A Solution to the Roadef 2022 Challenge

Andrea Balogh, Sharmi Dev Gupta,
Jheisson Argiro Lopez Restrepo, Filipe Souza,
Barry O'Sullivan, Helmut Simonis,
Diarmuid Grimes
Insight SFI Centre for Data Analytics
School of Computer Science and Information Technology
University College Cork, Cork, Ireland

And now for something completely different...

- Using Constraint Programming/Optimization for decision support
- AI, but not *that* kind of AI
- Not rule based
- Done with a group of PhD students as a side activity

Insight Overview

4 Co-Lead Universities 9 partner institutions	Built on **20** years of research in Data Analytics and AI
450+ Academics, Postdocs, PhDs, RAs	**3400+** Scientific conference and journal papers
175+ Funded collaborations with industry partners	**350+** Research Awards
16 Spin out companies 72 license agreements	**135+** H2020 consortia, 500+ collaborations, 40+ countries
1,137+ school visits, 28,000 students	**276** PhDs graduated

Insight

Overview

- Participation in French OR Competition *Roadef Challenge*
- "Hobby" project driven by PhD students
- Aim: Achieve good results with limited, low risk effort
- Use Optimization/Constraint Programming where possible
- 4th in qualification, 8th in main evaluation, out of 52 teams
- Two more stages to come

Table of Contents

Background: Roadef Challenge

- Competition run by French OR Society (Roadef) since 1999
  - Since 2010 in cooperation with Euro (European Operational Research Society)
- Different topics every two years
- Multi-stage competition with open and hidden datasets
- Problem provided by industrial partner (2022: Renault)
- Very close to industrial reality

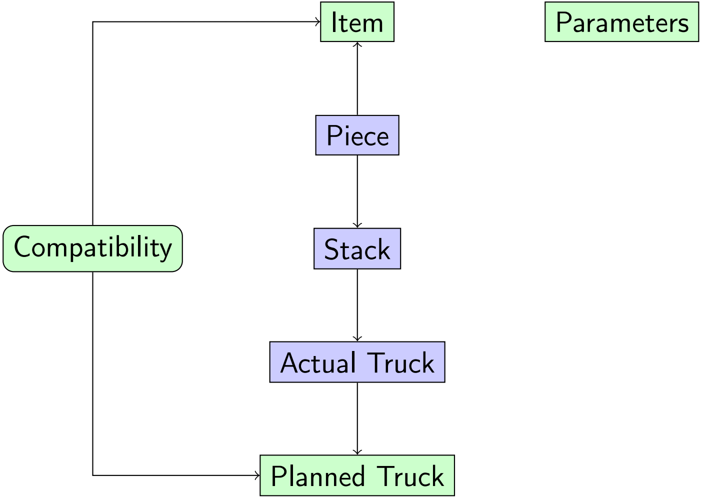
Past Challenges

1999 Inventory Management (Bouygues)
2001 Frequency Assignment (CELAR)
2003 Satellite Mission Planning (ONERA)
2005 Car Sequencing Problem (Renault)
2007 Intervention Planning (France Telecom)
2009 Disruption Management (Amadeus)
2010 Energy Management (EDF)
2012 Datacentre Machine Reassignment (Google)
2014 Rolling Stock Management (SNCF)
2016 Inventory Routing Problem (Air Liquide)
2018 Flat Glass Cutting Problem (Saint-Gobain)
2020 Maintenance Planning (RTE)

The 2022 Problem

- Transport items from suppliers to factories in semi-trucks
- Items cannot be delivered late, but can be transported early
- Each factory is one instance (20-50 instances per dataset)
- Items are placed in crates, which can be stacked
- Different types of trucks are used
- Given calendar of planned trucks, additional copies can be used when needed
- Overall cost is cost of trucks used plus inventory cost of items delivered early

