A FRAMEWORK FOR SEMANTIC QUERYING OF DISTRIBUTED DATA-GRAPHS VIA INFORMATION GRANULES

Davide Bacciu and Alessio Botta
IMT - Institutions, Market, Technologies
Lucca Institute for Advanced Studies, Italy
email: {d.bacciu,a.botta}@imtlucca.it

Dan C. Stefanescu
Mathematics and Computer Science Department
Suffolk University, Boston, MA, USA
email: dan@mcs.suffolk.edu

ABSTRACT
Regular path queries (RPQ) represent a common and convenient way to access and extract knowledge represented as labeled and weighted data-graphs. In this paper, we look to enhance the information representation in data-graphs and RPQs by augmenting their expressive power with the use of semantically meaningful knowledge in the form of information granules. We extended a recent distributed algorithm for the evaluation of RPQs on spatial networks by introducing fuzzy weights in place of crisp values both in the data-graphs and the query formulation. Moreover, we describe two alternative strategies for determining the costs of the paths computed by the fuzzy RPQ evaluation process. A spatial network case-study is used to illustrate the soundness of the approach.

KEY WORDS
Distributed computing, knowledge representation, fuzzy set theory, regular path queries, weighted data-graphs.

1 Introduction

The advent of ubiquitous fast networks, cheap storage and processing cycles allows for the accumulation, organization and access of large collections of data. In many areas such as communications and traffic networks, biological data management, cartography and web information systems, large databases are represented as labeled graphs for which regular path queries (RPQs) [1] represent a common and convenient way to access and extract knowledge. Users can further specify the desired knowledge by expressing their queries in terms of weighted RPQs (essentially weighted automata), e.g. requesting the cheapest of the queried paths. Recent work [11, 17, 18] explored computational aspects of evaluating regular path queries on large, weighted distributed data-graphs and, in particular, single and multiple source distributed evaluation, termination detection, fault tolerance and computation in grid environments. In this work we further look to enhance the practicality of extracting information from distributed data-graphs by augmenting their expressive power with the use of *information granules* [13].

For instance, consider the example of a *spatial network*, e.g., a road map [11]. A typical RPQ for this application could specify, among others, the destination, some intermediate locations and the specific kind of connection (road) between locations (cities). An RPQ provide us with enough descriptive power to express a request such as "*I would like to go from Pisa to Florence via Lucca, passing through a road between Pisa and Lucca, and through a road or an highway between Lucca and Florence*". One can further limit the paths returned by the evaluation of the afore mentioned query by designing a real numbered weighted RPQ, but this approach is too rigid, lacks convenience and will not be able to allow requests such as "*I would like to go from Pisa to Florence via Lucca, passing through a road between Pisa and Lucca in less than 20 minutes, and through an highway in about 30 minutes or a road in more than 40 minutes between Lucca and Florence*".

To allow for such requests we propose to enhance RPQs with elements of *granular computing* [13], a theory that deals with operations performed over information granules, rather than singular and exact values. Granular computing models the abstraction process of the human mind, and allows to associate a *semantic meaning* to data, i.e., to "*compute with words*" [10]. In the literature, many theories dealing with the representation of information granules have been developed. Fuzzy set theory (FST) [10] is the most established theory of information granulation. The basic computational units of FST are the *fuzzy sets*, i.e. sets with elements whose *degree of membership* ranges in the $[0, 1]$ real interval. For instance, a fuzzy set can be used to model easily the unprecise time distances expressed by "*less than 20 minutes*", "*about 30 minutes*" and "*more than 40 minutes*" in the previous example. In our work, we represent information granules in the form of *fuzzy quantities*, i.e. fuzzy sets defined over a real-valued universe of discourse [19].

In the following, we introduce a framework for the distributed evaluation of fuzzy weighted RPQs over data-graphs. We first define a basic model that recalls the key aspects of an algorithm that performs the distributed evaluation of single source queries on data-graphs [11, 17]. Then, we present an extended model that exploits fuzzy weighted automata for querying data-graphs whose edges are weighted by fuzzy quantities. In particular, we describe two strategies for determining the costs of the paths computed by the query evaluation process. Finally, we compare the two strategies on a spatial network example.

