Boosting Ontology Alignment Extraction with flow theory

Touati Chahira, Benaissa Moussa and Lebbeh Yahia

Computer Science Department
LITIO Laboratory
Oran, Algeria
Chahira40,moussabenaissa,ylebbah@yahoo.fr

Abstract—**Ontology matching is a key interoperability enabler for the Semantic Web since it takes the ontologies as input and determines as output correspondences between the semantically related entities of those ontologies. We present in this paper a graph based approach to tackle the ontology matching problem. The objective is to address the combinatorial aspects related to this issue. More precisely, our approach[1] consists in modeling the problem of extracting an alignment (matching) which satisfies a cardinality constraints, as minimizing some cost on feasible flow problem defined on a bipartite graph. The found solution represents the best alignment which maximizes the global similarity between the entities of the two ontologies.**

Keywords-Ontology alignment; ontology matching; Graph based approach; feasible flow; minimum cost flow; global similarity.

I. INTRODUCTION

Ontologies are at the present time in the middle of the work undertaken in the semantic Web. Aiming at establishing representations through which the machines can handle the semantics of information, the construction of ontologies requires at the same time a study of human knowledge and the definition of representation languages, as well as the development of systems to handle them.

Ontology allows describing a domain by defining vocabulary and axioms that govern it. In this context, ontology defines a set of concepts and their relationships with other concepts by specialization or through properties.

Ontologies are often different because they were conceived with a view to achieve various goals and describe sometimes numerous fields. Nevertheless, it is not rare to state the existence of common information between these ontologies. For example, given two ontologies, the same concept can be indicated via several terms, or the case of two ontologies which express the same knowledge by using different specification (see Fig. 1).

Moreover users of ontologies don't use only their own ontologies, they must often integrate or adapt other ontologies to solve their problems. Unfortunately it is very difficult to use in a simultaneous way these ontologies for a new application. This problem of heterogeneity between knowledge expressed within each of them must be resolved. Therefore, semantic links between entities belonging to two ontologies different must be established, which is the purpose of the ontology alignment [14].

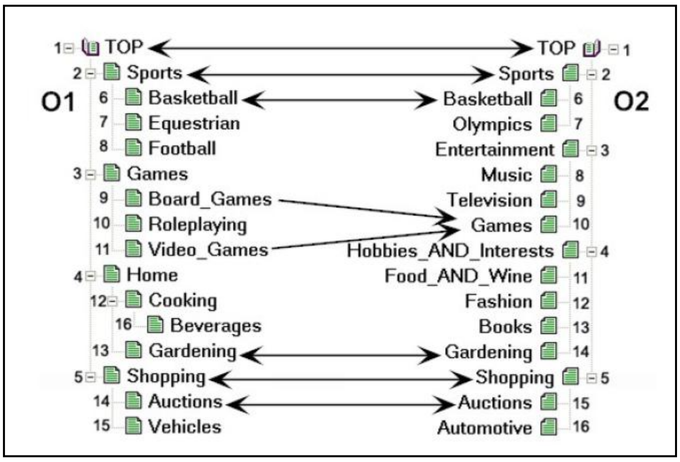

Figure 1. Matching between two ontologies.

Given two ontologies, the alignment generates a set of matches each linking two entities (eg concepts, instances, properties, terms, etc..) by a relation (equivalence, subsumption, incompatibility, etc..), which may include a degree of confidence. All correspondences, also called alignment, can be used to merge ontologies, migrate data between ontologies or translating queries formulated in terms of ontology to another.

The idea of this paper consists in exploiting the graph theory and in particular the minimum cost flow algorithm to solve the problem of identifying an ontology alignment which satisfies the cardinality constraints and has a maximum global similarity. It is necessary to note on this level that the work presented here approaches the alignment extraction problem which has formal properties. On the other hand, it does not approach the aspect of similarities calculation which is supposed to be calculated in addition. In this paper the focus is on the experimental part. The results presented in [21] clearly show a better performance of our approach compared to the standard approach used (Section VI). The objective of this paper is to push this particular empirical investigation by comparing our results with those obtained using an alternative approach [22] cited in [3] and appears to be very effective. The results are detailed in Section VI.

