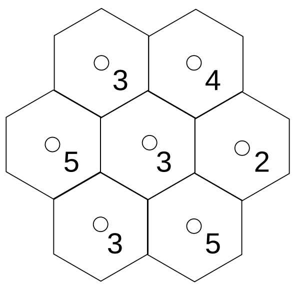
Feasible Cellular Frequency Assignment Using Constraint Programming Abstractions

Joachim P. Walser
Programming Systems Lab
Universität des Saarlandes
walser@ps.uni-sb.de

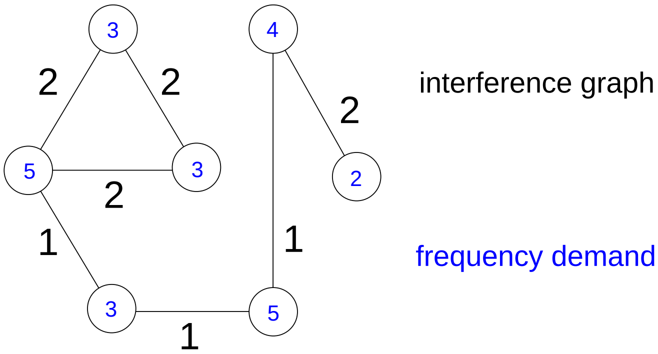
Feasible frequency assignment



cells with frequency demands

Goal: Assign frequencies to each cell, such that no interference occurs between frequencies

Problem formulation

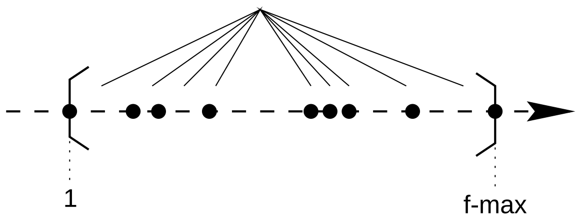


Given: Frequency demands (n_i),
interference constraints (d_{ij}): $|f_i - f_j| \geq d_{ij}$

Goal: Find assignment $(f_1, \ldots, f_M)$ that violates no interference-constraints

Optimization objectives

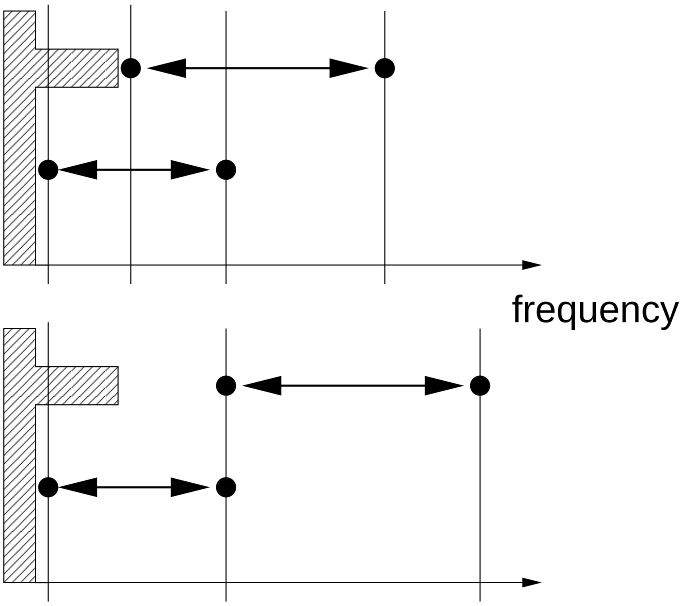
number of different frequencies



frequency span (largest frequency)

