Decision Analysis with Logic and Mathematical Programming

Mauricio Osorio, Alejandro Santoyo

Universidad de las Américas, Sta. Catarina Mártir, Cholula, Puebla,
72820 México
alejandro.santoyorz@udlap.mx

Abstract. Logical and mathematical programming are closely related, however, for some reasons, this has not generated widespread interest in the scientific community to study more deeply this connection, and perhaps one reason is that the scientific community has focused only on the direct benefits of each paradigm. At the same time, logic programming has been used for addressing typical decision analysis's concerns, such as solving decision problems under uncertainty (which are addressed by decision theory), and certain optimization problems (which are addressed by mathematical programming methods). Therefore, this paper proposes to make research on structural similarities between logical and mathematical programming, as well as algorithms, and technological and theoretical foundation that help us to create synergy between them lead us to have a new approach, for solving decision analysis problems.

Keywords: Answer set programming, logic programming, mixed integer programming, decision analysis.

1 Introduction

The title "Decision Analysis with Logical and Mathematical Programming" incorporates two paradigms which are much related. Indeed, relations between logical and mathematical programming have been established and used long time ago. However, they have not created widespread interest in the scientific community, which is very interesting, because even though these paradigms grew with different roots, and addressing different problems, it seems that they might converge on solving Decision Analysis Problems (DAP).

On the other hand, it seems to us that even though logic programming has not been used for addressing all decision analysis's concerns as a whole, there is a significant amount of research on each of them separately [1][2][3], so that it is possible to develop a conceptual framework for logic programming to solve DAP.

Therefore, this project seeks to make a rigorous and detailed analysis of these paradigms to find mechanisms for their incorporation into one, so that it can address DAP by using some form of cooperation between logical and mathematical programming.

With the idea of having a defined framework, we will try to define one set of DAP, which should contain an appropriate mix of decision factors that allows us to create the necessary experiments to find out the required structural similarities between logical and mathematical programming, as well as algorithms, and technological and theoretical foundation which as a whole can define a new approach for solving this kind of problems.

The organization of this document is as follows:

Section 2. Facts and relationships between paradigms: will state the likenesses between logical and mathematical programming, and its convergences which constitute the basic assumption to think that these paradigms could collaborate to create synergy on DAP resolution.

Section 3. New approach: Since decision analysis's concerns have already been addressed by logical programming, and due to the close relationship between logical and mathematical programming, this section will explain the project's purpose and argue the new approach's advantages solving DAP.

Section 4. The research project: this section will provide more insight into the new approach and its involved challenges, in order to have a comprehensible problem statement and objective of the research project.

Section 5. Conclusions: finally this section states some conclusions in order to have a final and better understanding of the whole research project.

2 Facts and Relationships between paradigms

This project was born when it was found that the inequalities of an optimization problem could be expressed in terms of logical propositions, and in fact, it could also be the opposite, i.e. a logical proposition can be expressed in algebraic terms.

On the other hand, it happens that logic programming has addressed particular aspects of decision analysis' concerns [1][2][3], and it seems that logic programming could be the required paradigm for automating the analysis decision process [4].

Therefore, we believe it is necessary to show the series of similarities and facts that were found on the road.

2.1 Logic programming and decision analysis

Globalization has imposed the need for organizations to be efficient in all aspects of their activities, so that decisions they must face imply more number of choices with high economic impact. Therefore, it is required to evaluate each option and make the best decision. As a consequence, it was required to have a formal procedure to address such complex decision problems, and Ronald A. Howard called this procedure "decision analysis" [5]. Such a procedure incorporates uncertainties, values and preferences in a model which should be adequate for computer manipulation.

Thus, our interest is only in DAP with options to choose between, which should involve an optimization problem, the decision analysis procedure, values and uncertainties, its rationality, and the resulting optimal decision. Since all these topics have been addressed by logic programming though in a separately fashion, it was thought it is possible to develop a conceptual framework for logic programming, in particular nonmonotonic logic, to solve DAP.

2.2 Logic programming

Human beings have long dreamed of creating intelligence for solving any kind of problems, but it was until 1956 when John McCarthy coined the term artificial intelligence (AI). Then, AI became a branch of computer science which aims to solve problems related to deduction, reasoning, knowledge representation, planning, learning, natural language, motions and manipulation [6].

