Using Ant Colony Optimization and Genetic Algorithms for the Linguistic Summarization of Creep Data

Carlos A. Donis-Díaz[1], Rafael Bello[1], and Janusz Kacprzyk[2]

[1] Informatic Studies Center, Universidad Central Marta Abreu de Las Villas, Santa Clara, Cuba
{cadonis,rbellop}@uclv.edu.cu
[2] Systems Research Institute, Polish Academy of Sciences, Warsaw, Poland
Janusz.Kacprzyk@ibspan.waw.pl

Abstract. Some models using metaheuristics based in an "improvement of solutions" procedure, specifically Genetic Algorithms (GA), have been proposed previously to the linguistic summarization of numerical data (LDS). In the present work is proposed a new model for LDS based in Ant Colony Optimization (ACO), a metaheuristic that use a "construction of solution" procedure. Both models are compared in LDS over *creep* data. Results show how the ACO based model overcomes the measures of goodness of the final summary but fails to improve the results of the GA based model in relation to the diversity of the summary. Features of both models are considered to explain the results.

Keywords: Linguistic Data Summarization, Ant Colony Optimization, Genetic Algorithms, Fuzzy Logic, Creep rupture stress.

1 Introduction

Linguistic data summarization has been for a long time a subject of intensive research, and various tools and techniques from computational linguistics, natural language generation, etc. have been proposed. The use of the fuzzy logic with linguistic quantifiers is one of the most conceptually simple, developed and used approaches for the linguistic summarization of numerical data (LDS). The concept of a linguistic data summary, using fuzzy logic with linguistic quantifiers, which will be employed in this paper, was introduced by Yager [1], then considerably advanced in [2, 3] and presented in an implementable way in [4].

The process of generating linguistic data summaries for a given set of numerical data, usually a relational numerical database, can conveniently be represented as an optimization problem in which the best summaries from a large set of candidates are selected, and the basic objective function is assumed to be the truth degree of a linguistic summary that is equated with the degree of truth of a linguistically quantified proposition that is conceptually equivalent to the linguistic summary in question. Several works to deal with this problem have been developed [5-9]. Most of them obtain linguistic summaries by using heuristics based in an "improvement of solutions" procedure, specifically using evolutionary heuristics like Genetic Algorithms

P. Angelov et al. (eds.), *Intelligent Systems'2014*,
Advances in Intelligent Systems and Computing 322, DOI: 10.1007/978-3-319-11313-5_8

(GA). In a recent related work [10] has been proposed a hybrid model of GA with local search which improves the results obtained with the basic version of GA.

In the present work is proposed a different way of obtaining a linguistic summary by using a metaheuristic based in a "constructing of solutions" procedure, specifically an Ant Colony Optimization (ACO) based model. To the best of our knowledge, this metaheuristic has not been used previously for LDS although it has been used in problems with some commonalities like classification rules discovery and fuzzy rules learning. As final objective is compared the behavior of both metaheuristics when are applied in LDS over *creep* data.

2 Theoretical Background

This section presents the necessary theoretical precepts related to: the linguistic summarization of data, the hybrid GA proposed in [10] and used for comparison purposes in the present work, and finally some basics features of the ACO metaheuristic.

2.1 Linguistic Data Summarization Using Fuzzy Logic with Linguistic Quantifiers

In this paper is considered the linguistic data summarization approach proposed in [2].

Having: $Y=\{y_1, \ldots, y_n\}$ a set of objects in a database, e.g., the set of workers, and $A = \{A_1, \ldots, A_m\}$ a set of attributes (fuzzy variables) characterizing objects from Y, e.g., salary, age, etc. in a database D of workers, and $A_j(y_i)$ denotes the value of attribute A_j for object y_i. A linguistic summary from D consists of:

- a summarizer S, i.e. a linguistic expression composed by an attribute together with a linguistic value defined on the domain of attribute A_j (e.g. 'low salary');
- a quantifier Q (a linguistic quantifier), i.e. a fuzzy set with universe of discourse in the interval [0, 1] expressing a quantity in agreement, e.g. *most*;
- a qualifier R, i.e. a fuzzy filter determining a fuzzy subset of Y; can be composed for one or several linguistic expressions (e.g. 'young' for attribute 'age').
- a truth degree T (validity) of the summary, i.e. a number from the interval [0, 1] assessing the truth of the summary (e.g. 0.7); usually, only summaries with a high value of T are interesting;

Thus, linguistic summaries may be exemplified by

$$T\ (Most\ of\ young\ employees\ earn\ low\ salary) = 0.7$$

and their foundation is Zadeh's [11] linguistically quantified proposition corresponding to *QRy's are S*.

