A Comparison of Optimization Methods for Multi-Objective Constrained Bin Packing Problems

P. Olivier[1], A. Lodi[1], G. Pesant[1]

[1]École polytechnique de Montréal, Montreal, Canada
{philippe.olivier, andrea.lodi, gilles.pesant}@polymtl.ca

JOPT 2018

8 May 2018

Outline

1. The Wedding Seating Problem
2. Existing Methods
3. Constraint Programming Model

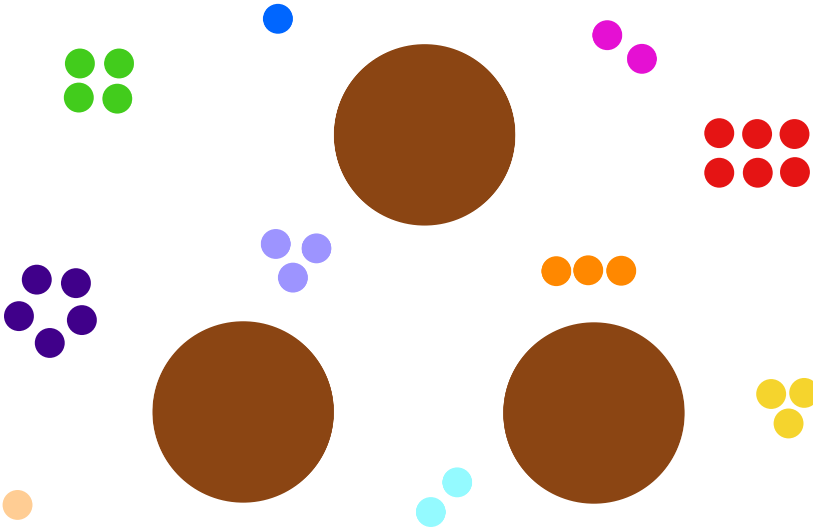
Outline

1. The Wedding Seating Problem

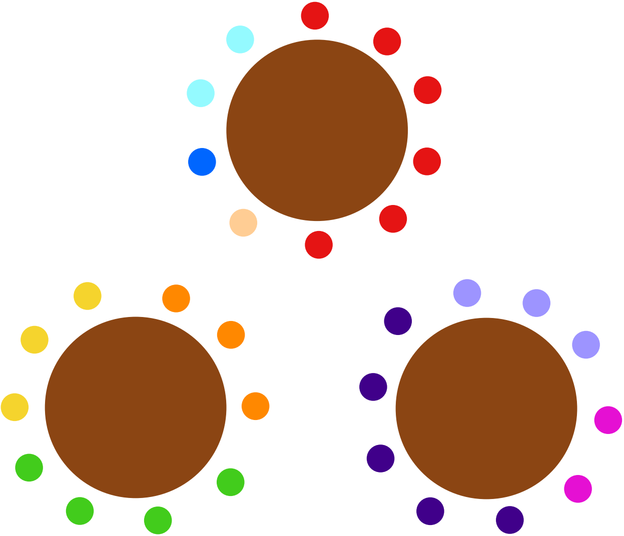
2. Existing Methods

3. Constraint Programming Model

The Wedding Seating Problem



The Wedding Seating Problem

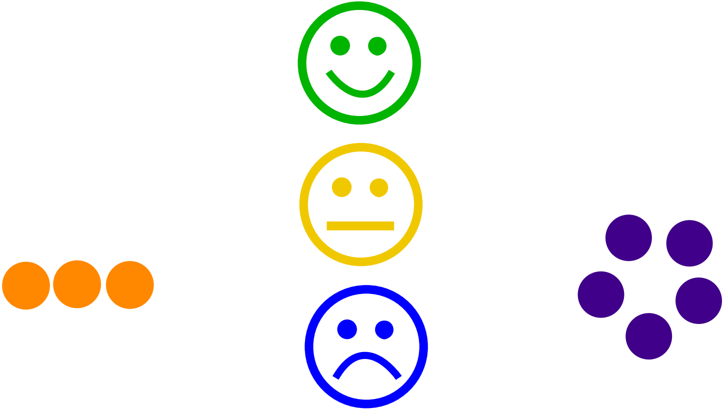


The Wedding Seating Problem

The Wedding Seating Problem

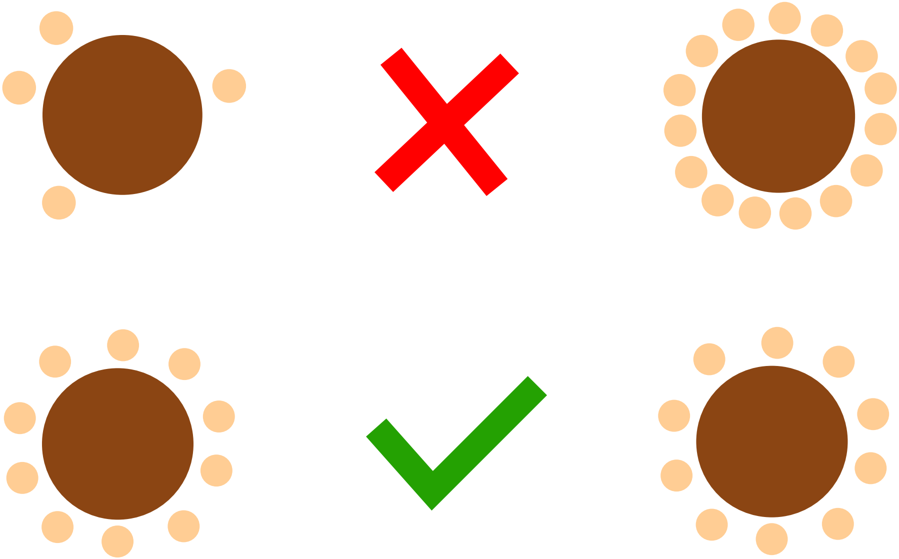
The Wedding Seating Problem

The Wedding Seating Problem



The Wedding Seating Problem

The Wedding Seating Problem



Outline

1. The Wedding Seating Problem

2. Existing Methods

3. Constraint Programming Model

Existing Methods

Overview

- Original IP model [Bellows and Petersen, *Annals of Improbable Research*, 2012].
- Two-stage algorithm using tabu search [Lewis, *WorldComp International Conference Proceedings*, 2013].
- Improved IP model [Lewis and Carroll, *Journal of the Operational Research Society*, 2016].

Existing Methods

Two-stage algorithm using tabu search [Lewis, 2013]

Build a graph where nodes represent groups, edges represent relations, and colors represent tables.

Existing Methods

Two-stage algorithm using tabu search [Lewis, 2013]

Build a graph where nodes represent groups, edges represent relations, and colors represent tables.

- Stage 1: Color nodes to find an initial feasible solution

Existing Methods

Two-stage algorithm using tabu search [Lewis, 2013]

Build a graph where nodes represent groups, edges represent relations, and colors represent tables.

- Stage 1: Color nodes to find an initial feasible solution
- Stage 2: Improve this feasible solution with a tabu search

