ASP Challenge Problem:
Insurance Referee Assignment Problem

Satyam Raj

Arizona State University, Tempe, United States

Abstract

The goal of this project is to illustrate the insurance referee assignment issue and attempt to resolve it using knowledge representation and reasoning while maintaining all such limitations in mind.

1.Problem Statement

Our problem is related to the insurance companies which play a crucial role in our life. Whenever an insurance company receives any claims from its customer it checks whether the claim is justified or not. In order to check the eligibility of the claim insurance companies have referees. Referees can be both internal and external. Internal referees have a fixed salary whereas external referees are paid per case and the amount of payments depends on the case. Each referee has a maximum workload which is defined in minutes, beyond which they cannot work. Furthermore, each referee has a preference for a region and a specialized domain. Every insurance case is defined by three factors i.e. the expected effort to handle, the damage amount in euros, and the payment that an external referee would get if it is assigned to them. Our task here is to help the insurance in assigning these cases to the referee considering the constraints defined in the problem [1].

2.Project Background

There are various insurance available today such as life insurance, car insurance, travel insurance, home insurance, and so on. When an individual purchases an insurance plan from an insurance company then it's the responsibility of the insurance company to provide financial protection to its customer. Insurance is a way to manage our risk. In fact, in many countries insurance is required under the law, for example, in the United States of America, it's necessary for you to have car insurance under federal law. Since insurance can accrue cash value or be exchanged into cash, it can be viewed as a financial asset. Considering the importance of insurance in someone's life, it's really important for insurance companies to check the eligibility of an insurance claim properly. Our problem revolve around this only. We have to help insurance companies so that they can assign the claim cases to referees who can verify whether it is justified or not.

In order to solve this problem I have developed an automated solution that would help us to solve our problem of insurance referral with virtually no error. I have chosen ASP to solve it. ASP stands for Answer Set Programming and is a kind of declarative programming targeted toward challenging search problems, notably NP-hard ones [3]. Answer set solvers, which are programs for creating stable models, are used to conduct searches in ASP since search issues are simplified to calculating stable models. Many answer set solvers employ search algorithms that are improved versions of the Davis-Putnam-Logemann-Loveland method [2]. A set of atoms may be viewed from the perspective of knowledge representation as a description of a whole state of knowledge: the atoms that are a part of the set are known to be true, and the atoms that are not a part of the set are known to be false. A consistent but potentially incomplete collection of literals can be used to express a state of knowledge. I decided to solve this problem using ASP as it is reliable, there is transparency and it provides a short solution. ASP presents a special chance to digitize and save existing knowledge. Additionally, we are clingo as it is very easy to establish the actual time restriction in our scenario using Clingo.

In this project, we have the instance where we are provided with the set of cases that has to be handled, a set of regions, a set of domains, a threshold for the external referee, and the set of the available referee.

3.Problem Solving Approach

In order to solve this problem I had to develop a step-by-step approach to the problem. In addition to that, I also had to consider the hard and weak constraints defined in the problem. I always had to ensure that the sum of all the efforts of the cases which were handled by the referee should not exceed the maximum number of minutes that referee is supposed to work. Furthermore, another constraint I had to consider was that, if a referee has a preference of zero for a region then I made sure that the referee should not get the cases from that area. Similarly, if the referee has zero preference for a specific domain then I wrote the logic in a way such that they won't get cases related to that domain. For example, if a referee has a 0 preference for the cases related to cars then they won't get cases related to cars. As mentioned earlier, we have both internal and external referees in our case but if the amount of damage exceeds a certain threshold then it can be only assigned to the internal referee. These were the hard constraints that I had to keep in mind while writing the logic.

In addition to the hard constraints, I also had to deal with the weak constraints. One of the difficult constraints to handle is that, that the internal referee has to be given preference so that we can minimize the cost but at the same time I also had to make sure that the overall payment to both the external and internal referee should be balanced. Similarly, I made sure that the overall workload for both the external and internal referees is balanced. I have written the logic in such a way that referees should get the type of case for which they have a high preference. In a similar manner, my logic ensures that the referee will get the cases from the postal code for which they have a high preference.

The referee in the problem is given as the facts of the form where I had multiple pieces of information. I got to know about the unique identifier of the referee, their previous payment which is zero for the internal referee, and the sum of all the payments an external referee got. The threshold is also defined for the referee beyond which they can't work. This threshold is defined in minutes. Furthermore, we have information about the sum of all the previous efforts they made. We also have an identifier that will help us to know whether the referee is an internal one or the external one.

Just like the referee, cases are also defined as the facts of forms. Each case has a unique identifier and type which tells us about the domain of the cases. There can be various domains such as passenger car, truck, life, etc. Additionally, each case has an estimated duration in minutes and the amount of the damage given in euros. We can also get to know about the postal code to which the case belongs and the amount of money we need to give if the case is handled by an external referee.