Main Concepts

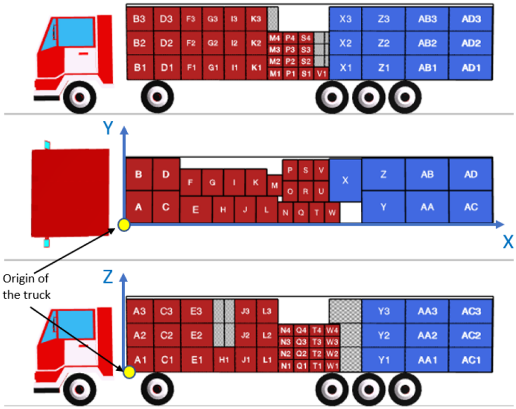


Temporal Constraints

- Each item has earliest and latest arrival window
- Latest timepoint corresponds to some planned truck
- Item cannot arrive late (even on same day)
- If item arrives early, inventory cost (per day) must be paid
    - Cost per day varies widely
    - No cost if delivered earlier on the last day

3D Placement Problem

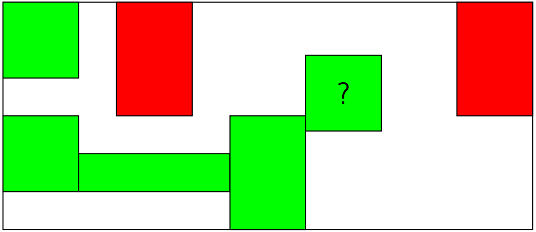
- Fill trailer of semi-truck with pieces
- Pieces must stay upright, may be rotated lengthwise or widthwise
- Only pieces with identical footprint can be stacked
- All pieces must fit within trailer volume
- Weight of all pieces must be below trailer capacity



Source: Roadef Challenge Description

Support

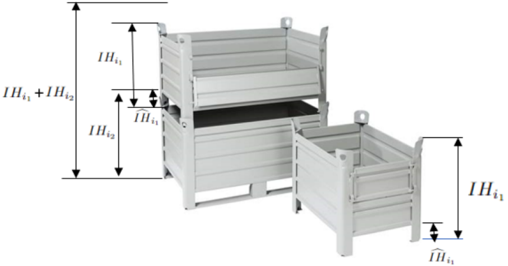
- Each item must be supported in z and x direction
- For z direction
  - Either item is placed on floor of truck
  - Or is is placed on top of another item with the same footprint and orientation
  - Crate corners connect together
- For x direction
  - Either item is placed against front (left) wall of truck
  - Or it is (partially) supported by another item immediately to its left



- Floorplan
- Green: Supported
- Red: Not supported

Stacking of Crates

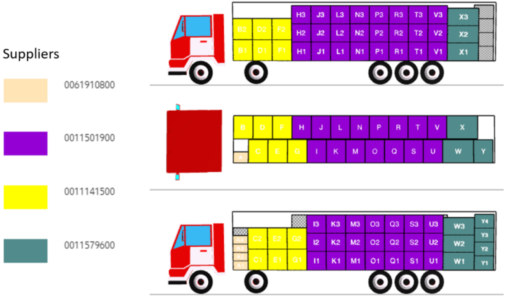
- Pieces are packed in crates
- Only crates with same footprint can be stacked
- Stacked crates may be nesting
  - Total height is smaller than sum of heights
- Most crates can be placed lengthwise or widthwise
  - Some have fixed orientation



Source: Roadef Challenge Description

Loading Order

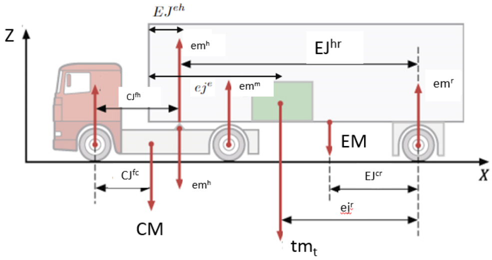
- Trucks are visiting suppliers in given order
- Not all trucks visit all suppliers
- Items must be loaded in sequence of visit
- Earlier visits in front, later visits in back
- Details more complex



