Optimization Applications

Helmut Simonis

Insight SFI Centre for Data Analytics, UCC, Ireland

A World Leading SFI Research Centre

Science Foundation Ireland For what's next

Insight SFI RESEARCH CENTRE FOR DATA ANALYTICS

HOST INSTITUTIONS

PARTNER INSTITUTIONS

FUNDED BY:

Take-Away Message

- Problem led research can be fun and rewarding
- Very different types of applications and domains
- From research prototypes to fielded systems
- Variety of tools and methods
- Provides structure to fundamental research

Problems Shown

- Flexible Flow-Shop with Transportation Times
  - J&J, studies future factory design
- Outpatient Waitlist Management
  - Now commercialized with Stimul.AI
- Elevator Maintenance Planning and Scheduling
  - Combination with simulation
- CAT Constraint Acquisition
  - Part of ASSISTANT EU project, aimed at scheduling
- Selection of other problem types
  - Only summary slide shown

Hybrid Flexible Flowshop with Transportation Times

Joint work with

- Michele Garraffa
- Barry O'Sullivan
- Eddie Armstrong (J&J Research, Limerick)

Hybrid Flexible Flowshop with Transportation Times

Real-World Problem

- Manufacturing Industry
- Move away from dedicated, high volume standard production
- Allow for increasing customization of product to customer needs
- Take advantage of more flexible, universal machines
- Decentralize production

Research Challenges

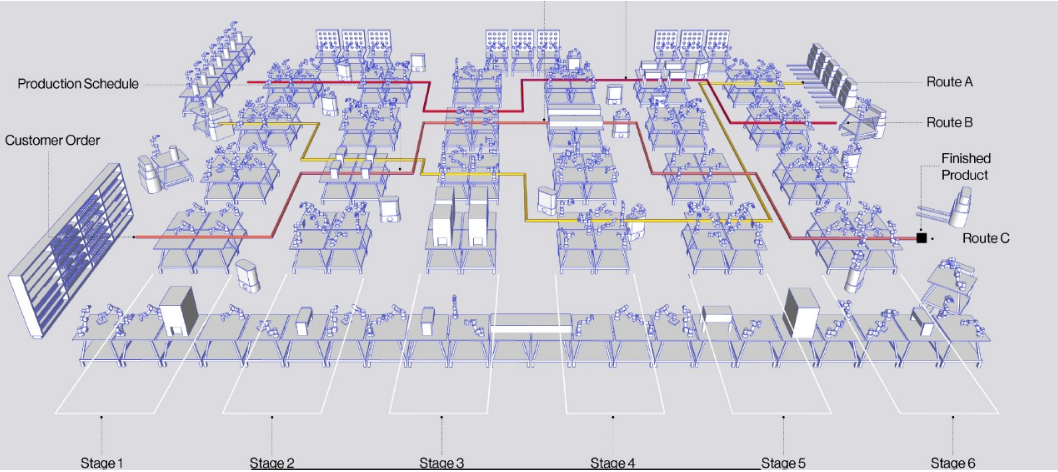
- Consider transport time in flowshop scheduling
- Choose appropriate technology to solve problem
- Study realistic scenarios at scale

A bit of Background

- Johnson&Johnson is a large multi-national company
  - Strong production and research presence in Ireland
  - Focus on consumer health, medical devices, pharmaceuticals
- Confirm
  - Irish National SFI Centre focussed on Manufacturing
  - Includes groups from multiple universities
  - Our focus is on analytics/optimization
  - Complements our work in the Insight SFI Centre for Data Analytics

Hybrid Flexible Flowshop with Transportation Times

Flexible Factory Structure (Including Transport Between Machines)



Main Elements of Problem

- Flow shop
  - Jobs run through production in the same sequence
- Hybrid
  - Multiple, identical machines available in each stage
- Flexible
  - Some production stages may be skipped for certain jobs
- Transportation Time
  - Time for transport between stage is significant, but not a resource limit
  - Many robots to handle transport tasks
  - Typical machine layout in lanes
- Objective makespan
  - Production not driven by due dates

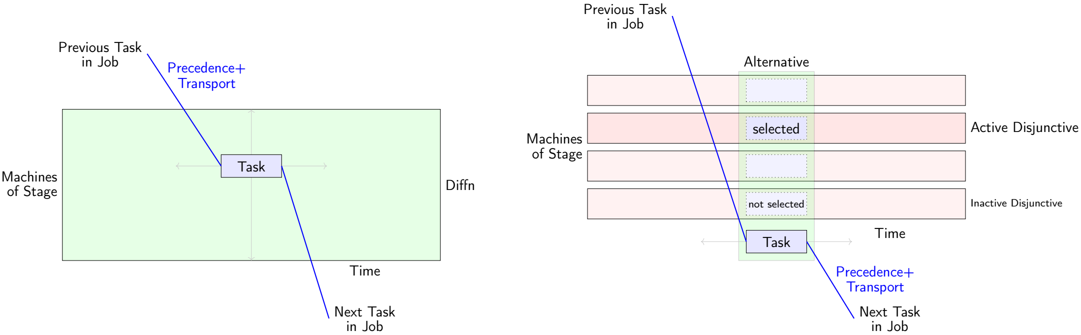
Objectives of Project

- Identify best tools to schedule new plant
  - Explore variety of different approaches and techniques
  - Do not just focus on your preferred solution method/solver
- Answer some design questions before committing to one approach
  - Is it better to have one or multiple facilities?
  - How far should the transport reach between lanes?
  - How can we exploit flexibility in new machines to offer better products?
    - Semi custom production
- Provide some quantitative comparison based on typical production data
  - Not currently for operational scheduling

CP Models

- Two main modelling alternatives
  - Diffn model to handle machine choice
  - Interval Task Variables with optional tasks on all alternative machines
- Transportation time handled by table constraint
  - Transportation between machines for tasks of the same job
  - Much simpler case than sequence dependent setup
- Precedences between tasks of jobs
- Objective Cmax makespan

CP Model Main Alternative



Dedicated MIP Models

- Four alternatives based on literature for hybrid flexible flowshop
- Adding transportation time grows model complexity
- Picked best alternative on small scale test cases
- None of the methods scale to expected problem sizes

Dispatch Rule/Local Search

- To provide baseline result/ initial upper bound
- Schedule jobs in random order
- Assign each task to first available machine
- Dispatch Rule
  - Explore different initial job permutations
- Local Search
  - Also explore swaps/insertion of jobs in sequence
- Written in Java

Implementations

- MiniZinc, Chuffed, free search
  - Diffn constraint
- MiniZinc, Chuffed, priority search
- MiniZinc (interval task variables)
- MiniZinc, Cplex
- MIP model, Cplex
- CP Optimizer (interval task variables, black box search)
- SICStus Prolog diffn model, custom search)

Hybrid Flexible Flowshop with Transportation Times

Instance Generator

- Produce sequence of test cases with increasing number of jobs
  - 20, 25, 30, 40, 50, 100, 200, 300, 400 jobs
  - 25 instances per problem size
- Parameters chosen to reflect real world factory
  - 8 stages, 10 machines/stage, some skipped stages
  - Discrete power law for job types
    - A few products are quite common, many are rare in order set
  - Transport times based on lanes
- Instances available on line
  - `https://zenodo.org/record/5168966`

Experimental Setup

- Experiments run on single core of Windows 10 laptop
- Timeout 300s
- Upper bound provided by 10s of Local Search
- Best lower bound provided to stop search for optimal solutions
  - Optimal solutions found for many smaller (20 30 jobs) instances

Cmax Results with Different Models (average over 25 instances, 300s timeout)

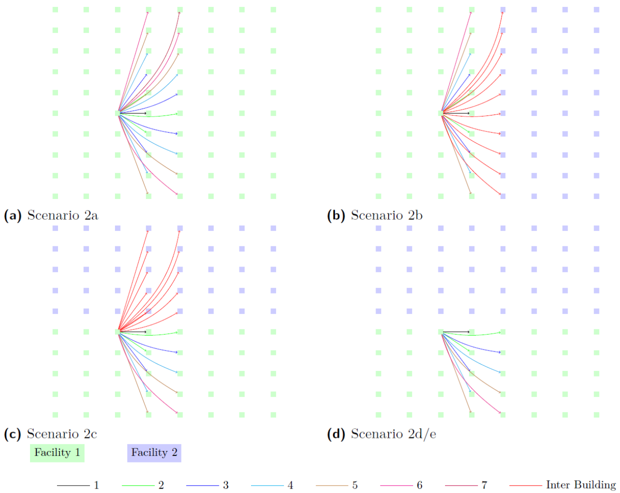
Size	Lower Bound	Upper Bound	CP Opt	Chuffed Free	Chuffed Priority	Dispatch Rule	Local Search	SICStus
20	61.88	63.56	62.72	63.48	63.04	63.28	63.20	62.72
25	62.84	65.96	64.24	-	64.76	65.20	64.84	64.16
30	64.12	70.24	66.68	-	68.44	69.16	68.24	66.84
40	65.32	77.36	72.56	-	75.40	76.08	75.28	73.28
50	67.24	84.52	78.40	-	82.24	83.16	82.24	79.40
100	94.72	120.12	115.16	-	116.96	118.28	118.92	113.04
200	153.08	185.16	180.48	-	181.32	182.80	184.76	176.72
300	214.96	249.12	248.96	-	248.76	246.96	248.88	240.96
400	275.36	311.60	311.28	-	-	308.76	311.40	303.16

Comments

- CP Optimizer and SICStus perform best
  - CP Optimizer better for small/medium instances
  - SICStus does scale better
  - Note: SICStus uses hand made search routine
  - Chuffed free search does not scale at all
    - Very poor improvements on makespan
  - Chuffed priority search: good initial solutions only
- Dispatch Rule and Local Search perform quite well
  - Further development potential
- MIP does not work at all
  - Limited to smaller instances not shown here