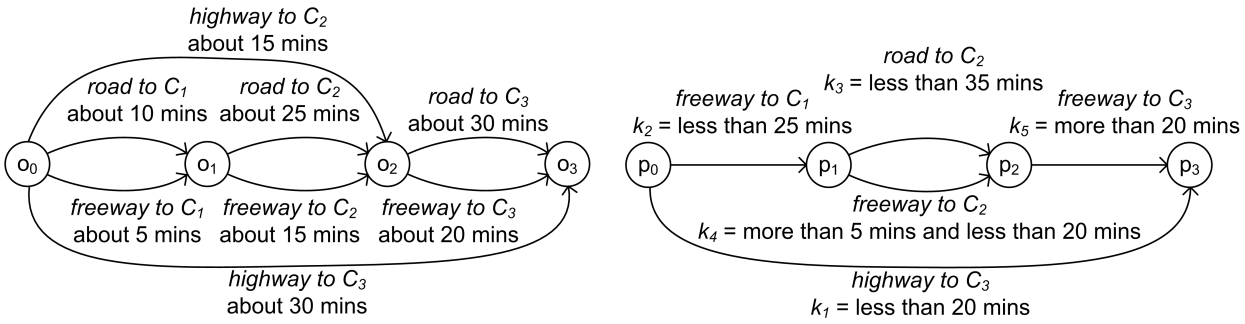


Figure 1. Sample (a) data-graph and (b) RPQ with fuzzy weights.

2 Previous Work

The problem of evaluating fuzzy weighted RPQs on a distributed data-graph is related to at least three different topics: fuzzy databases, fuzzy shortest path problems, and parallel and distributed computation over data-graphs.

Many attempts have been done in the literature to augment the traditional database models with fuzzy techniques [3, 4]. Most of the previous work focused on the entity-relations and on the object-oriented models. Work in the field of fuzzy databases explores many different issues, ranging from the soft evaluation of the matching between a query and the database content, to the representation of attributes via fuzzy concepts [4]. For instance, [3] details a classification of the possible attribute representations by means of FST. Our model gathers inspiration from both approaches, defining a soft matching procedure that compares fuzzy query descriptions with the fuzzy concepts in the database.

Another topic related to data-graph's weights granulation is the fuzzy shortest path problem (FSPP) [8, 12]. This topic has been deeply analyzed: it mostly consist in augmenting the weights of a graph with fuzzy quantities rather than crisp values, and in expanding algorithms for the evaluation of shortest paths on such graphs. The conversion from crisp numbers to fuzzy quantities leads to some difficult theoretical problems that make the search for efficient algorithms difficult.

A number of recent works [1, 9, 16, 11, 17, 18] dealt with different aspects of parallel and distributed evaluation of RPQs on graph databases. The algorithms presented in [1, 16] use unweighted data-graphs and their generalization is not immediate. The approach of [9] describes a parallel implementation of shortest path on data-graphs and discusses practical techniques for improving performance. The works of [17, 18] develop algorithms for the single and multiple source distributed evaluation of real numbered weighted RPQs on data-graphs, while [11] discusses practical aspects of distributed generalized query evaluations in grid environments.

3 The Basic Model

Let us consider a data-graph DB that can be represented as a weighted and labeled graph $DB = (V, E, \Delta, \mathbb{K})$, where $V = \{o_0, \ldots, o_N\}$ is the set of vertices representing data-graph objects and $E \subseteq V \times \Delta \times \mathbb{K} \times V$ is the set of edges. The signature Δ defines an alphabet of symbols in a specific domain. For instance, in the domain of spatial networks, values in Δ can be *road to* C_i, *highway to* C_i, *freeway to* C_i, etc. The weight set $\mathbb{K}$ contains elements in the domain knowledge associated to the edges. More formally, a graph edge $e_j = (o_j, \delta_j, \omega_j, o'_j)$ represents a relationship between objects o_j and o'_j, identified by the label $\delta_j \in \Delta$ and tied to the domain knowledge associated to the information granule $\omega_j \in \mathbb{K}$. Figure 1(a) shows a sample spatial network weighted by fuzzy quantities that model approximate time distances.

The weights ω_j defined over $\mathbb{K}$ are used to compute a cost $c(\pi)$ of each path $\pi = e_0 \ldots e_j \ldots e_F$, starting from vertex o_0 and ending at vertex o_F. In the extended model the cost $c(\pi)$ will be computed in a general domain $\mathbb{C}$, whose nature depends on the specific application. We associate with domain $\mathbb{C}$ a relation $\preceq_{\mathbb{C}}$ denoting an ordering between the costs paths in $\mathbb{C}$ as determined by an aggregation *cost sum* operator $\oplus_{\mathbb{C}} : \mathbb{C} \times \mathbb{C} \to \mathbb{C}$. For the basic model, we set $\mathbb{K} \equiv \mathbb{C}$: hence the total cost of a path $\pi = e_0 \ldots e_F$ can be calculated as the summation of the edge weights along the given path

$$c(\pi) = \omega_0 \oplus_{\mathbb{C}} \cdots \oplus_{\mathbb{C}} \omega_j \cdots \oplus_{\mathbb{C}} \omega_F. \qquad (1)$$

In the settings of [11, 17], both weights and costs are real numbers, modeling, e.g., the length in miles of a road and the total length of the trip. Formally, we have $\mathbb{K} \equiv \mathbb{C} \equiv \mathbb{R}$: hence, the cost sum $\oplus_{\mathbb{C}}$ is the real $+_{\mathbb{R}}$ operator, while $\preceq_{\mathbb{C}}$ is the standard ordering relation $\leq_{\mathbb{R}}$ between reals.