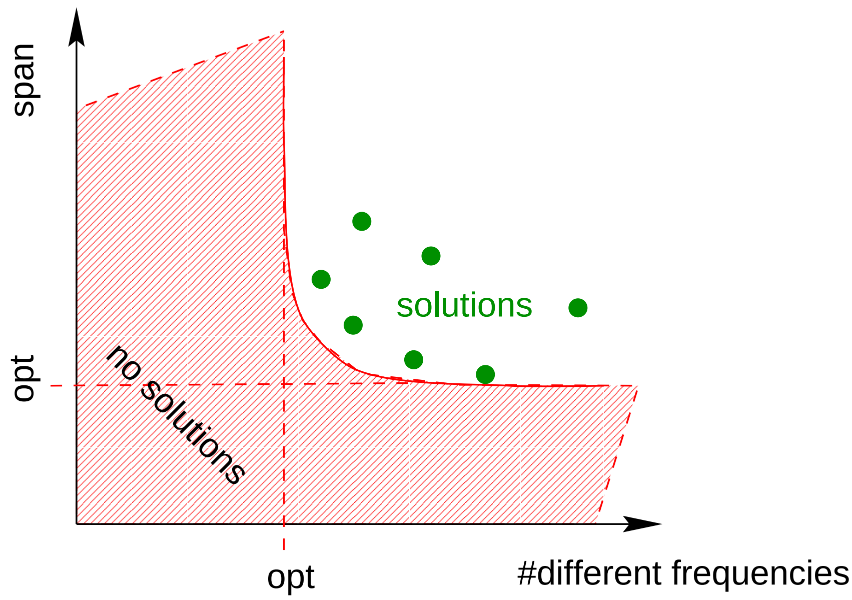
Both criteria studied in the literature
(Metzger, 1970) – (Tiourine et. al, 1995)

Tradeoff between criteria



Number of frequencies vs. span

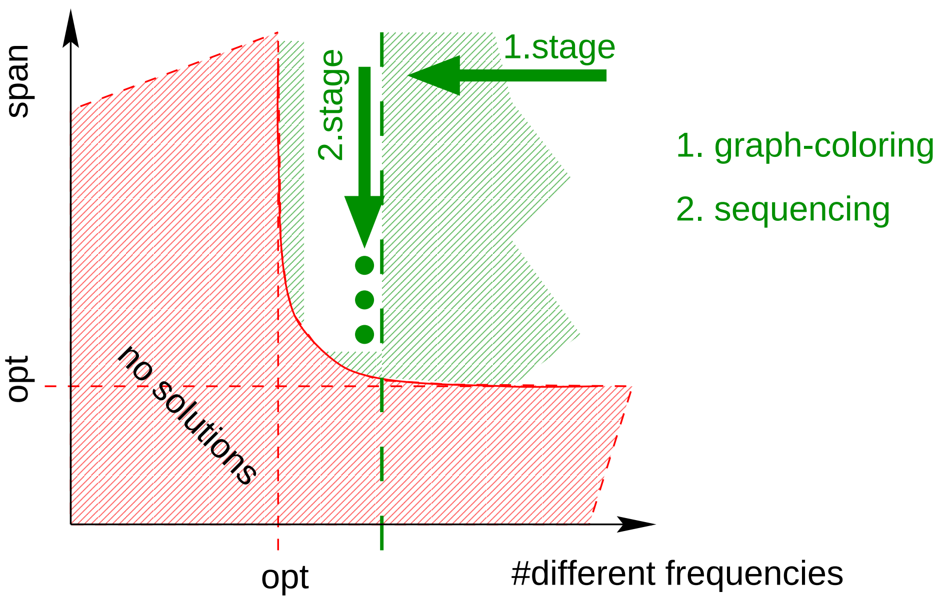
Two-criteria optimization



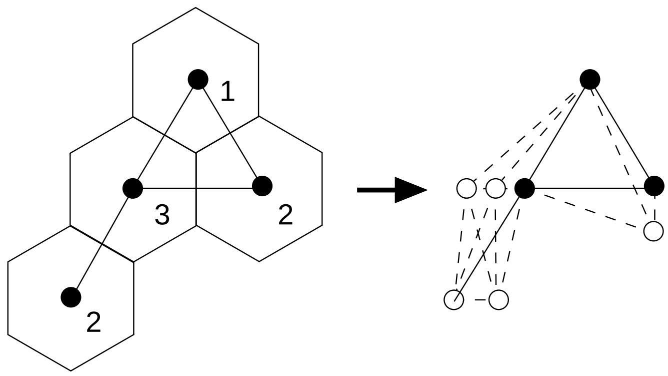
Optimization of both criteria

Can we **factor** the problem
in order to optimize both?