The truth value (T) may be calculated by using either original Zadeh's calculus of linguistically quantified statements [11] where a (proportional, nondecreasing) linguistic quantifier Q is assumed to be a fuzzy set in [0, 1] and the values of T are calculated as $T(QRy's\ are\ S) = \mu_Q(r)$ where

$$r = \frac{\sum_{i=1}^{n} (\mu_R(y_i) \wedge \mu_S(y_i))}{\sum_{i=1}^{n} \mu_R(y_i)} \tag{1}$$

Besides T (truth), other quality measures have been proposed to determine the quality of summaries. In [3] are described: the truth degree (*T1*) that correspond with the mentioned *T*, the degree of imprecision (*T2*), the degree of covering (*T3*) and the degree of appropriateness (*T4*).

In the present work is used the term proposition, to be more specific a linguistically quantified proposition, to refer a linguistic summary, and the term (linguistic) summary will be referred to a set of propositions. This is basically consistent with [2, 4] and in particular, the modern natural language generation (NLG) based approach.

2.2 A Hybrid Model of GA with Local Search for LDS

In this section are summarized the basic features of the hybrid model of GA with local search (HybridGA-LDS) proposed in [10] for LDS. It will be used for comparison purposes in this work.

Genetic Representation. Chromosomes in HybridGA-LDS model represent a whole linguistic summary (i.e. a set of linguistically quantified propositions) and each gene codifies just one of such propositions.

Fitness Function. The HybridGA-LDS model searches for a summary containing linguistic propositions with high values of quality (goodness) and high diversity between them. The fitness function to be maximized for a chromosome i is defined in the interval [0, 1] as $F_i = m_g G_i + m_d D_i$ where the term G_i and D_i represent the *Goodness* and *Diversity* of the summary respectively. *Goodness* is calculated as the mean value of the individual goodness $g_j = T_j\ .\ St_j$ of propositions (genes) according to: $G_i = \frac{1}{n} \sum_{j=1}^{n} T_j \cdot St_j$ where T_j is the truth degree of the proposition j, St_j represents a called linguistic strength indicator and n is the total number of propositions in the summary. The term D_i expresses the degree of diversity between the propositions forming the summary; is calculate as $D_i = C_i / n$ where C_i represents the number of clusters of propositions existing in the summary.

Genetic Operators. The HybridGA-LDS model mixes the use of traditional operators like selection, crossover and mutation with two specifics operators proposed to improve the search.

The first proposed operator was the *Cleaning Operator*: this operator was introduced to "clean" those propositions inside a summary having no opportunities to evolve towards better solutions during the process due to the inexistence of cases in the data set to cover them. The operator substitutes the propositions with $T = 0$ by others randomly generated.

The second operator was the *Propositions Improver Operator*; it was introduced to overcome the weakness of basic operators (crossover and mutation) to improve the quality of individual propositions. Through the evolutionary process, the basic crossover operator improves the quality of the chromosomes (summary) as a whole but do not improve the quality of genes (propositions), i.e. improves the diversity degree of the summary but does not improves individual propositions. In other hand, the mutation operator does not guarantee the sufficient perturbation in the search to solve the above problem. The *Propositions Improver Operator* implements a randomly greedy search using a best first strategy based in six possible transformations of the linguistic proposition: four to modify the quantifier, one to modify the summarizer and one to modify the quantifier. Two parameters control the deep of the search: the length of the search (*ls*), (i.e. the total number of new considered propositions) and the maximum number of searches without improve the quality ($s_w i$).