An RPQ on such a DB is described by a *finite state automaton* (FSA) $A = (P, \Delta, \tau, p_0, P_F)$, where P is a finite set of states, Δ is the signature, τ is the transition relation, p_0 is the initial state, and P_F is the set of final states. Each edge in the automaton identifies a query term and every path leading from the initial state p_0 to a final

state $p_f \in P_F$ determines an admissible instance of the query, i.e., an acceptable path.

The original algorithm looks for the cheapest acceptable paths defined by an RPQ A over a DB distributed among machines in a grid environment [11]. The algorithm starts from the root vertex o_0 and proceeds by incrementally building the set of optimal acceptable solutions in a distributed fashion. Each time a new (partially) acceptable path π^*_{RPQ} is matched on the graph, its (partial) cost $c(\pi, \pi^*_{RPQ})$ is computed by aggregating the costs of the edges via $\oplus_{\mathbb{C}}$. Here, π^*_{RPQ} denotes any admissible partial or complete path of the RPQ A. If the newly matched π^*_{RPQ} is better than the (possibly) already existing one, then π^*_{RPQ} replaces the previous partial path in the set of optimal acceptable solutions. Evaluation of the cheapest path is performed via $\preceq_{\mathbb{C}}$. At each time instant, each machine of the grid is aware of the best partial solutions discovered until then that reach at least one vertex stored in its local memory. Although the algorithm is based on a greedy strategy, it is shown to return the optimal complete paths [11, 17]. Furthermore, the algorithm includes techniques to deal with fault recovery and termination detection over the whole grid. Due to lack of space, we do not report the full algorithm, whose details can be found in [11].

In the next section, we introduce a generalized version of the RPQ matching algorithm that extends the model to distributed querying on DB graphs weighted by information granules. In particular, we derive the conditions under which we can generalize the results obtained in [11, 17] for real-valued weights, to fuzzy edge coefficients. With this respect, we specify the properties of the cost set $\mathbb{C}$, the sum $\oplus_{\mathbb{C}}$ and the ordering relation $\preceq_{\mathbb{C}}$.

4 The Extended Model

As stated above, in our extended model, both data-graphs and RPQs are weighted by fuzzy quantities for enhancing their expressive power. The algorithm is generalized so as to perform a *semantic matching* of the information granules defined over the RPQ paths against the knowledge stored in the DB and to compute on-the-fly the actual $c(\pi)$, by using a measure of *dissimilarity* between the DB and RPQ weights. In this extended model, a fuzzy weighted RPQ is represented by a weighted FSA $A = (P, \Delta, \tau, \mathbb{K}, p_0, P_F)$, where $\mathbb{K}$ identifies the set of the edge weights and τ is a transition relation such that $\tau \subseteq P \times \Delta \times \mathbb{K} \times P$.

A simple example of an RPQ weighted by fuzzy quantities is shown in Fig. 1(b). Obviously, the characteristics of the weight set $\mathbb{K}$ depend on the kind of information granulation that is chosen for knowledge representation. In our particular setting, $\mathbb{K}$ is a semi-ring whose elements are fuzzy quantities defined over a bounded universe of discourse $X \subset \mathbb{R}$ [2], with commutative sum $\oplus$ and distributive product $\otimes$ implementing the fuzzy union $\cup$ and intersection $\cap$ operators, respectively. We refer to this semi-ring as $\mathcal{F}$. In the light of this fuzzy interpretation, we associate each fuzzy quantity $k_i \in \mathcal{F}$ to a linguistic term t_i (e.g. $t_1 =$ *about 5 mins*) by means of a mapping function $M : \mathbb{T} \rightarrow \mathcal{F}$, such that $M(t_i) = k_i$. We remark that the use of this association allows to enforce the transparency and interpretability of the model by using meaningful linguistic terms in place of mathematical notations. In particular, the linguistic approach can be exploited to generate articulated fuzzy descriptions by applying linguistic modifiers [10] (such as "*less than*") to the primitive fuzzy sets, such as "*about 20 minutes*". Linguistic hedges have a clear mathematical formulation and act by modifying the shape of the fuzzy sets to which they are applied. Therefore, it is possible for the user to reason using only linguistic terms and modifiers, thus hiding the complexity of the underlying mathematics.

In [3], the authors proposed an off-the-shelf set of fuzzy data types to represent attribute values in databases. The datatypes map linguistic expressions to fuzzy sets in a standard way. In the following, we will exploit them to represent information granules associated to graph edges.