Source: Roadef Challenge Description

Truck Constraints

- Total weight limit is not enough
- Non-linear axle weight constraints
  - Placing too many items in front of truck exceeds mid-axle weight capacity
  - Constraints not incremental, adding one more item may reduce an axle weight
- Axle weight limits apply for each load step, not just for fully loaded truck
- Checker will reject infeasible placement



Source: Roadef Challenge Description

Instance Parameters

- Relative cost of transport vs. inventory
  - Ratios 1:5, 1:1, 5:1
- Cost increment of extra trucks
- Runtime limit (1,800 or 3,600s)

What is provided?

- Problem description (pdf)
- Competition rules (pdf)
- Instance files (multiple files per factory, French csv)
- Result checker (Java, decides if solution is correct, at which cost)
- 3D Visualizer (Java, not used by our team)

How is the Competition run?

- Competitors provide binary versions of their programs
- Programs run on Roadef machines against known/unseen instances
- Competition machines only allow certain commercial tools, limit resource use
- Solutions are checked with given checker
- Points allocated for each instance, comparing with solutions by other competitors
- Total ranking based on sum of points achieved
    - Problem size does not matter
    - Ranking not based on total cost
- Only best cost value for each instance is made available at end of phase

Stages of the Competition

- Challenge starts, early July, 2022
- Sprint, Dataset A, October 2022 (DNA)
- Qualification, Dataset B, January 2023
- Test phase, Datasets C and X, June 2023
- (Unlimited Evaluation), Datasets C and X, End September 2023
- Scientific Contribution, November 2023

Important Observation

- Early datasets are quite different from later ones
  - Number of items and trucks relatively small
  - Trucks spaced in calendar, little opportunity for early delivery
  - Daily inventory cost high
  - Not all constraints are present
- Challenging to develop tool based on early instances that performs well on later ones
- Easy to only focus on improving known instance solutions
- You don’t know how good your solution needs to be

Samples of Instances (Dataset X)

Name	Nr	Inventory Cost	Transport Cost	Extra Truck Cost	Time Limit	Different Items	Total Items	Trailer Types	Shortest Trailer	Longest Trailer	Planned Trucks	Truck Entries	Suppliers	Supplier Docks	Plant Docks	Products	Package Codes	Stacka-bility
AS	0	1.00	1.00	0.20	3,600	14,648	51,025	8	4,080	24,840	1,474	33,489	65	65	5	813	82	32
AS2	1	5.00	1.00	0.20	3,600	14,648	51,025	8	4,080	24,840	1,474	33,489	65	65	5	813	82	32
AS3	2	1.00	5.00	0.20	3,600	14,648	51,025	8	4,080	24,840	1,474	33,489	65	65	5	813	82	32
AS4	3	1.00	1.00	0.20	3,600	14,850	51,923	8	4,080	24,840	1,508	33,964	64	64	5	806	80	31
AS5	4	5.00	1.00	0.20	3,600	14,850	51,923	8	4,080	24,840	1,508	33,964	64	64	5	806	80	31
AS6	5	1.00	5.00	0.20	3,600	14,850	51,923	8	4,080	24,840	1,508	33,964	64	64	5	806	80	31
AS7	6	1.00	1.00	0.00	3,600	14,850	51,923	8	4,080	24,840	1,508	33,964	64	64	5	806	80	31
BY	7	1.00	1.00	0.20	3,600	46,070	185,348	15	13,400	13,500	5,868	107,869	374	375	14	2,431	236	90
BY2	8	5.00	1.00	0.20	3,600	46,070	185,348	15	13,400	13,500	5,868	107,869	374	375	14	2,431	236	90
BY3	9	1.00	5.00	0.20	3,600	46,070	185,348	15	13,400	13,500	5,868	107,869	374	375	14	2,431	236	90
BY4	10	1.00	1.00	0.20	3,600	44,995	193,304	15	13,400	13,500	5,629	102,462	377	378	14	2,446	233	90
BY5	11	5.00	1.00	0.20	3,600	44,995	193,304	15	13,400	13,500	5,629	102,462	377	378	14	2,446	233	90
BY6	12	1.00	5.00	0.20	3,600	44,995	193,304	15	13,400	13,500	5,629	102,462	377	378	14	2,446	233	90
BY7	13	1.00	1.00	0.00	3,600	44,995	193,304	15	13,400	13,500	5,629	102,462	377	378	14	2,446	233	90