[1]This work is supported by TASSILI research program 11MDU839 (France,Algeria).

The paper is organized as follows: we begin by giving some related work and then in section III we introduce an illustrating example to present the overall approach. In section IV we present some preliminary notions of ontology alignment necessary to understand the next sections. We present successively the notions of correspondence, alignment, alignment properties and alignment extraction methods. Then, in section V, we present our approach which consists of a flow model of the alignment extraction problem which satisfies the cardinality constraints and has a maximum global similarity. In this section, we present our contribution which consists in the construction of network on which we apply the minimum cost algorithm to select the alignment with the necessary properties. Finally we present our experimental results and conclude the paper.

II. RELATED WORK

There has been important background work that can be used for ontology alignment: in discrete mathematics for matching graphs and trees [9], [12], in databases for reconciling and merging schemas [7], in machine learning for clustering compound objects described in a restricted first order logic [2].

In this section, we will look at related systems with a special focus on the topics of extraction methods:

Globally we can distinguish two approaches for handling the alignment extraction problem.

1) Interactive approach: The user is involved in the alignment extraction process. One way to implement this approach consists of displaying all entity pairs with their confidence measures and those judged the most relevant by the user are selected. This approach seems more relevant than the automatic one especially in traditional applications where large data sets are handled [4]. In this case we can quote [1]

2) Automatic approach: which our approach is based. Correspondences between entities are extracted automatically without the user intervention. Within this approach, various methods have been proposed in literature. These methods depend heavily on the properties of the target alignment. In this case we can quote two works:

- [19] The principle of this method is based on the idea that it is possible to infer logical constraints by comparing subsumption relations between concepts of the ontologies to be matched. A standard algorithm to solve the problem of extracting correspondences is known as the Hungarian method [17]. This method expects a real-valued matrix as input and creates a one to one assignment, such that the sum of the chosen entries is minimal. To use the Hungarian method the input mapping M’ has to be transformed into a corresponding matrix H. Each concept of the source ontology corresponds to a row and each target concept corresponds to a column. Since the Hungarian method finds a minimal assignment an entry in the matrix has to be interpreted as distance between two concepts, where the distance between C_1 and C_2 is defined as 1 − similarity. If there exists no such correspondence in M’ the distance is set to 1. In most matching situations it will not be possible to match all or even the majority of concepts. Matching candidates will thus not be available. Therefore, the input matrix has to be extended by additional concepts that play the role of alternative matching candidates.
- [3] They provide an efficient solution to this problem by reducing it to the maximum weight matching in a bipartite graph and by adopting the Shortest Augmenting Path algorithm (SAP) [22]. They provide an alternative solution to this problem by reducing it to the maximum weight matching in the bipartite graph G = (S∪T, E), where S contains the source ontology concepts, T contains the target ontology concepts, and E contains an edge oriented from S to T for each correspondence with a similarity value higher than the threshold, weighted with the threshold value itself. They recall that a maximum weight matching M is a subset of the edges in E such that for each vertex in G at most one adjacent edge is contained in M and the sum of the weights (i.e., the similarity values) of the selected edges is maximized. Thanks to this transformation, they can adopt the Shortest Augmenting Path algorithm (SAP) to find the optimal solution in polynomial time.

Many diverse solutions of matching have been proposed so far; see [6], [8], [10], [11], [20]. But in our contribution we provide the best alignment between two ontologies which maximizes the global similarity and verifies the cardinality constraints.

III. ILLUSTRATIVE EXAMPLE

Let O and O’ be two ontologies. O contains the concepts $\{C_1,C_2,C_3,C_4\}$ and O’ contains the concepts $\{C'_1, C'_2, C'_3, C'_4, C'_5, C'_6\}$. We assume that some technique computing similarities between concepts of O and O’ has produced the following similarity matrix S. We assume in this example that the source ontology cardinality O is equal to 3 and the target ontology cardinality O’ equal to 2.