Hybrid Flexible Flowshop with Transportation Times

Four Layout Alternatives (One or Two Locations)



Insight

Five Scenarios Tested

2a Single facility organized in lanes

2b Two facilities in sequence (sequential for all jobs)

2c Two facilities in parallel with transport between facilities allowed

2d Two facilities in parallel, transport only within each facility

2e Two factories in parallel, with 80% of jobs preassigned to a factory

Scenario Comparison

Solver	Size	Scenario 2a	2b	2c	2d	2e
SICStus	200	176.84	184.84	178.28	180.52	180.48
% over Best		0.00	4.52	0.81	2.08	
CPOptimizer	200	184.40	190.92	186.00	183.52	183.52
% over Best		1.23	4.81	2.11	0.75	
Dispatch	200	182.76	190.44	184.28	184.60	184.64
% over Best		0.00	4.20	0.83	1.01	
Local Search	200	184.68	192.24	185.76	186.08	185.96
% over Best		0.13	4.23	0.72	0.89	

Hybrid Flexible Flowshop with Transportation Times

Summary

- New variant of known scheduling problem
  - Arising from flexible new factory design
  - Transportation between machines/locations important element of schedule
  - Good solutions are obtained with CP for large problem instances
  - Not all CP models achieve the same solution quality
  - MIP results weak
  - Remaining, open gap between best lower bound and best solution found
- Scheduling model used for factory design study
  - Which layout gives the best overall results?
  - Explores four design alternatives

Results Scale to Hundreds of Jobs (shown: SICStus 1000 jobs, 80 machines)

Outpatient Waitlist Management

Joint work with...

- Mike O'Keeffe
- Adrian O'Leary
- Barry O'Sullivan
- At Insight Centre for Data Analytics, University College Cork

Outpatient Waitlist Management

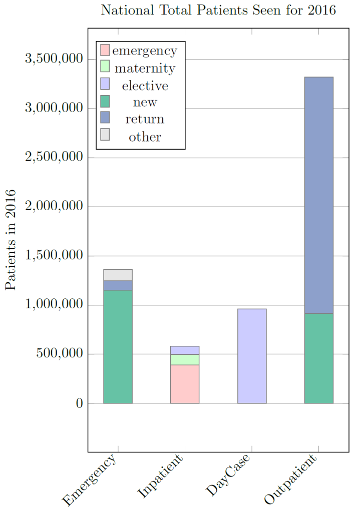
Real-World Problem

- Healthcare in Ireland
- Wait times for patients are out of control, even before Covid-19
- Longer wait times, poorer patient outcomes
- Critical to understand where to invest
- Currently: no tools to understand how changes affect performance

Research Challenges

- How to model hospital environment, many independent actors
- Deal with uncertain demand, and uncertain outcomes
- Understand where capacity is lost/not used

Hospital Services Overview



Data: HSE Management Data Report, Dec 2016

Outpatient Types

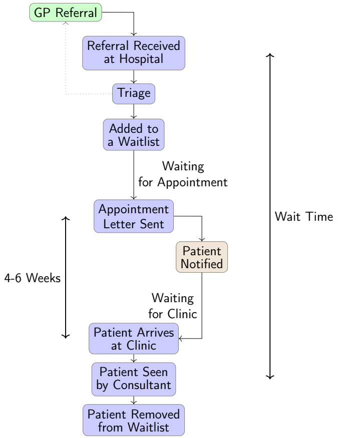
Rapid access seen within 14 days

Urgent seen within 28 days

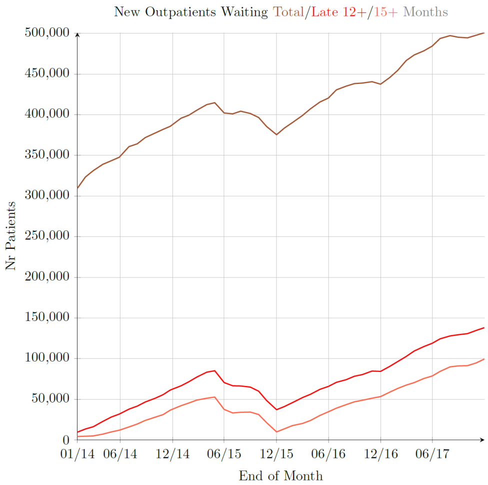
Soon seen within 3 months

Routine seen within 12 months (13 weeks, 15 months, 18 months?)

Outpatient Waitlist Management Process (Simplified)

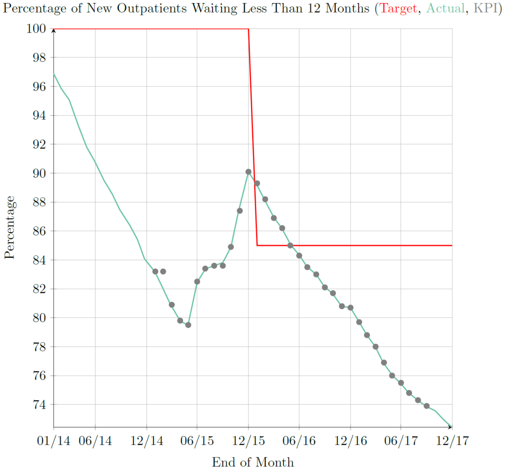


The Bad News



Data: NTPF

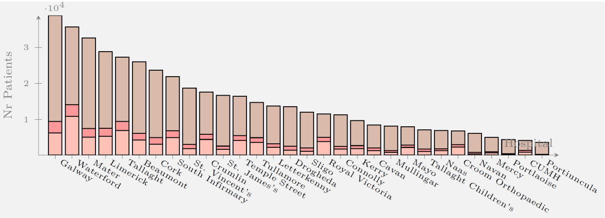
KPI: Waiting Time Percentage



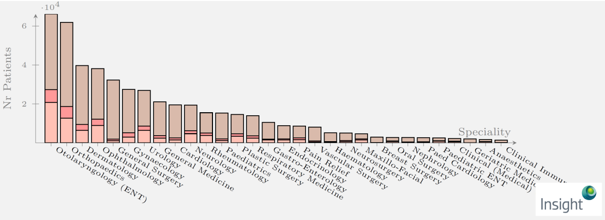
Data: HSE

A Near Universal Problem in Ireland

By Hospital



By Speciality



Outpatient Waitlist Management

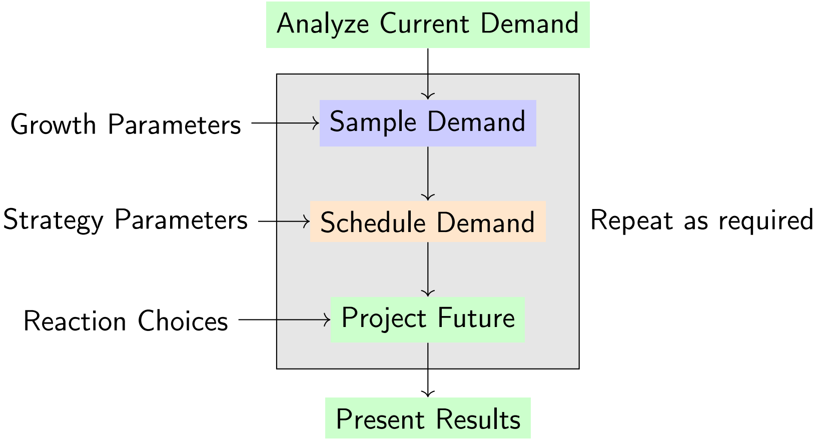
Our Brief

- Concentrate on Outpatients
- Develop strategy for appointment decision making
- What-if tool to understand the impact of decisions
- Support current stakeholders
- Not: Build automated appointment scheduling tool

The Appointment Conundrum

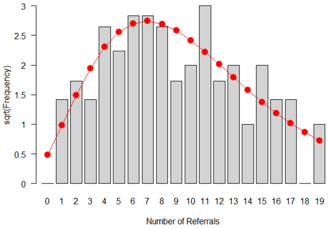
- We have to give “routine” appointment before knowing “urgent” demand
- There is limited capacity
- No overtime allowed (Croke Park agreement)
- How much capacity to set aside for urgent cases?
- How much overbooking is possible?

Methodology

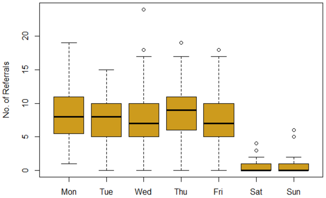


Demand Data (Not Public)

Received
Per Day

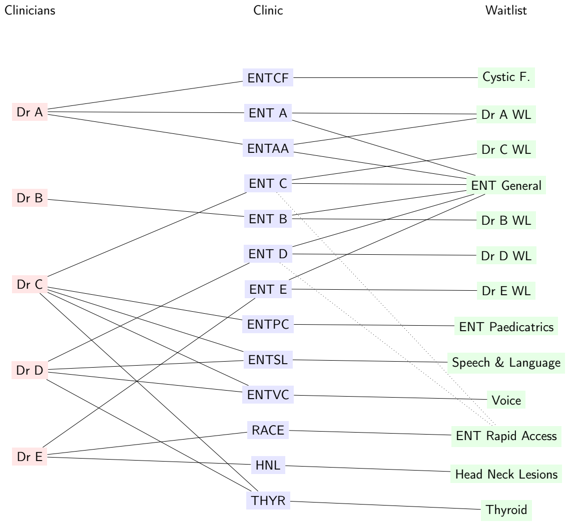


Received
Per Day of Week

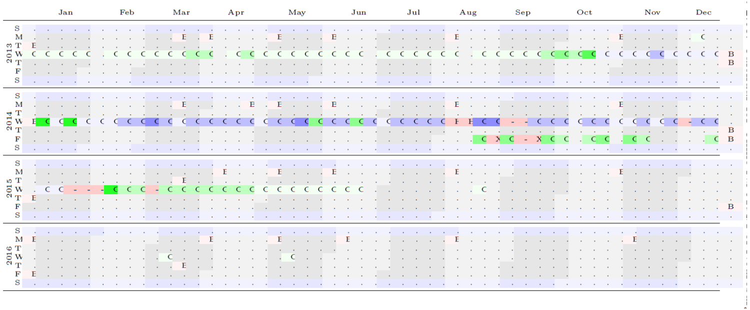


- Fitting distributions
  - Poisson, not good fit
  - Negative Binomial
- Limited Seasonality (unlike Emergency Department)