- 200k pieces, 10k trucks, 300+ suppliers, 2k products per instance
- Different parameter combinations

Table of Contents

Our Involvement

- A group of four PhD students wanted to participate
- Background: UCC came second in 2012 challenge (Google)
- Involved some of their supervisors
- Asked me to help with overall design (end October 2022)
- Weekly meeting (remote/in-person) to discuss progress
- Uses our Java development framework
  - Code generator for most boiler-plate code
  - Visualization/reporting tools
  - Used for $\approx$ 50 projects so far

Aims

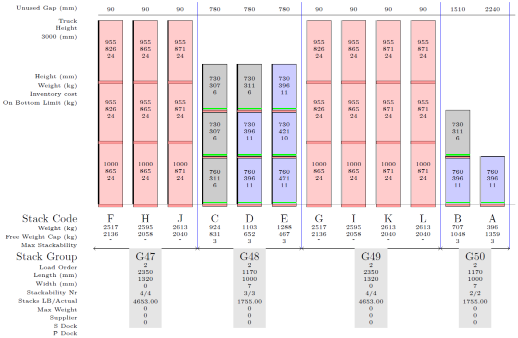
- This is a complex problem, not easy to find feasible solutions
- Problems range from quite small to very large, limited solution time
- First aim is to find feasible solutions, then improve solution quality
- Use decomposition to create sub-problems that can be designed on their own
    - Ideally, decompose into separate, independent sub-problems
    - Or, create sequence of sub-problems that chain intermediate results
- Come up with lower bounds for instances to judge solution quality
- Avoid high-risk, high reward approaches due to limited resources
- Cannot afford to spend much time in any sub-problem

First Step: Minimal Viable Product

- Place every item on last possible truck
  - This creates independent sub-problems for each planned truck
- Create minimal number of stacks of compatible items
  - Bin packing variant
- Place all stacks in the truck
  - 2D placement, since we deal with stacks, not items
  - May need more than one truck to take all items
- No inventory cost, everything is delivered just in time
- Produce results files, check against provided checker
- Validates understanding of problem, provides baseline

Building Stacks

- Group items that can be stacked together
  - Footprint
  - Load stage
- Use available height/weight
- Respect stackability constraints
- Respect forced orientation
- Cannot afford to run full optimization model
- Use lower bound to check optimality

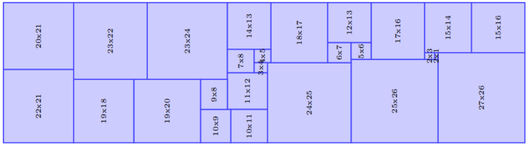


Colored by Inventory Cost

Insight

2D Placement as an Optimization Problem

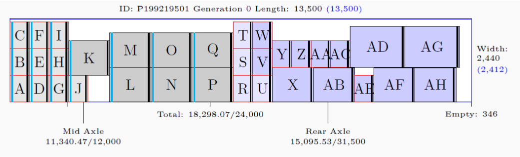
- We have done a lot of work on using Constraint Programming for placement problems
- Initial idea was to use CP for the placement sub-problems here
- We decided against this:
  - Thousands of sub-problems to solve, needs too much time
  - Requires tools that are not installed on Roadef machines
  - Axle weight constraints would need new constraint
  - Support only handled by a few solvers



from Simonis, O'Sullivan: Almost Square Packing, CPAIOR 2011

Placement Heuristic

- Heuristic placement
  - Order items by load stage and increasing weight
  - Place next item in left-most, lowest free corner
- Guarantees support
- Use lower bound to check optimality
- No extra points for neat packing!