Thereafter, Artificial intelligence developed a logic-based approach, when John McCarthy felt that machines did not need to simulate human thought, but should instead try to find the essence of abstract reasoning and problem solving [7], he focused on using formal logic to solve a wide variety of problems, including knowledge representation, planning and learning [8]. Finally all these efforts led to the development of a new paradigm: logic programming [9].

Logic programming has had to overcome a lot of obstacles in order to be able to provide support for knowledge representation, common sense reasoning, abductive and inductive reasoning, decision making under uncertainty, causation and action, planning and problem solving, theory of beliefs, etc., which could mean that logic programming is ready for addressing DAP.

2.3 Logic and mathematical programming

As it was stated, logic programming can deal with deduction, reasoning, knowledge representation, uncertainty, planning problems, etc. by a declarative language that uses logical formulas for creating logic programs, and then the inference engine, by an inference process, deduce (search for) the logical consequences of the program.

On the other hand, no much people know that logic can be used for solving optimization problems modeled by mathematical programming, particularly integer and combinatorial programming (Mixed Integer Programming MIP). However, other branch of artificial intelligence, constrain programming, can deal with other areas of mathematical programming [10], and there is research on matching constrain and logic programming [11].

Since there is a close relationship between logic programming and MIP [12], it could be possible to use logic programming for solving MIP problems and vice versa, but it is not clear to what extent it is possible. However, this possibility can enhance one´s insight into the problems to solve.

Within this context, it turns out that an optimization problem is closely related to an inference problem since there is a strong link between logic and optimization since an optimization process has two stages: To find a feasible solution which is a search problem, and to prove that the feasible solution is optimal, which in turn is a logical inference problem. Additionally, optimization methods are actually specialized logical inference methods. For example, a cutting plane is an inequality that is implied by other inequalities [13].

On the other hand, a logical inference process is an optimization problem [14], since logical inference can be a very hard combinatorial problem, and the amount of computation required grows rapidly with the size of the knowledge base.

Finally, there is a strong relationship between models because they have almost the same structure, use a formal language, and they allow to deduce facts; Insomuch that a mathematical model can be defined as a logical model and vice versa [14]. However, it is required an example in order to explain what we are looking for with all these similarities.

3 New approach

As it was stated, the project aims to use logical and mathematical programming, and particularly nonmonotonic logic and MIP, for solving decision analysis problems in order to find out the required mechanisms that make possible to create that both paradigms can work together. This section explains the whole idea.

3.1 The tourist problem

As stated above, logic and mathematical programming, despite having different roots they seem to converge, and the scenario in which this convergence has taken place is decision analysis, which is the ideal setting to do the research that this paper proposes. The following paragraph describes the problem of *the tourist* which is a simplified version of the type of problems that this new approach would allow to address.

Suppose a tourist arrives in a city which has many tourist attractions, however, he has only one day to know as many places as possible. Additionally, the tourist has a series of preferences on the tourist attractions, and additionally there are circumstances that could affect these preferences.

With regard to the preferences, they could be indicated in any way, for example: the tourist may prefer museums to parks and historic sites, historical places to parks and casinos, and theater to nightclubs, etc. Preferences could also be indicated by saying that the tourist prefers to visit the most diverse places as possible, however, he does not like parks, nightclubs and casinos.

With regard to the circumstances that may affect preferences we can mention the following: The weather can affect preferences associated with open spaces like parks, zoos, outdoor concerts, lakes, etc. And mass demonstrations, traffic accidents and closures of roads that cause traffic chaos which could affect preferences related to nearby tourist sites.

Moreover, as the tourist has only a day he cannot visit all the places he wanted since he will have the limitation of time. In this sense, given the set of all the tourist attractions, the tourist will have to select a subset that will be determined according to his/her preferences and circumstances. Once the subset has been found then the optimal tour must be defined, but if the required time to make the tour is longer than the available time, then the tour should be cut accordingly taking again into account preferences and circumstances.

This problem differs from that proposed in [15] since it does not seek to determine a tourist's vacation destination, but to determine the attractions that a tourist should visit in the city where he is already, considering his preferences and available time, which means that there also exists an associated optimization problem.

3.2 Conventional approach

The first step using a traditional approach is to create some kind of model to elicit and understand the preferences and their relationship to the circumstances. However, as mentioned above, preferences could be provided using very different kind of expressions, which implies a different way to understand and shape them.