Waitlist/Clinic Model



Learning Capacity from Historical Data

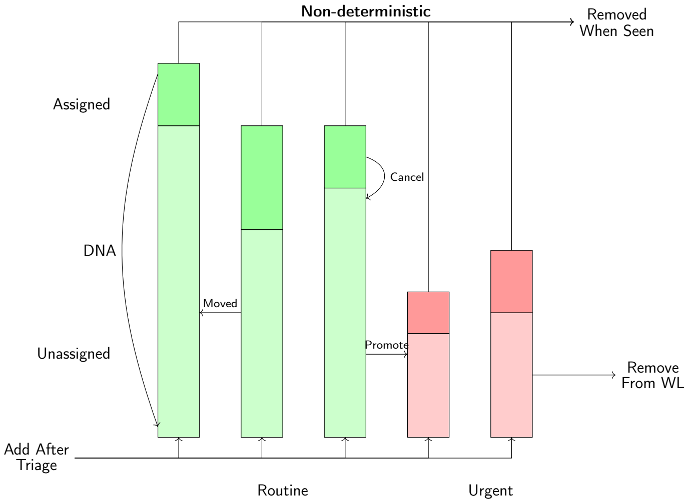


- Repeat frequency
- Capacity
- Cancellation frequency
- Replacement clinics

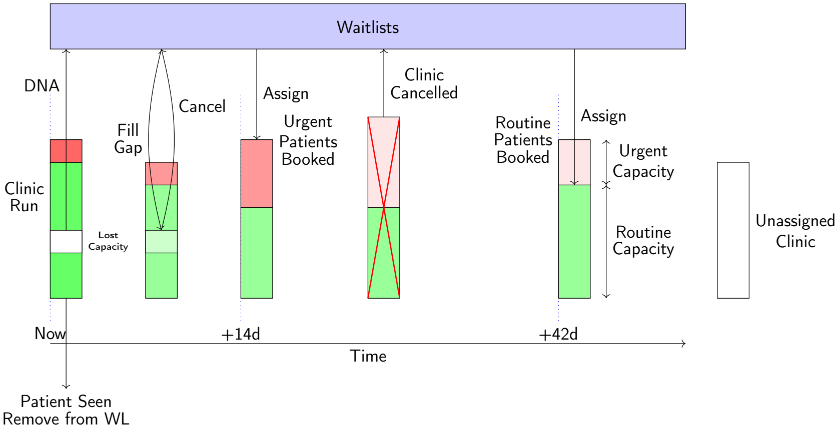
Optimization Problem

- Assign waiting patients to slots in clinics
- Use appropriate clinic for given patient
- Make appointments k_p days in advance
- Free and reuse slots when patients cancel
- Reschedule patients when clinic cancelled
- Do not change appointments otherwise
- Reserve u slots for urgent cases
- Solved for each day

Waitlist Actions



Clinic Allocation



Outpatient Waitlist Management

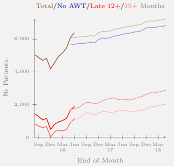
Introduction

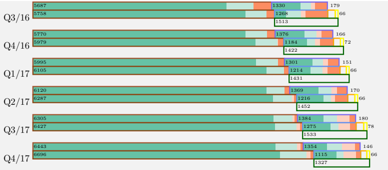
Solution Approach

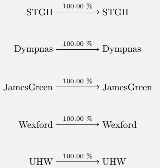
Results

Summary

Baseline Analysis, Management View

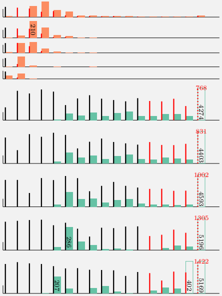






List/Category	Date	Patients Waiting	Breaching on Date	Patients in Breach	Avg	90%	Max
All Urgent	One Month Ago	596	464	584	142.84	250	515
	Now	432↘	302↘	428↘	90.72↘	128↘	228↘
	In One Month	352↘	227↘	343↘	73.50↘	115↘	228–
	In Six Months	150↘	38↘	147↘	45.45↘	63↘	228–
	In One Year	109↘	11↘	51↘	33.19↘	56↘	70↘
Routine	One Month Ago	5080	1635	4514	682.66	1215	1512
	Now	5255↗	1737↗	4613↗	686.44↗	1219↗	1554↗
	In One Month	5390↗	1801↗	4687↗	696.07↗	1226↗	1623↗
	In Six Months	5845↗	2090↗	5315↗	743.39↗	1293↗	1718↗
	In One Year	6196↗	2317↗	5737↗	787.76↗	1505↗	1835↗

Waiting Time Distribution

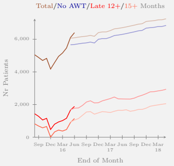


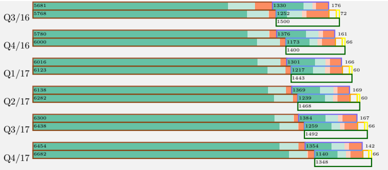
OWL2 (baseline): Thursday 4th August, 2016, at 22:58

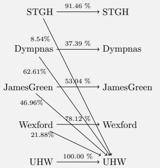
Insight

Insight

Scenario: Balance Patients Between Hospitals

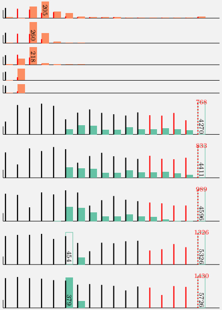






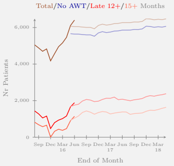
List/Category	Date	Patients Waiting	Breaching on Date	Patients in Breach	Avg	90%	Max
All Urgent	One Month Ago	596	464	584	140.17	250	515
	Now	412↘	281↘	408↘	84.85↘	106↘	200↘
	In One Month	339↘	217↘	330↘	69.21↘	92↘	200–
	In Six Months	146↘	36↘	143↘	36.86↘	42↘	46↘
	In One Year	112↘	15↘	109↘	33.42↘	37↘	41↘
Routine	One Month Ago	5080	1635	4511	608.66	718	796
	Now	5269↗	1758↗	4629↗	609.15↗	743↗	797↗
	In One Month	5390↗	1784↗	4661↗	612.68↗	757↗	798↗
	In Six Months	5870↗	2111↗	5326↗	632.37↗	781↗	803↗
	In One Year	6188↗	2325↗	5726↗	632.10↘	780↘	803–

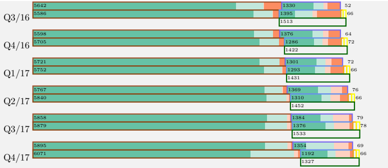
Waiting Time Distribution

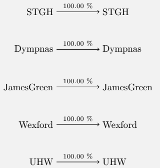


OWL2 (balanced; ENTMH can serve all hospitals with overall capacity): Thursday 4th August, 2016, at 17:02

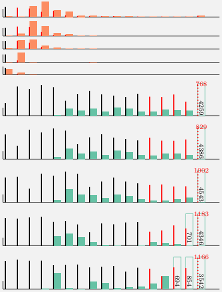
Scenario: Reduce DNA (Did not attend) to 5%







List/Category	Date	Patients Waiting	Breaching on Date	Patients in Breach	Avg	90%	Max	Waiting Time Distribution
All Urgent	One Month Ago	596	464	584	142.64	250	515	
	Now	404↘	300↘	401↘	91.83↘	128↘	228↘	
	In One Month	301↘	199↘	293↘	74.12↘	120↘	228–	
	In Six Months	130↘	27↘	127↘	42.88↘	56↘	228–	
	In One Year	92↘	8↘	21↘	30.87↘	47↘	64↘	
Routine	One Month Ago	5080	1635	4509	655.06	1154	1473	
	Now	5238↗	1735↗	4606↗	658.91↗	1161↗	1490↗	
	In One Month	5346↗	1796↗	4678↗	668.77↗	1166↗	1540↗	
	In Six Months	5591↗	1968↗	5145↗	714.68↗	1277↗	1669↗	
	In One Year	5766↗	2061↗	5314↗	757.54↗	1471↗	1779↗	

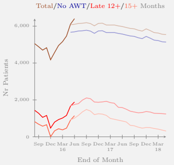


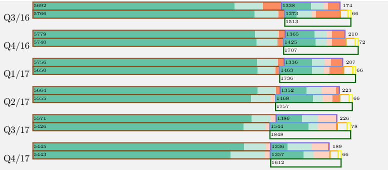
OWL2 (DNA 5 percent for routine patients): Thursday 4th August, 2016, at 16:59

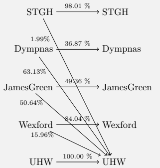
Insight

Insight

Scenario: Add Capacity

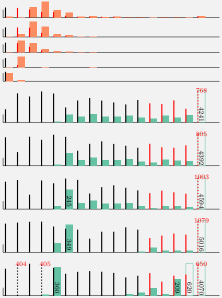






List/Category	Date	Patients Waiting	Breaching on Date	Patients in Breach	Avg	90%	Max
All Urgent	One Month Ago	596	464	583	141.70	248	513
	Now	424↘	301↘	421↘	90.75↘	130↘	231↘
	In One Month	335↘	210↘	330↘	71.78↘	123↘	231–
	In Six Months	146↘	29↘	140↘	38.77↘	36↘	231–
	In One Year	117↘	0↘	0↘	26.30↘	30↘	30↘
Routine	One Month Ago	5080	1644	4508	540.10	654	672
	Now	5268↗	1734↗	4624↗	534.89↘	653↘	672–
	In One Month	5409↗	1795↗	4701↗	531.09↘	653–	672–
	In Six Months	5610↗	1867↗	5134↗	514.78↘	629↘	672–
	In One Year	5454↘	1573↘	5004↘	474.74↘	547↘	566↘

Waiting Time Distribution



OWL2 (25 New Patients per Week from 1/10/2016): Thursday 4th August, 2016, at 17:06

Insight

Outpatient Waitlist Management

Status

- Initially developed with industrial partner
- Tested and evaluated at hospital
- Actual data used, but manual feed
- Startup company Stimul.AI to commercialize solution

STIMUL.AI

Summary

- Presented case study from Irish health system
- Strategy for outpatient appointments
- Mix of analytics, simulation, and optimization
- Nation-wide analysis of available data
- What-if tool for selected departments

Joint work with...