To make the model clearer, consider the RPQ in Fig. 1(b): its weighted automaton is described by the state set $P = \{p_0, p_1, p_2, p_3\}$, the signature $\Delta = \{$*freeway to* C_1, *road to* $C_2, \dots\}$, the initial state p_0, the final state set $P_F = \{p_3\}$, and by the transition relation $\tau = \{(p_0,$ *highway to* $C_3, k_1, p_3),\ \dots,\ (p_2,$ *freeway to* $C_3, k_5, p_3)\}$, where $\{k_1 =$ *M*(*less than 20 mins*)$, \dots, k_5 =$ *M*(*more than 20 mins*)$\}$ defines the mapping between the linguistic terms and the fuzzy weights $k_i \in \mathcal{F}$.

Clearly, the three alternative paths defined by the RPQ in Fig. 1(b) over the spatial network of Fig. 1(a) are all acceptable. As stated above, from an information-granulation point-of-view, the challenge relies in discerning which of the three alternatives is the most similar to the request, i.e., in performing the semantic matching of the paths. To this aim, we exploit the notion of similarity of information granules [6], i.e., we use a similarity index to compute a dissimilarity measure $c_{ij} = \overline{\mathrm{Sim}}(k_i, \omega_j) : \mathbb{K} \times \mathbb{K} \rightarrow \mathbb{C}$ between each weight ω_j of the acceptable path on the DB and the corresponding weight k_i of the RPQ. Each time a new partial path is discovered by the algorithm, the dissimilarity measure between the weights of the matching edges is computed on-the-fly in the domain $\mathbb{C}$. Consider, for instance, the matching between the RPQ path $\pi_{RPQ} = t_0 \dots t_i \dots t_{F'}$ and the corresponding, acceptable, DB-graph path $\pi = e_0 \dots e_j \dots e_F$. The total matching cost of (π, π_{RPQ}) is then calculated as

$$c(\pi, \pi_{RPQ}) = c_{00} \oplus_{\mathbb{C}} \cdots \oplus_{\mathbb{C}} c_{ij} \cdots \oplus_{\mathbb{C}} c_{F'F}. \quad (2)$$

The choice of the similarity index is dictated by the restrictions placed on the cost domain $\mathbb{C}$ and on the related binary operators. In particular, we require $(\mathbb{C}, \oplus_{\mathbb{C}})$ to be an *additive semigroup*, where $\mathbb{C}$ is a set closed under the associative and commutative sum operator $\oplus_{\mathbb{C}}$. Moreover, we require $\mathbb{C}$ to be an *ordered semigroup* [5] equipped with a complete ordering relation $\preceq_{\mathbb{C}}$ satisfying the *isotonicity* property, that is, for all $a, b, c \in \mathbb{C}$ it holds

$$a \preceq_{\mathbb{C}} b \Rightarrow (a \oplus_{\mathbb{C}} c \preceq_{\mathbb{C}} b \oplus_{\mathbb{C}} c) \wedge (c \oplus_{\mathbb{C}} a \preceq_{\mathbb{C}} c \oplus_{\mathbb{C}} b). \quad (3)$$

In addition, we require $\mathbb{C}$ to have an identity element $\overline{0}_{\mathbb{C}}$ acting as left and right neutral element of $\oplus_{\mathbb{C}}$: hence, $\mathbb{C}$ is an *ordered monoid*.

Under the conditions described so far, we can seamlessly extend the distributed RPQ matching algorithms in [11, 17] to the more general case of fuzzy-weighted datagraphs. In this approach, the crisp-valued edge costs are substituted with elements from the monoid $\mathbb{C}$, while the aggregated costs are computed using the sum operator $\oplus_{\mathbb{C}}$ and the minimum cost conditions are determined by means of the order relation $\preceq_{\mathbb{C}}$. Notice that the very nature of distributed computing requires the isotonicity property in (3) which becomes the most critical of the six desirable properties identified by Wang and Kerre [19] in their study on ordering relations of fuzzy quantities.

The final form of the semigroup $\mathbb{C}$ is determined by the implementation of the cost sum operator. For instance, if $\oplus_{\mathbb{C}}$ is chosen as a the average sum over the elements in $\mathbb{C}$, then $\forall a \in \mathbb{C}$ we have $a \oplus_{\mathbb{C}} a = a$, i.e. idempotency holds for every element of the semigroup and, therefore, $\mathbb{C}$ is a *semi-lattice*. On the other hand, if $\oplus_{\mathbb{C}}$ is implemented as the standard summation operator we have that $a \preceq_{\mathbb{C}} a \oplus_{\mathbb{C}} a$ and $\mathbb{C}$ is a *positive ordered semigroup* [5].

