CSE579 Final Project
Insurance Assignment Problem

Ben Rafalski

Problem Statement

The insurance assignment problem is a problem of finding the best insurance referee to check if a submitted customer claim is justified. The problem involves a few entities:

- Internal and external referees who can be assigned to a case.
- Cases that are assigned to referees.
- Regions where cases are located.

Assigning the best insurance referee is defined by the following constraints:

Hard Constraints

- The total number of minutes for the cases assigned to a referee must be below their maximum working minutes.
- Cases are not allowed to be assigned to referees not in charge of a region at all.
- Cases must not be assigned to a referee who is not in charge of that case type at all.
- Cases that exceed the damage threshold must be assigned to internal employees.

Weak Constraints

- Internal referees are preferred over external ones.
- Payments among external referees should be balanced.
- Workload should be balanced among referees.
- Referees should be assigned types of cases with higher preference.
- Referees should be assigned cases in regions with higher preference.

Background

To solve this problem, it is most suitable to use answer set programming (ASP). Answer set programming is part of the declarative paradigm. It is suitable for NP-hard search problems. This is because they are reduced to computing stable models and answering set solvers (programs to generate stable models). Most answer set solvers are based on the DPLL algorithm for computing stable models. In this implementation, the Clingo answer set solver is used.

An ASP program consists of a list of rules and facts. Each rule is of the form

HEAD :- BODY.

and will resolve to either true or false and the head and body are made up of atoms (such as p,q, or r). A fact is a rule with an empty body, these rules do not have the $:-$ ("if"/implication) symbol. Another type of rule is the choice rule. These rules are of the form

$$\{atom_1, ..., atom_i\}.$$

and represent an arbitrary choice that can be made among which atoms to include in the stable model. Many ASP solver also support cardinality constraints on choice rules. These are of the form

$$n\{atom_1, ..., atom_i\}m.$$

and say that the stable model must include at least n and at most m of the atoms in the choice rule. These cardinality constraints are also allowed to appear on the body of the rule as well. Variables in ASP are capitalized and can be passed within predicates or used in rules with a pattern. Ranges are also allowed in ASP and represent a range of values that a literal can be assigned. These ranges are of the form

$$(start..end)$$

where $start$ is the first value of the range and end is the last value in the range.

In Clingo, stable models for an ASP program can be generated by feeding the program to the Clingo command line binary file. Using 0 as input will give all stable models for the program while not supplying any arguments will only produce one stable model.

Lastly, constraints are a form of rule that are useful in constructing ASP programs. There are two types of constraint rules: hard and weak constraints. Hard constraints are of the form

:- BODY.

and represent rules that have a true head. If the body of the rule is false then the rule will always evaluate to false and no stable model will be produced for the respective set. Weak constraints are of the form

$$:\sim \text{BODY. } [w@p]$$

where w is a weight and p is a level. If the body if the rule is false then a weight of value w is added to the level p for this model. Levels with higher values will take precedence over ones with smaller values. When optimizing among multiple stable models, these rules allow the program to chose between stable models using the minimum values found at different levels.

Solution to the problem

To solve the problem described above, I implemented each of the hard constraints and each of the weak constraints as well as a general assignment rule. For general assignments, I formed a rule that has the meaning: every case must be assigned to one referee. Then I used ASP hard and weak constraints to remove the invalid stable models. For the first hard constraint, I used the #sum predicate to sum up all of the minutes for each case that was assigned to each referee and compare the value to the respective referee's max workload. For the next two hard constraints, I just removed all of the stable models with referees that has invalid preferences for type and region (i.e. has 0 as their preference). Finally, for the last hard constraint, I compared the damage for each assigned case to the damage threshold and removed the stable model if the damage was too high and the referee was external.

For the weak constraints, I created the first constraint on level five with highest preference. I added a weight to this constraint each time an external referee appeared in a stable model. For the next weak constraint, I balanced the pay among external employees on level four by calculating the total payment for each employee in the stable model and minimizing it using their previous pay at that point. For the third constraint, I did a similar approach on level three by computing the total workload for each referee and minimizing it using the previous workload they had. For the fourth constraint, I maximized the preference value for each stable model on level two so that higher preferences would be chosen. Similarly, for the final constraint I maximized the preferences of region on level one so that the highest region preferences were selected.

