Mechanical & Industrial Engineering
UNIVERSITY OF TORONTO

CERIDIAN

Explore Alternative Algorithms for Employee Rostering

Client: Ceridian Canada
Supervisor: Prof. Merve Bodur

Team: Jiahua Chen, Jiaru Li, Sijia Li, Mingkun Wang

Project Background

Ceridian is a global human capital management software company that provides work intelligence solutions for organizations of all sizes and from multiple industries. The rostering algorithm currently employed by the client is a modification of the metaheuristics Greedy Randomized Adaptive Search Procedure (GRASP), but it can be **slow** and **has no guarantee of solution quality**.

The client is looking for alternatives that can

- Generate **better quality** rosters
- **Minimize** demand coverage penalty
- **Meet** fundamental rostering constraints
- Solvable in **reasonable amount of time**

Mathematical Optimization can achieve ALL!

Performance Comparison Between GRASP and IP Model

Solution Quality (Objective Function Value) and Time Comparison

Time Interval / Model	15-Min	30-Min	60-Min
	Penalty / Time	Penalty / Time	Penalty / Time
GRASP (Local Diner)	230.3 / 63 s	154 / 18 s	227 / 9 s
IP Model (Local Diner)	0 / 6624 s	0 / 395 s	0 / 72 s
GRASP (Cosmetics Store)	967.8 / 92 s	957.5 / 23 s	939 / 6 s
IP Model (Cosmetics Store)	19 / 21539 s	11 / 695 s	6 / 216 s

Problem Size Comparison (Number of Variables)

Time Interval / Example	15-Min	30-Min	60-Min
	Integer (Binary)	Integer (Binary)	Integer (Binary)
Local Diner	185,682 (181,650)	92,946 (90,930)	46,578 (45,570)
Cosmetics Store	269,262 (266,574)	134,862 (133,518)	67,662 (66,990)

Methodology and Problem Instances

Optimization Model Formulation

Objective Function

$$min \sum_{j\in J}\sum_{k\in K}\sum_{l\in L}(p^o_{jkl}s^o_{jkl}+p^u_{jkl}s^u_{jkl})$$

Decision Variables (in Objective)

s^o_{jkl} : Number of overstaffing on day j, slot k, task l
s^u_{jkl} : Number of understaffing on day j, slot k, task l

Parameters (in Objective)

p^o_{jkl} : Penalty for overstaffing on day j, slot k, task l
p^u_{jkl} : Penalty for understaffing on day j, slot k, task l

Sets (in Objective)

$j \in J$: Set of all days
$l \in L$: Set of all tasks
$k \in K$: Set of all time slots

