Individual Progress Report: 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

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.

Overall Progress

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. Clingo, a high-performance implementation of ASP that provides support for advanced solving techniques has been explored and discussed in this work.

Background 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, and Aggregates* 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.

Understanding the Problem Statement. The referee assignment problem can be a very complex and demanding optimization problem, but 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 one referee can be assigned to multiple cases. The hard constraints are intuitive and has correspondence to real-life scenarios, for example, each referee's workload must not exceed their maximum working minutes, or cases must not be assigned to referees who have zero expertise in the specific domain of the case. The weak constraints are the optimization concerns, and also intuitive, for instance, assigning internal referees is preferred as it is more cost-effective.

Challenges

The primary challenge associated with the referee assignment problem is its inherent *complexity* due to the large number of factors that are needed to be considered for each case, including the domain, required effort, geographic location, and claim amount, along with the type of the referee, their preferences, and availability for handling the cases. Even though from a top-level perspective, the problem might seem a simple mapping problem between referees and cases, in reality, it is rich with a variety of complex factors. Moreover, the problem addresses some of the very sophisticated subproblems such as *workload balancing, cost optimization, fair distribution of tasks*, and *preference matching*, which make the problem more challenging.

Solution. Addressing these challenges requires a sophisticated optimization algorithm and a comprehensive understanding of the constraints and the company's objectives. The solution approach taken in this work to tackle these challenges utilizes the *Elaboration Tolerance* feature of ASP that allows modifying the existing program in an incremental fashion, simultaneously ensuring that the new program is consistent with the previous answers. This feature brings more flexibility to the problem specification as it allows to consider one constraint at a time, making the comprehension of the problem a lot simpler, and as new optimization objectives are added, the same program can be incrementally modified to accommodate these changes. This approach allows to break down the complex problem into smaller easy-to-understand pieces, simultaneously maintaining the consistency of the solution as the problem specification evolves.

Tasks Completed

The following is a detailed list of the tasks that have been completed to date.

- Necessary background work on Clingo and ASP paradigm has been done to understand how Clingo can be used to model the problem and find optimized solutions for different problem instances.
- The problem statement along with the hard and weak constraints has been thoroughly comprehended to start programming.
- The development environment has been set up with necessary version control and a private remote repository in GitHub for future correspondence.
- The initial version of the code with instructions for generating the potential solutions and the following hard constraints are in place, as specified in Listing 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.
  - Cases must not be assigned to referees who have zero preference for the region in which the case occurred.
  - Cases must not be assigned to referees who have zero expertise in the specific domain of the case.
  - Cases with claim amounts higher than a certain threshold must only be assigned to internal referees.
- The initial version of the code is a correct (but not optimal) solution to the referee assignment problem as it satisfies all the hard constraints specified in the problem statement. Listing 2 and 3, respectively, specify an example problem instance, and the output of the current version of the program. It demonstrates that the solution matches the expected correct outcome.
- The initial version of the code is committed and pushed into the remote repository on GitHub.

Tasks To Be Done

This section describes the things that are yet to be done for the successful completion of the work, and the plan of execution of the future work.

- The immediate target is to understand the optimization factors (the weak constraints) to start modifying the initial code to find not only a correct but also an optimal solution for the referee assignment problem.
- It is required to understand the relative weightage of the following weak constraints in the final objective function and incorporate them into the code.
  - More cases should be assigned to internal referees to minimize the overall cost.
  - The assignment of cases to external referees should be fairly distributed to ensure that their overall payment is balanced.
  - The assignment of cases to all referees should be fairly distributed to ensure that their overall workload is balanced.
  - Referees should handle more cases with types they prefer more.
  - Referees should handle more cases in regions they prefer more.
- Once the modified code is in place, rigorous testing of the code with more example instances should be done to ensure the correctness of the solution.
- The limitations of the implementation is needed to be identified and addressed in the final report as a scope of future work.

The plan is to distribute the tasks in hand evenly between the upcoming two weeks; the first week is to be devoted purely to the comprehensive understanding of the weak constraints, their relative weightage in the final objective function, and the technical details to implement the same in Clingo, while the second week is assigned for the incremental implementation of the code and testing against an extensive set of test cases. The last week is kept as a buffer for debugging (if needed), documentation, and the final report preparation.

Appendix

Listing 1: Solution Clingo Program (Initial Version)

```
%% Definition of the predicates

% case(id, type, workload, damage, zipcode,
    payment)
% referee(id, type, max_workload,
    prev_workload, prev_payment)
% prefType(referee_id, case_type, pref)
% prefRegion(referee_id, zipcode, pref)
% externalMaxDamage(damage)
% assign(case_id, referee_id)


%% Potential Solution
% Every case is assigned to exactly one
    referee
1{assign(CID, RID): referee(RID, _, _, _, _)
    }1 :- case(CID, _, _, _, _, _).


%% Hard Constraints

% MaxWorkLoad of a Referee cannot exceed by
    the TotalWorkLoad
totalWorkLoad(RID, TWL) :- referee(RID, _, _
    , PWL, _), TWL = #sum{WL, CID: assign(
    CID, RID), case(CID, _, WL, _, _, _)}.
:- assign(CID, RID), referee(RID, _, MWL, _,
     _), case(CID, _, _, _, _, _),
    totalWorkLoad(RID, TWL), TWL > MWL.

% Cases must not be assigned to referees who
     have zero preference in the specific
    case type
:- assign(CID, RID), referee(RID, _, _, _, _
    ), case(CID, CTYPE, _, _, _, _),
    prefType(RID, CTYPE, 0).

% Cases must not be assigned to referees who
     have zero preference in the specific
    case region
:- assign(CID, RID), referee(RID, _, _, _, _
    ), case(CID, _, _, _, ZIP, _),
    prefRegion(RID, ZIP, 0).

% Cases with damage > externalMaxDamage can
    only be assigned to internal referees
:- assign(CID, RID), referee(RID, RTYPE, _,
    _, _), case(CID, _, _, DMG, _, _),
    externalMaxDamage(EMDMG), DMG > EMDMG,
    RTYPE = e.


%% Display
#show assign/2.
```

Listing 2: Example Problem Instance

```
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).

% expected result (optimum): assign(5, 7)
```

Listing 3: Solution Output

```
clingo version 5.4.1
Reading from solution.asp ...
Solving...
Answer: 1
assign(5,7)
SATISFIABLE

Models    : 1
Calls     : 1
Time      : 0.001s (Solving: 0.00s)
CPU Time  : 0.001s
```