TABLE 1 Similarity matrix

O O’	$C'_1 : 2$	$C'_2 : 2$	$C'_3 : 2$	$C'_4 : 2$	$C'_5 : 2$	$C'_6 : 2$
$C_1 : 3$	0.81	0.61	0.73	0.61	0.50	0.44
$C_2 : 3$	0,92	0.83	0.39	0.52	0.84	0.12
$C_3 : 3$	0.64	0.62	0.26	0.74	0.94	0.31
$C_4 : 3$	0.23	0.96	0.32	0.25	0.60	0.82

We require again that all similarity values must be greater or equal to a threshold s = 0.5. Filtering S over the above threshold gives the following matrix (Table 2):

The problem here is to extract automatically an alignment with the maximum global similarity between concepts of O and O’ which satisfies the cardinality constraints. To handle efficiently this problem we propose an algorithm based on graph theory.

TABLE 2. Similarity matrix after filtering

O O'	C'_1 : 2	C'_2: 2	C'_3 : 2	C'_4 : 2	C'_5 : 2	C'_6 : 2
C_1 : 3	0.81	0.61	0.73	0.61	0.50	
C_2 : 3	0,92	0.83		0.52	0.84	
C_3 : 3	0.64	0.62		0.74	0.94	
C_4 : 3		0.96			0.60	0.82

The algorithm proposed in our approach is the minimum cost flow algorithm. We give hereafter the flow graph for the above example Fig 2.

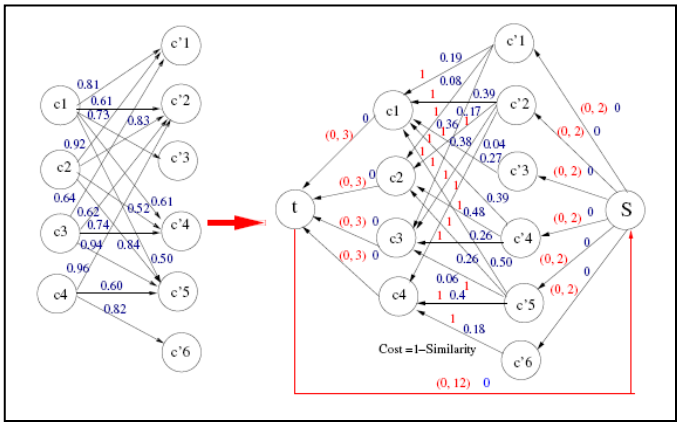

Figure 2. Network flow of the example

We can find several solutions where the flow is max, for example {(C_1,C'_3), (C_1,C'_4), (C_1,C'_5), (C_2,C'_1), (C_2,C'_2), (C_2,C'_4),(C_3,C'_1), (C_3,C'_2), (C_4,C'_5), (C_4,C'_6)} with global similarity equal to 6.79. But after the application of the minimum cost flow algorithm, we obtain the following alignment: {(C_1,C'_1), (C_1,C'_3), (C_1,C'_4), (C_2,C'_1), (C_2,C'_2), (C_2,C'_5), (C_3,C'_4), (C_3,C'_5), (C_4,C'_2), (C_3,C'_6)} with global similarity equal to 8.2. Here is the contribution of this paper: providing the best alignment between two ontologies which maximizes the global similarity and verifies the cardinality constraints.

IV. PRELIMENAIRES ON ONTOLOGY ALIGNMENT

The process of alignment between ontologies aims to identify semantic correspondences between their entities. In this section we give some definitions on the key materials of this work [14].

A. Correspondence notion

Let O and O' be two ontologies. A correspondence M between O and O' is a quintuple <id, e, e', R, n> where:

- id is a unique identifier for the correspondence M;
- e and e' are entities of O and O' respectively (e.g., concepts, roles or instances);
- R is a semantic relation holding between e and e' (for example, equivalence, more general, more specific, disjointness);
- n is a confidence measure (typically a value in [0,1]).