2.3 Ant Colony Optimization

ACO is a metaheuristic inspired in the behavior of real ants to forage for food. This metaheuristic has been widely used in many optimization problems and fields including applications with some commonalities to the objective in this work like classification rules discovery [12, 13] and fuzzy rules learning [14]. The implementation of ACO used in the present work is Max-Min Ant System (MaxMin AS) [15] an extension of the Ant System (AS) [16] implementation in order to improve its performance. For an overview and recent reviews on ACO can be consulted [17, 18], following are presented some basics features of MaxMin AS:

- The pheromone update is applied offline and evaporated according to: $\tau_{(v_i, v_j)} = \rho \cdot \tau_{(v_i, v_j)}$ where $\tau_{(v_i, v_j)}$ is the pheromone value of arc a_{v_i, v_j} between nodes v_i and v_j; ρ is the persistence factor, a parameter defined by the user; $(1\text{-}\rho)$ is the evaporation factor. After the evaporation, the pheromone is deposited on each arc corresponding with the solution of the best ant A_{best} as:

$$\tau_{(v_i, v_j)} = \tau_{(v_i, v_j)} + f(Q(A_{best})), \quad \forall a_{v_i, v_j} \in A_{best}$$

  where $f(Q(A_{best}))$ represents a function based on the quality of the solution in A_{best}. The ant that is permitted to add pheromone can be the ant with the best solution of the current iteration or the ant with the best global solution. Furthermore, it is common that ant solutions are improved by local searches before pheromone update.

- The possible values for pheromone trails are limited to the interval $[\tau_{min}, \tau_{max}]$.
- The initial pheromone trail of each arc is set to a high value.

3 ACO-LDS: A New Model for LDS

In this section is described the proposed ACO based model for searching linguistic data summaries (ACO-LDS). Similar to the HybridGA-LDS model, ACO-LDS aims to search not only propositions with good (possible, the best) qualities but also search propositions forming a good (possible, the best) summary in reference to the diversity among them. To meet this aim, an ant in ACO-LDS represents a whole summary and each iteration of the process discovers one summary, probably the best one between all iterations. Following subsections describes main aspects of the proposed model.

3.1 The General Algorithm of ACO-LDS

The high-level pseudo code of the algorithm is presented in Fig. 1.

```
Input: dataset
Output: best discovered summary
ComputeLocalHeuristicInformation()
InitPheromones()
gbSummary  = null          // Global best summary
currentIt = 1              // Current iteration
while (currentIt < max.iter.) and (not stagnation) do
   ibSummary = null   // Iteration best Summary
   for  a=1 to colony size do
     summary_a = null
     for i=1 to propositions per summary do
       proposition_i = CreateProposition(dataset)
       summary_a = summary_a + proposition_i
     end for
     if Fitness(summary_a) > Fitness(ibSummary) then
       ibSummary = summary_a
     end if
   end for
   Improve(ibSummary)
   if Fitness(ibSummary) > Fitness(gbSummary) then
     gbSummary = ibSummary
   end if
   UpdatePheromone(gbSummary)
   currentIt = currentIt + 1
end while
return gbSummary
```

Fig. 1. High-level pseudocode for ACO-LDS

As mentioned before, ACO-LDS use the MaxMin AS implementation of ACO. The procedure starts computing the local heuristic information for each node in the graph and initializing the pheromones to a high value. Then each iteration of the algorithm (*while* loop) produces a summary that correspond to the best summary obtained from the construction process developed by the colony (*for* loop, lines 7 to 16). This best summary of the iteration (*ibSummary*) is improved by performing a local search on each of its propositions; this is a difference with most applications that use ACO for discovering classification or fuzzy rules where the local search is applied to all constructed solutions. The best global summary is updated with the best summary of the iteration if this latter has a better value of fitness. Finally, the pheromones are updated using the best global summary. The *while* loop iterates until the maximum number of iterations or the stagnation condition is reached. This latter condition occurs when more than 90 percent of propositions are stagnant. A proposition is stagnant if all nodes visited by the ant when constructing the proposition have the pheromone value equal to τ_{max} and the remaining nodes in the graph have τ_{min}.

