Insurance Referee Assignment Problem

1. Problem Description

An insurance company needs to check if customer claims in insurance cases are justified. To this end, it sends referees to different locations to inspect damages (e.g. damaged cars) and write reports. The insurance company employs its own referees (internal referees) but can also authorize external referees to handle a case. The overall task is to assign referees to insurance cases according to various hard and weak constraints.

Referees have a maximum workload per day, which is given by the maximum number of working minutes per day. Referees are in charge of certain geographical regions, which are identified by their postal code, where the assignment of referees to regions is gradual to support preferences (e.g. referees must not, can but prefer not to, or prefer to take a case in a certain region, etc); the degree of preference is represented by an integer. Moreover, referees can be specialized in certain domains (e.g. passenger cars, trucks, etc). As for the regions, the assignment of referees to types of cases supports degrees of preferences. While internal referees receive a fixed salary per month, external referees are paid per case, where the amount of payment depends on the case.

Insurance cases are defined by the expected effort to handle it (number of working minutes), the amount of damage (in Euros), and the payment an external referee receives if assigned to this case. Note that these attributes do not necessarily correlate, i.e., there can be cases with smaller expected effort and/or amount of damage but higher payment for the referee.

Our scenario focuses on an assignment of referees to cases on a single working day. An instance consists of the set of available referees and the set of cases to handle. The task is to assign exactly one referee to each case (but a referee can be assigned to multiple cases) according to the following constraints.

Hard constraints:

- The maximum number of working minutes of a referee must not be exceeded by the actual workload, where the actual workload is the sum of the efforts of all cases assigned to this referee.
- A case must not be assigned to a referee who is not in charge of the region at all (i.e., who has preference 0; see below).
- A case must not be assigned to a referee who is not in charge of the type of the case at all (i.e., who has preference 0; see below).
- Cases with an amount of damage that exceeds a certain threshold can only be assigned to internal referees.

Weak constraints:

- Internal referees are preferred in order to minimize the costs of external ones.
- The assignment of cases to external referees should be fair in the sense that their overall payment should be balanced (i.e., they should all have the chance to handle cases such that their overall payments are similar).
- The assignment of cases to (internal and external) referees should be fair in the sense that their overall workload should be balanced.
- Referees should handle types of cases with higher preference.
- Referees should handle cases in regions with higher preference.

2. Input Description

We focus on an assignment for a single working day. An instance consists of a set of available *referees*, a set of *cases* to be handled, a threshold for external referees, and preferences of regions and types of cases. The input format is as follows.

2.1. Referees are given as facts of form

referee(*rid*, *type*, *max_workload*, *prev_workload*, *prev_payment*).

where

- *rid* is a positive integer that servers as unique identifier of the referee,
- *type* defines the type of the referee and is one of the constants *internal* or *external*,
- *max_workload* is an integer in the range [0,720] that defines the maximum working minutes of the referee per day,
- *prev_workload* is an integer that represents the previous workload of the referee (i.e., the sum of the efforts of all cases handled so far),
- *prev_payment* is an integer that represents the sum of previous payments of external referees (is 0 for internal ones).

2.2. Cases are given as facts of form

case(*cid*, *type*, *effort*, *damage*, *postc*, *payment*).

where

- *cid* is a positive integer that servers as unique identifier of the case,
- *type* is a constant that defines the domain of the case (e.g. *passenger car*, *truck*, etc)
- *effort* is an integer that defines the expected number of working minutes needed to handle the case,
- *damage* is an integer that defines the amount of damage in Euros,
- *postc* postal code of the location of this case,
- *payment* is an integer that defines how much payment an external referee receives for handling this case.

2.3. The maximum damage of cases assigned to external referees is given by a single fact of form

externalMaxDamage(*d*).

where *d* is an integer. Cases with a damage greater than *d* must be handled by internal referees.