B. Alignment notion

The alignment can be defined as a set of correspondences. The alignment process has two ontologies O and O' as input and produces an alignment A between entities of O and O' as output (See Fig. 3). Other parameters can complete this definition, namely:

1) A preliminary alignment A' to be completed or refined by the process.
2) External resources r such as a thesaurus for example.
3) Parameters p such as thresholds or weights for example.

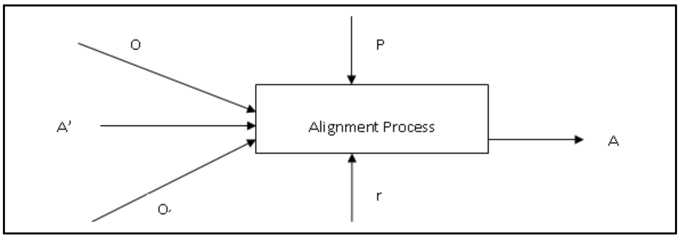

Figure 3. Alignment process

C. Alignment properties

Total alignment: An alignment A is said to be total (or complete) from O to O' if and only if each entity of O has a corresponding entity in O'.

Injective alignment: An alignment A is said to be injective from O to O' if and only if for each correspondences C = <id_C, e_1, e_2, n, R> Є A and C' =<id_C, e'_1, e'_2, n, R> Є A we must have: if $e_2 = e'_2$ then $e_1 = e'_1$.

D. *The cardinality constraints and ontology alignment*

After the selection process, the result alignment must have the following properties:

1) The global similarity should be maximal. We mean by global similarity, the sum of the values of the similarities of different correspondences that forms the alignment.

2) The cardinality constraints must be verified. We distinguish in general the following cases:

a) Case 1: 1−1 constraints: each entity of the ontology source must correspond to one entity of the target ontology and each entity of the target ontology must correspond to one entity of the source ontology.

b) Case 2: n − m constraints: each entity of the ontology source must correspond to at most m entities of the target ontology and each entity of the target ontology must correspond to at most n entities of the ontology source.

*c) Case 3: n − *, * − m, * − ** constraints: in this case we use the symbol * to mean that we don't impose cardinality constraints.

We propose in this paper, a flow based model that can extract an alignment that satisfies the two conditions above. This model can be easily adapted to understand the case of total and injective alignment and consider more general cardinality constraints where an entity of a given ontology (source or target) must correspond to at most n entities and at least m entities of the other ontology.

In Fig. 4, we give some examples of the configurations of multiplicity between two ontologies.

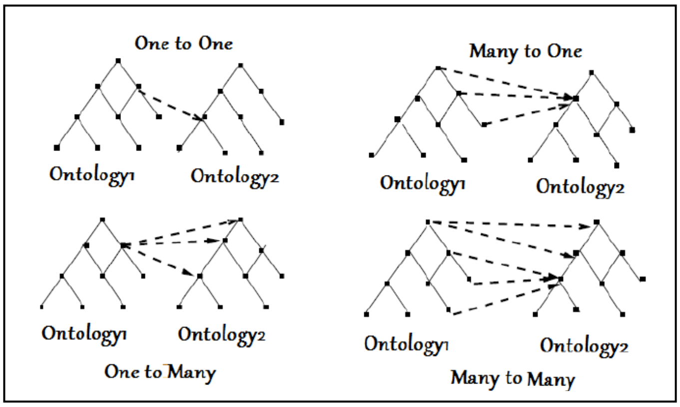

Figure 4. Different kinds of alignments

E. *Alignment extraction methods*

We can distinguish two categories of methods for extracting an alignment.

- Methods based on local optimizations: The methods of this category extract the target alignment by iterating over correspondences belonging to the initial alignment (typically the similarity matrix). At each step, similarity within each pair of entities is locally maximized [13].
- Methods based on global optimization: The methods of this category proceed by optimizing a global criterion rather than optimizing locally. Typically, we consider the global similarity between corresponding entities as an objective function to be maximized: $f = \sum_{C \in A} conf(C)$ where $conf(C) = conf<id_C, e_1, e_2, n, R >= n$ [16].