- Mark Antunes, Vincent Armant, Kenneth N. Brown, Gabriel G. Castane, Daniel Desmond, Guillaume Escamocher, Michele Garraffa, Anne-Marie George, Diarmuid Grimes, Mike O'Keefe, Yiqing Lin, Barry O'Sullivan, Cemalettin Ozturk, Luis Quesada, Mohamed Siala, Helmut Simonis and Nic Wilson

Elevator Maintenance Planning and Scheduling

Real World Problem

- Manufacturing Industry, after sales support
- Maintenance is crucial for safety/availability of product
- Preventive/Predictive/Reactive Maintenance influence each other
- How to organize service, what to do?

Research Challenge

- How to plan/schedule if events interrupt planned work
- How to use predictive maintenance to avoid problems before they occur
- What is the right problem decomposition?

Travelling Repair Person (TRP)

- Providing service for devices at customer premises
- Planned preventive maintenance and testing, regular visits
- Technicians travel to multiple, but few customers per day
- Unplanned repair work after faults, response-time critical
- Service times quite variable
- Impact of skills and local knowledge

Why is this important? (1)

Law and Crime

Lift firm Otis fined HK$320,000 over Hong Kong mall escalator accident that injured 18

Company, which pleaded guilty to four summonses, could have discovered safety issues with escalator three months before malfunction, court told

Jasmine Siu
Published: 8:15pm, 9 Mar, 2018

Why is this important? (2)

BIG STORY 10 APRIL 5, 2016 / 6:40 AM / 3 YEARS AGO

Schindler sells Japanese business to Otis after accident

2 MIN READ

ZURICH (Reuters) - Elevator maker Schindler is selling its Japanese business to United Technologies' Otis unit after its new installations in the country were halted following a 2006 accident.

Source: REUTERS

Why is this important? (3)

Elevator at one of Chicago's tallest skyscrapers plunges 84 floors after hoist rope breaks

NOVEMBER 19, 2018 / 10:47 AM / CBS NEWS

Source: By Chris6d - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=78201640

Elevator Maintenance Planning and Scheduling

High-level View

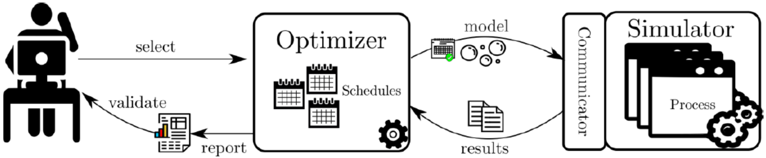


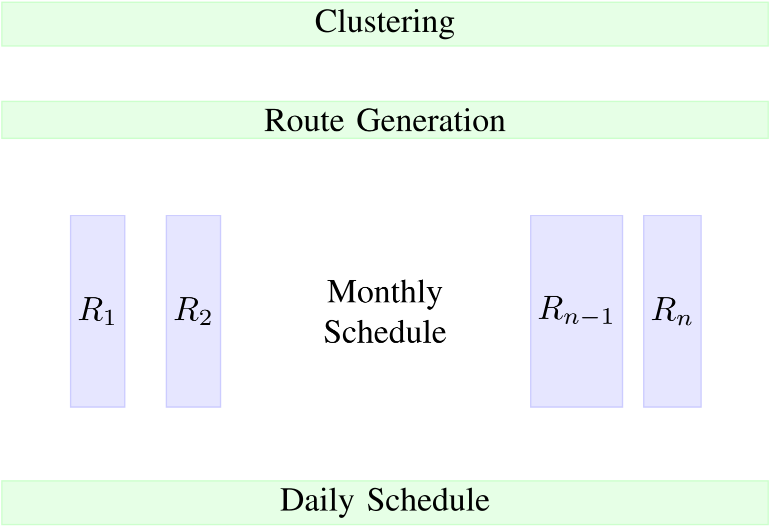
Figure 1 High level overview of the framework

- Optimizer deals with planning, load balancing, efficient schedules
- Simulator explores how to react to changes
- Simulator also provides one result as assumed reality

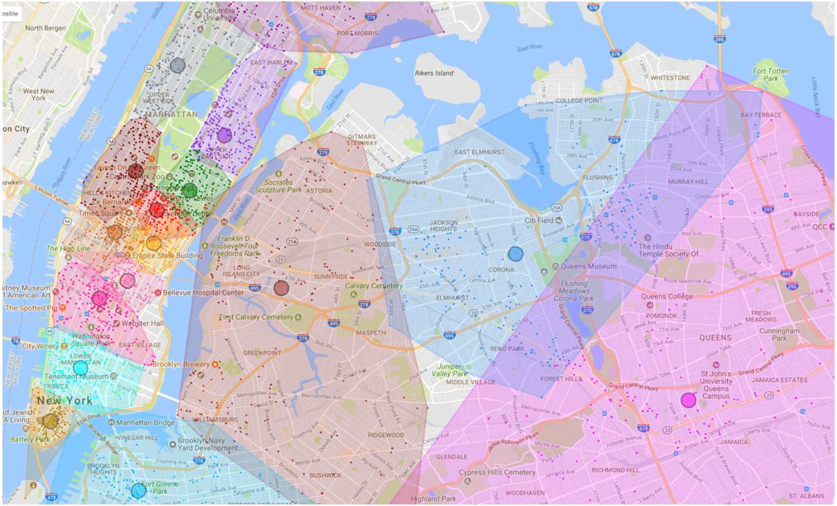
Optimizer Design

- Infeasible to build homogenuous model for complete problem
- Added business process constraint
  - Technicians should be responsible for "their" buildings
  - Improves service quality
  - Customers see familiar face
- All work in one building should be performed by the same engineer, if possible
- Engineers should be assigned compact areas of work
- Balanced workload within the same depot

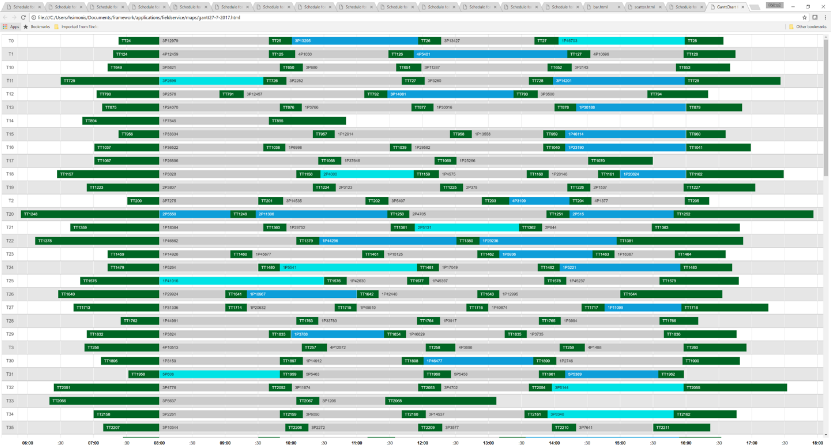
Optimizer Decomposition



Clustering and Depot Assignment



Scheduling: One Day of Monthly Plan



Insight

Methods Used

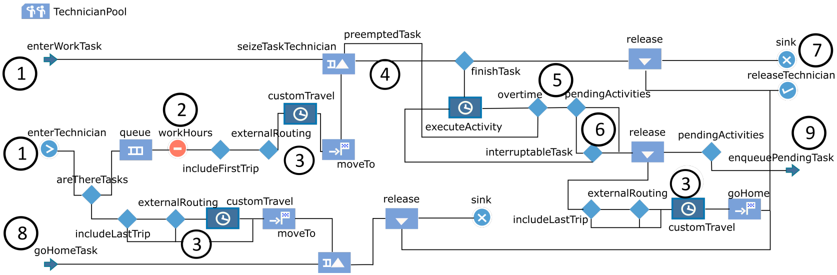
Clustering Connected components on generated graph

Routing Which places to visit in one trip

- Core MIP Model
- Iterative MIP inside Clustering
- Two stage grouping of locations to reduce expected travel
- Local Search

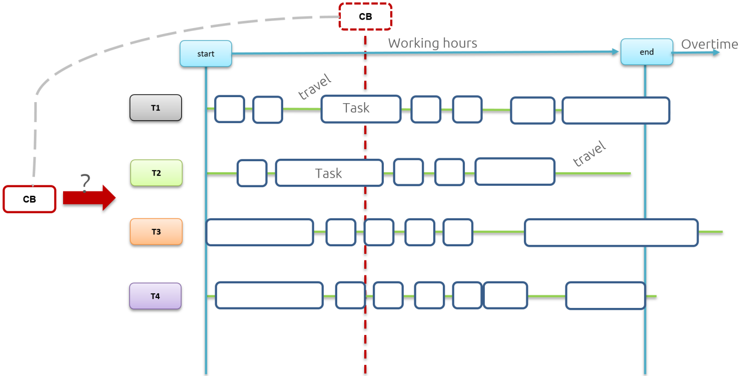
Scheduling Dynamic Programming and Set Partitioning

Simulator Process Modelling



Insight

Dealing with Unplanned Callbacks



- Who is dealing with the callback?
- How to adjust the schedule after callback?

