Chapter 8

Conclusions and future work

'Our imagination is the only limit to what we can hope to have in the future.'

Charles F. Kettering

8.1 Conclusions

In this thesis, we addressed the problem of recognition of structures in images using graph representations and inexact graph matching. One of the main contributions of our work is to express this task as a combinatorial optimization problem with constraints, and to propose methods to solve it based on EDAs and their parallelization.

A discussion on different representations of individuals has been provided. In particular, we proposed representations in both the discrete and continuous domains. Some of the constraints imposed to the matching could be introduced directly in the representations.

Different types of fitness functions have been presented. Our contribution here is twofold. First an experimental comparison of their behavior has been performed, and second new fitness functions based on probability theory have been designed.

The main focus of our thesis was on the optimization itself. A new approach based on estimation of distribution algorithms was introduced for solving the graph matching problem. Its foundations rely on an evolutionary computation paradigm that applies learning and simulation of probabilistic graphical models (i.e. Bayesian networks in the discrete domain and Gaussian networks in the continuous one) as an important part of the search process. Our contribution in this part was to adapt these algorithms to the inexact graph matching problem with constraints, which to our knowledge have never been addressed before. In particular we proposed original solutions to take the constraints into account. This contribution can certainly be exploited in other combinatorial optimization problems with constraints, thereby enlarging the potential application field of EDAs.

Finally another contribution relies in the parallelization of EDAs. Up-to-date parallelization techniques have been applied to these algorithms, resulting in two different programs suitable for execution on multiprocessors with shared memory and cluster of workstations under windows or GNU-Linux systems. The use of shared memory libraries with threads –using *pthreads*– as well as high-level parallelization libraries based on message passing –such as MPI– have been analyzed in detail. The particular case of EBNA$_{BIC}$ has been detailed, and each of its steps has been analyzed in terms of parallelization and computation costs. A parallel version of this algorithm is proposed for the BIC metric. This contribution allows now to use EDAs to solve problems with higher complexity.

From an experimental point of view, our contribution lies in the comparison of the performance of EDAs in both discrete and continuous domains with other evolutionary computation techniques such as genetic algorithms and evolutionary strategies. These experiments were performed for the different types of individual representations, different types of fitness functions, and applied to synthetic and real graph matching problems. Results show that our approach obtains better results and that converge to a solution by having to evaluate less individuals than other more usual evolutionary computation methods such as genetic algorithms. These differences in the results have been proved to be statistically significant after applying non-parametric tests.

8.2 Future work

Many different adaptations, tests, and experiments have been left for the future due to lack of time (i.e. the experiments with real data are usually very time consuming, requiring even days to finish a single run). Future work concerns deeper analysis of particular mechanisms, new proposals to try different methods, or simply curiosity.

There are some ideas that I would have liked to try during the description and the development of the fitness functions in Chapter 3. This thesis has been mainly focused on the use of EDAs for graph matching, and most of the fitness functions used to find the best result where obtained from the literature of adapted from these, leaving the study of fitness functions outside the scope of the thesis. The following ideas could be tested:

1. It could be interesting to consider the regions in the model and data images with different importance, depending on their size or their specific meaning with respect to the recognition process. This mechanism would for instance aid to distinguish in very complex problems which are the regions that are essential to be found, the ones that sometimes appear, and the ones that rarely do.

2. The way the model is constructed could be also changed: instead of using one typical image (prototype), it could be based on different images, in order to provide some information on the variability among the different images, and introduce it in the attributes. Unfortunately, in the type of images that we have taken as real examples the construction of a model from each image is a tedious task and no further study in this direction could be performed.

Obviously, the use of other types of individual representations and fitness functions could be investigated since they have an important influence on the results obtained at the end. New approaches in this direction can be induced from techniques described in the literature such as [Bloch, 1999a,b, Rangarajan et al., 1999a, Sanfeliu and King-Sun, 1983].

The performances of all the fitness functions described in Section 3.4.3 have not been compared on a same problem. The main reason was that some fitness functions are very complex to compute and require a considerable execution time to evaluate each individual. Parallelization techniques have been applied to the learning step in EDAs, but not for the evaluation of individuals, and such a mechanism could help at reducing execution times. Nevertheless, we are already designing and running experiments to compare the performance of our newly proposed probability theory-based fitness function $f_4(h)$ and $f_5(h)$ to such of the fitness functions defined previously in this section. The preliminary results of these experiments do not seem to be satisfactory, and further study is still required in order to understand the behavior of these two fitness functions and improve it.

Concerning the results for both applications (brain and facial features), we can also expect to improve them by having richer graphs, with more attributes.

In the definition of the EDAs in Chapter 4, there are also many ideas that could be exploited to try to obtain a most effective convergence towards the best solution. An example of this is the use of a mechanism that could be understood as a learning depending on the fitness value of the individual: in the learning proposed for EDAs all the selected individuals are used for the learning equally regardless of their fitness value. This means that the fitness value is just considered for selecting the best individuals, but differences between the values among these individuals are not considered in the learning process. A similar idea to this is proposed in the Bit-Based Simulated Crossover algorithm (BSC) [Syswerda, 1993], but this idea could be extended to any EDA. One of the disadvantages that this new type of learning can have is that by accelerating the convergence the search is too focused to the main individuals, and therefore EDAs could lead to local maxima. However, this idea is still a possibility that could be analyzed in the future to check whether local maxima are avoided or not and how to improve it for specific problems such as inexact graph matching.

The initial population in all EDAs has been built using a uniform distribution. Other methods could be also tested, as sometimes a pre-processing step could be added so that the search can also start with some specific individuals. Also, other types of statistical initializations such as greedy probabilistic methods could help at directing the search from the beginning, leading to less evaluations.

Regarding the application of parallelism to EDAs, an extension for the near future is the use of more powerful multicomputers in order to improve the parallelization: the computers we used had at most only 2 processors, and therefore no more than 4 workers were created per computer so that all the workers do not compete for CPU use with the corresponding thrashing problem. An additional task to perform is the parallelization of other algorithms such as EGNA_{ee} and EMNA, which are also susceptible of being parallelized due to the high number of tasks that can be performed in parallel on different processors.