V. A FLOW MODEL OF THE ALIGNMENT EXTRACTION PROBLEM

In this work, we use the automatic approach and the method based on global optimization. Our objective is not to extract just an alignment but is to extract the best one which maximizes the objective function f.

The minimum cost flow algorithm is the algorithm used in our approach to solve the problem of extracting an alignment between two ontologies. The minimum cost flow problem is a generalization of the maximum flow problem. It is one of the most fundamental network flow problems.

Networks are especially convenient for modeling because of their simple nonmathematical structure that can be easily depicted with a graph. This simplicity also reaps benefits with regard to algorithmic efficiency.

Suppose that we have a network G(V,E) with nodes V = 1, .., n, directed edges $E = (i, j) \in V \times V$. Network G has two special nodes s and t called the source and the sink, respectively. For every directed edge $(i, j) \in E$, the cost of pushing one unit of flow from node i to node j is c(i, j), and the positive capacity is u(i, j). The minimum cost flow problem is to find a maximum flow of minimum cost from the source node s to the sink node t.

Different approaches have been proposed to solve the minimum cost flow problem. Extensive discussion of this problem and its applications can be found in the book and paper of Ford and Fulkerson [15], Edmonds and Karp [5].

Given a flow network G(V,E) with source $s \in V$ and sink $t \in V$, where edge $(i, j) \in E$ has capacity u(i, j), flow f(i, j) and cost c(i, j). The cost of sending this flow is f(i, j). c(i,j). You are required to send an amount of flow d from s to t.

The objective is to find a feasible flow which has the minimum cost $\sum_{(i,j) \in E} f(i, j) \cdot c(i, j)$.

The algorithm runs in pseudo polynomial time. However, suppose the costs c(i, j) are integers, which are less than or equal to an integer C, Edmonds and Karp [5] have proven that the algorithm halts after at most $1 + (1/4)(n^3 - n)(n - 1)C$ flow augmentations, which can be equivalently rewritten as $O(n^4C)$. For more details we return the readers to [18].

We present in this section our contribution. It is initially about the construction of a network which models the cardinality constraints with in particular a wise choice of the capacity constraints. Then the choice of the costs and the research of the maximum flow which has the minimum cost and ensures the optimality of the global similarity. We detail below network construction rules.

- Orienting each edge from each concept of ontology 2 to concept of ontology 1. For such an arc (u, v) : the lower and the upper bound are initiated as follows: $l_{uv} = 0$ and $u_{uv} = 1$. The cost associated with the edge (u, v) is : $c(u, v) = 1 - similarity(u, v)$.
- Adding a vertex s and an arc from s to each concept of ontology 2. For such an arc (s, a_i): the bounds of capacities are defined as follows: $l_{sai} = 0$, $u_{sai} = n$. The cost associated with each arc (s, a_i) is equal to 0. The value n represents the cardinality with source ontology O.
- Adding a vertex t and an arc from each concept of ontology 1 to t. For such an arc (x, t) : the bounds of capacities are defined as follows: $l_{xt} = 0$, $u_{xt} = m$. The cost associated with each arc (x, t) is equal to 0. The value m represents the cardinality with target ontology O'.
- Adding an arc (t, s) with $l_{ts} = 0$. The cost associated with the arc (t, s) is equal to 0. And $u_{ts} = m \times |$ the number of concepts of ontology 1 $|$.

After the construction of the network, we apply the minimum cost flow algorithm. This algorithm ensures that the low obtained is compatible (it checks the capacity constraints, thus it checks also the cardinality constraints). On another side, it ensures that the global costs of the flow are minimum, therefore that the total similarity of alignment is maximum since the cost is equal to 1−sim, with sim the similarity, and that the maximum value of the similarity between two entities is equal to 1.