Insight

Elevator Maintenance Planning and Scheduling

Use Cases

- Compare variants of problem to understand impact of changes
- Examples
  - Where to place depots and their area?
  - How many technicians are needed in which depots?
  - Should technicians do both planned and unplanned work?
  - When is overtime the better choice?

Scenario Evaluation: KPI Comparison

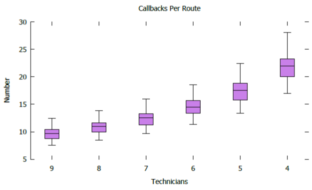


Figure 6 Callbacks per route (technician)

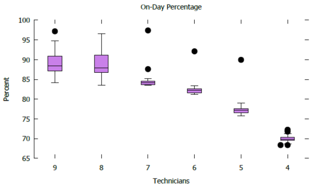


Figure 7 Percentage of tasks performed per tasks

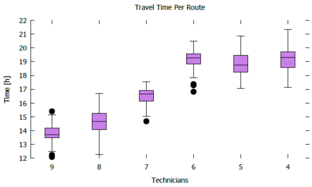


Figure 9 Travel time per technician

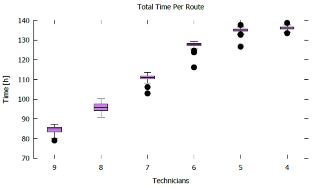
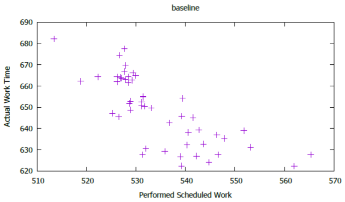
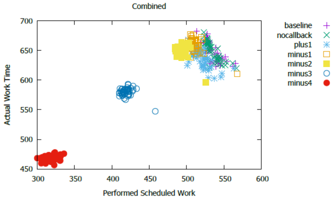


Figure 8 Total time per technician

Insight

Scenario Evaluation: Qualitative Differences





- On left, each point shows the outcome of one month of optimization+simulation
- On right, compare outcomes for different scenarios, clear clustering of results

Elevator Maintenance Planning and Scheduling

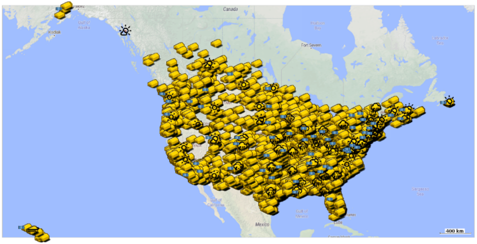
Challenges: Data

- We need company internal data to understand problem
- Problem for publication, for continued work
- Open data as alternatives
  - New York City
    - 76,000 elevators with locations
  - Toronto, ON
    - 40,000 elevators
    - Inspection dates, outcomes
    - Accident and injury reports

Challenges: Scalability

Data source	Locations
Arby's	3347
Burger King	7269
Dairy Queen	5189
Dominos Pizza	3261
Dunking donuts	8134
KFC	5637
Little Caesars	4019
Mc Donald's	15474
Papa John's	3089
Pizza Hut	6672
Starbucks	11788
Subway	2213
Taco bell	6996
Wendy's	6140
Walmart	22212

- 136663 total locations (cleaned noisy data)
- 1 Unit per location
- 63 areas (US + Canada)
- Experiments definition:
  - 1, 10, 100, 1000, 10000 technicians per area



Challenges: Tools and Results

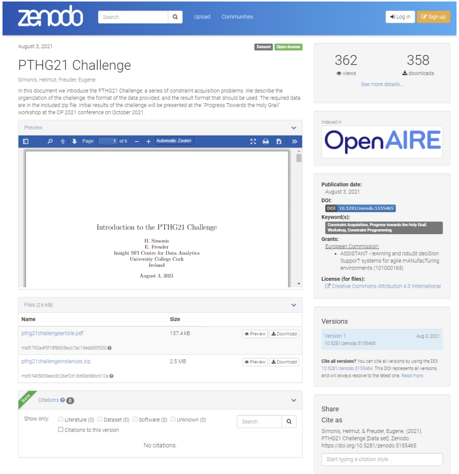
- We provide research and experimental software
- **Not** a solution
- End-user would like applicable results
- Managing expectations is important

Conclusions

- We presented the Travelling Repair Person Problem
- Important as an industrial problem
- Interesting as a research challenge
- We use combination of optimization and simulation to deal with novel properties of problem
- System transferred to customer in 2019

Based on previous work with

- N. Beldiceanu, IMT Atlantique
- M. Carlsson, SICS



PTHG21 Challenge co-organized with E. Freuder

CAT Constraint Acquisition

Industrial Problem

- Industry
- Optimization Projects are hard to manage
- Skilled experts are not easily found
- Communication between domain experts and programmers key
- Easy to miss key constraint during design

Research Challenge

- How can we make optimization more accessible?
- Lower barriers to entry
- or, make existing experts more productive
- Bridge gap between application domain and abstract optimization concepts

Take-Away Points

- Constraint Acquisition provides a way to learn constraint model from data
- Questions about use cases
- Transferable, executable models
- Common benchmark set: PTHG21 Challenge
- CAT System shows feasibility of approach

Background

- ASSISTANT project (EU H2020, ICT-38 project, `https://assistant-project.eu/`)
- Constraint Acquisition part of WP 4
- Making Constraint Acquisition relevant in real world, scheduling setting
- Based on case studies from Siemens Energy and Atlas Copco

CAT Constraint Acquisition

Introduction

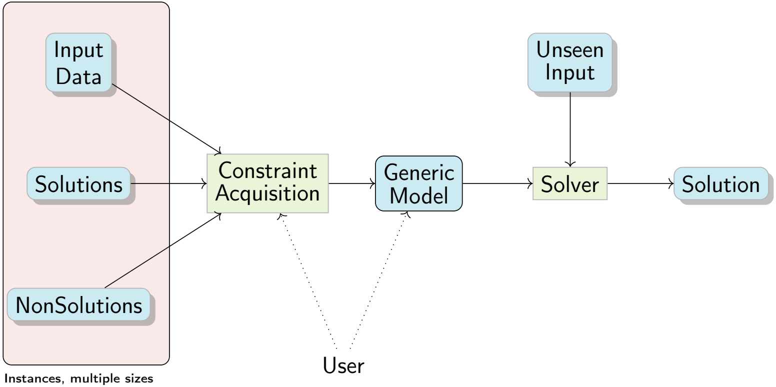
Solution Approach

Results

Constraint Acquisition - What is it?

- Learn Constraint Models from data
  - Given positive and negative examples ("Passive")
  - Asking questions to user ("Active")
- Useful to
  - Understand problem
  - Classify new examples as solutions or non-solutions
  - Use generated model to find solutions

Intended Use Case

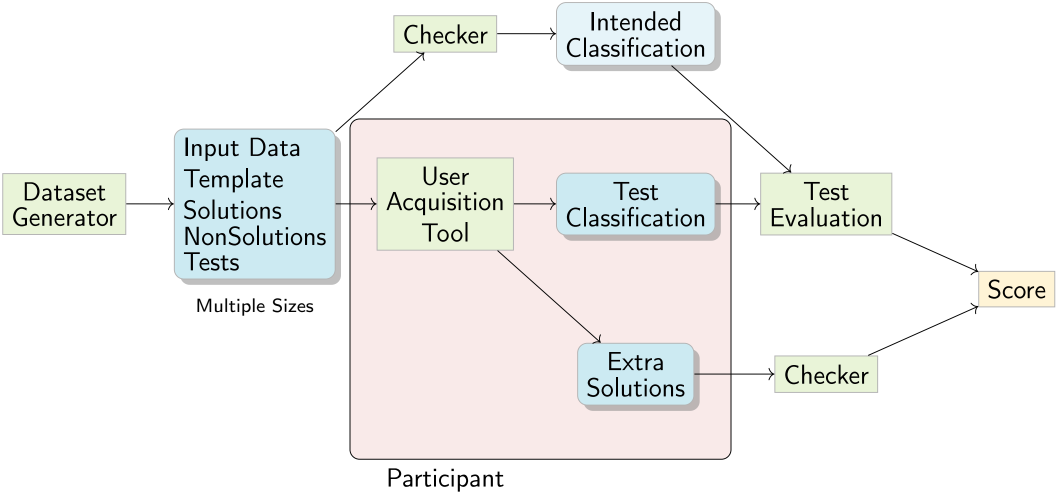


- Aim: demonstrate feasibility of Constraint Acquisition as an end-to-end tool chain

Properties

- Generated model must be transferable to new data
- Problem size varies from day to day
- Some variables of model may not be exposed in solution provided
  - Auxiliary variables not interesting to user
  - Individual cost elements
- Constraints are there for a reason
  - Due to structure of problem (think: Sudoku)
  - Due to input data (think: Graph Colouring)

PTHG21 Challenge



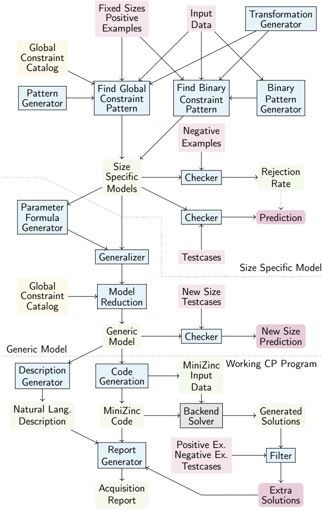
Challenge Problems (Set 1)