Until now we have focused on the definition of a general framework that allows to derive sound instances of the triplet $\mathbb{Q} = (\mathbb{K}, \mathbb{C}, \overline{\mathrm{Sim}})$. In the remainder of the work, we analyze two alternative formulations, restricted to the special case $\mathbb{K} \equiv \mathcal{F}$.

4.1 Early Defuzzification

The *early defuzzification* approach is a seamless extension of the one proposed in [11]. We calculate the dissimilarity between the RPQ and the data-graph weights as a crisp-valued cost. To this aim, we exploit the well-known set-theoretic Jaccard index [6], which computes a crisp measure of similarity between two fuzzy quantities. More precisely, we set:

- $\mathbb{C} = U$, with $U = [0, 1] \subset \mathbb{R}$, $\oplus_{\mathbb{C}} = \tilde{+}_{\mathbb{R}}$ and $\preceq_{\mathbb{C}} = \leq_{\mathbb{R}}$. The cost sum $\tilde{+}_{\mathbb{R}}$ is the average operation over reals, i.e. given $x, y \in U$, we have $x \tilde{+}_{\mathbb{R}} y = 1/2 \cdot (x +_{\mathbb{R}} y)$;
- $\overline{\mathrm{Sim}}(k_1, k_2) : \mathcal{F} \times \mathcal{F} \to U = \overline{\mathrm{Sim}}_J(k_1, k_2) = 1 - \mathrm{Sim}_J(k_1, k_2)$, where $\mathrm{Sim}_J(k_1, k_2)$ is the Jaccard index computed on the two fuzzy quantities $k_1, k_2 \in \mathcal{F}$. We recall that

$$\mathrm{Sim}_J(k_1, k_2) = |k_1 \cap k_2| / |k_1 \cup k_2|. \quad (4)$$

Instantiating $\oplus_{\mathbb{C}}$ to the average operator $\tilde{+}_{\mathbb{R}}$ serves to unbias cost aggregation with respect to the path length. Alternatively, if we intend to penalize longer paths as in [11, 17], we can define the $\oplus_{\mathbb{C}}$ operator as the sum $+_{\mathbb{R}}$.

In this approach, the defuzzification step, i.e. the transformation of the fuzzy representation into a single crisp value, is performed by the $\overline{\mathrm{Sim}}$ operator each time the edge cost c_{ij} is computed on-the-fly. The early defuzzification approach can be implemented easily, but it has the drawback of loosing too early much of the knowledge represented by the information granules.

4.2 Late Defuzzification

In the *late defuzzification* approach, we delay the transformation of the information granules into crisp values until the evaluation of $c(\pi_a, \pi^*_{RPQa}) \preceq_{\mathbb{C}} c(\pi_b, \pi^*_{RPQb})$ is performed by the distributed algorithm between two matching partial paths. To this aim, we require the edge costs c_{ij} and the path costs $c(\pi, \pi^*_{RPQ})$ to be fuzzy quantities: thus, we need an index that assesses dissimilarity of information granules in terms of fuzzy sets rather than crisp values, i.e., a fuzzy-valued dissimilarity measure. Hence, we define:

- $\mathbb{C} = \mathcal{F}_U$, with $\oplus_{\mathbb{C}} = \tilde{+}_{\mathcal{F}}$ and $\preceq_{\mathbb{C}} = \leq_{\mathcal{F}}$. $\mathcal{F}_U \subset \mathcal{F}$ is the set of fuzzy sets defined on the interval U. The cost sum $\tilde{+}_{\mathcal{F}}$ is the fuzzy extension of the average operation over reals $\tilde{+}_{\mathbb{R}}$, computed using fuzzy arithmetics as

$$\mu_{\tilde{k}_1 \tilde{+}_{\mathcal{F}} \tilde{k}_2}(z) = \sup_{z = x \tilde{+}_{\mathbb{R}} y} \min\left(\mu_{\tilde{k}_1}(x), \mu_{\tilde{k}_2}(y)\right), \quad (5)$$

  where $x, y, z \in U$, $\tilde{k}_1, \tilde{k}_2 \in \mathcal{F}_U$, and $\mu_{\tilde{k}_i}(x) : U \to U$ is the *membership function* that defines the degree of memberships of the fuzzy quantity $\tilde{k}_i$ on U [15]. The $\leq_{\mathcal{F}}$ is an ordering relation of fuzzy quantities [19], properly chosen so as to enforce the ordered semigroup condition of Section 4;
- $\overline{\mathrm{Sim}}(k_1, k_2) : \mathcal{F} \times \mathcal{F} \to \mathcal{F}_U$ is a fuzzy evaluation of the dissimilarity of two fuzzy quantities.