3.2 ACO Representation for LDS

In ACO-LDS each ant represents/constructs/modifies a summary with a fixed number of linguistically quantified propositions. Considered propositions have the form:

$<Q>(<a_1 = l_{1j}>$ *and* $[a_2 = l_{2j}]$ *and* … $[a_i = l_{ij}]$ are/have/… $<a_s = l_{sj}>) = <g>$

where $<Q>$ is the linguistic quantifier; terms $<a_i = l_{ij}>$ represents the linguistic expressions used in the qualifier being a_i the *i-th* fuzzy variable and l_{ij} the *j-th* linguistic term selected for a_i , observe that the qualifier require at least one linguistic expression; the summarizer is represented by only one linguistic expression $<a_s = l_{sj}>$ being a_s the fuzzy variable selected for that purpose and l_{sj} the linguistic term used for a_s; finally $<g>$ represents the goodness (quality) of the proposition. The "*and*" operator is calculated as the minimum membership degree of both concatenated linguistic expressions (in general, is a *t-norm*).

The graph used by ants to construct a linguistic proposition is composed by nodes representing the possible linguistic expressions for the qualifier $<a_1 = l_{1j}>$ and the summarizer $<a_s = l_{sj}>$. In the graph will exist, for each fuzzy variable a_i, as many nodes as linguistics terms have been defined for a_i. To define an arcs between to nodes one rule apply: from the group of linguistic expressions corresponding to a fuzzy variable can only be selected one when construction a proposition, i.e. can not be established arcs between nodes representing linguistics expressions belonging to the same fuzzy variable.

3.3 Constructing a Proposition

When building a summary, the ants create a fixed number of propositions. In the construction process of one proposition (referred in line 10 of Figure 1), the ant selects linguistics expressions (nodes) for the summarizer and the qualifier in a tour through the graph. The ant start selecting only one node for the summarizer from the group of linguistic terms defined for it in the graph. Then the ant selects the first node for the qualifier guaranteeing that the subset of objects from the database meeting this partial qualifier is not null. The process continues adding nodes to the qualifier while the subsequent subsets contain one or more examples or all possible nodes have been added. Finally, the model selects the quantifier that better value of goodness (g) cause in the proposition.

Transition Rule. When selecting a node is important to note that arcs between nodes have not a special means. For the final proposition, the important thing is if a specific node is selected or not; the precedence relationship (that arcs represents) between two nodes is irrelevant in this case. This is why the pheromone is stored in the nodes and not in the arcs.

As ants have to construct several propositions, is necessary to keep different trails of pheromone, one for each proposition. To satisfy this condition in the pheromone matrix was included an additional index indicating the number of the proposition (*tour*) for which the trail is maintained.

In ACO-LDS is used a pseudo-random transition rule, similarly as does ACS. During the tour t, a node v_{ij} (representing the linguistic expression <$a_i = l_{ij}$>) is randomly selected using a probability distribution first calculated as:

if $q \leq q_0$

$$P_{v_{ij}} = \begin{cases} 1, & \text{if } v_{ij} = \arg\max\{(\alpha \cdot \tau_{(t,v_{ij})} + (1-\alpha)\eta_{(v_{ij})}), \quad \forall i \in N\} \\ 0, & \text{in other case} \end{cases}$$

else ($q > q_0$)

$$P_{v_{ij}} = \begin{cases} \dfrac{\alpha \cdot \tau_{(t,v_{ij})} + (1-\alpha)\eta_{(v_{ij})}}{\sum_{k=1}^{m}\sum_{j=1}^{n_k} \alpha \cdot \tau_{(t,v_{kj})} + (1-\alpha)\eta_{(v_{kj})}}, & \forall\, k \in N \\ 0, & \text{in other case} \end{cases}$$

where q_0 is a parameter in [0, 1] and q a random value in [0, 1], $\tau_{(t,v_{ij})}$ is the pheromone value accumulated by node v_{ij} in the trail of the t-th tour, $\eta_{(v_{ij})}$ is the local heuristic value for node v_{ij}, m is the number of attributes, n_k is the number of linguistic terms of attribute k, N represents the set of selectable attributes, i.e. attributes not yet used by the ant, α is a parameter to control de importance given to τ and η in the equation. Before to be used, pheromone and heuristic values are normalized in [0, 1].