Colored by Load Stage

Extensions of Minimal System

- Portfolio of placement methods
- Provide counterfactual explanations
- Allow stacks to be transported earlier
  - Clustering of trucks
  - Movement of stacks to earlier trucks
  - Trade number of trucks against inventory cost

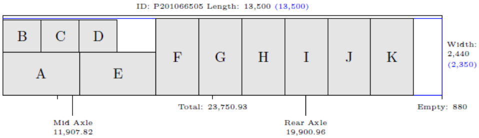
Placement Portfolio

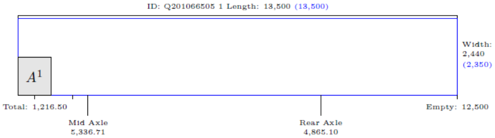
- Heuristic is very fast, but incomplete
- Can afford to run many variants
  - Random reordering of items
  - Orientation preference
  - Maintain axle weight limits
- Try complete models for special cases
- Add fallback heuristics to always find a solution
  - Quality may be poor
  - Better to have bad solution rather than none

Table 1: Problem Strategy Count

	CC	CC2	CC3	CC4	CC5	CC6	CC7	CI	CI2	CI3	CI4	CI5	CI6	CI7	DO	DO2	DO3	DO4	DO5	DO6	DO7
	63	47	76	67	53	77	84	391	344	415	318	277	335	644	554	533	561	793	779	797	860
Conservative Lengthwise Density	1	2	1			2															
Conservative Lengthwise Size	1	3	1	2	1	1	2	4	2	3	5	4	3	5	1	1	2				
Conservative Lengthwise Weight	3	6	2	1	3	1	3		1				1	1							
Conservative Preference Density																1					
Conservative Preference Size								1		2			1								
Conservative Preference Weight								2	2	1	1	1	1	1	1						
Conservative Random Density	2	1	3	2	2	1	1	4	6	4	6	4	3	7	4	1	1	1	2	2	1
Conservative Random Size				1		1		19	21	13	11	10	13	10	8	7	6	4	3	12	3
Conservative Random Weight	2	1			1	2	1	3	13	17	7	3	6	5	2	2	1	1			3
Conservative Widthwise Density	1	2	3	4	2	2	1	1	2					1							
Conservative Widthwise Size	13	4	15	14	7	9	10	3	4	5	4	4	3	5	1	1	1				
Conservative Widthwise Weight	3	3	3	3	3	2	4	2	3	1	1	1	2	2		1		1	1		1
Corners Lengthwise Density	2	3			3	2	1	8	6	11	3	8	5	2		1	5	3	2	3	3
Corners Lengthwise Size	10	4	7	7	9	4	7	179	165	173	154	134	141	136	60	79	81	58	56	57	55
Corners Lengthwise Weight	8	10	6	7	13	4	8	25	23	23	14	21	16	18	5	3	5	2	3	6	3
Corners Preference Density								54	44	44	47	46	54	44	19	10	10	13	16	9	11
Corners Preference Size					1			294	285	297	274	265	298	234	145	135	135	101	93	111	94
Corners Preference Weight			1					25	28	31	25	28	30	24	10	7	7	8	6	6	8
Corners Random Density	10	7	10	13	9	11	12	10	11	11	10	3	13	10	2	4	3	1	2	2	2
Corners Random Size	7	14	12	20	14	17	15	29	30	27	26	31	21	26	9	10	12	9	6	10	3
Corners Random Weight	18	12	16	17	11	24	25	10	15	11	9	8	12	12	2	1	1	1	2	2	4
Corners Widthwise Density	4	2	4	4		6	5	63	73	72	61	71	61	55	15	9	17	14	9	14	9
Corners Widthwise Size	95	127	87	91	120	90	85	1160	1206	1163	1003	1038	973	869	343	362	323	295	322	280	292
Corners Widthwise Weight	4	6	6		3	2	1	184	147	164	162	145	159	92	26	21	34	31	23	23	12
Manyprecomputed Precomputed Weight	1	1	1	2	2	1	2	2	2	3	2	1	2	3	1	4	2				3
SingleFile Widthwise Size	2		2	4	2	1	5	3	2	5	5	3	1	5	3	1	1	1	1	1	2
SingleFile Widthwise Weight								1			3	1	3	3							
SinglePrecomputed Precomputed Weight	183	186	183	197	194	202	181	817	834	805	688	724	700	608	224	243	224	199	206	197	158
TwinPrecomputed Precomputed Weight	8	14	3	8	17	5	20	61	89	55	67	76	51	86	19	17	22	9	11	10	16