Main Results and Analysis

Instance 1 Results

clingo version 5.4.0
Reading from solver.lp ...
Solving...
Answer: 1
assign(4,5)
Optimization : 0 0 230 -2 -2
OPTIMUM FOUND

Models : 1
Optimum : yes
Optimization : 0 0 230 -2 -2
Calls : 1
Time : 0.003s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time : 0.016s

Instance 1 Analysis This instance was the simplest of all the instances. There was only one case of type c and in region 2000. It was 90 minutes long, had 3000 in damages, and payed $90. There were three referees in this scenario. Referee 4 and 5 were internal and referee 6 was external. Each one of the referees have a maximum minutes allowed to work larger than the 90 minutes for the single case so the first constraint does not apply to this test. There are type preferences for each one of the referees corresponding to the type c. Referee 4 has a preference of 0 so they will not be assigned to the case. Referee 5 has a preference of 2 and referee 6 has a preference of 3, so referee 6 will be chosen here is optimizing is needed. For region preferences, each of the referees has a preference defined for the region. Referee 4 has a preference of 3 but is not able to be included in the stable model already so it is irrelevant. The other two referees both have a preference of 2 for the region. The last hard constraint applies to referee 6 here as well since the max damage is larger than the 1500 threshold set in the instance. Thus, referee 5 is assigned the case.

This instance was testing for the hard constraint on type preferences and the hard constraint for the external damage threshold.

Instance 2 Results

clingo version 5.4.0
Reading from solver.lp ...
Solving...
Answer: 1
assign(5,7)
Optimization : 0 0 700 -3 -2
OPTIMUM FOUND

Models : 1
Optimum : yes
Optimization : 0 0 700 -3 -2
Calls : 1
Time : 0.003s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time : 0.000s

Instance 2 Analysis This instance was similar to instance 1 but test for different constraints. There was only one case with type a and in region 1000. It was 45 minutes long, had 700 in damages, and payed $60. There were three referees in this scenario. Referee 8 and 9 were external and referee 7 was internal. Each one of the referees have a maximum minutes allowed to work larger than the 45 minutes for the single case so the first constraint does not apply to this test. There are type preferences for each one of the referees corresponding to the type a. Referee 7 has a preference of 1, while referee 8 and referee 9 have a preference of 3 so they will get higher preferences. For region preferences, each of the referees has a preference defined for the region. Referee 7 has a preference of 3 and will get higher preferences. The other two referees both have a preference of 0 so they are automatically excluded form the stable model. The last hard constraint does not apply to this scenario since the damages are less than 1500. Thus, referee 7 is assigned the case.

This instance was testing for the hard constraint on region preferences.

Instance 3 Results
clingo version 5.4.0
Reading from solver.lp ...
Solving...
Answer: 1
assign(6,12)
Optimization: 1 4240 6480 -3 -2
Answer: 2
assign(6,11)
Optimization: 1 4240 6480 -3 -3
OPTIMUM FOUND

Models : 2
Optimum : yes
Optimization : 1 4240 6480 -3 -3
Calls : 1
Time : 0.004s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time : 0.000s

Instance 3 Analysis This instance was similar to instance 1 but test for different constraints. There was only one case with type b and in region 1000. It was 200 minutes long, had 2500 in damages, and payed $80. There were three referees in this scenario. All of the referees were external. Referee 10 has a max workload of 120 which is less than the 200 that is required for the case, so they are excluded from being assigned the case by the first hard constraint. There are type preferences for each one of the referees corresponding to the type b. Referee 10 has a preference of 3 and will get higher preferences for this constraint, while referee 11 and referee 12 have a preference of 2 so they will get lower preferences. For region preferences, each of the referees has a preference defined for the region. Referee 10 has a preference of 3, referee 11 has a preference of 2, and referee 12 has a preference of 1 so referee 10 will have the highest region preferences while referee 12 will have the lowest. The last hard constraint does not apply to this scenario since the damages are less than 3000. Both referee 12 and 11 have valid stable models but referee 11 is assigned the case since they have a higher preference set for this region.

This instance was testing for the hard constraint on max workload for a referee and the weak constraint on region preferences.

Instance 4 Results
clingo version 5.4.0
Reading from solver.lp ...
Solving...
Answer: 1
assign(7,14)
Optimization : 0 0 700 -3 -2
OPTIMUM FOUND

Models : 1
Optimum : yes
Optimization: 0 0 700 -3 -2
Calls : 1
Time : 0.003s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time : 0.016s

Instance 4 Analysis This instance was similar to instance 1 but test for different constraints. There was only one case with type b and in region 4000. It was 250 minutes long, had 2500 in damages, and payed $160. There were three referees in this scenario. Referee 13 and 14 were internal and referee 15 was external. Each one of the referees have a maximum minutes allowed to work larger than the 250 minutes for the single case so the first constraint does not apply to this test. There are type preferences for each one of the referees corresponding to the type b. Each one of the referees have a preference of 3 for this case type so the type preference constraints do not apply to this instance. For region preferences, each of the referees has a preference defined for the region. Referee 15 has a preference of 3 and will have the highest preference for this optimization. The other two referees both have a preference of 2 for the region, so their preference weights will be lower. The last hard constraint applies to this scenario since the the damages are more than 1500, so referee 15 will not be chosen. Thus, referee 14 is assigned the case since they have less hours worked than referee 13, so they need more work to balance the workload.

