

Denise,

petit clin d'oeil. Voici une modélisation avec des contraintes en GNU Prolog.

On commence par initialiser le solveur FD pour lui dire le plus grand entier qui nous intéresse (ici 60000 = le plus grand nombre pair N dont on veut la décomposition).

```
| ?- fd_set_vector_max(60000).  
yes
```

Ensuite on dit juste les contraintes à vérifier (N est pair et il est la somme de R et S qui sont premiers,  $R \leq S$  évite les symétries).

Rappel, lorsque Prolog affiche une solution, au '?' tu peux répondre:

- ENTREE pour dire que tu veux arrêter là,
- ';' pour dire que tu veux la solution suivante,
- 'a' pour lui demander toutes les solutions restantes (Attention ça peut être long).

```
| ?- N #= 2 * _, N #= R + S, R #=<= S, fd_prime(R), fd_prime(S),  
fd_labeling(N), fd_labeling([R,S]).
```

```
N = 4  
R = 2  
S = 2 ? ;
```

```
N = 6  
R = 3  
S = 3 ? ;
```

```
N = 8  
R = 3  
S = 5 ? ;
```

```
N = 10  
R = 3  
S = 7 ? ;
```

```
N = 10  
R = 5  
S = 5 ? ;
```

```
N = 12  
R = 5  
S = 7 ? ;
```

```
N = 14  
R = 3
```

$$S = 11 ? ;$$

$$N = 14$$

$$R = 7$$

$$S = 7 ? ;$$

$$N = 16$$

$$R = 3$$

$$S = 13 ? ;$$

$$N = 16$$

$$R = 5$$

$$S = 11 ? ;$$

$$N = 18$$

$$R = 5$$

$$S = 13 ?$$

Si tu veux une seule décomposition par nombre pair N:

yes

```
| ?- N #= 2 * _, N #= R + S, R #=< S, fd_prime(R), fd_prime(S),  
fd_labeling(N), once(fd_labeling([R,S])).
```

$$N = 4$$

$$R = 2$$

$$S = 2 ? ;$$

$$N = 6$$

$$R = 3$$

$$S = 3 ? ;$$

$$N = 8$$

$$R = 3$$

$$S = 5 ? ;$$

$$N = 10$$

$$R = 3$$

$$S = 7 ? ;$$

$$N = 12$$

$$R = 5$$

$$S = 7 ? ;$$

$$N = 14$$

$$R = 3$$

$$S = 11 ? ;$$

$$N = 16$$

$$R = 3$$

$$S = 13 ? ;$$

N = 18  
R = 5  
S = 13 ? ;

N = 20  
R = 3  
S = 17 ?

Si tu veux toutes les décomposition pour un N donné (ex: 1802) avec un affichage sympa (on utilise format = printf en C):

```
| ?- N=1802, N #= 2 * _, N #= R + S, R #=<= S, fd_prime(R),  
fd_prime(S), fd_labeling(N), fd_labeling([R,S]), format('%d = %d +  
%d\n', [N,R,S]), fail.  
1802 = 13 + 1789  
1802 = 19 + 1783  
1802 = 43 + 1759  
1802 = 61 + 1741  
1802 = 79 + 1723  
1802 = 103 + 1699  
1802 = 109 + 1693  
1802 = 139 + 1663  
1802 = 181 + 1621  
1802 = 193 + 1609  
1802 = 223 + 1579  
1802 = 271 + 1531  
1802 = 313 + 1489  
1802 = 331 + 1471  
1802 = 349 + 1453  
1802 = 373 + 1429  
1802 = 379 + 1423  
1802 = 421 + 1381  
1802 = 499 + 1303  
1802 = 523 + 1279  
1802 = 571 + 1231  
1802 = 601 + 1201  
1802 = 631 + 1171  
1802 = 673 + 1129  
1802 = 709 + 1093  
1802 = 733 + 1069  
1802 = 739 + 1063  
1802 = 751 + 1051  
1802 = 769 + 1033  
1802 = 811 + 991  
1802 = 883 + 919
```

Si tu veux restreindre la plage du nombre N :

```
fd_domain(N, 20902, 20924), N #= 2 * _, N #= R + S, R #=<= S,  
fd_prime(R), fd_prime(S), fd_labeling(N),
```

once(fd\_labeling([R,S])).

N = 20902  
R = 3  
S = 20899 ? a

N = 20904  
R = 5  
S = 20899

N = 20906  
R = 3  
S = 20903

N = 20908  
R = 5  
S = 20903

N = 20910  
R = 7  
S = 20903

N = 20912  
R = 13  
S = 20899

N = 20914  
R = 11  
S = 20903

N = 20916  
R = 13  
S = 20903

N = 20918  
R = 19  
S = 20899

N = 20920  
R = 17  
S = 20903

N = 20922  
R = 19  
S = 20903

N = 20924  
R = 3  
S = 20921

Pour un affichage plus sympa (correspondant à ton papier 'partager.pdf') :

```
| ?- fd_set_vector_max(60000), fd_domain(N, 20902, 20924), N #= 2
* _, N #= R + S, R #=< S, fd_prime(R), fd_prime(S),
fd_labeling(N), once(fd_labeling([R,S])), format('%d = %-5d +
%d\n', [N,R,S]), fail.
20902 = 3      + 20899
20904 = 5      + 20899
20906 = 3      + 20903
20908 = 5      + 20903
20910 = 7      + 20903
20912 = 13     + 20899
20914 = 11     + 20903
20916 = 13     + 20903
20918 = 19     + 20899
20920 = 17     + 20903
20922 = 19     + 20903
20924 = 3      + 20921
```

Sympa non ?

Enjoy !

Daniel