Constraints

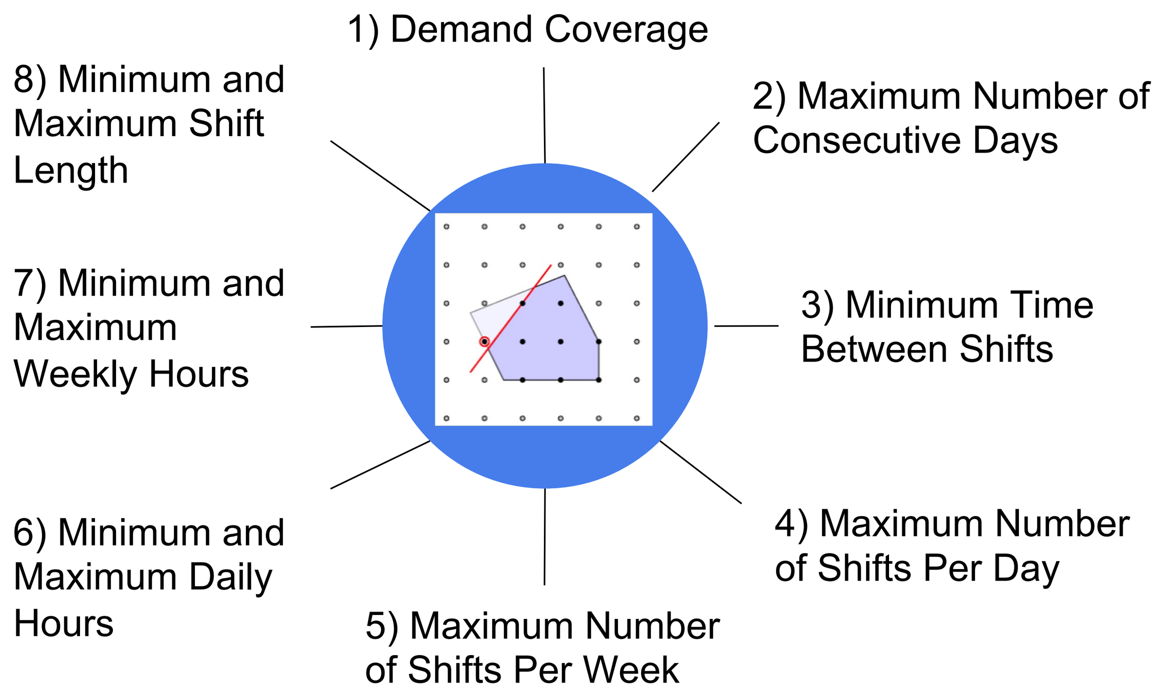

Illustration of Methodology

Inputs:
1. Data: Demand, Employee Availabilities, Employee Skills
2. Constraints Specifications

Solving IP model using Gurobi

Output:
Employee Roster Schedule

Local Diner

3 Types of Work (Cashier, Server, Cook)
30 Employees
24/7 Working Hours

	Day1	Day2	...	Day6	Day7
Employee0	0.0	0.0	...	0 - 7 as cook	0 - 5 as cook
Employee1	0 - 5 as server	11 - 15 as cook	...	11 - 15 as cook	5 - 10 as cook
Employee2	0 - 4 as cook	0 - 7 as server	...	0 - 4 as server	0 - 7 as server
Employee3	5 - 12 as server	10 - 14 as cashier	...	8 - 14 as server	2 - 8 as cashier
Employee4	11 - 16 as cashier	14 - 20 as cashier	...	0.0	0.0
Employee5	0.0	0.0	...	10 - 14 as cashier	8 - 14 as cashier
...	...	...	...	...	...
Employee24	0.0	8 - 13 as server	...	0 - 5 as server	8 - 15 as server
Employee25	0 - 5 as server	0 - 4 as server	...	0.0	0 - 4 as server
Employee26	0.0	7 - 12 as server	...	9 - 15 as server	11 - 15 as server
Employee27	0.0	0.0	...	18 - tomorrow 0 as server	18 - 24 as server
Employee28	8 - 14 as cook	0 - 7 as cook	...	0.0	0.0
Employee29	5 - 11 as server	7 - 14 as cook	...	18 - 22 as server	0.0

An Example Local Diner Roster (First and last 2 days, first and last 6 employees)

Cosmetics Store

2 Types of Work (Zone 1, Zone 3)
66 Employees
12/7 Working Hours

	Day1	Day2	...	Day6	Day7
Employee0	11 - 18 as zone3	14 - 20 as zone3	...	0.0	0.0
Employee1	0.0	15 - 19 as zone1	...	11 - 19 as zone1	0.0
Employee2	0.0	0.0	...	0.0	0.0
Employee3	0.0	13 - 19 as zone3	...	14 - 20 as zone3	0.0
Employee4	10 - 18 as zone1	9 - 15 as zone1	...	10 - 18 as zone1	0.0
Employee5	0.0	0.0	...	0.0	10 - 18 as zone3
...	...	...	...	...	...
Employee60	16 - 20 as zone3	0.0	...	8 - 15 as zone3	9 - 16 as zone3
Employee61	0.0	0.0	...	13 - 20 as zone3	12 - 19 as zone3
Employee62	10 - 18 as zone3	12 - 19 as zone3	...	0.0	0.0
Employee63	9 - 14 as zone3	15 - 19 as zone3	...	0.0	13 - 19 as zone3
Employee64	0.0	0.0	...	9 - 17 as zone3	12 - 19 as zone3
Employee65	16 - 20 as zone3	15 - 20 as zone3	...	0.0	10 - 18 as zone3

An Example Cosmetics Store Roster (First and last 2 days, first and last 6 employees)

Project Impact

- Develops the **first generalized optimization approach** to tackle rostering problems as a variation of the classical nurse scheduling problem
- **Saves money** for the client's customers (avoid paying for overstaffing / losing sales from understaffing)
- **Saves time** by decreasing the effort required later on to adjust the produced schedules manually
- Establishes a **flexible starting framework** that can be extended in the future to accommodate various user inputs

Future Work

The mathematical optimization may take longer for larger/difficult problem instance.

- Develop a hybrid model (optimization on relaxed settings + a heuristic algorithm to improve)
- Incorporate employee preferences to improve their satisfaction
- Explore inverse optimization methods