Counterfactual Explanations (Why didn't you do that?)

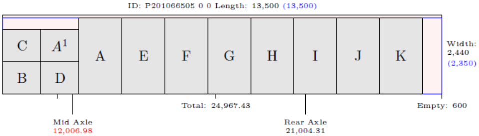




2.4.1 Counter Factuals

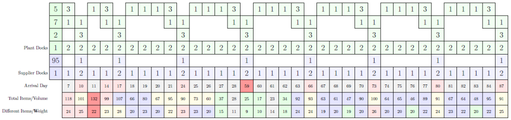
Table 12: Counter Factuals for P201066505

Name	Truck	Cost	Nr Errors	Excess Weight	Assigned Trucks	Excess Trucks	Assignment Strategy	Orientation Strategy	Order Strategy
P201066505 0	P201066505	1500.00	1.00	6.98	1	0	Corners	Lengthwise	Weight
P201066505 0	P201066505	1500.00	1.00	351.98	1	0	Corners	Widthwise	Density
P201066505 0	P201066505	1500.00	1.00	351.98	1	0	Corners	Preference	Density



Pre-processing: Clustering

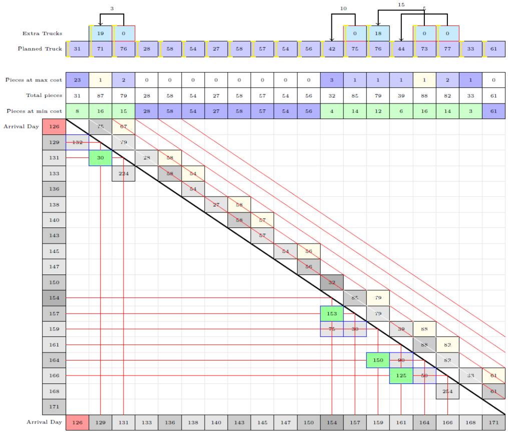
- We can partition the overall instance into clusters
- Two planned trucks are in same cluster, if some item can be transported on either truck
- Find connected components in generated graph
- We only have to consider interactions inside one cluster at a time



Not all trucks visit all suppliers/docks (or in same sequence)