The definition of $\overline{\mathrm{Sim}}$ and of $\preceq_{\mathbb{C}}$ is not immediate, as it requires a sound choice of $\leq_{\mathcal{F}}$ among the many alternative options in the literature [19], as well as the definition of a proper $\overline{\mathrm{Sim}}$.

To the best of our knowledge, the only approach to fuzzy-valued similarity of fuzzy quantities has been introduced by Dubois and Prade [6, 7]. The proposed approach builds the fuzzy-valued similarity by evaluating the Jaccard index on the sets obtained by α-cutting the fuzzy quantities being compared. We recall that an α-cut of a fuzzy quantity k is the crisp set $k_\alpha = \{x | \mu_k(x) \geq \alpha\}$, with $\alpha \in [0, 1]$. The value of the Jaccard index is then used as the membership degree of α. Hence, given two fuzzy quantities k_1 and k_2, the fuzzy-valued similarity Sim_{DB} is built on U as $\mu_{\mathrm{Sim}_{DB}}(\alpha) = \mathrm{Sim}_J(k_{1\alpha}, k_{2\alpha})$. Two equal fuzzy quantities $k_1 = k_2$ are evaluated to $\tilde{1}_{DB}$, with $\mu_{\tilde{1}_{DB}}(x) = 1, \forall x \in U$, while two completely different fuzzy quantities $k_1 \cap k_2 = \emptyset$ are evaluated to $\tilde{0}_{DB}$, with $\mu_{\tilde{0}_{DB}}(x) = 0, \forall x \in U$. We remark that $\tilde{0}_{DB} \subseteq \tilde{k} \subseteq \tilde{1}_{DB}$ holds $\forall \tilde{k} \in \mathcal{F}_U$.

The fuzzy-valued index of similarity introduced by Dubois and Prade cannot be effectively exploited in our approach to build $\overline{\mathrm{Sim}}$. Indeed, most of the existing ordering relations of fuzzy quantities $\leq_{\mathcal{F}}$ are not able to correctly

Table 1. Edge-cost comparison of the early and late defuzzification approaches for the evaluation of the RPQ of Fig. 1.

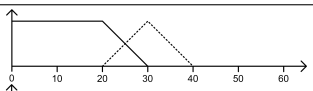
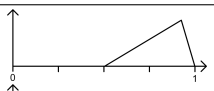
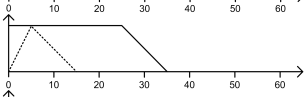
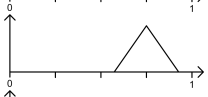
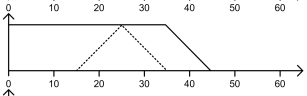
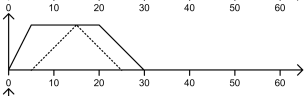
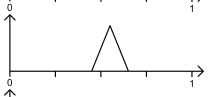
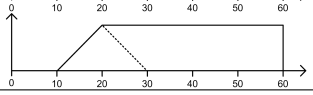
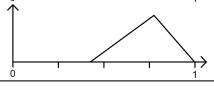
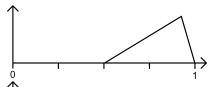
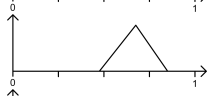
k_i	ω_j	Fuzzy quantities	c_{ij} $\mathbb{Q} = (\mathcal{F}, U, \overline{\mathrm{Sim}}_J)$	$\mathbb{Q} = (\mathcal{F}, \mathcal{F}_U, \overline{\mathrm{Sim}}_{OFM})$
k_1	$\omega_1(o_0 \overset{hway\ C_3}{\longrightarrow} o_3)$		0.9231	
k_2	$\omega_2(o_0 \overset{fway\ C_1}{\longrightarrow} o_1)$		0.7500	
k_3	$\omega_3(o_1 \overset{road\ C_2}{\longrightarrow} o_2)$		0.7500	
k_4	$\omega_4(o_1 \overset{fway\ C_2}{\longrightarrow} o_2)$		0.5556	
k_5	$\omega_5(o_2 \overset{fway\ C_3}{\longrightarrow} o_3)$		0.7778	

Table 2. Path-cost comparison of the early and late defuzzification approaches for the evaluation of the RPQ of Fig. 1.