The proposed model includes an extra heuristic called *Frequency of use* when selecting the next node v_{ij}. This heuristic aims to build the current proposition as different as possible in relation with the propositions previously added to the partial summary in construction, i.e. the *Frequency of use* contributes to increase the diversity of the summary. The heuristic calculates a term F_u using the number of times (u_{vij}) that v_{ij} has been used in the partial summary under construction: $F_u = 1 - (u_{v_{ij}} / p)^e$. Term p represents the amount of propositions added up to now to the partial summary and e is a parameter in [0, 1] to graduate the "power" of influence of F_u. Then the heuristic affects the final probability distribution according to: $P_{v_{ij}} = P_{v_{ij}} \cdot F_u$

Local Heuristics. For the transition rule, two local heuristics are used depending if the node to be selected is for the summarizer or for the qualifier. As known, the values for the local heuristics are calculated and stored as a pre-step (Line 1, Fig. 1).

For summarizer's nodes is used the degree of imprecision (ID) as local heuristic. This heuristic is calculated based in the degree of imprecision T2 mentioned in subsection 2.1 and proposed in [3]. This value depends only on the form of the summarizer and as in the present work, is considered the summarizer to has just one linguistic expression, the form of ID is: $ID = T2 = 1 - in(s)$ where in(s) defines the degree of fuzziness of the fuzzy set s defined for the fuzzy variable S as:

$in(s) = \dfrac{cardinality(x \in X_S : \mu_s(x) > 0)}{cardinality(X_S)}$ where X_S refers to the universe of discourse of the fuzzy variable S and $\mu_s(x)$ the membership degree of the element x to the fuzzy set s.

For qualifier's nodes is used a proposed *Relevance* degree (*RD*) as heuristic. The *Relevance* estimates the importance of a node v_{ij} for a given linguistic expression s in the summarizer by using the degree of covering (*T3*). *Relevance* is calculated as:

$$RD_{(v_{ij},s)} = \max(0, \quad T3_{(v_{ij} \wedge s)} - T3_{(s)}) \tag{2}$$

where $T3_{(v_{ij} \wedge s)}$ is calculated as in [3] and $T3_{(s)} = \dfrac{cardinality(\mu_s(y) > 0)}{number\ of\ examples}$. Equals values of $T3_{(v_{ij} \wedge s)}$ and $T3_{(s)}$ mean that "observing" node v_{ij} in the dataset has no influence to "observe" s, i.e. node v_{ij} is not relevant to s; in this case $RD_{(v_{ij},s)}$ is equal to cero. While higher the value of $RD_{(v_{ij},s)}$, more relevant is the node v_{ij} for the summarizer s. As this heuristic depends on previous selection of the summarizer, for each node should be calculated and stored many values as linguistic terms were defined for the fuzzy variable used in the summarizer.

Fitness Function. The fitness function for ACO-LDS was defined similar to that proposed in [10] and described in subsection 2.1 but incorporating other measures of quality in the calculus of goodness for individual propositions. The goodness g of a proposition j is calculated as: $g_j = w_1 T_1 St_j + w_2 T_2 + w_3 T_3 + w_4 T_4$, where $\sum_i w_i = 1$ and T_1, T_2, T_3, T_4 are obtained as proposed in [3].

Updating the Pheromone Trails. As defined for MaxMin AS the pheromone levels are bounded according to the interval $[\tau_{min}, \tau_{max}]$. In ACO-LDS is used an approach where limits are dynamically updated each time a new best solution is found, as detailed in [15]. The ant containing the best global solution is the only permitted to increases the pheromone level in nodes belonging to the solution. The general rule to calculate the new value of pheromone τ^{ts+1} for the tour t having a previous time stamp ts is expressed as

$$\tau_{(t,v_{ij})}^{ts+1} = \begin{cases} \max(\tau_{\min}, \rho \cdot \tau_{(t,v_{ij})}^{ts}), & if\ v_{ij} \notin gbSummary \\ \min(\tau_{\max}, \rho \cdot \tau_{(t,v_{ij})}^{ts} + \rho \cdot \tau_{(t,v_{ij})}^{ts} \cdot F_{gbSummary}), & if\ v_{ij} \in gbSummary \end{cases}$$

where τ^{ts} represents the pheromone value in the previous iteration and ρ is the pheromone persistent factor ($1 - \rho$ is the evaporation factor).