2.4. Preferences of regions are given as facts of form

prefRegion(*rid*, *postc*, *pref*).

where

- *rid* is a referee id,
- *postc* is the postal code of a region,
- *pref* is an integer value in the range [0,3].

It encodes that *rid* should take a case in region *postc* with preference *pref*, where a higher value of *pref* denotes higher preference. The special case *pref*=0 means that *rid* is not allowed to take a case in region *postc* at all (hard constraint). We assume that if for a certain pair of *rid* and *postc* no *pref* is specified, then it is 0 by default.

2.5. Preferences of case types are given as facts of form

prefType(*rid*, *caset*, *pref*).

where *rid* is a referee id, *caset* is a case type, and *pref* is an integer value in the range [0,3]. It encodes that *rid* should take a case with type *caset* with preference *pref*, where a higher value of *pref* denotes higher preference; the special case *pref*=0 means that *rid* is not allowed to take a case with type *caset* at all (hard constraint). We assume that if for a certain pair of *rid* and *caset* no *pref* is specified, then it is 0 by default.

3. Output Description

The output is an assignment of exactly one referee to each case, which is to be represented by atoms of form

assign(*cid*, *rid*)

where *cid* is the id of a case and *rid* is the id of a referee.

4. Examples

Assumed:

- referee(*rid*, *type*, *max_workload*, *prev_workload*, *prev_payment*) as laid out under 2.1

- case(*cid*, *type*, *effort*, damage, *postc*, *payment*) as laid out under 2.2
  i := internal referee; e:= external referee
- externalMaxDamage(*d*) as laid out under 2.3
- prefRegion(*rid*, *postc*, *pref*) as laid out under 2.4
- prefType(*rid*, *caset*, *pref*) as laid out under 2.5

Example 1

given:

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*).

assignment:

assign(4, 5)

Example 2

given:

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*).

assignment:

assign(5, 7)

Example 3

given:

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*).

assignment:

assign(6, 11)

Example 4

given:

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*).

assignment:

assign(7, 14)

Example 5

given:

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*).

assignment:

assign(8, 17)

Example 6

given:

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*).

assignment:

assign(8, 19)
assign(9, 19)
assign(10, 21)

Example 7

given:

case(*11*, a, 90, 10000, 3033, 65).
case(*12*, b, 90, 5000, 3033, 65).
case(*13*, a, 90, 5000, 3034, 65).

referee(23, i, 480, 3000, 0).
referee(24, i, 480, 3000, 0).
referee(25, e, 360, 1000, 1000).
referee(26, e, 120, 2500, 1700).

prefType(23, a, 2).
prefType(23, b, 2).
prefType(24, a, 1).
prefType(24, b, 3).
prefType(25, a, 1).
prefType(25, b, 3).
prefType(26, a, 3).
prefType(26, b, 0).

prefRegion(23, 3033, 1).
prefRegion(23, 3034, 3).

prefRegion(24, 3033, 2).
prefRegion(24, 3034, 1).
prefRegion(25, 3033, 2).
prefRegion(25, 3034, 1).
prefRegion(26, 3033, 3).
prefRegion(26, 3034, 1).

externalMaxDamage(*10000*).

assignment:

assign(11, 25)
assign(12, 25)
assign(13, 25)

Example 8

given:

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*).

assignment:

assign(14, 27)
assign(15, 27)

Example 9

given:

case(*16*, b, 180, 2000, 7013, 100).
case(*17*, b, 180, 2000, 7013, 100).

referee(30, i, 480, 20000, 0).
referee(31, i, 480, 10000, 0).
referee(32, i, 480, 1000, 0).

prefType(30, b, 3).
prefType(32, b, 2).

prefRegion(30, 7013, 2).
prefRegion(31, 7013, 3).
prefRegion(32, 7013, 3).

externalMaxDamage(*10000*).

assignment:

assign(16, 32)
assign(17, 32)

