Example (Doctor rostering)

	Mon	Tue	Wed	Thu	Fri	Sat	Sun
Doctor A	call	none	oper	none	oper	none	none
Doctor B	app	call	none	oper	none	none	call
Doctor C	oper	none	call	app	app	call	none
Doctor D	app	oper	none	call	oper	none	none
Doctor E	oper	none	oper	none	call	none	none

Constraints to be **satisfied**:

1. #doctors-on-call / day $= 1$
2. #operations / workday ≤ 2
3. #operations / week ≥ 7
4. #appointments / week ≥ 4
5. day off after operation day
6. . . .

Objective function to be **minimised**:

- Cost: . . .

Example (Doctor rostering)

```
set of int: Days = 1..7;
set of int: Mon2Fri = 1..5;
enum Doctors = {Dr_A, Dr_B, Dr_C, Dr_D, Dr_E};
enum ShiftTypes = {app, call, oper, none};

array[Doctors,Days] of var ShiftTypes: Roster;

solve minimize ...;  % plug in an objective function

constraint forall(d in Days)
   (count(Roster[..,d],call) = 1);
constraint forall(w in Mon2Fri)
   (count(Roster[..,w],oper) <= 2);
constraint count(array1d(Roster),oper) >= 7;
constraint count(array1d(Roster),app) >= 4;
constraint forall(D in Doctors)
   (regular(Roster[D,..], "((oper none)|app|call|none)*"));
...  % other constraints
```