# Optimizing Referee Assignment in Vehicle Insurance Companies using Answer Set Programming

**Souradip Nath**
Arizona State University
699 S Mill Ave
Tempe, AZ 85281
snath8@asu.edu

### Abstract

The Referee Assignment Problem in vehicle insurance companies involves assigning a suitable referee to a disputed claim. This problem is a blend of a decision and an optimization problem, where the company needs to consider various factors while making the assignment. In this work, an approach using Answer Set Programming (ASP) has been proposed to solve this problem and it has been demonstrated that the approach generates efficient and effective solutions while satisfying various constraints and optimizing specific objectives.

## Problem Statement

In the context of vehicle insurance companies, the *Referee Assignment Problem* involves assigning a suitable referee to a disputed claim. Such a problem arises when a claimant disagrees with an insurer's decision regarding the settlement of their claim, and the two parties cannot resolve the dispute directly. In such cases, a referee is appointed to review the case and make a final decision.

The referee assignment problem can be challenging for vehicle insurance companies, as they need to consider various factors while making the assignment. For instance, the company needs to ensure that the referee is qualified to review the specific type of claim, has no conflicts of interest, and is available to review the case within the required time frame. Moreover, the company needs to optimize certain factors associated with the assignment, such as the overall cost of the assignment, the balanced distribution of workload and payment, and so on. Hence, this problem is a perfect blend of a decision problem and an optimization problem. A brief discussion of the problem specification is described next.

***Referee***. The problem involves assigning two different kinds of referees to the disputed claims: *Internal* and *External*. The internal referees are the salaried employees of the insurance company, while the external referees are independent third parties who are compensated on a per-case basis. Each referee is characterized by their *Maximum Workload* in minutes per day, *Work Region Preference*, and *Domain Preference* in a four-point preference scale (0-3). Assigning internal referees could be more cost-effective for the insurance company, as they do not incur additional expenses. However, external referees could be more impartial and have more specialized expertise, depending on the type of claim.

***Case***. Each disputed case is defined by its *Domain*, *Required Effort* in minutes, *Geographic Location* identified by postal codes, *Claim Amount* in euros, and an optional *Payment* in euros that specifies how much an external referee will be compensated for handling the case.

***Constraints***. A correct solution to the referee assignment problem satisfies several *hard constraints* which are some ground conditions the solution should never violate, while an effective solution addresses several optimization factors specified as *weak constraints* such as cost minimization, workload distribution, and preference matching.

The ultimate objective is to develop an efficient and effective algorithm for assigning referees to claims while satisfying various constraints and optimizing specific objectives.

## Background Work

An efficient and effective solution to the referee assignment problem can be obtained using various approaches, including manual assignment, rule-based assignment, and optimization-based assignment.

*Answer Set Programming(ASP)* is an optimization-based approach that can be used to automate the referee assignment process. ASP is a form of declarative programming that allows for concise and modular encoding of complex problems. In ASP, a problem is formulated as a logic program consisting of a set of rules that define the problem domain and constraints on the solution, along with a set of assumptions about the initial state of the problem. The solver then searches for answer sets that satisfy the program, using algorithms such as conflict-driven clause learning (CDCL) and backtracking. *Clingo*, a high-performance implementation of ASP that provides support for advanced solving techniques has been explored and discussed in this work.

An extensive exploration of Clingo and the ASP paradigm has been conducted to gain an understanding of the internal workings of ASP, including the concepts of *Choice Rules and Constraints, Negation as Failure, Stable Models, Grounding, Aggregates, and Optimization* as part of the background work. Based on this comprehensive understanding and background work, Clingo is effectively utilized to find an optimal solution to the referee assignment problem.

## Main Approach

The Referee Assignment Problem is a complex problem that requires a systematic approach to solve it. In this work, ASP has been employed as the primary tool to solve the problem, with Clingo being used as a programming language to implement and test the solution.

The solution approach can be divided into three distinct phases. The first phase involves a comprehensive understanding of the problem statement and the requirements for the solution. In the second phase, the Referee Assignment World is modeled using the Clingo programming language by selecting the relevant predicates that are necessary to represent the world accurately. Finally, in the third phase, the solution program is written in Clingo. Each phase is described in detail next.

