

## Valutazione della Soluzione

La soluzione del tuo amico è molto elegante e dimostra una comprensione profonda dei CSP. La tecnica di symmetry breaking è il punto di forza:

**Strategia originale:**  $Tav1 \neq Tav2$  - garantisce solo che persone in conflitto stiano in tavoli diversi **Strategia ottimizzata:**  $Tav1 < Tav2$  - impone un ordinamento che elimina soluzioni simmetriche

Questo trade-off (23→20 punti per un drastico risparmio di tempo) è una scelta ingegneristica eccellente. Nel mondo reale, una soluzione sub-ottimale ma computazionalmente efficiente è spesso preferibile.

## Miglioramenti Possibili

La soluzione potrebbe essere raffinata con:

1. **Heuristic più sofisticate:** Invece di `labelingffc`, potresti implementare una euristica che assegna prima le persone con più conflitti
2. **Constraint ridondanti:** Aggiungere vincoli impliciti che aiutano la propagazione
3. **Symmetry breaking avanzato:** Ordinare anche sui tavoli delle persone che si conoscono

## Analisi della Performance

Il tuo amico ha ragione: il symmetry breaking accelera drasticamente la ricerca. Senza il vincolo  $Tav1 < Tav2$ , dovresti esplorare molte più soluzioni equivalenti.

## Miglioramenti Implementati

### 1. Symmetry Breaking Migliorato

```
(P1 < P2 -> Tav1 < Tav2 ; Tav2 < Tav1)
```

Ordina basandosi sull'ID delle persone, non solo sui conflitti.

### 2. Global Constraints per Efficienza

```
global_cardinality(Invitati, Pairs)
```

Più efficiente di `occurrences` multipli per vincoli di capacità.

**3. Euristică Most-Constrained-First** Il nuovo `smart_labeling` assegna prima le persone con più conflitti, riducendo il backtracking.

**4. Constraint Redundanti** Aggiunge vincoli di bilanciamento per distribuire meglio gli invitati.

**5. Utilities di Analisi** Funzioni per visualizzare e analizzare le soluzioni trovate.

## Come Testare

Salva il codice migliorato come `matr_enhanced.ecl` e prova:

```
[matrimonio].  
[matr_enhanced].  
compare_solutions.
```

## Ulteriori Estensioni Possibili

1. **Preferenze di vicinanza:** Aggiungere pesi diversi alle conoscenze
2. **Vincoli di diversità:** Bilanciare età, genere, etc.
3. **Ottimizzazione multi-obiettivo:** Bilanciare punteggio vs uniformità tavoli
4. **Interface grafica:** Visualizzazione della disposizione tavoli
5. **Parallelizzazione:** Esplorare rami diversi in parallelo

## Valore Pedagogico

Questa estensione è ottima per:

- Dimostrare tecniche avanzate di CSP
- Confrontare performance di euristiche diverse
- Mostrare l'importanza del symmetry breaking
- Insegnare l'uso di global constraints

Il codice originale del tuo amico è già molto buono - questi miglioramenti aggiungono sofisticazione mantenendo la chiarezza concettuale.