ASP Challenge Problem:
Insurance Referee Assignment Problem

Dhanraj Bhosale
Arizona State University

Abstract

Through knowledge representation and reasoning, this research tries to illustrate the difficulty of assigning insurance referees and explore various solutions while taking into consideration all current constraints.

1. Problem Statement

This issue concerns insurance firms and how they manage client claims. Insurance firms have referees, who may be internal or external, who evaluate claims to verify their validity. External referees are paid on a case-by-case basis, with the payment amount varying based on the case, whereas internal referees are given a predetermined wage. Each official has a maximum workload, a preferred area, and a preferred specialty do-main. Each insurance claim is determined by three elements: the anticipated work needed to resolve it, the damage in euros, and the compensation an outside referee would get if they were given the case. The objective is to assign cases to referees while adhering to these restrictions in the problem [1].

2. Project Background

A key instrument for risk management and self-protection against financial loss is insurance.

When a person purchases an insurance plan, it becomes the responsibility of the insurance company to provide financial protection to its customer. In case of an unfortunate event, the customer can file an insurance claim to receive financial compensation from the insurance company. However, it is essential for insurance companies to verify the eligibility of an insurance claim properly. However, it is essential for insurance companies to verify the eligibility of an insurance claim properly.

To facilitate Insurance firms frequently allocate the claim cases to referees in order to speed up the claim verification procedure. Referees are in charge of evaluating each claim to see if it satisfies the insurance company's requirements for payment. Therefore, selecting the appropriate referrals for the task is crucial for insurance businesses. With our help, insurance firms will be able to assign claim cases to referees who are most qualified for the position.

I have created an automated, practically error-free solution to our insurance referral issue. Answer Set Programming (ASP), a subset of declarative programming that focuses on resolving complicated search problems, especially those that are NP-hard [3], is the foundation of the solution. Since the search issues are simplified to calculating stable models, answer set solvers that produce stable models are employed to carry out searches in ASP. Many answer set solvers make use of sophisticated search algorithms that improve the Davis-Putnam-Logemann-Loveland technique [2]. An entire state of knowledge is described by a set of atoms from the viewpoint of knowledge representation. Any atom that is a part of the set is taken to be true, while those that aren't are taken to be false. A collection of literals can be used to consistently, but perhaps not entirely, convey a state of knowledge. I chose Answer Set Programming (ASP), a dependable and open method that might offer a succinct answer, to deal with this problem. The potential to digitize and preserve existing information is another exceptional benefit of ASP. Clingo, which makes it simple to create time limits precisely, was our choice for our case.

A set of instances that need our attention, a collection of regions, a choice of domains, a predefined upper limit for the external referee, and a list of qualified referees have all been provided.

3. Problem Solving Approach

I used a methodical strategy that includes segmenting the challenge into doable parts in order to successfully solve it.

I made sure to keep in mind the problem's severe limitations and requirements as I went forward. In particular, I made sure that the burden of the referee did not exceed the maximum time permitted by making sure that the overall effort of the cases given to them did not go over this limit. The referees' expressed preferences for particular regions were also taken into consideration, and if their preference value was zero, I made sure they were not given cases from such locations. I developed the algorithm to steer clear of assigning referees cases relating to a particular field if they have no affinity for it. For instance, a referee won't hear instances involving automobiles if they are not their favorite. Both internal and external referees can manage a case in our system, but if the harm surpasses a certain level, only internal referees may. These were the precise guidelines I had to take into account when developing the reasoning.

In addition to working with stringent limits, I also had to cope with softer constraints. Giving the internal referee priority in order to save expenses and maintain a balance in the overall payment to both internal and external referees was one such restriction. In a similar vein, I tried to keep the workload for both categories of referees balanced. To do this, I used reasoning that distributes cases to referees based on their preferences and areas of competence, as well as taking into account their preferred postal codes. This method optimizes employee pleasure and guarantees that they are given instances that are compatible with their skills and areas of interest.