The local Search. ACO-LDS applies a local search to improve the propositions. This local search is similar to that used in [10] but do not include the transformation that modify the quantifier because in ACO-LDS, the quantifier is selected as the best one, i.e. the quantifier that better value of linguistic strength produce in the proposition is used. The local search is only applied to propositions of the best summary of the iteration (*ibSummary*); this approach permits to develop a deeper local search since the whole process do not increases the total number of considered propositions when compared with the approach (mostly used in works done so far using ACO in similar problems) where the local search is applied to all considered propositions.

4 Comparing GA and ACO Metaheuristics in LDS

HybridGA-LDS and ACO-LDS models were applied over *creep* data in the present work. The creep rupture stress (*creep*) is an important mechanical property considered in the design of new alloys. It measures the stress level in which a steel structure fails when exposed to quite aggressive conditions over long periods of time. The data and fuzzy modeling used was the same as employed in [10]. Is important to note that for *creep* problem, the propositions having *Most* or *Much* as quantifier are more interesting, that is why the parameter *St* (linguistic strength) was set in both model preferring these quantifiers by using the same values as in [10]. In order to achieve uniformity in the processing of both models, experiments were developed so that both considered the same total number of propositions (250 000, representing the 6.91E-15 percent from the total for this problem) when obtaining its results. Both models used the same fitness function as described in the present work; values for w_i were: w_1=0.4, w_2= 0.1, w_3=0.25, w_4=0.25. To get the results, ten runs of each model were made. The Wilcoxon's test and Monte Carlo's technique were used to compare the results pairs to pairs and to calculate a more precise signification of the differences respectively.

4.1 Results and Analysis

Results of experiments are presented in Table 1. Columns (2) to (4) present general quality measures of a summary: (2) Goodness, (3) Diversity, (4) number of proposition having the desired quantifier (*Most* or *Much*); its values represent the mean value from ten runs of each model. In turn, the quality measures of individual propositions (Column (6) to (11)) represent mean values from propositions composing the summaries obtained in all runs of models.

Table 1. Quality measures of summaries and propositions obtained by models

Model	Fitness	Goodness	Quantifier	Diversity	Quality measures of individuals propositions				
					T1	*T1·St*	*T2*	*T3*	*T4*
(1)	(5)	(2)	(4)	(3)	(6)	(7)	(8)	(9)	(10)
HybridGA-LDS	0.6931	0.5616	16.30	1.0000	0.9566	0.5157	0.8960	0.5287	0.5343
ACO-LDS	0.7359	0.6984	21.00	0.8233	0.9439	0.6638	0.8453	0.7357	0.6576

Let first observe the behavior of general quality measures in obtained summaries: ACO-LDS produces a better value of fitness (with significant difference) respect to the value of HybridGA-LDS model. This result is supported by the “Goodness” component of the fitness value ($\text{Goodness}_{\text{HybridGA-LDS}} < \text{Goodness}_{\text{ACO-LDS}}$) but not by the “Diversity” component ($\text{Diversity}_{\text{HybridGA-LDS}} > \text{Diversity}_{\text{ACO-LDS}}$); the differences are significant in both cases. Results for “Goodness” are a direct consequence of results obtained for quality measures of individuals propositions (Columns 6 to 11): except for *T2*, ACO-LDS improves or equals (in *T1*, the differences are not significant) the results obtained by HybridGA-LDS in each quality measure.