Type	Problem	Source	Features
1	Graph Coloring	ALICE, CHIP	graph as data, optimization
2	N-Queens	CSPlib 054	
3	Warehouse Location	CSPlib 034	cost matrix/vector as data, implicit cost variables
4	Golomb Ruler	CSPlib 006	implicit decision variables, optimization
5	Sudoku	Pre-assignment	pre-assignment as data, single solution
6	Sudoku	No pre-assignment	many solutions
7	Schur's Lemma	CSPlib 015	non-standard variable pattern, ternary constraint
8	All Interval	CSPlib 007	auxiliary variables
10	Magic Squares	CSPlib 019	implicit formula
11	Orthogonal Latin Squares	Euler	constraint on tuples
12	BIBD	CSPlib 028	3 parameters, implicit formulas, symmetry breaking
13	Costas Array	CSPlib 076	auxiliary variables, constraint on tuples
14	N-Queens variant	fairy chess piece	non-traditional attack
15	N-Queens variant		
16	N-Queens variant		

The CAT System

- Find global constraint models comparable to hand-built solutions
- Assumption: All needed information is either given as data or in regular structure
- Transferable, executable models for class of instances
- Built bottom up, based on test cases
- Should extend to more complex cases
  in more narrow application domain

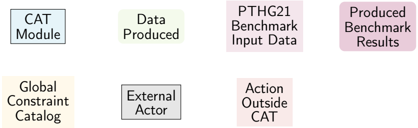
The CAT System Architecture



Acquire size specific model

Generalize acquired models

Produce working constraint program



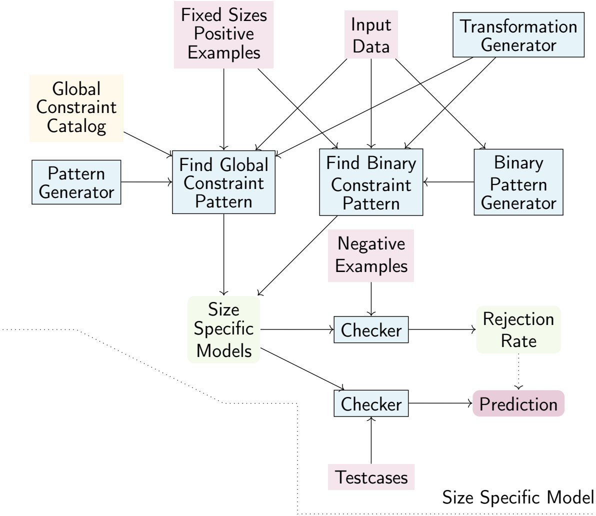
Example: Problem 12 (BIBD), Instance 0

```
"inputData": {
    "lambda": 2,
    "v": 6,
    "k": 3
},
```

Given Positive Sample:

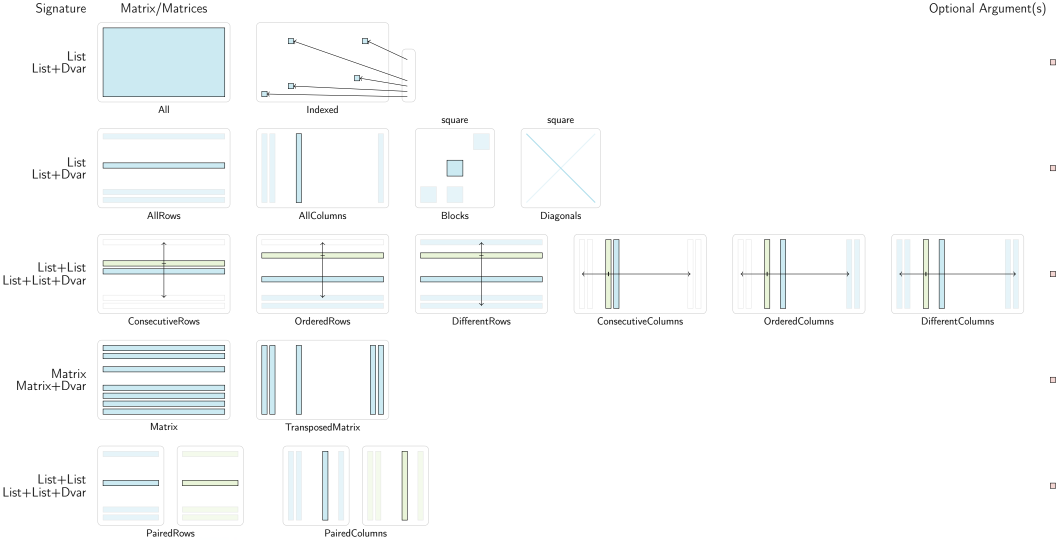
1	1	1	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	0	0
1	0	1	0	0	1	0	0	1	1
0	1	0	1	0	0	1	0	1	1
0	0	1	0	1	0	1	1	1	0
0	0	0	1	1	1	0	1	0	1

Stage 1: Acquire Size Specific Model



Insight

Pattern Generator Involving Matrix



Size Specific Model for Instance 0

Constraint	Signature	Pattern	Extra Arg
sum	List+Dvar	AllRows	5
sum	List+Dvar	AllColumns	3
scalar_product	List+List+Int	DifferentRows	2
lex_chain_greater	Matrix	Matrix	-
lex_chain_greater	Matrix	TransposedMatrix	-
lex_chain_geq	Matrix	TransposedMatrix	-

plus many others

1	1	1	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	0	0
1	0	1	0	0	1	0	0	1	1
0	1	0	1	0	0	1	0	1	1
0	0	1	0	1	0	1	1	1	0
0	0	0	1	1	1	0	1	0	1

Size Specific Model for Instance 0

Constraint	Signature	Pattern	Extra Arg
sum	List+Dvar	AllRows	5
sum	List+Dvar	AllColumns	3
scalar_product	List+List+Int	DifferentRows	2
lex_chain_greater	Matrix	Matrix	-
lex_chain_greater	Matrix	TransposedMatrix	-
lex_chain_geq	Matrix	TransposedMatrix	-

plus many others

1	1	1	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	0	0
1	0	1	0	0	1	0	0	1	1
0	1	0	1	0	0	1	0	1	1
0	0	1	0	1	0	1	1	1	0
0	0	0	1	1	1	0	1	0	1

Size Specific Model for Instance 0

Constraint	Signature	Pattern	Extra Arg
sum	List+Dvar	AllRows	5
sum	List+Dvar	AllColumns	3
scalar_product	List+List+Int	DifferentRows	2
lex_chain_greater	Matrix	Matrix	-
lex_chain_greater	Matrix	TransposedMatrix	-
lex_chain_geq	Matrix	TransposedMatrix	-

plus many others

1	1	1	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	0	0
1	0	1	0	0	1	0	0	1	1
0	1	0	1	0	0	1	0	1	1
0	0	1	0	1	0	1	1	1	0
0	0	0	1	1	1	0	1	0	1

Size Specific Model for Instance 0

Constraint	Signature	Pattern	Extra Arg
sum	List+Dvar	AllRows	5
sum	List+Dvar	AllColumns	3
scalar_product	List+List+Int	DifferentRows	2
lex_chain_greater	Matrix	Matrix	-
lex_chain_greater	Matrix	TransposedMatrix	-
lex_chain_geq	Matrix	TransposedMatrix	-

plus many others

1	1	1	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	0	0
1	0	1	0	0	1	0	0	1	1
0	1	0	1	0	0	1	0	1	1
0	0	1	0	1	0	1	1	1	0
0	0	0	1	1	1	0	1	0	1

Size Specific Model for Instance 0

Constraint	Signature	Pattern	Extra Arg
sum	List+Dvar	AllRows	5
sum	List+Dvar	AllColumns	3
scalar_product	List+List+Int	DifferentRows	2
lex_chain_greater	Matrix	Matrix	-
lex_chain_greater	Matrix	TransposedMatrix	-
lex_chain_geq	Matrix	TransposedMatrix	-

plus many others

1	1	1	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	0	0
1	0	1	0	0	1	0	0	1	1
0	1	0	1	0	0	1	0	1	1
0	0	1	0	1	0	1	1	1	0
0	0	0	1	1	1	0	1	0	1

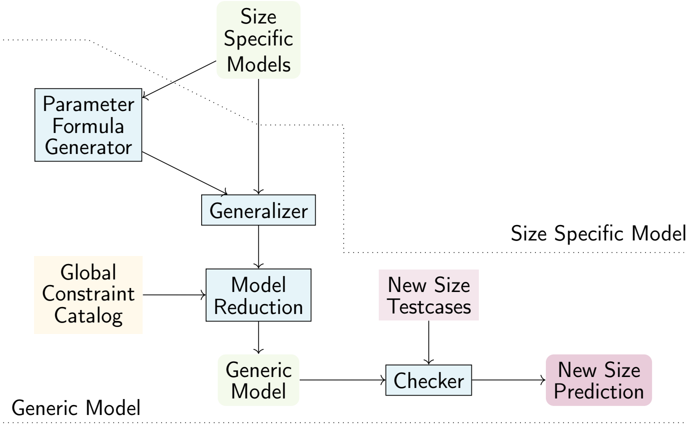
Size Specific Model for Instance 0

Constraint	Signature	Pattern	Extra Arg
sum	List+Dvar	AllRows	5
sum	List+Dvar	AllColumns	3
scalar_product	List+List+Int	DifferentRows	2
lex_chain_greater	Matrix	Matrix	-
lex_chain_greater	Matrix	TransposedMatrix	-
lex_chain_geq	Matrix	TransposedMatrix	-

plus many others

1	1	1	1	1	0	0	0	0	0
1	1	0	0	0	1	1	1	0	0
1	0	1	0	0	1	0	0	1	1
0	1	0	1	0	0	1	0	1	1
0	0	1	0	1	0	1	1	1	0
0	0	0	1	1	1	0	1	0	1