Once all the information has been elicited, it could be represented by a decision tree, and it will make possible to determine the set of tourist attractions the tourist would prefer to know. Afterwards, considering the travel times matrix it is possible to determine the shortest travel time route by formulating a mathematical model and by using MIP's methods for finding the optimal result.

If the shortest travel time route is larger than the available time, it will be necessary to cut the tour considering the tourist's preferences for getting a new set of sites to visit. Then the associated mathematical model will be formulated again to find the shortest travel time tour one more time, and this process will be repeated as many times as necessary.

It is important to point out that the whole approach depends on the human manipulation and interpretation of the information, which means that if there is a handling and/or interpretation error, perhaps you might be solving the wrong problem. Moreover, mistakes could come from human errors, but additionally they could come from interpreter's bias.

3.3 Proposed approach

On the other hand, the use of logical and mathematical programming to solve the same problem leads us to a new approach. Again, taking into account the information we can gather from the problem, it is required to represent it in terms of logical formulas, and this is not a human interpretation process, but a knowledge representation process.

This means that the main concern is not to interpret the information, but represent it in logical formulas terms, as well as the relationships between them. Therefore, there is no possibility of human bias, since problem's facts and their relationships are just represented without adding any meaning or interpretation.

Once the knowledge representation process has finished, the inference process can determine the set of sites the tourist would prefer to know as well as a logical model for finding the shortest time travel tour, which even though can be again solved as an inference process, it might be translated into an integer programming model formulation that, very likely, could be solved more efficiently [16].

However, even though it seems very straightforward to use logic programming to solve ADP, actually there are several theoretical and technological factors that make impractical and infeasible to deal with real-world decision analysis problems.

4 The research project

In order to be able to face and solve the tourist problem, it is required to join the knowledge representation and reasoning power of logic programming with the solving capabilities of MIP. Thus, it is expected that this new approach can deal with more complex problems, and even with problems from other mathematical programming subfields, however, there are several challenges to face even considering the simplified tourist problem such as to handle extensive use of numerical computations, and numerical reasoning, etc. [17].

Knowledge Representation is an area concerned with how knowledge can be represented symbolically and manipulated in an automated way by reasoning programs [18], in this case the reasoning program is an inference process. It is supposed then that any kind of facts or rules that currently are used in DAP, can be represented by the KR language in terms of logical formulas, consequently the following question arises: Is it possible to deal with uncertainty, preferences, and probabilities as decision analysis does?

In this sense, it has been defined that this project will use answer set programming [19], which is a declarative programming language and is derived from nonmonotonic reasoning in knowledge representation, and is based on the stable model semantics of logic programming [20], which seems to be the best choice, since it is capable to represent almost any kind of fact: recursive definitions, defaults, causal relations, special forms of self-reference [1], etc.

However, even though that some semantics has been developed to deal with probabilities [1], does it mean that can be used as decision analysis methods do?, does the required extension exist?, are they supported by the same solver?, and what about uncertainty and preferences?

4.1 Inference Process

As mentioned above, when the knowledge representation process has ended, and all is ready for finding the shortest time travel tour by an inference process, it will be so slow that it will be impractical due to several inference process problems related to the inference process itself, such as: grounding is a bottleneck, and there is no connection between problem type and search strategy, etc. [21].

However, and since an inference problem can be seen as an optimization one [14], it could be possible to use MIP techniques, considering that they might improve the process efficiency. Indeed, Hooker and Chandru also state that both models have the same kind of structure, and that there is a trend to merge them, because they complement each other.

In this sense, it seems to be desirable to be able to get advantage from using both models, merge them, complement them or interact between them. However, the following questions arise: To what extent could this be done?, what will it require to pass from one model to another?

Thus, it is necessary to work on ASP solver's issues in order to make possible to use them on real world problems. Such issues could be: languages extensions, ASP integration with other paradigms, and execution models [17], however, at this moment it is difficult to predict where this project could lead, but it is clear that a research project working on these issues will make possible to move between logical and mathematical models and methods so that eventually a new and powerful paradigm can arise.

4.2 Research objective

General Objective: To make a rigorous and detailed analysis on the contact point between the paradigms in order to define the required course of action to achieve a smooth interaction between mathematical and logical models for solving DAP, by investigating the structural similarities, languages extensions, ASP integration with other paradigms and execution models. However, this position paper does not propose to provide an alternative to traditional optimization, inference, and decision analysis methods, but to suggest a broader conceptual framework for interaction.