Using "externalMaxDamage" we define the threshold beyond which the case will be handled by the internal referee. It is given by a single fact of form. Preference for the cases and regions is also represented by the facts of form. In both of this preference, we have a unique identifier for the referee and the preference score which is in the range of 0 to 3 where 0 means that they have no preference for the same, and 3 means that they have a high preference for the case and the region. We also have the postal code and the case type.

In the optimization criteria, I have considered factors to implement a formula that will help me to minimize the overall cost. The first factor used in the formula is the sum of all the payments made to an external referee and my aim was to reduce this sum. I also kept in mind that the case which will be assigned should be balanced both in terms of duration and amount. I have represented it as "C.A".

Secondly I have taken "O" as the sum of all the payments that has been made to the external referee and "avg" as the average total compensation paid to all outside referees. The deviation from the average for each referee is penalized with fees in order to maintain the overall compensation to all external referees. I have taken the "C.B" as the sum of this cost.

I have used a similar concept in order to balance the workload of a referee. It is one of weak constraints of the problem that workload should be balanced whether it's an external or internal referee. I have taken "W" as the sum of all the workloads and "avgW" as the average of the workload over all the referees. The absolute value of the difference between avgW and W is the cost that will be penalized for the divergence from the average. The sum of this cost is represented as "C.C".

As we have defined earlier that each referee has a preference for certain types of the domain which is in the range of 0 to 3. If a referee has a zero preference for certain types of cases they won't be assigned to cases of that type. For the non zero value of preference, we have a cost which is the difference between 3 and the preference. We will represent that cost as "C.D". The "C.D" is the sum of these cost.

In a similar way we have preferences for certain types of postal code. These preferences are in the region of 0 to 3. If a referee has zero preference for certain postal codes then they won't be assigned cases related to that postal codes. For the non-zero value of preference, we have a cost which is the difference between 3 and the preference. We will represent that cost as "C.E". The "C.E" is the sum of this cost.

Based on the above sum, I used a formula that would help me to minimize the overall cost. The equation used is given below :

$$c = 16. (C.A) + 7.(C.B) + 9.(C.C) + 34. (C.D) + 34. (C.E)$$

The "c" is the overall minimized cost and the magnitude of the constant in the above formula depicts their importance.

