MapPSO Results for OAEI 2008

Jürgen Bock[1] and Jan Hettenhausen[2]

[1] FZI Research Center for Information Technology, Karlsruhe, Germany
bock@fzi.de
[2] Griffith University, Institute for Integrated and Intelligent Systems, Brisbane, Australia
j.hettenhausen@griffith.edu.au

Abstract. We present first results of an ontology alignment approach that is based on discrete particle swarm optimisation. In this paper we will firstly describe, how the algorithm approaches the ontology matching task as an optimisation problem, and briefly sketch how the specific technique of particle swarm optimisation is applied. Secondly, we will briefly discuss the results gained for the Benchmark data set of the 2008 Ontology Alignment Evaluation Initiative.

1 Presentation of the system

We introduce the Ontology **Map**ping by **P**article **S**warm **O**ptimisation (MapPSO) system as a novel research prototype, which is expected to become a highly scalable, massively parallel tool for ontology alignment. In the following subsection the basic idea of this approach will be sketched.

1.1 State, purpose, general statement

The MapPSO algorithm is being developed for the purpose of aligning large ontologies. Instance mapping however is not part of our efforts. Motivated by the observation that ontologies and schema information such as thesauri or dictionaries are not only getting numerous on the web, but also are becoming increasingly large in terms of the number of classes/concepts and properties/relations. This development raises the need for highly scalable tools to provide interoperability and integration of various heterogeneous sources. On the other hand the emergence of parallel architectures provide the basis for highly parallel and thus scalable algorithms which need to be adapted to these architectures.

For the presented MapPSO method we formulated the ontology alignment problem as an optimisation problem which allowed us to employ a discrete variant of particle swarm optimisation [1, 2], a population based optimisation paradigm inspired by social interaction between swarming animals. Particularly the population based structure of this method provides high scalability on parallel systems. Particle swarm optimisation furthermore belongs to the group of anytime algorithms, which allow for interruption at any time and will provide the best answer being available at that time. Particularly this property might be interesting when an alignment problem is subject to certain time constraints.

1.2 Specific techniques used

MapPSO utilises a discrete particle swarm optimisation (DPSO) algorithm, based in parts on the DPSO developed by Correa *et al.* [1, 2], to tackle the ontology matching problem as an optimisation problem. The core element of this optimisation problem is the objective function which supplies a fitness value for each candidate alignment.

To find solutions for the optimisation problem, MapPSO simulates a set of particles whereby each particle is a candidate alignment comprising a set of initially random mappings[3]. Each of these particles maintains a memory of previously found good mappings (*personal best*) and the swarm maintains a collective memory of the best known alignment so far (*global best*). In each iteration, particles are updated by changing their sets of correspondences in a guided random manner. Correspondences which are also present in the *global best* set are more likely to be kept, as are those with a very good evaluation. In addition the number of correspondences represented by each particle also changes according to the number of correspondences in the *global best* alignment in a self-adaptation process.

Each candidate alignment of two ontologies is scored based on a weighted sum of quality measures of the single correspondences, and the number of correspondences it consists of. The currently best alignment is the one with the best known fitness rating according to these criteria. According to this revisit of the ontology matching problem, a particle swarm can be applied to search for the optimal alignment.

For each correspondence the quality score is calculated based on an aggregation of scores from a configurable set of base matchers. Each base matcher provides a distance measure for each correspondence. Currently the following well known base matchers are used:

- SMOA string distance [3] for entity names
- SMOA string distance for entity labels
- WordNet distance for entity names
- WordNet distance for entity labels
- Vector space similarity [4] for entity comments
- Hierarchy distance to propagate similarity of superclasses / superproperties
- Structural similarity of classes derived from properties that have them as domain or range classes
- Structural similarity of properties derived from their domain and range classes

For each correspondence the available base distances are aggregated by applying the OWA operator [5]. The OWA operator performs an **O**rdered **W**eighted **A**verage aggregation of the base distances by ordering the base distances and applying a fixed weight vector. The evaluation of the overall alignment of each particle is computed by aggregating all its correspondence distances and accounting for the number of correspondence represented by this particle.

In the current implementation each of the particles runs in an individual thread and all fitness calculations and particle updates are performed in parallel. The only sequential portion on the algorithm is the synchronisation after each iteration to acquire the fitness value from each particle and determine the currently global best alignment.

[3] Currently only 1:1 alignments are supported.