This instance was testing for the hard constraint on max damage for an external referee and the weak constraint on balancing workload.

Instance 5 Results
clingo version 5.4.0
Reading from solver.lp ...
Solving...
Answer: 1
assign(8,18)
Optimization: 1 4240 6480 -3 -2
Answer: 2
assign(8,17)
Optimization: 1 4240 6480 -3 -3
OPTIMUM FOUND

Models : 2
Optimum : yes
Optimization : 1 4240 6480 -3 -3
Calls : 1
Time : 0.003s (Solving: 0.00s 1st Model: 0.00s Unsat: 0.00s)
CPU Time : 0.000s

Instance 4 Analysis This instance was similar to instance 4 but test for different constraints. There was only one case with type a and in region 4000. It was 480 minutes long, had 2500 in damages, and payed $240. There were three referees in this scenario. Referee 17 and 18 were external and referee 15 was internal. Each one of the referees have a maximum minutes allowed to work equal 480 minutes for the single case so the first constraint does not apply to this test. There are type preferences for each one of the referees corresponding to the type a. Referee 16 has a preference of 0 so they are excluded from the assignment. Both other referees have a preference of 3, so this constraint does not affect them. For region preferences, each of the referees has a preference defined for the region. Referee 17 has a preference of 3 and will have the highest preference for this optimization. The other two referees both have a preference of 2 for the region, so their preference weights will be lower. The last hard constraint does not apply to this scenario since the damages are not more than 2500. Thus, referee 17 is assigned the case since they have a larger region preference than referee 18.

This instance was testing for the hard constraint on region preferences and the weak constraint on region preferences.

Conclusions

Overall Conclusions To end, the insurance referee assigning problem is about finding referees to assign to check on the validity of customer claims. There are both internal and external referees and many preferences for each referee.

There are many hard and weak constraints that provide the basis for how referees are assigned. Hard constraints on referees are: workload constraints for assigned cases, region preferences, case type preferences, and external referee damage threshold constraints. The weak constraints consist of limiting external referees, balancing external employee payments, balancing referee workloads, and preference constraints on region and case type.

The solution to the problem can easily be implemented in answer set programming (ASP). This paradigm allows for the programmer to define a set of rules and fact to produce stable models. These stable models are the output and serve as the answer to the problem being addresses.

By efficiently allocating referees to cases, the insurance company can streamline the claim verification process, enhance customer satisfaction, and optimize resource utilization. It also allows for fair and balanced distribution of workload and payments among internal and external referees.

Overall, the referee assigning problem plays a crucial role in ensuring the efficient and effective handling of insurance cases, promoting fairness, and maintaining customer trust in the claims evaluation process.

In the end, the referee assigning problem is a great example of how ASP can be utilized to solve real world problems affecting industries such as insurance agencies. This strict programming environment ensures that claims are evaluated with the highest of confidence and reassures customers of the trust they have with their agents.

Self-Assessment I believe I met all the requirements for this project and sufficiently answered the questions. For each instance, I identified the correct stable model(s) and optimizations. I have implemented each one of the hard and each one of the weak constraints. My work should be of good use to insurance agencies who wish to adhere to strict assigning policies. If I were to redo the project over again, I would start by analyzing the given test cases first before programming. In this iteration I began programming first and implemented an extra weak constraint that was not being used in any of the test cases (payment in external employees need to be balanced).

Opportunities for Future Work

There is a wide range of possibilities that are able to be assessed in the future for better advancements in the are of assigning the best referee to the best case such as:

- In the future, further research can be done into the optimization of the algorithms used for the referee assigning problem. They can use linear programming, integer programming, and other kinds of algorithms to make the process more accurate and more efficient.
- The insurance companies can also look into dynamic and real time assignments. They can set the program to run with a front end and allow referees to set their preferences through there. Then they can use the algorithm to assign referees to cases automatically.
- More research can also be done using machine learning. They can use data to assess preferences and referee attributes such as time worked or payments received. This can add to the algorithm and make it even more optimal than it already is using ASP.
- More preferences and areas can also be assesses using the same algorithm provided here. These preference can also be aimed at customers and providing the best service possible. Different customers can become a part of the algorithm and have their own preferences and attributes like the referees have.
- Adding more constraints can also be done in the future to make the algorithm more strict to the needs of the insurance agencies. This can optimize for attributes such as travel distance and referee level to save on traveling and to assign better qualified referees to harder cases.