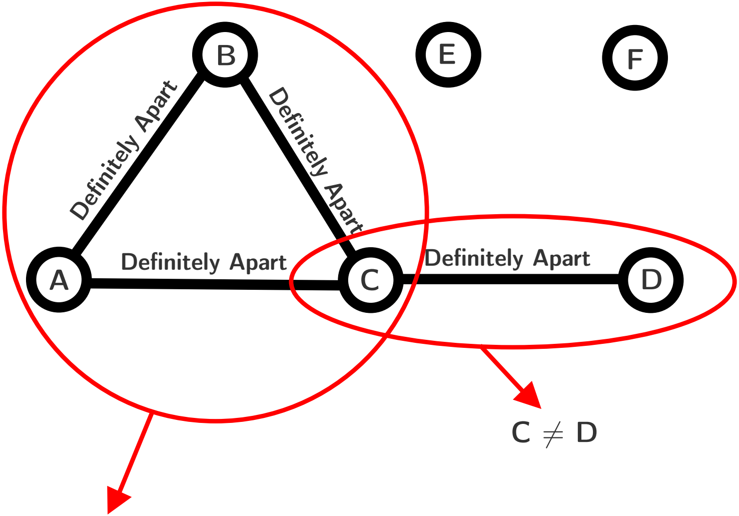
Outline

1. The Wedding Seating Problem

2. Existing Methods

3. Constraint Programming Model

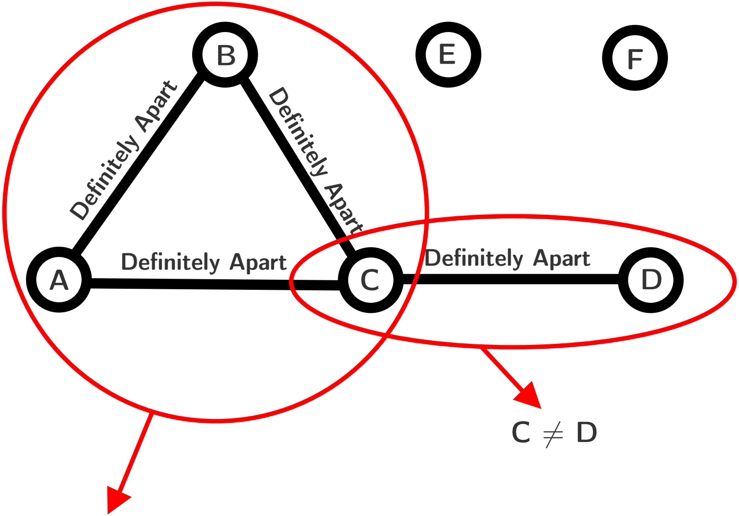
Constraint Programming Model



$C \neq D$

alldifferent(A, B, C)

Constraint Programming Model



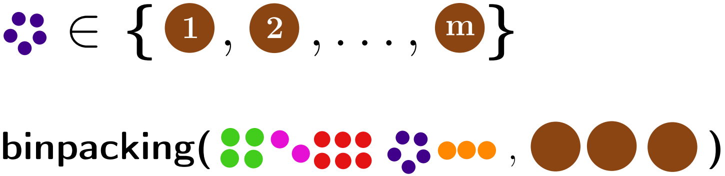
$C \neq D$

alldifferent(A, B, C) → Needs 3+ tables

Constraint Programming Model

⁘ ∈ {①, ②, …, ⓜ}

Constraint Programming Model



Constraint Programming Model

$⁙ \in \{①, ②, \ldots, ⓜ\}$

binpacking(∷ ⁚ ⠿ ⁙ ••• , ● ● ● **)**

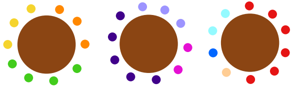
alldifferent(☹ **)**

Constraint Programming Model

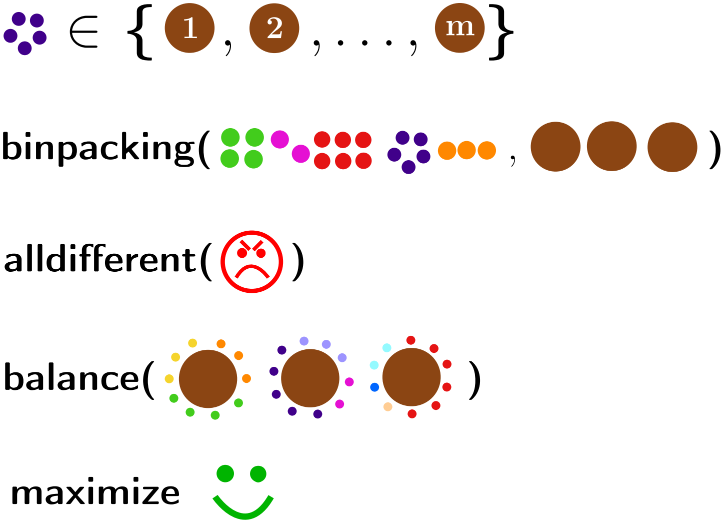
$\in \{$ ❶, ❷, …, ⓜ $\}$

binpacking(,)

alldifferent( )

balance()

Constraint Programming Model



$\in \{ 1, 2, \ldots, m \}$

binpacking(,)

alldifferent( )

balance( )

maximize

Constraint Programming Model

Branching Heuristic

Similar to *best fit decreasing*:

Constraint Programming Model

Branching Heuristic

Similar to *best fit decreasing*:

1. Pick the largest group yet unassigned

Constraint Programming Model

Branching Heuristic

Similar to *best fit decreasing*:

1. Pick the largest group yet unassigned
2. Sit that group at the best possible table

Constraint Programming Model

Branching Heuristic

Similar to *best fit decreasing*:

1. Pick the largest group yet unassigned
2. Sit that group at the best possible table
3. If no table has enough room, start a new table