4.Analysis and Result

My aim here was to assign the case to the referee considering various constraints in mind. I have tested my logic against various cases provided and it provided the correct solution every time. The log shows that the code executes flawlessly for several instances. Nine instances have been run to evaluate various scenarios. With optimizations, the tests are successfully running. Please refer to the screenshot for the results I have gotten for the various instances.

```
Answer: 1
assign(4, 5)
case(4, c, 90, 3000, 2000, 90) referee(4, i, 480, 220, 0) referee(5, i, 360, 140, 0) referee(6, e, 480, 40, 700) prefType(4, c, 0)
prefType(5, c, 2) prefType(6, c, 3) prefRegion(4,2000,3) prefRegion(5,2000,2) prefRegion(6,2000,2) externalMaxDamage(1500)
Optimization: 185
OPTIMUM FOUND

Models       : 1
  Optimum    : yes
Optimization : 185
Calls        : 1
Time         : 0.011s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.011s
```

```
Answer: 1
assign(5, 7)
case(5, a, 45, 700, 1000, 60) referee(7, i, 480, 220, 0) referee(8, e, 240, 0, 0) referee(9, e, 480, 220, 4000) prefType(7, a, 1)
prefType(8, a, 3) prefType(9, a, 3) prefRegion(7,1000,3) prefRegion(8,1000,0) prefRegion(9,1000,0) externalMaxDamage(1500)
Optimization: 187
OPTIMUM FOUND

Models       : 1
  Optimum    : yes
Optimization : 187
Calls        : 1
Time         : 0.010s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.010s
```

```
Answer: 1
assign(6, 11)
case(6, b, 200, 2500, 1000, 80) referee(10, e, 120, 140, 2800) referee(11, e, 480, 10, 300) referee(12, e, 480, 140, 2800)
prefType(10, b, 3) prefType(11, b, 2) prefType(12, b, 2) prefRegion(10,1000,3) prefRegion(11,1000,2) prefRegion(12,1000,1)
externalMaxDamage(3000)
Optimization: 187
OPTIMUM FOUND

Models       : 1
  Optimum    : yes
Optimization : 187
Calls        : 1
Time         : 0.012s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.012s
```

```
Answer: 1
assign(7, 14)
case(7, b, 250, 2500, 4000, 160) referee(13, i, 480, 6000, 0) referee(14, i, 480, 450, 0) referee(15, e, 480, 500, 270)
prefType(13, b, 3) prefType(14, b, 3) prefType(15, b, 3) prefRegion(13,4000,2) prefRegion(14,4000,2) prefRegion(15,4000,3)
externalMaxDamage(1500)
Optimization: 185
OPTIMUM FOUND

Models       : 1
  Optimum    : yes
Optimization : 185
Calls        : 1
Time         : 0.011s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.011s
```

```
Answer: 1
assign(8, 17)
case(8, a, 480, 2500, 4000, 240) referee(16, i, 480, 6000, 0) referee(17, e, 480, 6000, 4000) referee(18, e, 480, 6000, 4000)
prefType(16, a, 0) prefType(17, a, 3) prefType(18, a, 3) prefRegion(16,4000,2) prefRegion(17,4000,3) prefRegion(18,4000,2)
externalMaxDamage(2500)
Optimization: 185
OPTIMUM FOUND

Models       : 1
  Optimum    : yes
Optimization : 185
Calls        : 1
Time         : 0.013s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.013s
```

```
Answer: 3
assign(8, 19)
assign(9, 19)
assign(10, 21)
case(8, a, 90, 5000, 3033, 65) case(9, a, 240, 5000, 3033, 160) case(10, a, 40, 5000, 3033, 25) referee(19, i, 360, 2000, 0)
referee(20, i, 600, 6000, 0) referee(21, e, 480, 2000, 1200) referee(22, e, 480, 6000, 4000) prefType(19, a, 3) prefType(20, a, 1)
prefType(21, a, 2) prefType(22, a, 3) prefRegion(19, 3033,3) prefRegion(20, 3033,1) prefRegion(21, 3033,3) prefRegion(22, 3033,3)
externalMaxDamage(10000)
Optimization: 187
OPTIMUM FOUND

Models       : 1+
  Optimum    : yes
Optimization : 187
Calls        : 1
Time         : 0.015s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.015s
```

```
Answer: 2
assign(14, 27)
assign(15, 27)
case(14, a, 180, 2000, 5026, 100) case(15, c, 180, 2000, 5026, 100) referee(27, e, 480, 1400, 770) referee(28, e, 480, 2800, 1500)
referee(29, e, 480, 20400, 2000) prefType(27, a, 3) prefType(27, c, 1) prefType(28, a, 2) prefType(28, c, 3) prefType(29, a, 3)
prefType(29, c, 3) prefRegion(27, 5026, 3) prefRegion(28, 5026, 1) prefRegion(29, 5026, 3) externalMaxDamage(3000)
Optimization: 185
OPTIMUM FOUND

Models       : 1+
  Optimum    : yes
Optimization : 185
Calls        : 1
Time         : 0.013s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.013s
```

```
Answer: 2
assign(16, 32)
assign(17, 32)
case(16, b, 180, 2000, 7013, 100) case(17, b, 180, 2000, 7013, 100) referee(30, i, 480, 20000, 0) referee(
referee(32, i, 480, 1000, 0) prefType(30, b, 3) prefType(32, b, 2) prefRegion(30, 7013, 2) prefRegion(31,
externalMaxDamage(10000)
Optimization: 185
OPTIMUM FOUND

Models       : 1+
  Optimum    : yes
Optimization : 185
Calls        : 1
Time         : 0.013s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time     : 0.013s
```

It is clear that the program executes flawlessly and creates ideal models in every situation.

5. Conclusion

The Automatic Warehouse Scenario project is an excellent illustration of how to put the knowledge you learned in the Knowledge Representation and Reasoning course to use. It has also shown to be a fantastic representation of the difficulties that huge businesses and organizations actually face and how they employ KRR to solve them. I had the chance to learn how to overview a project's needs, divide challenges into smaller ones, and then turn those smaller difficulties into usable code while working on this project. Working on this project gave me the chance to develop my ability to analyze a project's requirements, break down problems into smaller ones, and then transform those smaller problems into useful code.

6. Future Work

There is a vast opportunity for KRR in today's industry and one such use is for the insurance companies in assigning cases to the referees. We have solved this problem using Clingo and Answer Set Programming. As the businesses expand the complexity of the problem will increase because of which we will have to improve the constraint checking and improve the overall performance of the program. It will also provide the opportunity to improve the time complexity of the program. KRR has vast opportunities in many fields such as Insurance, Artificial Intelligence, Robotics, and Bioinformatics [4].

7. References

1. Satyam Raj , Course Project – Milestone 3: Individual Progress Report

2. Peter Baumgartner, 2012, A First-Order Davis-Putnam-Logemann-Loveland Procedure. http://www.cs.toronto.edu/~sheila/2542/w06/readings/stavros_slides_fdpll.pdf

3. Vladimir Lifschitz, 2018, What Is Answer Set Programming?, https://www.cs.utexas.edu/users/vl/papers/wiasp.pdf

4. Charalampos Nikolaou, Egor V. Kostylev, George Konstantinidis, Mark Kaminski, Bernardo Cuenca Grau, and Ian Horrocks. Foundations of Ontology-Based Data Access under Bag Semantics. Artificial Intelligence, 2019. https://linkinghub.elsevier.com/retrieve/pii/S0004370219300426