The referee's unique identification, prior payment (which is zero for internal referees), and the total amount of compensation earned by external referees are all included in the problem. There is a time restriction over which the referee is not permitted to work. Additionally, data on the total of their prior efforts is supplied, along with a code to distinguish between internal and external referees.

The distinct identities and kinds of each case—such as passenger vehicle, truck, or life—indicate the domain to which it belongs and are used to group cases. Each scenario has an estimated time frame in minutes and a damage calculation in euros. Along with the cost that must be paid for an outside referee to oversee the case, information regarding the case's postal code is also accessible. In conclusion, cases are categorized according to their identities and kinds, and they can also be categorised according to their duration, damage, postal code, and referee costs.

When determining when a case should be handled by an internal referee, we utilize the phrase "externalMaxDamage". One fact establishes this restriction. In addition, we describe preferences for cases and areas using facts that comprise the referee's unique identify and a preference score between 0 and 3. The difference between a score of 0 and a score of 3 denotes the strength of the liking for the case and area. In addition to the postal code and case type, these choices contain them.

I considered a number of aspects in order to create a formula that would help save expenditures. The main aspect taken into account was the sum of fees made to an outside referee, which I tried to lower. Additionally, I made an effort to maintain a balance between the length and weight of each case assignment, which is indicated in the formula by the letter "C.A."

I assumed that the variable "O" stands for the total sum of payments made to the external referee and that the variable "avg" stands for the average payment made to all external referees. To preserve the overall pay provided to all external referees, any referee's divergence from the average salary is penalized with fees. The letter "C.B" stands for the total amount of these fees.

I added a mild restriction to my strategy to make sure that referees' workloads were distributed fairly. Referees who are internal or external must adhere to this restriction. I totaled up each workload individually to come up with the average workload, which is represented by the letter "avgW," and then calculated the cost of deviation from the average workload. The penalty is shown as "C.C." and is calculated as the difference between the average workload (avgW) and the total workload (W). C.C. essentially stands for the entire expense incurred as a result of the variance from the average workload.

In our previous definition, we demonstrated that referees have a preference between 0 and 3 for certain case types inside the domain. Referees will not be allocated to cases of a certain category if they have a preference of 0 for that type. However, for non-zero preferences, we may calculate a cost that is equal to the discrepancy between the referee's preference and the maximum preference value of 3. "C.D." will be used to indicate this expense. The total of these charges is used to determine the "C.D".

From 0 to 3, we have particular postal code preferences. Referees will not be assigned cases involving certain postal codes if their preference is 0. We determine the cost as the difference between 3 and a referee's preference if they have one and it is more than zero. The total of all such expenditures will be shown by the letter "C.E." for this cost.

I used a calculation to cut the overall cost. The formula is written as follows:

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

The size of the constant in the formula above indicates how important they are. The "c" stands for the total minimized cost.

4. Analysis and Result

I've successfully put my referee selection logic, which takes into consideration a number of restrictions, to the test. I have confirmed that the code operates flawlessly and consistently returns the right answer by executing a number of situations. Nine cases in all were analyzed to evaluate various scenarios, and with improvements, the tests were successful. For the outcomes of these situations, please see the screenshot.

```
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
```

Without any doubt, the program operates seamlessly and produces optimal models under all circumstances.

5. Conclusion

One excellent example of how the Knowledge Representation and Reasoning course may be used in real-world situations is the Automatic Warehouse Scenario project. It illustrates the difficulties big enterprises and organizations encounter and how they use KRR to deal with them. I learned a lot about establishing project needs, breaking down difficult challenges into smaller, more manageable problems, and turning those smaller problems into useful code as a result of my involvement in this project. My abilities in project analysis, issue decomposition, and coding have improved as a result of this project.

6. Future Work

The insurance business now has a method for allocating cases to referees thanks to Answer Set Programming and Clingo. The complexity of the problem will, however, rise as organizations grow, necessitating the improvement of constraint checking and overall program performance. Furthermore, the program's temporal complexity has room for improvement. Significant prospects are presented by KRR in a number of industries, including robotics, bioinformatics, insurance, and artificial intelligence [4].

7. References

1. Dhanraj, 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