(π, π_{RPQ})	τ	$c(\pi, \pi_{RPQ})$ $\mathbb{Q} = (\mathcal{F}, U, \overline{\mathrm{Sim}}_J)$	$\mathbb{Q} = (\mathcal{F}, \mathcal{F}_U, \overline{\mathrm{Sim}}_{OFM})$
(π_a, π_{RPQa})	$p_0 \overset{hway\ C_3}{\longrightarrow} p_3$	0.9231	
(π_b, π_{RPQb})	$p_0 \overset{fway\ C_1}{\longrightarrow} p_1 \overset{road\ C_2}{\longrightarrow} p_2 \overset{fway\ C_3}{\longrightarrow} p_3$	0.7593	
(π_c, π_{RPQc})	$p_0 \overset{fway\ C_1}{\longrightarrow} p_1 \overset{fway\ C_2}{\longrightarrow} p_2 \overset{fway\ C_3}{\longrightarrow} p_3$	0.6945	

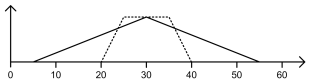


Figure 2. Fuzzy sets $\hat{k}_1$ (solid) and $\hat{k}_2$ (dotted).

recognize $\tilde{1}_{DB}$ as the supremum element of $\mathcal{F}_U$. For instance, let us consider the fuzzy quantities $\hat{k}_1$ and $\hat{k}_2$ shown in Fig. 2. Clearly, $\hat{k}_1$ and $\hat{k}_2$ are not equal. As an example, we choose Yager's second index [20] Y_2 to assess the ordering between $\mathrm{Sim}_{DB}(\hat{k}_1, \hat{k}_2)$ and $\tilde{1}_{DB}$. As proven in [19], Y_2 enforces the isotonicity property required in Section 4. We recall the definition of Y_2

$$Y_2(k) : \mathcal{F} \to \mathbb{R} = \int_0^1 \frac{\sup(k_\alpha) + \inf(k_\alpha)}{2} d\alpha. \quad (6)$$

By definition, we have that $Y_2(k_1) \leq_{\mathbb{R}} Y_2(k_2) \Rightarrow k_1 \leq_{\mathcal{F}} k_2$. But then we have $\tilde{1}_{DB} \leq_{\mathcal{F}} \mathrm{Sim}_{DB}(\hat{k}_1, \hat{k}_2)$ since it can be easily verified that $(Y_2(\mathrm{Sim}_{DB}(\hat{k}_1, \hat{k}_2)) = 0.6055 \wedge Y_2(\tilde{1}_{DB}) = 0.5)$. This trivial example proves that Sim_{DB} is not a good choice for our approach. In the application example of Section 5, we employed an *ordered fuzzy-valued dissimilarity measure* $\overline{\mathrm{Sim}}_{OFM}$ built upon two crisp-valued similarity measures, i.e., the the set-theoretic Jaccard index Sim_J of Eq. (4) and the proximity-based Minkowski 1-metric Sim_M [6], computed as

$$\mathrm{Sim}_M(k_1, k_2) = 1 - \frac{\int_X |\mu_{k_1}(x) - \mu_{k_2}(x)|\, dx}{\int_X dx}. \quad (7)$$

$\overline{\mathrm{Sim}}_{OFM}$ is a triangular fuzzy quantity with core in $\overline{\mathrm{Sim}}_J(k_1, k_2)$. Triangle extremes are computed by

$$\int_U \overline{\mathrm{Sim}}_{OFM}(k_1, k_2) dx = \frac{1}{2} \cdot (1 - \mathrm{Sim}_M(k_1, k_2)). \quad (8)$$

A formal analysis of such measure is beyond the scope of the this paper, but, roughly speaking, the proximity-based index is used to assess the uncertainty of the dissimilarity measured by the set-theoretic index. The infimum and the supremum elements $\tilde{0}_{OFM}$ and $\tilde{1}_{OFM}$ are the singletons in 0 and 1, respectively. More formally, we have $\mu_{\tilde{1}_{OFM}}(1) = 1$ and $\mu_{\tilde{1}_{OFM}}(x) = 0\ \forall x \neq 1$, and similarly for $\mu_{\tilde{0}_{OFM}}(x)$. We remark that, choosing Y_2 as $\leq_{\mathcal{F}}$, $\tilde{0}_{OFM} \leq_{\mathcal{F}} \tilde{k} \leq_{\mathcal{F}} \tilde{1}_{OFM}$ holds $\forall \tilde{k} \in \mathcal{F}_U$.

5 Application Example

We applied both the early and late defuzzification approaches on the matching paths resulting from the evaluation of the RPQ in Fig. 1(b) over the DB in Fig. 1(a). Table 1 shows the comparison between the edge costs computed by using the two approaches. Table 2 shows the aggregated results over the three alternative acceptable complete paths.

Both approaches recognize π_{RPQ_c} as the cheapest path, i.e. the most similar to the user request. However, the late defuzzification approach is able to characterize more sharply the actual semantic difference between the information granules associated with the query and with the DB. For instance, consider the edge costs c_{22} and c_{33}: in early defuzzification the costs are identical, whereas late defuzzification produces a wider support for c_{33}, taking into account the larger difference between the support widths of k_3 and ω_3 with respect to k_2 and ω_2. This can be interpreted as an higher degree of fuzzyness in the evaluation of the cost.