We notice that this network makes it possible to model all the cardinality constraints. An injective alignment, for example, can be obtained by fixing the maximum capacities at 1. The completeness property is assured if we set the maximum and minimum capacities of all the arcs (x, t) to 1. Moreover, this model makes it possible to take into account more general cardinality constraints. Indeed, it is possible to represent the following constraint in particular: ”each entity of an ontology can be associated with at most n entities and at least m entities of other ontology”.

VI. EXPERIMENTATION AND DISCUSSION

We detail in this section the results obtained. Indeed, we did not find in the specialized literature other systems having tackled the question of the extraction of an alignment with multiple cardinality constraints and having provided detailed results being able to be used like support as comparison with our approach. Only work that we know is described in [3] but which does not provide detailed results only for square matrices. For this reason, we implement the algorithm [22] used in this work to be able to compare the two approaches.

In the figures 5, 6 and 7, we present the results of comparison between the minimum cost flow algorithm used in our approach and Karp algorithm used in [3].

We noticed two main results from our experimentations:

1) Case 1-1: Rectangular similarity matrices, our min cost flow algorithm and the algorithm used in [3] return a result in the same time. For example with some matrix 100×1000 our algorithm gives a solution after 1 second and the algorithm used in [3] gives a solution after 0.5 second. But on square matrices, the algorithm used in [3] is better than our algorithm.

2) Case n-m: Rectangular similarity matrices, for example with some matrix 200×2000 with cardinality constraints equal to 4 for ontology1 and 3 for ontology2, the min cost flow algorithm gives a solution after 11 seconds because it reacted directly on the matrix such as it is But the Karp algorithm gives a solution after 28 seconds. Because for the n-m selection case they reuse the algorithm for the 1-1 matching case several times sequentially. They keep track of the number of mappings found for each vertex, and at the end of each iteration, they remove from the bipartite graph all the vertices together with their adjacent edges that have reached the maximal cardinality. The algorithm terminates when the graph is empty. In other words, our approach gives results better for ontologies where the difference between the number of concepts is important. Whereas that in the case of large ontologies and with weak variation the results are less powerful than in the Karp algorithm.

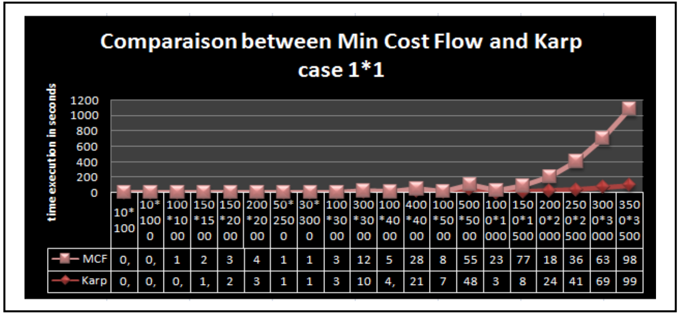

Figure 5. Comparison between Min Cost Flow algorithm and Karp algorithm used in[3]: case 1*1

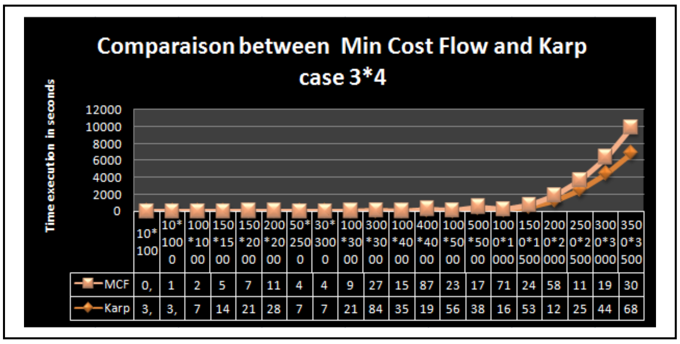

Figure 6. Comparison between Min Cost Flow algorithm and Karp algorithm used in[3]: case 3*4

3) Our algorithm depends on the cardinality constraints: If we change cardinality constraints we obtain the results represented in Fig. 7. With cardinality constraints equal to 20 for ontology1 and 10 for ontology2, the Min Cost algorithm reacted more slowly than we use the algorithm with cardinality constraints equal to 4 for ontology1 and 3 for ontology2.