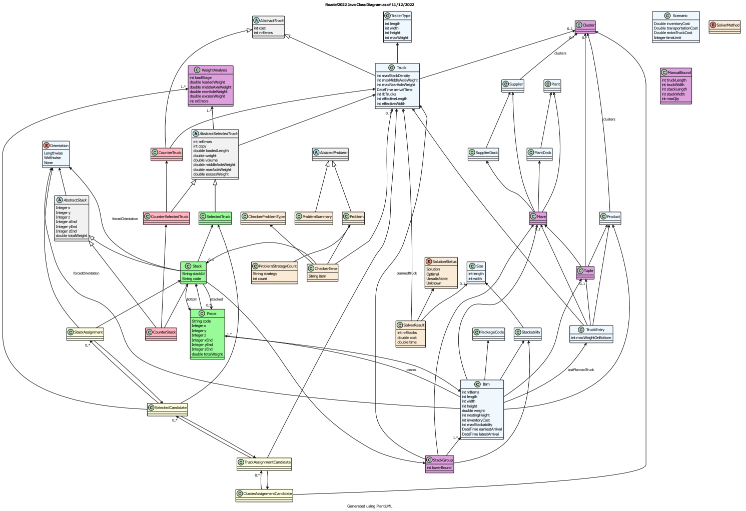
Post-Processing: Movement Solver

- Free up some sparsely used trucks
- Move all its stacks to another, earlier truck
- Saves cost of truck, costs inventory cost
- MIP Model
  - Find all possible moves
  - Select subset of moves that are compatible
  - Maximize cost savings



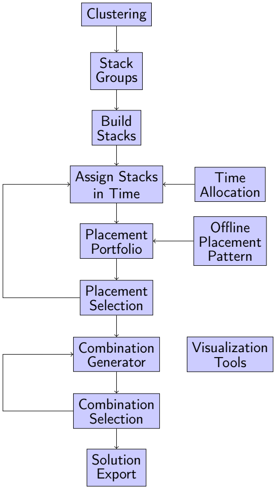
Insight

Underlying Object Model (December Code Review)



Current High Level Decomposition

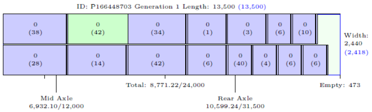
- Redesign after qualification phase
- New datasets have many more trucks on same day, or consecutive days
- Remove assumption of transporting items as late as possible
- Instead, solve optimization problem
  - Assign items in time to minimize number of trucks
  - While minimizing resulting inventory cost
  - Ignore detailed placement, only approximation of constraints



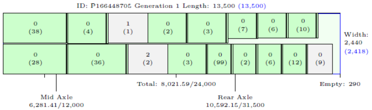
Insight

Example Assignment in Time (Dataset X/CI)

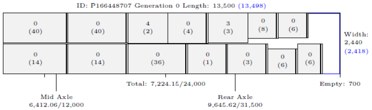
7.3 Truck P166448703, Arrival Time 15/6/2023 13:35, LB 1, Sizes 1200 x 1000, 1206 x 1010, 1209 x 1007, 1600 x 1200, 1610 x 1208, 2400 x 1200, 2600 x 1200



7.4 Truck P166448705, Arrival Time 15/6/2023 17:35, LB 1, Sizes 1200 x 1000, 1206 x 1010, 1209 x 1007, 1600 x 1200, 1610 x 1208, 2400 x 1200, 2600 x 1200



7.5 Truck P166448707, Arrival Time 15/6/2023 21:05, LB 1, Sizes 1200 x 1000, 1206 x 1010, 1209 x 1007, 1600 x 1200, 1610 x 1208, 2400 x 1200, 2600 x 1200

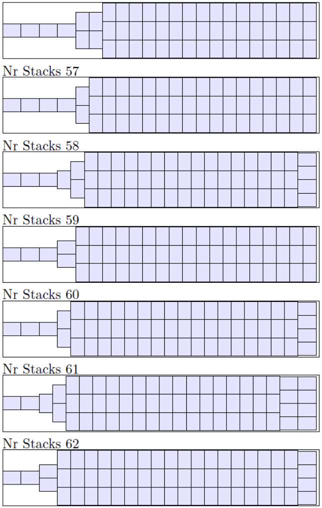


Observations

- Many trucks on same day
- Each truck is nearly full
  - Unfortunately, not always true
- Each truck contains items of many different sizes
- Only deliver items early that have low inventory cost
- No inventory cost for delivery on same day

Precomputed Solutions

- Some truck and item sizes occur very frequently
- We can precompute optimal solutions for special cases
- For a given set of items to be placed
  - Maximize use of space
  - Provide support
  - Push weight to end of truck