Stage 2: Generalize Acquired Models



Generalized Model

Constraint	Signature	Pattern	Extra Arg
sum	List+Dvar	AllRows	rowsum
sum	List+Dvar	AllColumns	colsum
scalar_product	List+List+Int	DifferentRows	scalarproduct
lex_chain_greater	Matrix	Matrix	-
lex_chain_greater	Matrix	TransposedMatrix	-
lex_chain_geq	Matrix	TransposedMatrix	-

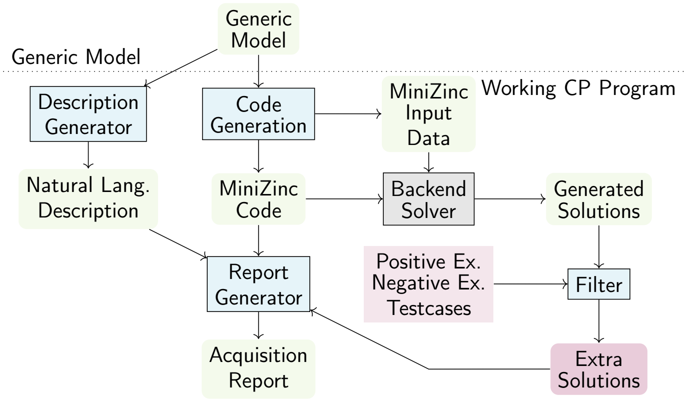
- Symbolic value, needs to be explained
- Dropped, since not present in all instances

Model Reduction (follows ModelSeeker)

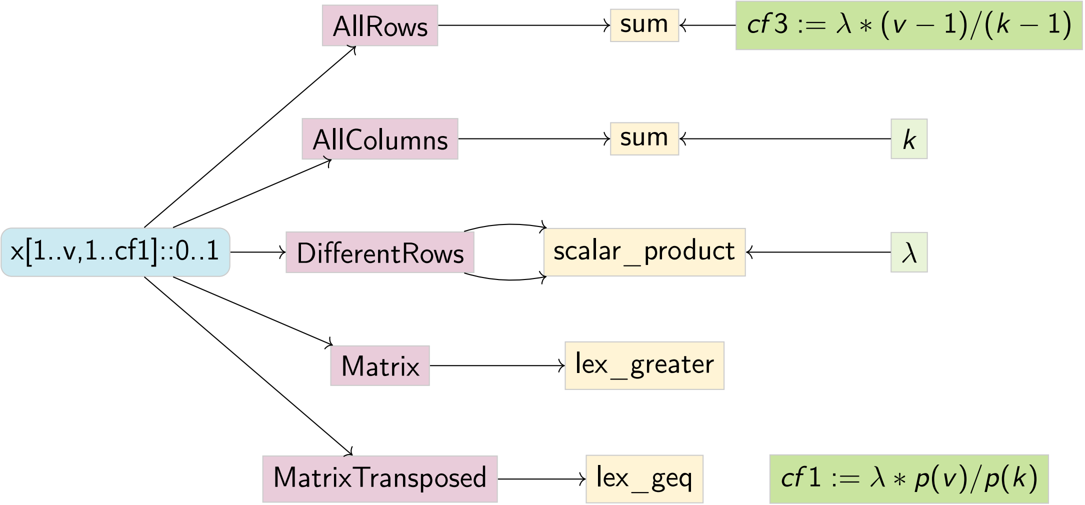
Nr	Constraint	Pattern	Trans.	Signature	Key	Key2	Key3 Formula	Offset Op	Sub. by	Signif	Trivial	Comment
1	lexchaingreater	Matrix	Id	Matrix	matrix					1100000	false	
2	lexalldifferent	Matrix	Id	Matrix	matrix				1	1000000	false	
3	lexchaingeq	Matrix	Id	Matrix	matrix				1	1000000	false	
4	lexchaingeq	MatrixTransposed	Id	Matrix	matrix					1000000	false	
5	sum	AllRows	Id	ListFD	matrix		cf3			100000	false	
6	sum	AllColumns	Id	ListFD	matrix		k			100000	false	
7	scalarproduct	DifferentRows	Id	ListListInt	matrix		inputData:lambda			100000	false	
8	scalarproduct	OrderedRows	Id	ListListInt	matrix		inputData:lambda		7	100000	false	
9	scalarproduct	ConsecutiveRows	Id	ListListInt	matrix		inputData:lambda		7	100000	false	
10	lexgreater	OrderedRows	Id	ListList	matrix				1	20000	false	
11	lexgreater	ConsecutiveRows	Id	ListList	matrix				10	20000	false	
12	lexgeq	OrderedRows	Id	ListList	matrix				10	15000	false	
13	lexgeq	ConsecutiveRows	Id	ListList	matrix				11	15000	false	
14	lexgeq	OrderedColumns	Id	ListList	matrix				4	15000	false	
15	lexgeq	ConsecutiveColumns	Id	ListList	matrix				14	15000	false	
16	lexdifferent	DifferentRows	Id	ListList	matrix				2	10000	false	
17	lexdifferent	OrderedRows	Id	ListList	matrix				10	10000	false	
18	lexdifferent	ConsecutiveRows	Id	ListList	matrix				11	10000	false	
19	equalsum	DifferentRows	Id	ListList	matrix				5	10000	false	
20	equalsum	OrderedRows	Id	ListList	matrix				19	10000	false	
21	equalsum	ConsecutiveRows	Id	ListList	matrix				19	10000	false	
22	equalsum	DifferentColumns	Id	ListList	matrix				6	10000	false	
23	equalsum	OrderedColumns	Id	ListList	matrix				22	10000	false	
24	equalsum	ConsecutiveColumns	Id	ListList	matrix				22	10000	false	
25	notallequal	All	Id	List	matrix				26	0	false	
26	notallequal	AllRows	Id	List	matrix					0	false	
27	notallequal	AllColumns	Id	List	matrix					0	false	
28	someequal	All	Id	List	matrix				29	0	false	
29	someequal	AllRows	Id	List	matrix					0	false	
30	someequal	AllColumns	Id	List	matrix					0	false	
31	minimum	All	Id	ListFD	matrix		0			0	true	
32	minimum	AllRows	Id	ListFD	matrix		0			0	true	
33	minimum	AllColumns	Id	ListFD	matrix		0			0	true	
34	maximum	All	Id	ListFD	matrix		1			0	true	
35	maximum	AllRows	Id	ListFD	matrix		1			0	true	
36	maximum	AllColumns	Id	ListFD	matrix		1			0	true	
37	sum	All	Id	ListFD	matrix		cf2			0	true	
38	nvalue	All	Id	ListFD	matrix		2			0	true	
39	nvalue	AllRows	Id	ListFD	matrix		2			0	true	
40	nvalue	AllColumns	Id	ListFD	matrix		2			0	true	

- Subsumption properties of Global Constraint Catalog
- Trivial constraint recognition
- Heuristic ranking of interest

Stage 3: Produce Working CP Program



Graph Based Model



Generated Code (Data Declarations)

```
% Generated Constraint Model for Problem type12
% Produced by CAT Constraint Acquisition Tool
include "globals.mzn";
include "../minizinclibrary/cat.mzn";

% Integer Parameters
int:k; % inputData:k
int:lambda; % inputData:lambda
int:v; % inputData:v

% Generic Formulas
int:cf1 = lambda*npairs(v) div npairs(k);
int:cf2 = 2*lambda*npairs(v) div nminus1(k);
int:cf3 = lambda*nminus1(v) div nminus1(k);
% Variables
array[1..v,1..cf1] of var 0..1:x; % matrix
```

Sample MiniZinc Data File

```
% Data for problem type12 instance 0
k = 3; % inputData:k
lambda = 2; % inputData:lambda
v = 6; % inputData:v
```

Generated Code (Variables and Constraints)

```
% Variables
array[1..v,1..cf1] of var 0..1:x; % matrix

% Constraints
% Constraint 1
constraint lex_chain_greater(x);
% Constraint 4
constraint lex_chain_greatereq(transpose(x));
% Constraint 5
constraint forall(i in 1..v)
  (sum([x[i,j]|j in 1..cf1])=cf3);
% Constraint 6
constraint forall(j in 1..cf1)
  (sum([x[i,j]|i in 1..v])=k);
% Constraint 7
constraint forall(i1, i2 in 1..v where i1 != i2)
  (scalarproduct([x[i1,j]|j in 1..cf1],[x[i2,j]|j in 1..cf1],lambda));

% Objective
solve satisfy;
```

Generated Description

Given integer parameters *k*, *lambda*, and *v*, and introducing symbols

cf1 = lambda*npairs(v) div npairs(k)

cf3 = lambda*nminus1(v) div nminus1(k)

find an assignment for a matrix *x* with *v* rows and *cf1* columns, where each element ranges between *0* and *1*,
such that

1 the constraint *lexchaingreater(x)* holds for the matrix *x*.

4 the constraint *lexchaingeq(transpose(x))* holds for the transposed matrix *x*.

5 the sum of each row of *x* is equal to *cf3*.

6 the sum of each column of *x* is equal to *k*.

7 the scalar product of every pair of different rows of *x* is equal to *lambda*.