### Understanding the Problem Statement

A clear understanding of the problem can help realize that the problem is a simple mapping problem; exactly one referee is to be mapped to each case, but multiple cases can be mapped to each referee.

The hard constraints impose certain strict restrictions on this mapping. But, they are intuitive and correspond to real-life scenarios. A brief understanding of the hard constraints is itemized below.

1. The referee's workload must not exceed their maximum working minutes, which is the sum of the efforts of all cases assigned to them.
2. Cases must not be assigned to referees who have zero preference for the region in which the case occurred.
3. Cases must not be assigned to referees who have zero expertise in the specific domain of the case.
4. Cases with claim amounts higher than a certain threshold must only be assigned to internal referees.

The assignment satisfying the hard constraints is further optimized by the following weak constraints which are consistent with real-world scenarios.

1. More cases should be assigned to internal referees to minimize the overall cost.
2. The assignment of cases to external referees should be fairly distributed to ensure that their overall payment is balanced.
3. The assignment of cases to all referees should be fairly distributed to ensure that their overall workload is balanced.
4. Referees should handle more cases with types they prefer more.
5. Referees should handle more cases in regions they prefer more.

It is worth noting that the weak constraints in the Referee Assignment Problem have varying degrees of relative importance. Among them, constraints 4 and 5 hold the highest level of relative importance as they deal with *Preference Matching* which is crucial for achieving desirable outcomes. Constraint 1, which aims to minimize the overall cost of the vehicle insurance company, is also considerably significant. Meanwhile, constraints 2 and 3, which serve as optimization criteria, are comparatively less important but cannot be overlooked.

Understanding the problem statement helps in identifying the critical factors that are needed to be considered and forms the basis of the next phases of modeling the solution.

### Modeling the Referee Assignment World

During the second phase, the Referee Assignment World is represented in Clingo by defining predicates that accurately and consistently describe the key components of the problem. This step involves careful consideration of the problem statement and ensuring that the naming conventions used for the predicates are intuitive and meaningful. The predicates defined in this phase encapsulate the essential aspects of the Referee Assignment Problem and roughly belong to either of these three classes: *Predicates defined as Facts*, *Auxiliary Predicates*, and *Solution Predicate*.

- The *Predicates defined as Facts* include the ones explicitly mentioned in the problem statement, which are given as inputs to the program, such as *case, referee, prefType, prefRegion*, etc.
- The *Auxiliary Predicates* are additional predicates that provide more implicit information about the problem world, for instance, the *Payment, Workload, Numbers of Internal and External Referees, etc*. These are created to facilitate solving the problem and are discussed in detail in the implementation phase.
- Finally, the *Solution Predicate (Assign)* is a binary predicate that describes the expected output of the program, i.e. the optimal assignment of referees to cases, *Assign(case id, referee id)*.

The sole purpose of carefully defining and structuring these predicates is to ensure that the problem can be effectively represented in a way that is both human-readable and machine-interpretable.

### Implementation

The final phase involves implementing the solution to the problem using Clingo, which can be divided into three main sections: *Generate, Define*, and *Test*. To restrict the output to the set of solution atoms and remove auxiliary predicates, an additional *Display* section is included.

**Generate.** The *Generate* section is responsible for generating potential solution candidates using a constrained choice rule, which ensures that each case is assigned to only one referee.

```
{assign(CID, RID): referee(RID, _, _, _, _)
    }=1 :- case(CID, _, _, _, _, _).
```

This rule forms the basis of the solutions and generates the solution domain.

**Define.** The *Define* section includes rules that define auxiliary predicates, which provide more implicit information about a solution candidate. These properties are crucial for obtaining the solution and are investigated in the Test section. The Clingo aggregate `#sum`, and the built-in arithmetic functions, − (subtraction), / (integer division), and $|\,.\,|$ (absolute value) have been used to define the following auxiliary predicates.