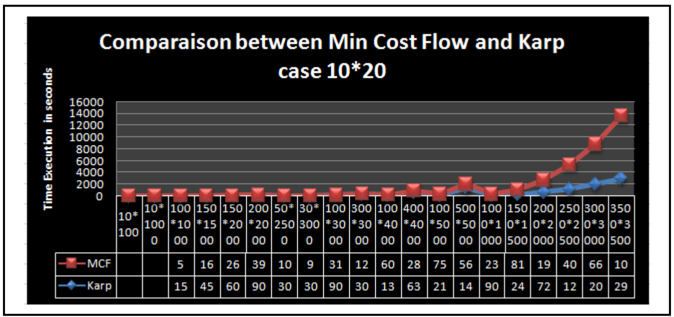

Figure 7. Comparison between Min Cost Flow algorithm and Karp algorithm used in[3]: case 10*20

In Fig. 8, we compare the results of the Hungarian[2] algorithm [17] used in other alignment system with the results of the minimum cost algorithm used in our approach (the cardinality constraints considered in this cases are of type (1-1) [21]).

The Hungarian algorithm works only on square matrices, and to adapt it and make it applicable to any type of matrix, we used the approach proposed in [19].

We noted two main results of our experiments:

[2] The implementation is available at http://code.google.com/p/hungarianassignment/

1) Rectangular similarity matrices: our min cost flow algorithm treats effectively the problem than the Hungarian algorithm. For example of a matrix 100 × 1000 the min cost flow algorithm finds a solution after 12 seconds because it reacts directly on the matrix. One of the reasons for this behavior is that the Hungarian algorithm transforms the matrix into a matrix 1100× 1100. It returns a solution after 38 seconds. So for large matrices the Hungarian algorithm is less efficient than the algorithm based on the flow.

2) Square similarity matrices: The Hungarian algorithm is better. Since this algorithm is established in practice to find an optimal allocation of such matrices.

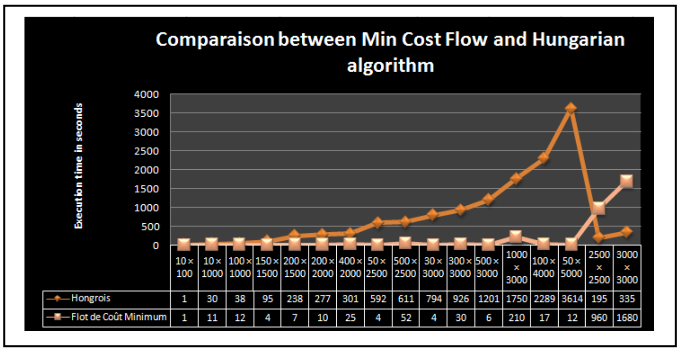

Figure. 8 Performance comparison between the Minimum Cost Flow and the Hungarian method on different input sizes

By exploiting complexities given of these two algorithms (i.e., Hungarian $O(n^3)$, and the minimum cost algorithm $O(n^4C)$), and since all the capacities are lower than 1, the complexity of the minimum cost flow algorithm can be written under $O(n^4)$. Therefore, it is provable that on the square matrices, the Hungarian algorithm is better. On the other hand it is not preferable in the contrary case i.e., on the rectangular matrices what is confirmed by our experimentation.

Finally, we can conclude that the minimum cost algorithm has an unquestionable advantage compared to the Hungarian algorithm or the Karp algorithm used in [3] in the context of the ontology matching. Indeed, ontologies generally correspond to rectangular matrices and it is very seldom to have in the reality ontologies with the same number of concepts.