1.3 Adaptations made for the evaluation

Since MapPSO is an early prototype, we did use the OAEI 2008 Benchmark test data during the development process. No specific adaptations have been made.

1.4 Link to the system and parameters file

The release of MapPSO for OAEI 2008 is located in the package `MapPSO` at
`http://ontoware.org/projects/mappso/`

1.5 Link to the set of provided alignments (in align format)

The alignment results of MapPSO for the Benchmark test case of OAEI 2008 are located in the package `alignResults` at
`http://ontoware.org/projects/mappso/`

2 Results

Since MapPSO is in an early development stage, we only participate in the Benchmark test case in the OAEI 2008.

2.1 benchmark

The Benchmark test case is designed to provide a number of data sets systematically revealing strengths and weaknesses of the matching algorithm. In the case of MapPSO the experiences were as follows:

The MapPSO algorithm is highly adjustable via its parameter file and can be tuned to perform well on specific problems, as well as to perform well for precision or recall. To obtain the results presented in table 1 we used a compromised parameter configuration.

For **tests 101-104** MapPSO achieves precision values of around 90 % and recall values of 100 %. Test 102 with a totally irrelevant ontology, however, still determines a number of wrong correspondences.

As for **tests 201-210** results are not as positive, as the quality of the alignment decreases with the number of features that provide linguistic features to exploit. For test case 202 where all names and comments are unavailable, MapPSO performs worst in this group of tests.

In **tests 221-247**, where the structure of the ontologies varies, the results are similar to the 10x tests. Since the main focus of the current implementation of MapPSO's base matchers is on linguistic features, such as string distance and WordNet distance.

The **tests 248-266** combine linguistic and structural problems. As the results show, the quality of the alignments is decreasing with the decreasing number of features available in the ontologies.

For the real-life cases, **tests 301-304**, no uniform results can be derived as the algorithm's precision and recall values vary between 0 and 60 %.

Table 1. MapPSO results for benchmark test cases.

Test Name	Precision	Recall	Test Name	Precision	Recall	Test Name	Precision	Recall
101	0.9	1	241	0.79	1	254-8	0.71	0.15
102	0	NaN	246	0.81	1	257	0.05	0.06
103	0.94	1	247	0.73	0.82	257-2	0.91	0.61
104	0.92	1	248	0.04	0.04	257-4	0.53	0.61
201	0.12	0.13	248-2	0.75	0.79	257-6	0.4	0.52
201-2	0.79	0.88	248-4	0.48	0.54	257-8	0.23	0.27
201-4	0.66	0.7	248-6	0.36	0.4	258	0.08	0.09
201-6	0.5	0.56	248-8	0.16	0.18	258-2	0.74	0.74
201-8	0.28	0.31	249	0.06	0.07	258-4	0.49	0.53
202	0.05	0.05	249-2	0.73	0.82	258-6	0.34	0.39
202-2	0.72	0.81	249-4	0.53	0.59	258-8	0.2	0.23
202-4	0.55	0.6	249-6	0.34	0.38	259	0.01	0.01
202-6	0.34	0.37	249-8	0.16	0.18	259-2	0.68	0.76
202-8	0.2	0.23	250	0.07	0.09	259-4	0.64	0.72
203	0.95	0.94	250-2	0.78	0.85	259-6	0.66	0.74
204	0.85	0.93	250-4	0.67	0.48	259-8	0.66	0.73
205	0.3	0.33	250-6	0.38	0.48	260	0.03	0.03
206	0.35	0.38	250-8	0.21	0.27	260-2	0.67	0.76
207	0.35	0.39	251	0.07	0.08	260-4	0.53	0.72
208	0.78	0.88	251-2	0.76	0.8	260-6	0.64	0.31
209	0.22	0.25	251-4	0.47	0.53	260-8	0.21	0.28
210	0.18	0.2	251-6	0.28	0.3	261	0.04	0.06
221	0.9	1	251-8	0.22	0.24	261-2	0.86	0.36
222	0.91	1	252	0.06	0.06	261-4	0.82	0.27
223	0.96	0.89	252-2	0.62	0.7	261-6	0.75	0.45
224	0.9	1	252-4	0.63	0.71	261-8	0.68	0.79
225	0.9	1	252-6	0.63	0.69	262	0.07	0.09
228	0.8	1	252-8	0.63	0.71	262-2	0.86	0.76
230	0.86	1	253	0.06	0.07	262-4	0.5	0.55
231	0.92	1	253-2	0.75	0.71	262-6	0.79	0.33
232	0.94	1	253-4	0.5	0.56	262-8	0.16	0.21
233	0.79	1	253-6	0.38	0.42	265	0.03	0.03
236	0.8	1	253-8	0.17	0.19	266	0.02	0.03
237	0.93	1	254	0	0	301	NaN	0
238	0.9	0.95	254-2	0.85	0.7	302	0.22	0.21
239	0.89	0.86	254-4	0.83	0.45	303	NaN	0
240	0.71	0.82	254-6	0.37	0.39	304	0.65	0.64

3 General comments

In the following we will provide a few statements on our experiences from participating in the OAEI 2008 competition and briefly discuss future work on the MapPSO algorithm.