Factoring: Graph-coloring & sequencing

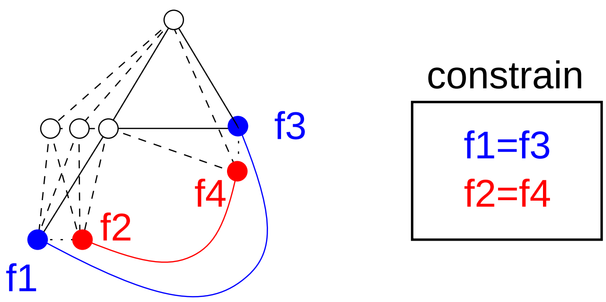


Close relation of FAP and graph coloring



The minimal number of colors needed
is the minimal number of different frequencies
(Metzger, 1970)

First stage: Restrict solutions by coloring



Instead of assigning frequencies directly:
Only use colors to **restrict solution space**

Second stage: Assign frequencies, compress span

- Idea from the scheduling domain: task-ordering
- Cell-sequencing: A **sequence** of cells

$$(c_1, c_2, \ldots, c_M)$$

  induces a frequency assignment.

- Use FAP heuristics to generate sequences
- **Goal:**
  Find a **sequence** that minimizes the **span**

Experimental results

	channels	2-stage (#freqs,span)	% above lower bound #freqs	span
FAP1	148	(95,145)	0%	2%
FAP2	165	(74,156)	7%	7%
FAP3	150	(86,137)	4%	5%
FAP4	161	(91,159)	1%	9%
FAP5	222	(122,274)	17%	0%
FAP6	222	(222,222)	0%	0%

FAP1 from ILOG user's manual, FAP6 from [Gamst 88]
FAP2-5, randomized problems, each 25 cells

- approximate two-criteria optimization
- cell sequencing alone: 1–2% within lower bounds
- runtime: approx. 40s for 40 LDS probes

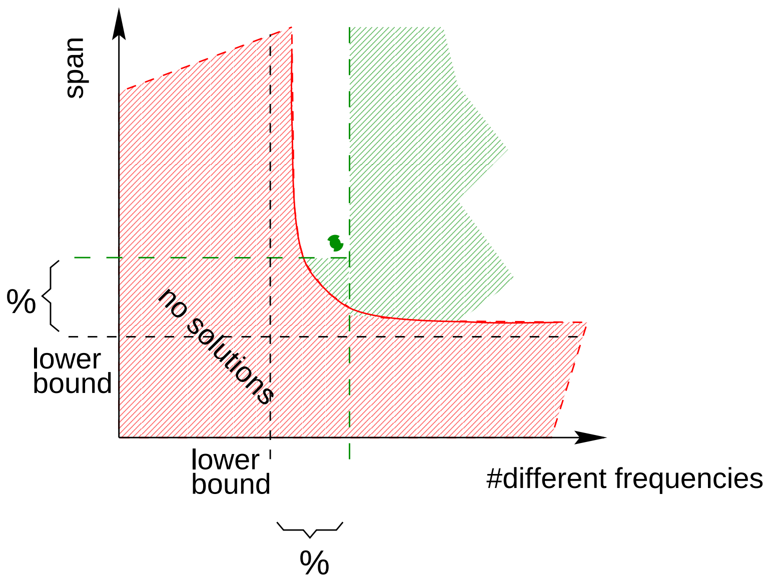
Constraint programming abstractions

- Graph-coloring using standard CP techniques (Dincbas, Simonis, Hentenryck, 1990)
- Abstractions:
  - Stage 1: finte domain propagation, ordering heuristics, branch-and-bound search
  - Stage 2: limited discrepancy search (Harvey&Ginsberg, 1995)
- Implementation in **Oz**

