consistent;
- $C^{rat}_{justif}(X, d_i)$: the set $CC(X) \cup \{\beta_i\}$ is consistent.

f^{type}_2 rates the level of groundness of the candidate generating default set.
$f^{type}_2(X) = card(Y)$, where Y is the biggest grounded set $Y \subseteq X \in CGD$.
f^{type}_3 checks the groundness property of X:

$$f^{type}_3(X) = \begin{cases} 0 & \text{if } X \text{ is grounded} \\ 1 & \text{otherwise} \end{cases}$$

Definition 6. [7] For a default theory (D, W) the *evaluation function* for a candidate generating default set $X \in CGD$ of an extension of $type \in \{clas, just, cons, rat\}$ is defined by:

$eval^{type} : CGD \longmapsto Z \cup \{\perp, \top\}$ with $\forall z \in Z, \perp < z < \top$
if $f^{type}_0(X) = 1$
  then $eval^{type}(X) = \top$
  else if $f^{type}_1(X) = 0$ and $f^{type}_3(X) = 0$
    then $eval^{type}(X) = \perp$
    else $eval^{type}(X) = f^{type}_1(X) - f^{type}_2(X)$
  endif
endif

The following theorem provides a necessary and sufficient condition for a set of defaults to be a generating set for an extension, using $eval^{type}$.

Theorem 7.[7] Let (D, W) be a default theory. A candidate generating default set $X \in CGD$ generates an extension of $type \in \{clas, just, cons, rat\}$ if and only if $eval^{type}(X) = \perp$.

This evaluation function can be used by different non-deterministic approaches as Genetic Algorithms and Ant Colony Optimization, to evaluate the candidate generating default sets from the search space. The efficiency of these approaches derives from the fact that the search space is not entirely explored. An initial candidate is progressively improved in order to obtain a solution for the *extension computation problem* (ECP).

4. Computing default extensions using Ant Colony Optimization

Based on the theoretical considerations of ANTDEL [9], in this section we propose a uniform theoretical approach of the extension computation problem for all versions of default logic using Ant Colony Optimization.

Ant Colony Optimization (ACO) [3, 4] is a population-based metaheuristic successfully used to solve difficult optimization problems which can be reduced to finding good paths through graphs. The collective behavior of ants, seeking for food and cooperating via the environment (the pheromone deposited on the paths), was the inspiration for this optimization technique.

Informally, the extension computation problem is represented as a search problem and it is solved using the ACO metaheuristics as follows:

- Given a default theory, a default graph, representing all the candidate generating default sets, is built. The default rules and two particular vertices: *in* and *out* form the set of vertices. The arcs connect the vertices containing compatible defaults. Each arc is weighted by pheromone which is initialized to 1 and is updated (deposited and evaporated) during the search process.

- An ant colony must find an optimal path from *in* to *out* in the graph, path which corresponds to a generating default set for an extension.

- The ants individually build their paths from *in* to *out* (corresponding to candidate generating default sets), using a probabilistic choice biased on the pheromone deposited on the arcs and a local evaluation function.

- The pheromone evaporates in time and increases on better paths. Therefore, during the optimization process, the paths are progressively improved in order to find an optimal solution (according to the evaluation function).

The following definitions formalize the above description using the concepts defined in the previous section.

Definition 7. Let (D, W) be a default theory. The *default graph of type*$\in \{clas, just, cons, rat\}$ associated to the default theory is $G^{type}(D, W) = (D \cup \{in, out\}, A^{type}, \varphi)$. The *arc set*, A^{type}, is defined as follows:

$$A^{type} = \{(in, out)\} \cup \{(d, out), \forall d \in D\} \cup$$
$$\cup \left\{(d, d') \in D^2 | d \neq d' \; and \; C^{type}(d, d') \; is \; true\right\} \cup$$
$$\cup \left\{(in, d), \forall d = \tfrac{\alpha:\beta}{\gamma} \in D | W \vdash \alpha \; and \; W \cup \{\beta, \gamma\} \, consistent\right\},$$

where: $d = \frac{\alpha:\beta}{\gamma}, d' = \frac{\alpha':\beta'}{\gamma'}$ and the conditions C^{type} are:

$C^{clas}(d, d') = C^{just}(d, d') : W \cup \{\beta, \gamma, \gamma'\} \; and \; W \cup \{\beta', \gamma, \gamma'\}$ consistent;

$C^{cons}(d, d') = C^{rat}(d, d') : W \cup \{\beta, \beta', \gamma, \gamma'\}$ consistent;

Each arc $(i, j) \in A^{type}$ is weighted by a positive real number $\varphi_{i,j}$, called *artificial pheromone*.

In order to decrease the search space, the arc set of the default graph is built using the following observations:

- The arc (in, d) is added to the arc set if d is applicable to W and its application will not lead to a contradiction. This condition is the same for all versions of default logic.