- *totalExReferee*: counts the number of external referees
- *totalReferee*: counts the total number of referees
- *newWorkLoad*: sums the efforts of all newly assigned cases for *each* referee
- *overallWorkLoad*: sums the previous workload and the new workload for *each* referee
- *avgOverallWorkLoad*: calculates the average of all overall workloads, truncated to an integer
- *divergenceOverallWorkLoad*: calculates the divergence between individual overall workloads
- *newExPay*: sums the payments of all newly assigned cases for *each* external referee
- *totalNewExPay*: sums all payments of all newly assigned cases for *all* external referees
- *overallExPay*: sums the previous and new payments of *each* external referee
- *avgOverallExPay*: calculates the average of all overall external payments, truncated to an integer
- *divergenceOverallExPay*: calculates the divergence between individual overall external payments
- *caseTypeMismatchFactor*: sums all `(3-pref)` for all assignments, where pref is the preference of the referee for the case type
- *caseRegionMismatchFactor*: sums all `(3-pref)` for all assignments, where pref is the preference of the referee for the case region.

**Test.** Finally, the *Test* section consists of the *Hard constraints* that eliminate unintended solution candidates and the *Weak constraints* that prioritize certain solutions based on their optimality.

The ***hard constraints*** have a specific structure where each rule only contains a body, and specifies the conditions for solution candidates that violate the hard constraints. For example, the hard constraint 1 is encoded in the following way:

```
:- assign(CID, RID), referee(RID, _, MWL, _,
     _), case(CID, _, _, _, _, _),
    newWorkLoad(NWL, RID), NWL > MWL.
```

This rule instructs the ASP solver to eliminate all those solution *assign* candidates for which the corresponding *referee*'s *newWorkLoad* exceeds their *max_workload*. The rest of the hard constraints are defined in a similar way.

The ***weak constraints***, in this work, have been addressed through an *optimization* approach. The idea is to quantify the underlying implication of each weak constraint to a cost value (represented by the defined auxiliary predicates). Then, an objective function is constructed by taking a positive weighted sum of all these costs, and finally, the weighted sum is minimized to address the underlying optimization criterion represented by each weak constraint.

The following is the intuitive mapping between the weak constraints discussed earlier and the auxiliary predicates.

1. Minimize *totalNewExPay* to minimize overall cost
2. Minimize *divergenceOverallExPay* to fairly distribute payments to external referees
3. Minimize *divergenceOverallWorkLoad* to fairly distribute overall workloads
4. Minimize *caseTypeMismatchFactor* to maximize case type preference matching
5. Minimize *caseRegionMismatchFactor* to maximize case region preference matching

Next, the optimization function is calculated by taking a weighted sum of all these individual costs. The weights are calculated based on the relative importance of the weak constraints discussed earlier.

```
f = (
   16 * totalNewExPay +
    7 * divergenceOverallExPay +
    9 * divergenceOverallWorkLoad +
   34 * caseTypeMismatchFactor +
   34 * caseRegionMismatchFactor
)
```

Lastly, this weighted sum `f` is minimized using the Clingo aggregate `#minimize`.

**Display.** Finally, the *Display* section indicates that only atoms over the binary predicate *assign* ought to be printed, thus suppressing the predicates used to describe an instance as well as the auxiliary predicates.

## Results and Analysis

To understand and validate the correctness of the solution, the Clingo program was thoroughly tested against various sample instances with varying complexity levels, and for each test case, the result was validated against an expected solution, provided by a human expert. In all cases, the program is able to come up with not only a correct but also an optimal solution. In this section, one of the sample instances and the corresponding output of the program are described in detail.

One of the sample instances is depicted in Figure 1, where there is only one case with a case id *8*, type *a*, a required effort of *480 minutes*, a damage claim of *2500 euros*, a zip code *4000*, and a payment amount of *240 euros* if the case is assigned to an external referee.

```
pwn_dp@sefcom-OptiPlex-7050:~/Desktop/CSE 579$ more simpleInstances/instance5.asp
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).

% expected result (optimum): assign(8, 17)
```

Figure 1: A Sample Instance of the Referee Assignment Problem

Moreover, there are three referees with ids *16, 17, and 18*, respectively, with all of them having a maximum workload of *480 minutes* and a previous workload of *6000 minutes*. Referee 16 is an *internal* referee, and both Referee 17 and 18 are *external* referees with a previous payment of *4000 euros* each. Referee 16 has *zero preference* towards case type *a*, while both the external referees have the highest *preference 3* for the same. Referee 17 prefers cases in the region with a zip code of 4000 more than the rest of the referees. Lastly, damage claims of up to *2500 euros* can be assigned to external referees.