Importance of Visualization

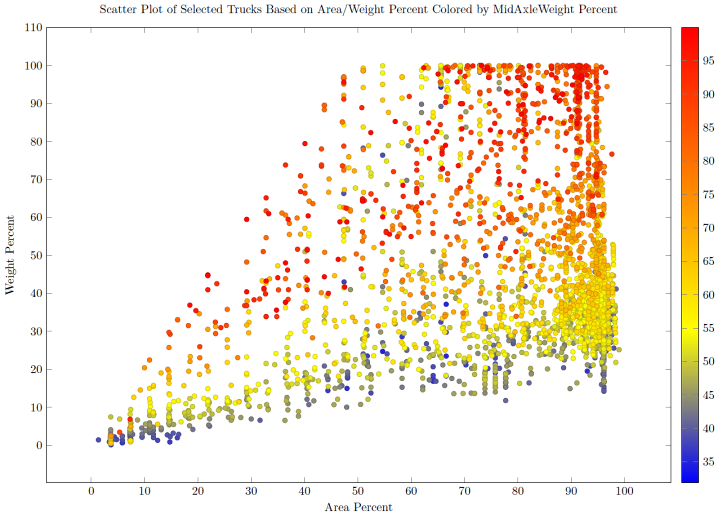
- We can only understand results by visualizing them
- Information overload: Show only what is needed
  - That may depend on situation, user
- Uses library of predefined visualizations in Java
  - Part of our Java development framework
- Add application specific custom visualizations
- Produce LaTeX reports

Table of Contents

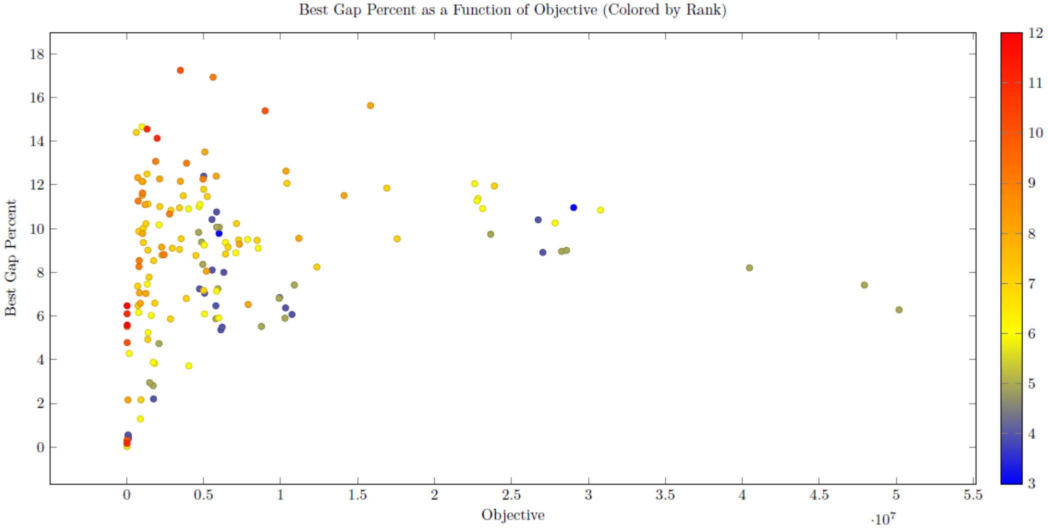
Code Size (Java)

Function	Package	LoC
Generated	datamodel	88k
	controller	22k
	view	4k
I/O	import	1,500
	export	200
Solvers	all	6,000
Visualization	all	4,300

Example Result Quality (Instance X/CI)



How do we compare?



Student Experience

- Round-table discussion of project at CP Training Week
  - Valuable learning experience
  - How to approach large-scale problem
  - Use of visualization to understand problems
  - Understand effort/reward compromises
  - Lots of fun as well

Summary

- Presented results for our entry to Roadef Challenge 2022
- Driven by students, limited time available
- Very challenging problem, both in complexity and scale
- Decomposition to create manageable sub-problems
- Fair to middlin' results
- Some stages of competition still on-going