- There is an arc between two defaults d and d' only if they are "compatible", meaning that they can belong together to a candidate generating default set. $C^{clas}(d, d'), C^{just}(d, d'), C^{cons}(d, d'), C^{rat}(d, d')$ express these "compatibility" conditions which are particular cases of the weak/strong regularity properties for the default logics.

Definition 8. For the default theory (D, W) and a path $P = (in, ..., out)$ in $G^{type}(D, W)$, $D_P = D \cap P \in CGD$ represents a candidate generating default set of $type \in \{clas, just, cons, rat\}$.

Definition 9. Let P be a path in the default graph $G^{type}(D, W)$ and $d = \frac{\alpha:\beta}{\gamma} \in D - P$.

- d is grounded in P if $W \cup Conseq(D_P) \vdash \alpha$,

- d is *type*-compatible with P if $C^{type}(P, d)$ is true, where:

$D_P = P \cap D$, $type \in \{clas, just, cons, rat\}$

$C^{clas}(P, d) : W \cup Conseq(D_P) \cup \{\beta\}$ consistent;

$C^{just}(P, d) : \forall \eta \in Justif(D_P) \cup \{\beta\} : W \cup Conseq(D_P) \cup \{\eta, \gamma\}$ consistent;

$C^{cons}(P, d) : W \cup Conseq(D_P) \cup Justif(D_P) \cup \{\beta, \gamma\}$ consistent;

$C^{rat}(P, d) :\ W \cup Conseq(D_P) \cup Justif(D_P) \cup \{\beta\}$ consistent;

- *the local evaluation function* is defined by:

$$loc^{type}(P, d) = \begin{cases} 1 & \text{if } d \text{ is grounded in } P \text{ and} \\ & \quad d \text{ is } \textit{type}\text{-compatible with } P \\ 0 & \text{otherwise} \end{cases}$$

Remarks:

1. The compatibility conditions for all types of versions are the apllicability conditions of the defaults and are used to apply one by one the defaults and to build candidate generating default sets (paths from *in* to *out* in the default graph).

2. The local evaluation function is used to choose efficiently the next vertex in the path (the next default to be applied) in order to reach the *out* vertex.

3. Due to the semi-monotonicity property of justified/constrained default logics, applying new defaults will not contradict previously applied defaults. If the $loc^{just/cons}(P, d) = 1$, then $D_P = P \cap D$ is a partial generating default set and P is a "good" path, which will lead to an optimal solution.

4. For classical/rational default logics, which do not satisfy the semi-monotonicity property, $loc^{clas/rat}(P, d) = 1$ will not guarantee that P is a

"good" path, because the application of new defaults can lead to a contradiction and the defaults from $D_P = P \cap D$ can not be generating defaults.

Definition 10. Let (D, W) be a default theory and $P = (in, ..., v_i)$ a path in the default graph $G^{type}(D, W) = (V, A^{type}, \varphi)$. The set of *vertices reachable from* v_i is $R(v_i, P) = \{v_j \in V - P | (v_i, v_j) \in A^{type}\}$. The *attractivity* of each reachable vertex v_j from $v_i \in P$ is defined as follows:

$$at^{type}(v_i, v_j, P, \alpha, \beta) = \frac{\varphi_{i,j}^{\alpha} * (loc^{type}(P, v_j))^{\beta}}{\sum_{v_k \in R(v_i, P)} \varphi_{i,k}^{\alpha} * (loc^{type}(P, v_k))^{\beta}},\ \alpha, \beta > 0$$

An ant chooses the next vertex on its path using the probability given by the attractivity function. α and β are used to give more or less influence of the pheromone or local evaluation.
The following ACO-based procedure, computes default extensions of any type.

Procedure Extension-Computation-Problem-ACO

Input data:
- (D, W) - a default theory
- $type$ - the type $\in \{clas, just, cons, rat\}$ of default extension
- na - the number of ants in the colony
- ni - the maximum number of iterations
- α, β - used to give more or less influence of the pheromone or local evaluation
- e - the evaporation coefficient
- k - the number of the best paths used for reinforcement

Output data:
- a generating default set for a *type* default extension of (D, W) *or*
- the best candidate, after ni iterations, for a generating default set of a *type* default extension of (D, W)

build $G = (V, A, \varphi)$ // the default graph: $G^{type}(D, W)$;
$it \leftarrow 1$; $sol \leftarrow false$;
while ($it <= ni$ and $not\ sol$) do
    for $i = 1$ to na do
        $P[i] \leftarrow$ **path-ant**$(G, type, \alpha, \beta)$;
        $ev[i] \leftarrow$ **eval**$^{type}(D_{P[i]})$;
    endfor
    order the arrays $ev[i], P[i], i = 1, na$ ascending with respect to $ev[i]$
    $bestP = P[1]$;
    if ($ev[1] = \perp$) then $sol \leftarrow true$; break; endif
    $\varphi \leftarrow$**update**(φ, P, k);
    $\varphi \leftarrow$**evaporation**(φ, e);
    $it \leftarrow it + 1$;
endwhile
if ($sol = true$) then

write "D_{bestP} is the generating default set of the *type*- default extension $CE(D_{bestP})$
else
write "D_{bestP} is the best candidate, after ni iterations, for a generating default set of a *type* default extension of (D, W)"
endif
endprocedure