5 Conclusions

There are closely related relationships between logical and mathematical programming which can lead to an even more powerful paradigm for addressing DAP, therefore it is needed to research the theoretical and technological foundations for delivering their potential in a practical fashion.

The research project propose to use a new approach for addressing DAP, which implies that both paradigms can collaborate each other, and to this end, it is needed to study, analyze and propose the required language extensions, ASP integration with other paradigm, and execution models that such a synergy requires.

6 Bibliography

1. Baral, C., Gelfond, M., Rushton, N., and Lifschitz, V.: Probabilistic reasoning with answer sets. Logic Programming and Nonmonotonic Reasoning, 21-33, Springer-Berlin (2004).
2. Confalonieri, R. and Prade, H.: Answer set programming for computing decisions under uncertainty. In proceedings of the 11^{th} European conference on Symbolic and quantitative Approaches to Reasoning with Uncertainty, 485-496, Springer-Verlag (2011).

3. Costantini, S. and Formisano, A.: Preferences and Priorities in ASP. In: Proceedings of the 9th Italian Convention on Computational Logic, 47-58, ceur-us.org (2012).
4. Doyle, J.: Background to qualitative decision theory. AI magazine, 20, 55-68 (1999).
5. Howard, R.: The foundations of decision analysis. Systems Science and Cybernetics, IEEE Transactions on, *4*(3), 211-219. (1968) doi:10.1109/TSSC.1968.300115
6. Luger, G. F., and Stubblefield, W. A. Artificial Intelligence Structures and Strategies for Complex Problem Solving (Trhird Edi.). MA USA: Addison Wesley Longman, Inc (1998).
7. Russell, Stuart J.; Norvig, P.: Artificial Intelligence: A Modern Approach (Second Edi.). Upper Saddle River, New Jersey: Prentice Hall (2003).
8. McCorduck, P.: Machines Who Think (Second Edi.). Natick, MA: A. K. Peters, Ltd (2004).
9. Colmerauer, A,. and Roussel, P.: The birth of Prolog. History of programming languages---II, ACM, New York (1996).
10. Gong, J., and Ji, J.: Integration of Constraint Programming and Mathematcial Programming. In: Proceedings of the 3rd International Conference on Advanced Computer Theory and Engineering (ICACTE), 6, 451-454 (2010).
11. Mellarkod, V., Gelfond, M., and Zhang, Y.: Integrating answer set programming and constraint logic programming. Annals of Mathematics and Artificial, 53(1-4), 251-287 (2008). doi:10.1007/s10472-009-9116-y
12. Williams, H.: Logic and integer programming. Springer (2008).
13. Hooker, J.: Logic-Based Methods for Optimization: Combining Optimization and Constraint Satisfaction. Canada: John Wiley & Sons (2000).
14. Hooker, J. N., & Chandru, V.: Optimization Methods for Logical Inference. Canada: John Wiley & Sons (1999).
15. Ielpa, S., Iiritano, S., and Leone, N.: An ASP-based system for e-tourism. In: Proceedings of 10th International Conference on Logic Programming and Nonmonotonic Reasoning. 368-381 (2009).
16. Bell, C., Nerode, A., NG, R., T., and Subrahmanian, S.: Mixed Integer Programming Deductive Methods Databases for Computing Nonmonotonic approach. *ACM*, *41*(6), 1178-1215 (1994).
17. Dovier, A.: Present and Future Challenges for ASP Systems. In: Proceedings of 10th International Conference on Logic Programming and Nonmonotonic Reasoning, 622-624 (2009).
18. Brachman, R. J., and Levesque, H. J.: Knowledge Representation and Reasoning. Elsevier Inc. (2004).
19. Lifschitz, V.: What is answer set programming. In: Proceedings of 23th Conference on Artificial Intelligence. 1594-1597 (2008).
20. Gelfond, M., and Lifschitz, V.: The stable model semantics for logic programming. In: Proceedings of the International Conference on Logic programming. 1070-1080 (1988).
21. Delgrande, J.: What Next for ASP?. In: Proceedings of the 10th International Conference on Logic Programming and Nonmonotonic Reasoning, 616-618 (2009).