A further analysis of the different effects produced by the early and the late defuzzification approaches on large data-graphs will be subject of future works.

6 Conclusion

Dealing with semantically meaningful representations of knowledge is a key challenge for the development of innovative database applications. In this work, we introduced a general framework for the distributed evaluation of fuzzy weighted RPQs on data-graphs whose semantics is characterized by means of information granules. In particular, we detailed the application of this framework to the design of an algorithm for the distributed mining of fuzzy data-graphs.

The theoretical issues concerning the properties of the fuzzy ordering and similarity operator deserve further studies. In particular, we intend to analyze the retrieval performance of the algorithm with respect to different choices of such operators. It would also be interesting to instantiate the model to information granules that have different (non-fuzzy) representations. For instance, we plan to study the general case of $\mathbb{K}$ as a multi-dimensional space, that is $\mathbb{K} \subseteq \mathbb{K}_1 \times \mathbb{K}_2 \times ... \times \mathbb{K}_n$, where each $\mathbb{K}_i$ identifies a different domain knowledge, each represented by information granules of a (possibly) different type. This problem could be addressed by exploiting the recent results on the evaluation of proximity in heterogenous spaces [14]. On a more practical side, we intend to evaluate the performance of the two defuzzification strategies described in Section 4, with respect to distributed data-graph querying, as done in [11].

References

[1] S. Abiteboul, P. Buneman, and D. Suciu. *Data on the Web: from relations to semistructured data and XML*. Morgan Kaufmann, 1999.

[2] D. Bacciu, A. Botta, and H. Melgratti. A fuzzy approach for negotiating quality of services. In *Proc. of TGC 2006*, LNCS, 4661:200-217,Springer, 2007.

[3] A. Bahri, S. Chakhar, Y. Naija, and R. Bouaziz. Implementing imperfect information in fuzzy databases. In *Proc. of ISCIII 2005*, 2005.

[4] P. Bosc, D. Kraft, and F. Petry. Fuzzy sets in database and information systems: Status and opportunities. *Fuzzy Sets Syst.*, 156(3):418–426, 2005.

[5] P. Conrad. Ordered semigroups. *Nagoya Math. J.*, 16:51–64, 1960.

[6] V.V. Cross and T.A. Sudkamp. *Similarity and compatibility in fuzzy set theory*. Physica-Verlag, 2002.

[7] D. Dubois and H. Prade. A unifying view of comparison indices in a fuzzy set-theoretic framework. In R.R. Yager, editor, *Recent developments in fuzzy set and possibility theory*, pages 3–13. Pergamon, 1982.

[8] F. Hernandes, M.T. Lamata, J.L. Verdegay, and A. Yamakami. The shortest path problem on networks with fuzzy parameters. *Fuzzy Sets Syst.*, 158(14):1561–1570, 2007.

[9] M. Hribar, V. Taylor and D. Boyce. Implementing parallel shortest path for parallel transportation applications. *Parallel Comp.*, 27(12):1537–1568, 2001.

[10] G.J. Klir and B. Yuan, editors. *Fuzzy sets, fuzzy logic, and fuzzy systems: selected papers by L.A. Zadeh*. World Scientific, 1996.

[11] Z. Miao, D.C. Stefanescu, and A. Thomo. Grid-aware evaluation of regular path queries on spatial networks. In *Proc. of AINA07 (to appear)*, 2007.

[12] S. Okada. Fuzzy shortest path problems incorporating interactivity among paths. *Fuzzy Sets Syst.*, 142(3):335–357, 2004.

[13] W. Pedrycz. Granular Computing - the emerging paradigm. *J. of Uncertain Syst.*, 1(1):38–61, 2007.

[14] A. Ralescu and M. Minoh. Measuring proximity between heterogeneous data. *Proc. of FUZZ-IEEE'07*, 2007.

[15] T. J. Ross. *Fuzzy Logic with Engineering Applications*. Wiley, 2004.

[16] D. Suciu. Distributed query evaluation on semistructured data. *ACM TODS*, 27(1):1–62, 2002.

[17] D.C. Stefanescu, A. Thomo, and L. Thomo. Distributed evaluation of generalized path queries. In *Proc. of SAC'05*, pages 610–616, 2005.

[18] D.C. Stefanescu and A.Thomo. Enhanced regular path queries on semistructured databases. *EDBT workshops*, 700–711, 2006.

[19] X. Wang and E.E. Kerre. Reasonable properties for the ordering of fuzzy quantities. *Fuzzy Sets Syst.*, 118(3):375–405, 2001.

[20] R.R. Yager. On choosing between fuzzy subsets. *Kibernetes*, 9:151–154, 1980.