Conclusions

- Two-criteria optimization for feasible frequency assignment
- Factorization of problem into two stages:
  - Stage 1: Restriction of solution space
  - Stage 2: Assignment
- Preliminary experiments suggest: bounding criterion 1 still allows for near-optimal results for criterion 2

Solution quality

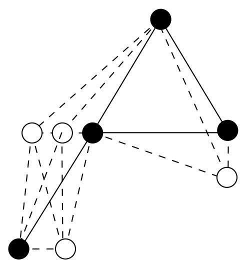


future work

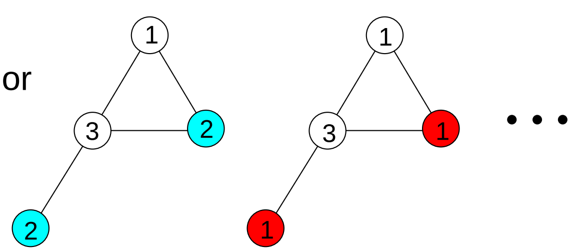
- Experiments on more realistic instances
- Empirical comparison with existing approaches
- Examine theoretical properties of **ICC**
- Extend **ICC** by search
- Improve heuristics in second stage

Exploit the structure of the cell graph

channel graph



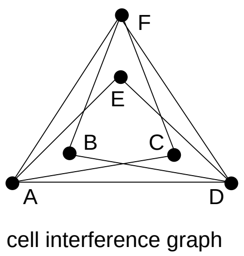
iteratively color
cell graph

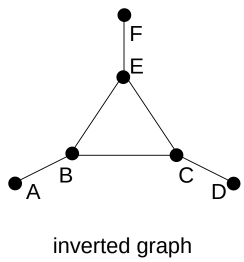


Approximation strategy: Iterative Clique Covering (ICC)

stage 1 – iterative clique covering (ICC)

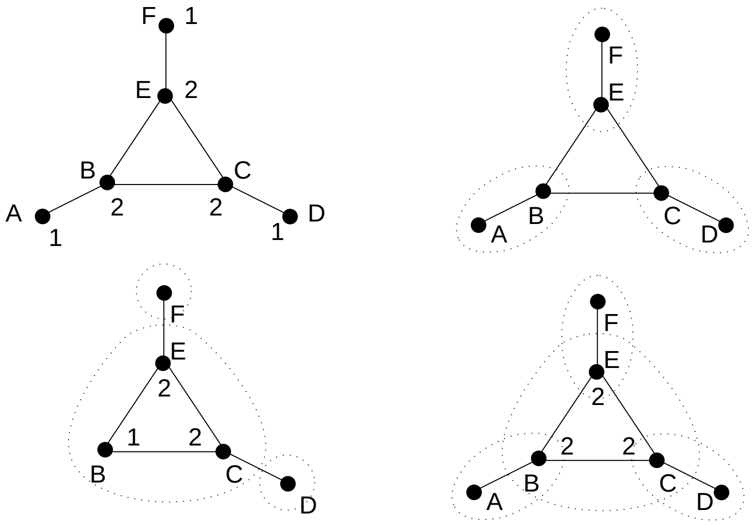
- graph coloring = clique-covering of inverted graph



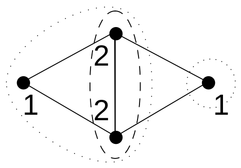


- cells without interference = independent set
  = clique in inverted graph
- channels from a largest clique (coloring): constrain to be equal

iterative clique covering, example

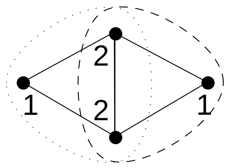


comparison with [Pennotti&Boorstyn '76]



3 colors

[Pennotti & Boorstyn '76]



2 colors

iterative clique covering