A System for Multi-Agent Belief Revision by Genetic Search

Evelina Lamma
Dipartimento di Ingegneria, Università di Ferrara,
Via Saragat 1, 44100 Ferrara, Italy, elamma@deis.unibo.it

Luís Moniz Pereira
Centro de Inteligência Artificial (CENTRIA),
Faculdade de Ciências e Tecnologia, Universidade Nova de Lisboa,
2829-516 Caparica, Portugal, lmp@di.fct.unl.pt

Fabrizio Riguzzi
Dipartimento di Ingegneria, Università di Ferrara,
Via Saragat 1, 44100 Ferrara, Italy, friguzzi@ing.unife.it

We consider a definition of the belief revision problem that consists in removing a contradiction from an extended logic program [9, 1, 2] by modifying the truth value of a selected set of literals called *revisables*. The program contains as well clauses with false ($\perp$) in the head, representing *integrity constraints*. Any model of the program must ensure that the body of integrity constraints be false for the program to be non-contradictory. Contradiction may also arise in an extended logic program when both a literal L and its opposite $\neg L$ are obtainable in the model of the program. Such a problem has been widely studied in the literature, and various solutions have been proposed [3, 5] that are based on abductive logic proof procedures.

The system performs belief revision in a society of logic-based agents, by means of a (distributed) genetic algorithm. The problem can be modeled by means of a genetic algorithm, by assigning to each revisable of a logic program a gene in a chromosome. In the case of a two-valued revision, the gene will have the value 1 if the corresponding revisable is true and the value 0 if the revisable is false. The fitness function that is used in this case is represented in part by the percentage of integrity constraints that are satisfied by a chromosome.

Each agent keeps a population of chromosomes and finds a solution to the revision problem by means of a genetic algorithm. We consider a formulation of the multi-agent revision problem where each agent has the same set of revisables and the same program, but is subjected to possibly different observations and constraints. Observations and constraints may vary over time, and can differ from agent to agent because agents may explore different regions of the world. Each agent by itself locally performs a genetic search in the space of possible revisions of its knowledge, and exchanges genetic information by crossing its revisable chromosomes with those of other agents. In this way, we achieve distribution in belief revision since chromosomes coming from different agents, through crossover, contribute to solve the problem.

In the genetic algorithm we also exploit computational logic techniques: the algorithm comprises a Lamarckian operator that differs from a Darwinian mutation operator because, instead of randomly modifying the genes, it modifies them in order to improve the fitness of the chromosome. Genes that are modified by this operator are also called "memes" [4]. The Lamarckian operator modifies the memes by means of a (logic-based) procedure inspired by [10]: the logical derivations leading to the inconsistency of belief are traced so as to remove these derivations' support on the meme coded assumptions, effectively by mutating the latter. In our algorithm, therefore, computational logic is used in order to find good revisions that are then distributed by means of the crossover genetic operator. A complete description of the algorithm can be found in [6, 8, 7].

We have performed experiments comparing the evolution in beliefs of a single agent informed of the whole of knowledge, to that of a society of agents, each agent accessing only part of the knowledge. The experiments have been performed on problems of model based diagnosis, a natural domain in which belief revision techniques apply [5], and on the n-queen problem. In spite that the distribution of knowledge increases the difficulty of the problem, experimental results show that the solutions found in the multi-agent case are comparable in terms of accuracy to those obtained in the single agent case. Moreover, we have seen that the adoption of computational logic methods in a genetic algorithm provides an improvement over purely genetic approaches. The results of the experiments can be found in [6, 8, 7].

The system is written in Sicstus Prolog. The system simulates the behaviour of a multi-agent system by keeping a separate population for each agent and by interleaving the application of genetic operators to the population of each agent. In order to return a single solution also for the multi-agent case, at the end of the computation the system considers the solutions found by each agent and computes the fitness for them by considering the integrity constraints in all agents (global fitness). Then, the chromosome with the highest global fitness is returned as the solution for the global belief revision problem.

The system allows the tuning of a number of parameters: the number of individuals in each agent, the maximum number of generations, the percentage of individuals to be selected, the percentage of individuals to be mutated, the percentage of individuals to be crossed over, the percentage of individuals to be mutated Lamarckianly, the type of fitness function to be used (either accuracy or a hybrid function, see [7] for details) and the type of selection operator (it can select chromosomes probabilistically from the population on the basis of the fitness function or select the best n chromosomes).

The Prolog code of the system, the instructions for its use and some examples of input files and can be found at `http://lia.deis.unibo.it/Software/gbr/`.

1 Acknowledgements

L. M. Pereira acknowledges the support of PRAXIS project FLUX "Flexible Logical Updates".

References

[1] J. J. Alferes and L. M. Pereira. *Reasoning with Logic Programming*, volume 1111 of *LNAI*. Springer-Verlag, 1996.

[2] J. J. Alferes, L. M. Pereira, and T. C. Przymusinski. "Classical" negation in non-monotonic reasoning and logic programming. *Journal of Automated Reasoning*, 20:107–142, 1998.

[3] J. J. Alferes, L. M. Pereira, and T. Swift. Well-founded abduction via tabled dual programs. In D. De Schreye, editor, *Procs. of the 16th International Conference on Logic Programming*, pages 426–440, Las Cruces, New Mexico, 1999. MIT Press.

[4] Susan Blackmore. *The Meme Machine*. Oxford U.P., Oxford, UK, 1999.

[5] C. V. Damásio, L. M. Pereira, and M. Schroeder. REVISE: Logic programming and diagnosis. In *Proceedings of Logic-Programming and Non-Monotonic Reasoning, LPNMR'97*, volume 1265 of *LNAI*, Germany, 1997. Springer-Verlag.

[6] E. Lamma, F. Riguzzi, and L. M. Pereira. Logic aided lamarckian learning. In *Proc. 5th Intl. Workshop on Multistrategy Learning*, 2000. Avaliable at `http://www.ing.unife.it/ docenti/ FabrizioRiguzzi/ msl00.ps`.

[7] E. Lamma, F. Riguzzi, and L. M. Pereira. Belief revision via lamarckian evolution. Submitted for publication, available at `http://www.ing.unife.it/ docenti/ FabrizioRiguzzi/ ga2001.ps`, 2001.

[8] Evelina Lamma, Fabrizio Riguzzi, and Luís Moniz Pereira. Belief revision by lamarckian evolution. In *First European Workshop on Evolutionary Learning (EvoLEARN2001)*, LNCS. Springer-Verlag, 2001.

[9] L. M. Pereira and J. J. Alferes. Well founded semantics for logic programs with explicit negation. In *Proceedings of the European Conference on Artificial Intelligenece ECAI92*, pages 102–106. John Wiley and Sons, 1992.

[10] L. M. Pereira, C. V. Damásio, and J. J. Alferes. Diagnosis and debugging as contradiction removal. In L. M. Pereira and A. Nerode, editors, *Proceedings of the 2nd International Workshop on Logic Programming and Non-monotonic Reasoning*, pages 316–330. MIT Press, 1993.