

Appendix

% every case must be assigned to one referee

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

% HARD CONSTRAINTS

% The maximum number of working minutes of a referee must not be exc% every case must be assigned to one referee

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

% 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.

```
:- assign(CID, RID), case(CID, _, EFFORT, _, _, _), referee(RID, _, MAX_WORKLOAD, PREV_WORKLOAD, _),  
#sum { X, C : C==CID, case(C, _, X, _, _, _) } > MAX_WORKLOAD.
```

% 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).

```
:- assign(CID, RID), case(CID, _, _, _, POSTC, _), referee(RID, _, _, _, _), prefRegion(RID, POSTC, 0).
```

% 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).

```
:- assign(CID, RID), case(CID, CASET, _, _, _, _), referee(RID, _, _, _, _), prefType(RID, CASET, 0).
```

% Cases with an amount of damage that exceeds a certain threshold can only be

% assigned to internal referees.

```
:- assign(CID, RID), case(CID, _, _, DAMAGE, _, _), referee(RID, e, _, _, _), externalMaxDamage(THRESHOLD),  
DAMAGE > THRESHOLD.
```

% WEAK CONSTRAINTS

% Internal referees are preferred in order to minimize the costs of external ones.

```
:- assign(CID, RID), case(CID, _, _, _, _, _), referee(RID, e, _, _, _). [1@5]
```

% 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).

```
total_payment(RID, TOTAL) :- referee(RID, _, _, _, _),  
#sum { P, C, R: assign(C, R), case(C, _, _, _, P), referee(R, _, _, _, _), R==RID } = TOTAL.
```

```
:- assign(CID, RID), case(CID, _, _, _, _), referee(RID, e, _, _, TOTAL_PAY), total_payment(RID, TOTAL), N = TOTAL + TOTAL_PAY. [N@4]
```

% The assignment of cases to (internal and external) referees should be fair in the

% sense that their overall workload should be balanced.

```
total_workload(RID, TOTAL) :- referee(RID, _, _, _, _),  
#sum { H, C, R: assign(C, R), case(C, _, H, _, _, _), referee(R, _, _, _, _), R==RID } = TOTAL.
```

```
:- assign(CID, RID), case(CID, _, _, _, _), referee(RID, _, _, PREV_WORKLOAD, _), total_workload(RID, TOTAL), N = TOTAL + PREV_WORKLOAD. [N@3]
```

% Referees should handle types of cases with higher preference.

```
:- assign(CID, RID), case(CID, CASET, _, _, _, _), referee(RID, _, _, _, _), prefType(RID, CASET, PREF). [PREF@2]
```

% Referees should handle cases in regions with higher preference.

```
:- assign(CID, RID), case(CID, _, _, _, POSTC, _), referee(RID, _, _, _, _), prefRegion(RID, POSTC, PREF). [PREF@1]
```

#show assign/2.

eeded by

% the actual workload, where the actual workload is the sum of the efforts of all

% cases assigned to this referee.

```
:- assign(CID, RID), case(CID, _, EFFORT, _, _, _), referee(RID, _, MAX_WORKLOAD, PREV_WORKLOAD, _),  
#sum { X, C : C==CID, case(C, _, X, _, _, _) } > MAX_WORKLOAD.
```

% 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).

```
:- assign(CID, RID), case(CID, _, _, _, POSTC, _), referee(RID, _, _, _, _), prefRegion(RID, POSTC, 0).
```

% 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).

```
:- assign(CID, RID), case(CID, CASET, _, _, _, _), referee(RID, _, _, _, _), prefType(RID, CASET, 0).
```

% Cases with an amount of damage that exceeds a certain threshold can only be

% assigned to internal referees.

```
:- assign(CID, RID), case(CID, _, _, DAMAGE, _, _), referee(RID, e, _, _, _), externalMaxDamage(THRESHOLD),
DAMAGE > THRESHOLD.
```

```
% WEAK CONSTRAINTS
```

```
% Internal referees are preferred in order to minimize the
costs of external ones.
```

```
:- assign(CID, RID), case(CID, _, _, _, _, _), referee(RID,
e, _, _, _). [1@5]
```

```
% 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).
```

```
total_payment(RID, TOTAL) :- referee(RID, _, _, _, _),
#sum { P, C, R: assign(C, R), case(C, _, _, _, _, P), referee(R, _, _, _, _), R==RID } = TOTAL.
```

```
:- assign(CID, RID), case(CID, _, _, _, _, _), referee(RID,
e, _, _, TOTAL_PAY), total_payment(RID, TOTAL), N =
TOTAL + TOTAL_PAY. [N@4]
```

```
% The assignment of cases to (internal and external) referees
should be fair in the
```

```
% sense that their overall workload should be balanced.
```

```
total_workload(RID, TOTAL) :- referee(RID, _, _, _, _),
#sum { H, C, R: assign(C, R), case(C, _, H, _, _, _), referee(R, _, _, _, _), R==RID } = TOTAL.
```

```
:- assign(CID, RID), case(CID, _, _, _, _, _), referee(RID,
_, _, PREV_WORKLOAD, _), total_workload(RID, TOTAL), N =
TOTAL + PREV_WORKLOAD. [N@3]
```

```
% Referees should handle types of cases with higher preference.
```

```
:- assign(CID, RID), case(CID, CASET, _, _, _, _), referee(RID,
_, _, _, _), prefType(RID, CASET, PREF). [-
PREF@2]
```

```
% Referees should handle cases in regions with higher preference.
```

```
:- assign(CID, RID), case(CID, _, _, _, POSTC, _), referee(RID,
_, _, _, _), prefRegion(RID, POSTC, PREF). [-
PREF@1]
```

```
#show assign/2.
```