As per the proposed solution, the problem will be tackled by following these steps:

– The potential solution candidates are *Assign*(8, 16), *Assign*(8, 17), and *Assign*(8, 18).

First, the hard constraints are applied to eliminate any unintended solution candidate.

– For neither of the referees does the workload exceed their maximum working minutes, hence they all can possibly be assigned case 8.

– Referee 16 has a zero preference for the case type *a* which makes *Assign*(8, 16) is an invalid solution. Hence, it is dropped.

– Neither referee 17 nor 18 has a zero region/type preference for the case, hence both of them can possibly be assigned the case.

– The claim damage amount doesn't exceed the specified threshold of 2500 euros, hence both the external referees are still eligible for the case.

Until this point, both *Assign*(8, 17) and *Assign*(8, 18) are correct solutions to the problem instance, but nothing could be commented about their optimality.

Intuitively, referee 17 has a higher preference for zip code 4000 compared to referee 18, hence an optimal solution should be *Assign*(8, 17). This can be further validated by the logical reasoning encoded in the proposed approach. As described in the previous section, the objective function values for the correct solutions *Assign*(8, 17) and *Assign*(8, 18) are calculated to be *11,280* and *11,314*, respectively. And, since the ultimate objective is to minimize the objective function, *Assign*(8, 17) is an optimal solution.

```
pwn_dp@sefcom-OptiPlex-7050:~/Desktop/CSE 579$ clingo solution.asp simpleInstances/instance5.asp
clingo version 5.4.1
Reading from solution.asp ...
Solving...
Answer: 1
assign(8,17)
Optimization: 11280
OPTIMUM FOUND

Models       : 1
  Optimum    : yes
Optimization : 11280
Calls        : 1
Time         : 103.020s (Solving: 62.12s 1st Model: 0.09s Unsat: 62.03s)
CPU Time     : 103.011s
```

Figure 2: The output of the proposed solution for the sample instance in Figure 1

Figure 2 depicts the output of the Clingo program for the same problem instance, and it is evident that it is able to find a correct and optimal solution within a reasonable amount of time.

## Conclusion

The Referee Assignment Problem for Vehicle Companies is a challenging optimization problem that aims to assign referees to cases in a way that minimizes various constraints such as workload, preferences, and location. In this work, an ASP-based solution using Clingo has been proposed that effectively addresses the problem and provides optimal solutions for a variety of instances.

The notable contributions of this work include providing a detailed problem statement for the Referee Assignment Problem for Vehicle Insurance Companies, proposing an ASP-based solution, and implementing an efficient and effective program using Clingo. Finally, the solution is tested against various instances, and an analysis of the results has been done to show its effectiveness and applicability in real-world scenarios.

## Future Work

While the proposed solution is effective in finding correct solutions by satisfying all the hard constraints, there is room for improvement in the optimization aspect. The optimization part of the solution involves evaluating individual correct solutions to find the optimal one based on weak constraints. Currently, the performance of the proposed solution in this optimization step is not as efficient as desired, especially when dealing with complex scenarios. Therefore, the main future scope of this work is to focus on enhancing the efficiency of the optimization algorithm so that it can be scaled up to handle more complex real-world scenarios.

Moreover, future research can explore additional aspects of the referee assignment problem that were not considered in the current study. For instance, the current scenario only focuses on assigning referees to cases on a single working day, but in practice, a more comprehensive solution should be provided for a longer period of time. Also, the assumption that only one referee can handle each case might not hold in some situations. For example, in some cases, assigning two novice referees to handle a case might result in better outcomes than assigning a single expert referee. Therefore, future work can take into account these more realistic aspects of the problem to develop more effective solutions.