The analysis of *T1·St*, the most significant component when calculating the quality of a proposition, has great importance in the explanation of results; this parameter combines the truth degree of a proposition with its linguistic strength. Let start the analysis taking into account the relation *r* expressed in equation (1) and the calculus of *T3* proposed in [3]: observing the components and relations they use, could be expected that build a proposition with high *r* by using linguistic expressions with high *T3* in the qualifier is more probable that build a proposition with high *r* by using linguistic expressions with low *T3* in the qualifier, i.e., in the process of constructing a proposition, select nodes with high *T3* has a positive influence in obtaining a proposition with high *r*. Having into account that *Relevance degree* rewards to nodes with high *T3* (see equation 2) and the fact that high values of *r* produce high values of membership to the linguistic terms *Most* and *Much* of the quantifier (i.e., produce high values of *T1* (see the calculus of *T=T1*) in propositions having *Most* and *Much* as quantifiers) can be concluded that using the *Relevance degree* as local heuristic when constructing propositions in ACO-LDS stimulates the production of propositions with high degree of truth (*T1*) and having *Most* and *Much* as quantifier. This analysis explain the results obtained by ACO-LDS for *T1·St* since the parameter *St* was set in the present application precisely, to stimulate propositions having quantifiers like *Most* or *Much*. Values obtained by HybridGA-LDS for this component are lower despite values for *T1* are high; the main reason for this result is that the model generates linguistic terms for the qualifier without check any relation with the linguistic term generated for the summarizer, so the model can find propositions with high *T1* but do not having *Most* or *Much* as quantifier necessarily. Concluding this analysis can be established that in the search of propositions with high values of *T1* and having *Most* or *Much* as qualifier, using the approach of ACO-LDS that constructs the solutions and therefore permits the use of a local heuristic like the *Relevance* degree, has advantage over using the approach of HybridGA-LDS that improves the solutions by using the genetic operators. Column 4 shows the number of propositions with *Most* or *Much* as quantifiers, this values are consistent with previous analysis.

Results obtained for components *T3* and *T4* can be explained in a similar way since both are favored by the use of the *Relevance* heuristic, i.e. its features are considered in some way by the *Relevance* degree.

When analyzing values obtained for the *Diversity* component of the fitness (column 3) can be noted that despite using an additional local heuristic (*Frequency of use*) specifically designed to ensure diversity in the summary, ACO-LDS fails to improve the performance of HybridGA-LDS (differences in values are significant). In this sense can be highlighted the effectiveness of the crossover operator whose main function in HybridGA-LDS is to ensure diversity in the summary. Results of ACO-LDS are conditioned by the strict (reduced) pool of nodes (linguistics expressions) that imposes the *Relevance* heuristic when selecting nodes to construct the proposition.

4.2 Values of Parameters Used in Experiments

Table 2. Parameters and values used in the experimentation process

Param.	Description	Interval	Value
ρ	the persistent factor used in the pheromone updating rule, $(1-\rho)$ is the evaporation factor	[0 – 1]	0.7
α	controls the importance given to the pheromone τ and the heuristic η in the calculus of the probability selection of a node	[0 – 1]	0.5
q_0	used when constructing the proposition to determine if the next node, will be selected in a deterministic or stochastic way	[0 – 1]	0.8
e	graduates the influence of the frequency of use (of a node in a summary) when calculating the probability selection of a node	[0 – 1]	0.3
ls	length of the local search in ACO-LDS, specify the maximum total number of considered propositions for the local search		20
$s_w i$	maximum number of searches without improvement in the local search in ACO-LDS		15

Presented values were the final ones obtained during an experimentation process.

5 Conclusions

In the present work has been proposed a new model for LDS based in ACO. This model (ACO-LDS) overcomes measures of goodness of the final summary but fails to improve the diversity of the summary obtained by HybridGA-LDS. When searching linguistic summaries on *creep* data, good results obtained in ACO-LDS for goodness are influenced by the constructive procedure used in ACO which allows the use of a local heuristic as *Relevance* that selects the qualifier based in a previous selection of the summarizer. Respect to the degree of diversity in the final summary, has been shown how the crossover operator used in Hybrid-LDS result a more effective approach than that used in ACO-LDS to meet this requirement in the final summary. Concluding can be established that ACO-LDS do not overcome completely the results of Hybrid-GA. In future works will be mixed the best features of both models.