CAT Constraint Acquisition

- Introduction
- Solution Approach
- Results

Insight

Evaluation

Problem	Description	Instances	Classified Correctly Positive Samples	Classified Correctly Negative Samples	Tests	Extra Solutions	Comment
1	Graph Coloring	10	100.00	100.00	100.00		
2	N-Queens	10	100.00	100.00	100.00		
3	Warehouse Location	10	100.00	100.00	100.00		
4	Golomb Ruler	8	100.00	100.00	100.00		Domain bound not generalized
5	Sudoku (with Hints)	10	100.00	100.00	100.00		
6	Sudoku (without Hints)	4	100.00	100.00	100.00		
7	Schur's Lemma	13	100.00	100.00	100.00		
8	All Interval	10	100.00	100.00	100.00		
10	Magic Squares	6	100.00	100.00	100.00		
11	Orthogonal Latin Squares	2	100.00	100.00	100.00		
12	BIBD	20	100.00	100.00	100.00		
13	Costas Array	11	100.00	100.00	100.00		
14	N-Queens Variant	11	100.00	100.00	100.00		
15	N-Queens Variant	7	100.00	100.00	100.00		
16	N-Queens Variant	13	100.00	100.00	100.00		Model incomplete

Summary and Next Steps

- Constraint Acquisition allows to generate models from examples
- Many different approaches, hard to compare
- Defining realistic use case
- Results on simple problems show promise
- Currently working on first realistic examples
- Next generation of PTHG Challenge problems

Other Applications

Other Noteworthy Applications

- NVD LoadBuilder
- Boliden Tara Mines Dewatering
- Dental School Timetabling
- Irish Naval Service Rostering
- Data Centre Load Consolidation
- Scheduling with Time Variable Energy Prices
- Characterizing EDF Power Plants with Timeseries Constraints
- Optical Network Design
- Supplier Selection Problem
- Optimizing UCC's CHP Plant Operation
- CP Conference Paper Assignment Tool

NVD LoadBuilder

- Real-World Problem
  - Deliver cars/vans from factory/ports to dealers
  - Group cars into loads for joint delivery
  - Using specialized transporters with complex configurations
  - Balance distance travelled, utilization of fleet, priority of orders
- Status
  - In daily use at customer since 2020
  - Start-up company CMC to further develop tool
- Research Challenges
  - Vehicle routing problem with complex capacity constraints
  - Decide which cars to deliver today
  - What impact does this have tomorrow
  - Explaining solutions to end-user
- Solution Approach
  - Decomposition
  - MIP, Constraint Programming, Local Search, Data Analytics

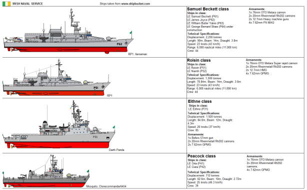
Boliden Tara Mines Dewatering

- Real-World Problem
  - When/how to pump water out of mine
  - Multiple pumps, reservoirs
  - Electricity cost major cost factor
  - Safe operation of mine paramount
- Status
  - Student-led project with DCU
  - Paper at AAAI 2016
  - Major flooding event in 2021
- Research Challenges
  - Scheduling with uncertain energy prices (real-time tariff)
  - Uncertain water ingress depends on operations
  - Capacity (min/max) constraints for storage
- Solution Approach
  - Electricity price prediction
  - Optimization

Dental School Timetabling

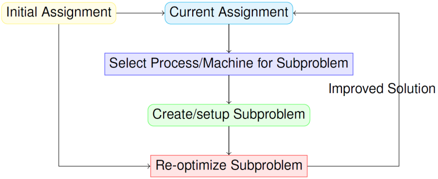
- Real-World Problem
  - Change time table during period of teaching capacity increase
  - Previous schedule no longer feasible
  - Multiple courses share same lab space (dental chairs) at the same time
  - Hard capacity limits on available resources and time slots
- Status
  - Used by dental school during transition period
  - Paper in IAAI 2013, AI Mag 2014
- Research Challenges
  - Very different from standard timetabling problem
  - Hard/soft capacity constraints
  - Tool cleaning setup time constraints
- Solution Approach
  - Optimization
  - Flexible prioritization of constraints

Irish Naval Service Yearly Rostering



- Real-World Problem
  - Decide which ships are performing which type of duty over the year
  - Budget limitations on total time at sea
  - Fair share of work across fleet
  - Fixed maintenance periods for certain ships
  - Special events (flotilla exercises, detached duty)
- Status
  - Prototype results produced for service
- Research Challenges
  - Finding the best tool and model for problem
  - Balanced assignment under budget constraints
  - Provide consistent force levels over whole year
  - Fair assignment of work/rest days across fleet
- Solution Approach
  - Optimization

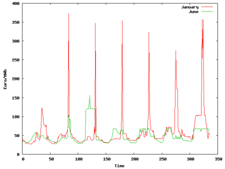
Data Centre Load Consolidation



- Real-World Problem
  - Move virtual machines between servers in a data centre
  - Balance/concentrate workload on multiple resource types
  - Extend to multiple data centres across world
- Status
  - 2nd price in Google Roadef/Euro Challenge 2012
  - Multiple papers
- Research Challenges
  - Reassignment problem
  - Multi-bin packing constraints
  - Large neighbourhood search to deal with problem size
- Solution Approach
  - Optimization
  - New tools/propagators

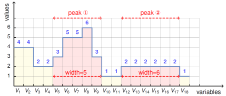
Insight

Scheduling with Time Variable Energy Prices



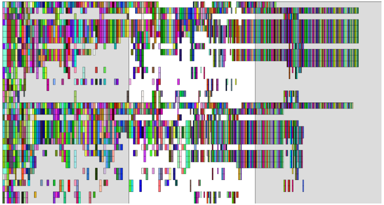
- Real-World Problem
  - How do time-variable electricity prices affect scheduling of use
  - Uncertainty of prices, sudden peak prices common in Ireland
  - In most cases, we have to commit to production before price is known
  - Deal with risk/possible rewards
- Status
  - Multiple papers
  - Continued work on price prediction with industry
- Research Challenges
  - Can we use time variable electricity prices to our advantage?
  - Which properties should a price prediction model have to help with scheduling?
  - Can we tune price prediction for the use case it is intended for?
- Solution Approach
  - Machine Learning
  - Optimization

Characterizing EDF Power Plants



- Real-World Problem
  - Unit Commitment Model for electricity supply
  - Decide which units to run when to satisfy demand/minimize cost
  - Change of production for different units is limited over time
  - Very error-prone integration into global model
- Status
  - Joint work with IMT-Atlantique, EDF Research
  - Series of papers on time-series constraints, Volume II of Global Constraint Catalog
- Research Challenges
  - Can we characterize the production limits of power plants as time-series constraints?
  - Learn constraints from historical data (planned/actual)
  - Create model of individual plants to describe their capabilities
  - Find redundant constraints to overcome limits of propagation
- Solution Approach
  - Machine Learning
  - Automata constraints
  - Generated code for propagators

Optical Network Design



- Real-World Problem
  - Core optical network design
  - Different from traditional IP network design
  - Define paths from source to sink
  - Use multiple frequency (light) bands over same fibre
- Status
  - paper ICTAI 2014
- Research Challenges
  - Modelling Choices
  - Amount of propagation achieved
  - Scalability of methods
- Solution Approach
  - Global Constraints

Supplier Selection Problem

Texas Instruments
SN7400N
Logic IC Operating temperature: 0...+70 °C Series: **SN7400** Function: Quad / 2-Input / NAND Manufacturer: Texas Instruments Package: DIL-14 Technology: TTL

€0.631 | + Add to BOM | Datasheet at TI.com | 19
CAD Models

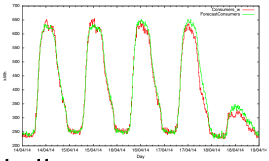
	Distributor	SKU	Stock	MOQ	Pkg		1	10	100	1,000	10,000
★	Digi-Key	296-14641-5-ND	3,644	1	Tube	EUR	1.39	1.24	0.967	0.631	0.559
★	Verical (2)	SN7400N	9,975	525		EUR				0.564	0.564
★	Avnet	SN7400N	513	1		EUR *	1.34	1.15	0.884	0.884	0.884
★	Farnell	1105937	874	1		EUR	1.32	1.32	0.893	0.617	0.549
★	TI store	SN7400N	5,787	1	Tube	EUR *	1.26	1.12	0.886	0.478	0.478

Show All | Specs | Descriptions | Manufacturer Page

- Real-World Problem
  - Which suppliers to select to provide list of components
  - Limit number of suppliers by ordering multiple items from same supplier
  - Price/lead time/quality of service are competing objectives
- Status
  - Work with industry partner
  - Paper in Annals of Operations Research
- Research Challenges
  - How do we learn which choices are preferred
  - Difficult to assign fixed weights to different aspects of solution quality
  - Iterative, interactive learning of preferences
- Solution Approach
  - Preference Learning
  - Optimization

Optimizing UCC's CHP Plant Operation

- Real-World Problem
  - When to run UCC's CHP plant to create electricity/heat on-site
  - Needs demand forecast for heat and electricity
  - Uncertain Real-time grid electricity price
  - Heat and electricity demand of campus not in sync
- Status
  - Tested for several weeks with operator of plant
  - Part of EU Discipl project



- Research Challenges
  - Heat and Electricity Demand prediction for campus
  - Price prediction for real-time grid price
  - Integration of plant operational constraints
  - Wider impact of heating strategy on campus
- Solution Approach
  - Machine Learning
  - Optimization

CP Conference Paper Assignment Tool

- Real-World Problem
  - Which reviewers to assign to papers
  - Consider bids by reviewers, avoid assigning unwanted papers
  - Deal with reviewers shared between multiple tracks
  - Balance assignment between reviewers
  - Allow pre-assignment, specific capacity constraints
- Status
  - Joint work with Data61, INRA
  - Used in 2020, 2021
  - Paper at ModRef 2020

- Research Challenges
  - Fair treatment of papers and reviewers
  - Finding mechanisms to allow Program Chair to control process
  - Not a black-box assignment
  - Integration with easychair
- Solution Approach
  - Optimization

Summary

Summary

- Provided details for some application work at Insight
- Shows the impact of practical problems on basic research
- Research can have a real impact
- It takes time to do application based research