3.1 Comments on the results

Firstly it shall be noted that MapPSO is a non-deterministic method and therefore on a set of independent runs the quality of the results and the number of mappings in the alignments will be subject to slight fluctuations.

For many of the benchmark test cases the current implementation of MapPSO could already provide reasonably good solutions. However, particularly alignments which are largely based on structural criteria currently impose a problem on the algorithm and require further development such as the addition of appropriate base matchers. This behaviour is particularly reflected in test cases, where lexical and linguistic information is omitted, such as in 201 and 202.

The submitted results were furthermore all acquired with an identical configuration file with a non-optimised and rather general set of parameters. For individual alignment problems, the quality of fitness values and thereby to some extend the efficiency of the algorithm can be improved by limiting the selection of base matchers to those that are most likely to provide useful ratings for the involved ontologies.

3.2 Discussions on the way to improve the proposed system

One of the most crucial component of MapPSO is the acquisition of fitness values for individual mappings and complete alignments. The MapPSO algorithm currently uses various base matchers, which are, in the current release naively implemented. It can be assumed that improving the current base matchers as well as adding further base matchers for an extended set of criteria will be highly beneficial for MapPSO. This regards in particular the aforementioned problem of taking structural properties of the alignments into account.

In addition, various other optimisations and extensions to the algorithm are conceivable. Particularly the extension of self-adaptation to the weight parameters and further optimisation of the currently implemented self-adapting length of candidate alignments appear to be promising. We hope to participate in next year's OAEI campaign demonstrating better performance on the benchmark test case and providing results for additional larger test cases on which we can demonstrate the scalability of the MapPSO approach.

4 Conclusion

In this paper we briefly introduced our ontology alignment system MapPSO and some results for the OAEI 2008 competition. Despite the fact that MapPSO is still at an early

stage of development we could achieve promising results for the majority of the benchmark alignments. Key features of the discrete particle swarm optimisation approach of MapPSO are high parallel scalability and the possibility to either set time constraints for the alignment or interrupt the alignment process at any time and acquire the best alignment MapPSO could find up to that point. Future work on MapPSO will focus on improving the weighting and scoring methods of the fitness function and improve usage of structural information of the ontologies as a mean of calculating score values for candidate alignments.

Acknowledgement

The presented research was partially funded by the German Federal Ministry of Economics (BMWi) under the project Theseus (number 01MQ07019). We would furthermore like to extend our gratitude to Florian Berghoff for his contributions to the implementation of the method.

References

1. Correa, E.S., Freitas, A.A., Johnson, C.G.: A New Discrete Particle Swarm Algorithm Applied to Attribute Selection in a Bioinformatics Data Set. In: Proceedings of the 8th Genetic and Evolutionary Computation Conference (GECCO-2006), New York, NY, USA, ACM (2006) 35–42
2. Correa, E.S., Freitas, A.A., Johnson, C.G.: Particle Swarm and Bayesian Networks Applied to Attribute Selection for Protein Functional Classification. In: Proceedings of the 9th Genetic and Evolutionary Computation Conference (GECCO-2007), New York, NY, USA, ACM (2007) 2651–2658
3. Stoilos, G., Stamou, G., Kollias, S.: A String Metric For Ontology Alignment. In: Proceedings of the 4rd International Semantic Web Conference. Volume 3729 of LNCS., Galway, Ireland, Springer (November 2005) 624–637
4. Salton, G., Wong, A., Yang, C.S.: A Vector Space Model for Automatic Indexing. Communications of the ACM **18**(11) (1975) 613–620
5. Ji, Q., Haase, P., Qi, G.: Combination of Similarity Measures in Ontology Matching using the OWA Operator. In: Proceedings of the 12th International Conference on Information Processing and Management of Uncertainty in Knowledge-Base Systems (IPMU'08). (2008)