Example 10

given:

case(*1*, a, 120, 1600, 1200, 73).
case(2, a, 440, 1000, 1190, 91).
case(3, b, 120, 800, 1190, 31).

referee(1, i, 600, 220, 0).
referee(2, e, 360, 140, 2800).
referee(3, e, 240, 40, 700).

prefType(1, a, 1).
prefType(1, b, 2).
prefType(2, *a*, 2).
prefType(3, b, 2).

prefRegion(1, 1200, 1).
prefRegion(1, 1190, 1).
prefRegion(2, 1200, 2).
prefRegion(2, 1190, 1).
prefRegion(3, 1190, 1).

externalMaxDamage(*1500*).

assignment:

assign(1, 1)
assign(2, 1)
assign(3, 3)

5. Scoring Schema

Optimization citeria. The optimization criteria are defined as follows:

- A: The sum of all payments of cases assigned to external referees should be minimized. We let c_A be the sum of all payments to external referees.
- B: Let the overall payment o_{rid} of an external referee *rid* be the sum of her/his *prev_payment* and the payment s/he receives for newly assigned cases, and let *avg* be the average* overall payment over all external referees. To balance the overall payments of all external referees, for each referee *rid*, the divergence from the average is penalized with costs $|avg - o_{rid}|$. We let c_B be the sum of all these costs.
- C: Let the overall workload w_{rid} of an (internal or external) referee *rid* be the sum of her/his *prev_workload* and the workload for newly assigned cases, and let *avg* be the average overall workload over all referees. To balance the workloads of all referees, for each referee *rid*, the divergence from the average is penalized with costs $|avg - w_{rid}|$. We let c_C be the sum of all these costs.
- D: For a referee id *rid*, a case type *caset*, and an integer value *pref* in the range [0,3], a fact of form *prefType(rid, caset, pref).* encodes that *rid* should take a case with type *caset* with preference *pref*, where a higher value of *pref* denotes higher preference; the special case *pref*=0 means that *rid* is not allowed to take a case with type *caset* at all (hard constraint). For values *pref* > 0, taking a case causes costs (3 - *pref*). We let c_D be the sum of all these costs.
- E: For a referee id *rid*, an integer postal code *postc*, and an integer value *pref* in the range [0,3], a fact of form *prefRegion(rid, postc, pref).* encodes that *rid* should take a case in region *postc* with preference *pref*, where a higher value of *pref* denotes higher preference; the special case *pref*=0 means that *rid* is not allowed to take a case in region *c* at all (hard constraint). For values *pref* > 0, taking a case causes costs (3 - *pref*). We let c_E be the sum of all these costs.

(* Here and in the following, the average is to be truncated to an integer.)

The overall costs to be minimized are given as follows:

$$c = 16 \cdot c_A + 7 \cdot c_B + 9 \cdot c_C + 34 \cdot c_D + 34 \cdot c_E$$

The weights of the individual costs capture different levels of importance the above criteria.

Ranking of participants. Given a solver s and an instance i, the score of s on the instance i, denotes as score(s, i), is computed as follows:

• score(s, i) = 1.5 if s proves the optimality of the solution

• score(s, i) = 0 if s found no solution
• score(s, i) = (cost(sbest,i)+1) / (cost(s,i)+1)

where cost(s, i) is the cost of the solution printed by the solver s on the instance i and cost(sbest, i) is the best cost printed on the instance i. The score is rounded off to three decimal digits. The overall score of a solver s is computed as the sum of the scores of s for each instance i. The solver with the maximum score is the winner.

Example. Let s_1, s_2, s_3 and s_4 be solvers and let i be an instance. Assume the following costs:

cost(s_1 , i) = 100 (opt), cost(s_2 , i) = 100, cost(s_3 , i) = 200, cost(s_4 , i) = 400

Then, their scores are the following:

score(s_1, i) = 1.5, score(s_2, i) = 1, score(s_3, i) = 0.502, score(s_4 , i) = 0.252