

Mathematical Programming: Modelling and Applications

Sonia Cafieri

LIX, École Polytechnique

`cafier@lix.polytechnique.fr`

November 2009

AMPL script - RandomWalk algorithm

```
param m, integer; # number of clauses
param n, integer; # number of variables (2n is the number of literals)
set M := 1..m;
set N := 1..n;
# phi[i,j] = 1 if x_j appears in clause i
#           = -1 if bar{x}_j appears in clause i
#           = 0 otherwise
param phi{M,N} integer, default 0;

param k >= 0 integer; # number of clauses you aim to satisfy
param x{N} binary;
param y{M} binary, default 0;

data max2sat_m9_n3.dat;

param fstar integer, >= 0, default 0;
param xstar{N} binary, default 0;
param ii integer;
param jj integer;
param l integer, default 1;
param ll integer, default 1;
param satisf integer, default 0;
param changedv{M,N} binary, default 0;
param flag binary, default 0;
param termination binary, default 0;
param iter integer, default 0;
param maxiter integer, default 100;

let k := m-1;
```

AMPL script - RandomWalk algorithm

```
# start with a random generated assignments
...to be completed...

repeat while (termination=0) {

    # check if the current assignment satisfies the clauses
    ...to be completed...
    # count the number of satisfied clauses
    ...to be completed...

    if (satisf< k) then {
        # choose the unsatisfied clause to be changed
        let ii :=1;
        let flag :=0;
        repeat while(ii <= m and flag= 0) {
            ...to be completed...
        }
        # here l is the unsatisfied clause to change, thus...
        # change the clause l
        let flag :=0;
        let jj :=1;
        # I look for a not previously changed variable
        ...to be completed...
        # change the value of variable ll
        ...to be completed...
    } else{
        print "problemsolved";
        let termination:= 1;
    }
    let iter := iter+1;
    if(iter> maxiter)then {
        print "iteration limit reached";
        let termination:= 1;
    }
} #endwhile
```

AMPL script - RandomWalk algorithm

```
for {i in 1..n}{  
  let xstar[i]:= x[i];  
}  
printf "iterations= %d\n", iter-1;  
printf "satisfied clauses =%d\n", satisf;  
display xstar;
```