Function path-ant$(G, type, \alpha, \beta)$
$v \leftarrow in$; $path \leftarrow \{in\}$;
while $(v \neq out)$ do
compute $R(v, path)$; // the vertices of V, reachable from v
for all $u \in R(v, path)$ do
compute $at^{type}(v, u, path, \alpha, \beta)$; // attractivity
endfor
choose $w \in R(v, path)$ with the probability $at^{type}(v, w, path, \alpha, \beta)$;
$path \leftarrow path \cup \{w\}$;
$v \leftarrow w$;
endwhile
return $path$;
endfunction

The **evaporation** function acts globally decreasing the pheromone on all the arcs: $\varphi(i, j) \leftarrow (1 - e) * \varphi(i, j)$, $\forall (i, j) \in A^{type}$.

In order to improve the paths in the next iterations, the pheromone on the best k paths is increased by the function **update** as follows:
$\varphi(i, j) \leftarrow \varphi(i, j) + 0.9^{k-m}$, $\forall (i, j) \in D \cap P[m], m = 1, \ldots, k$.

We remark that the **update** function can be improved, for justified/ constrained extensions, by reinforcing all the partial paths representing partial generating defaults, due to the semi-monotonicity property.

The fact that the existence of classical/rational extensions for a default theory is not guaranteed implies that in some cases the execution of the procedure will stop when the maximum number of iterations was executed, but we can not conclude if there is a classical/justified extension for the initial default theory or not.

For justified/constrained logics, in the worst case there is at least one extension for a default theory, corresponding to the path (in, out), which represents $\emptyset$ as the generating default set. If the maximum number of iterations is big enough, the procedure computes always a justified/constrained default extension.

For all types of extensions if the procedure stops without giving an extension, we obtain an approximate solution, the best candidate for a generating default set of an extension, that can be useful.

5. Conclusions and future work

In this paper we proposed a uniform theoretical approach of the extension computation problem for all versions (classical, justified, constrained, rational) of default logic using Ant Colony Optimization. Due to the high complexity (NP^{NP}) of this problem for non-trivial default theories, the ECP is solved as a search problem using this metaheuristic.

We are now working at the implementation of an automated system in order to obtain experimental results for non-trivial general default theories and to find the best combinations for the parameters of the proposed procedure. A first-order theorem prover [6], based on the sematic tableaux method will be used to check the consistency, inconsistency, derivability and groundness, needed for computing the local and general evaluation functions.

References

[1] G. Antoniou, A. P. Courtney, J. Ernst, J., M. A. Williams, "A System for Computing Constrained Default Logic Extensions", *Logics in Artificial Intelligence, Lecture Notes in Artificial Intelligence*, 1126, pp. 237–250, 1996.

[2] P. Cholewinski, W. Marek, M. Truszczynski, "Default reasoning system DeReS", *Proceedings of KR-96*, pp. 518–528, Morgan Kaufmann,1996.

[3] M. Dorigo, E. Bonabeau, G. Theraulaz, "Ant algorithms and stigmergy", *Future Generation Computer Systems*, no. 16, pp. 851–871, 2000.

[4] M. Dorigo, T.Stutzle, *Ant Colony Optimization*, MIT Press, 2004.

[5] W. Lukasiewicz, "Considerations on default logic - an alternative approach", *Computational Intelligence*, no. 4, pp. 1–16, 1988.

[6] M. Lupea, *Nonmonotonic Reasoning Using Default Logics, Ph.D. Thesis*, Babes-Bolyai University, Cluj-Napoca, 2002.

[7] M. Lupea, "Computing default extensions - a heuristic approach", *Studia Universitatis Babes-Bolyai, Informatica*, L, no. 2, pp. 49–58, 2005.

[8] A. Mikitiuk, M. Truszczynsky, "Constrained and Rational Default Logics", *Proceedings of IJCAI-95*, pp. 1509–1515, Morgan Kaufman, 1995.

[9] P. Nicolas, F. Saubion, I. Stephan, "New generation systems for non-monotonic reasoning", *International Conference on Logic Programming and NonMonotonic Reasoning*, pp. 309–321, 2001.

[10] R. Reiter, "A Logic for Default Reasoning", *Artificial Intelligence*, no. 13, pp. 81–132, 1980.

[11] T. H. Schaub, *Considerations on Default Logics, Ph.D.Thesis*, Technischen Hochschule Darmstadt, Germany, 1992.

[12] T. H. Schaub, "XRay system: An implementation platform for local query-answering in default logics", *Applications of Uncertainty Formalisms, Lecture Notes in Computer Science*, no. 455, pp. 254–378, Springer Verlag, 1998.

Babeş-Bolyai University, Faculty of Mathematics and Computer Science, Cluj-Napoca, Romania

E-mail address: `lupea@cs.ubbcluj.ro`