Acknowledgments. This research has been partially supported by the National Centre of Science under Grant No. UMO-2012/05/B/ST6/03068.

References

1. Yager, R.R.: A new approach to the summarization of data. Information Sciences 28, 69–86 (1982)
2. Kacprzyk, J.: Intelligent data analysis via linguistic data summaries: a fuzzy logic approach. In: Decker, R., Gaul, W. (eds.) Classification and Information Processing at the Turn of the Millennium, pp. 153–161. Springer, Heidelberg (2000)
3. Kacprzyk, J., Yager, R.R.: Linguistic summaries of data using fuzzy logic. International Journal of General Systems 30(2), 133–154 (2001)
4. Kacprzyk, J., Zadrożny, S.: Computing with words: towards a new generation of linguistic querying and summarization of databases. In: Sinčak, P., Vaščak, J. (eds.) Quo Vadis Computational Intelligence?, pp. 144–175. Physica-Verlag, Heidelberg (2000)
5. Castillo-Ortega, R., et al.: Linguistic Summarization of Time Series Data using Genetic Algorithms. In: 7th Conference of European Society for Fuzzy Logic and Technology - EUSFLAT 2011, Atlantis Press, Aix-les-Bains (2011)
6. Castillo-Ortega, R., et al.: A Multi-Objective Memetic Algorithm for the Linguistic Summarization of Time Series. In: 13th Annual Genetic and Evolutionary Computation Conference - GECCO 2011. ACM, Dublin (2011)
7. George, R., Srikanth, R.: Data summarization using genetic algorithms and fuzzy logic. In: Herrera, F., Verdegay, J.L. (eds.) Genetic Algorithms and Soft Computing, pp. 599–611. Physica-Verlag, Heidelberg (1996)
8. Kacprzyk, J., Wilbik, A., Zadrożny, S.: Using a Genetic Algorithm to Derive a Linguistic Summary of Trends in Numerical Time Series. In: International Symposium on Evolving Fuzzy Systems, Ambleside (2006)
9. Kacprzyk, J., Wilbik, A., Zadrożny, S.: Linguistic summarization of time series using a fuzzy quantifier driven aggregation. Fuzzy Sets and Systems 159(12), 1485–1499 (2008)
10. Donis-Diaz, C.A., et al.: A hybrid model of genetic algorithm with local search to discover linguistic data summaries from creep data. Expert System with Applications 41(4), 2035–2042 (2014)
11. Zadeh, L.: A computational approach to fuzzy quantifiers in natural languages. Computers and Mathematics with Applications 9, 149–184 (1983)
12. Parpinelli, R., Lopes, H., Freitas, A.: Data mining with an ant colony optimization algorithm. IEEE Transactions on Evolutionary Computation 6(4), 321–332 (2002)
13. Otero, F.B., Freitas, A., Johnson, C.G.: A New Sequential Covering Strategy for Inducing Classification Rules with Ant Colony Algorithms. IEEE Transactions on Evolutionary Computation 17(4), 64–76 (2013)
14. Alatas, B., Akin, E.: FCACO: Fuzzy Classification Rules Mining Algorithm with Ant Colony Optimization. In: Wang, L., Chen, K., S. Ong, Y. (eds.) ICNC 2005. LNCS, vol. 3612, pp. 787–797. Springer, Heidelberg (2005)
15. Stützle, T., Hoos, H.: MAX-MIN ant system. Future Generation Computer Systems 16(8), 889–914 (2000)
16. Dorigo, M., Colorni, A., Maniezzo, V.: The Ant System: Optimization by a colony of cooperating agents. IEEE Transactions on Systems, Man, and Cybernetics-Part B 26(1), 1–13 (1996)
17. Dorigo, M., Stützle, T.: Ant Colony Optimization: Overview and Recent Advances. In: Gendreau, M., Potvin, Y. (eds.) Handbook of Metaheuristics, pp. 227–263. Springer, New York (2010)
18. Dorigo, M., Birattari, M., Stützle, T.: Ant Colony Optimization- Artificial Ants as a Computational Intelligence. IEEE Computational Intelligence Magazine 1, 28–39 (2006)