VII. CONCLUSION

In this paper we showed that ontology alignment problem can benefit from the algorithmic techniques developed within flow theory. More precisely we modeled the problem of extracting an alignment which satisfies some cardinality constraints and the objective function defined as the global similarity between the ontologies entities as a graph network. In order to extract such an alignment we used the minimum cost flow algorithm. Then, we showed that the approach presented here handles efficiently the problem with rectangular similarity matrices.

REFERENCES

[1] M. Benaissa and Y. Lebbah, Optimisation de l'extraction de l'alignement des ontologies avec la contrainte de différence. Revue des Nouvelles Technologies de l'Information (RNTI). N_ E.20.Extraction et Gestion des Connaissances (EGC 2011), Editions Hermann.

[2] G. Bisson, Learning in fol with similarity measure. In Proc. 10th AAAI, San-Jose (CA US), pp 82–87, 1992.

[3] I. Cruz,F P. Antonelli and C. Stroe, Efficient selection of mappings and automatic quality-driven combination of matching methods. International Workshop on Ontology Matching , 2009.

[4] P. Shvaiko and J. Euzenat, Ten challenges for ontology matching. In The proceedings of the seventh International conference on Ontologies, Data Bases and application of Semantics (ODBASE), Monterry, Mexico, 2008.

[5] J. Edmonds and R. Karp, Theoretical improvements in algorithmic efficiency for network flow problems. Journal of ACM, pp 248–264, 1972.

[6] A. Doan and A. Halevy, Semantic integration research in the database community: A brief survey. Special Issue on Semantic Integration, pp:83–94, 2005.

[7] E. Rahm and P. Bernstein, A survey of approaches to automatic schema matching. VLDB Journal, pp. 334-350,2001.

[8] P. Shvaiko and J. Euzenat., A survey of schema-based matching approaches. Journal on Data Semantics, pp 146–171, 2005.

[9] C. Papadimitriou and K. Steiglitz, Combinatorial Optimization: Algorithms and complexity. Prentice-Hall, 1998.

[10] Y. Kalfoglou and M. Schorlemmer, Ontology mapping: the state of the art. Engineering Review, pp 1–31, 2003.

[11] E. Rahm and P. Bernstein, A survey of approaches to automatic schema matching. VLDB Journal, pp. 334–350, 2001.

[12] J. Hopcroft and R. Karp, An algorithm for maximum matching in bipartite graphs. SIAM Journal on Computing, pp. 225–231, 1973.

[13] J. Euzenat and P valtchev, Similarity-based ontology alignment in owllite. In The proceedings of the 15th European Conference on Artificial Intelligence (ECAI), pp. 333–337, valencia, spain, 2004.

[14] J. Euzenat and P. Shvaiko, Ontology matching. Springer, 2007.

[15] L. R. Ford and D. R. Fulkerson, Flows in networks. In Press,Princeton, NJ(1962), Princeton Univ, 1962.

[16] C.Meilicke and H.Stuckenschmidt. Analyzing mapping extraction approaches. In proceedings of the second Internationl workshop on ontology matching (OM-2007). Busan (Korea), 11 November 2007.

[17] H. W. Kuhn, The hungarian method for the assignment problem. Naval Research Logistic Quarterly, pp. 83–97, 1955.

[18] J. Liu, Algorithms for minimum cost flows. Computer Science Department, The University of Western Ontario, 2003.

[19] C. Meilicke and H, Stuckenschmidt. Applying logical constraints to ontology matching, Proceedings of the 30th German Conference on Artificial Intelligence (KI-07), Lecture Notes in Artificial Intelligence, Springer, pp. 99–113, 2007.

[20] N. Noy, Semantic integration, A survey of ontology-based approaches. SIGMOD Record, pp. 65–70, 2004.

[21] C. Touati, M. Benaissa and Y. Lebbah, Graph based approach for ontology matching, International Conference on Software Engineering and New Technologies (ICSENT 2012), Tunisia, 2012.

[22] R. M. Karp, An Algorithm to Solve the m*n Assignment Problem in Expected Time O(mnlog n). Networks, pp.143-152, 1980.