



MONASH
University

MONASH
INFORMATION
TECHNOLOGY



OPTiMA

ARC TRAINING CENTRE IN
OPTIMISATION TECHNOLOGIES
INTEGRATED METHODOLOGIES
AND APPLICATIONS

CrewAId

Interactive optimisation for human-in-the-loop
crew rostering and rerostering

Matthias Klapperstueck, Frits de Nijs, Ilankaikone Senthooran,
Matteo Miceli, **Michael Wybrow**



Team members



Visualisation



Matthias
Klapperstueck



Michael
Wybrow



Optimisation



Frits de Nijs



Ilankaikone
Senthoooran



Matteo Miceli

The Problem

- Rostering the 24/7 Operation of a Chemical Process Plant
- Set of crews alternating between day and night swings with recovery breaks (2 weeks on / 3 weeks off)
- Rerostering needs to deal with disruption by leave, sickness, training, and changed demand
- Supervisors need solutions they can inspect, adjust, and justify, to support operational decision-making

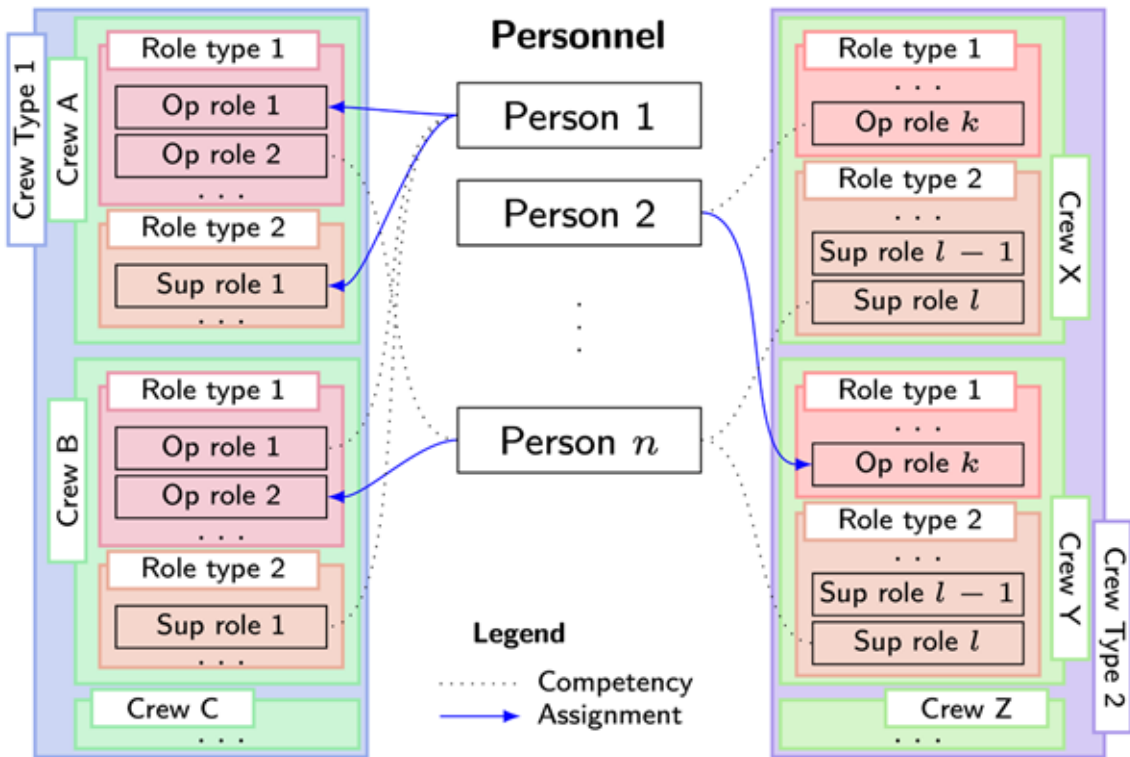


Interactive Crew Rostering

Problem we try and solve

- Assignment of personnel to crews and (multiple) roles within crews.
- Consider multi-skilled personnel, role bounds and training on the job
- Base rosters need to be stable for months or years
- Career stage and trainee status restrict valid assignments
- Good solutions must also be robust to future absences (leave, sickness, and training)
- Personnel can be marked as unavailable or external, or that they must be kept together or apart

Hierarchical nature of Rostering



Role Group	Role	A	B	C	D	E
LTO	LNG	Jonathan Patterson	Jaxon Sullivan	Maya Ward	Scarlett Nguyen	Nova Phillips
LTO	SL	Hunter Butler	Christian Long	Caleb Watson	Gabriel Murphy	Ellie Parker
LTO	UTES	Emilia Perry	Nathan Cruz	Grace Russell	Aurora Robinson	Willow Campbell
LTO	LIQ	Eliana Rogers	Lily Myers	Emily Kelly	John Bailey	Daniel Carter
LTO	ETP	Joshua Gutierrez	Lincoln Morales	Matthew Richardson	Joseph Cooper	Mila Roberts
LTO	PDFS	Serenity Simmons	Julian Howard	Violet Bell	Hazel Rivera	Benjamin Adams

Model for user control

- Hard constraints protect safety: competencies, career stage, role coverage, and availability
- Soft objectives express preferences: robustness, equity, training, and minimal disruption
- MiniZinc enabled CP and MIP-style models, solver comparison, and cross-checking
- Pareto and lexicographic modes expose trade-offs instead of a single opaque answer

Hard constraints

1. A person can be allocated to at most one crew
2. Regular, available personnel must be allocated a crew
3. A person can only take roles in their allocated crew
4. A person must take on a mandatory role in their crew
5. A person is only allocated roles they are skilled in
6. A person can be allocated at most one role per role type
7. Sufficient personnel are allocated to all roles in a crew
8. Certain roles are not over-subscribed
9. Personnel are assigned roles appropriate to their career stage
10. Trainees are assigned trainee roles

Two models

- Mixed Integer Programming (MIP)
- Constraint Programming (CP)
- Written in MiniZinc
- Common input and output parameters
- Benefits: Rapid prototyping, verify correctness, compare impacts, identify subtle misunderstandings of domain experts' requirements

Why interactive crew rostering?

- Rostering is still often manual, despite decades of optimisation research
- Transfer fails when tools ignore uncertainty, operational systems, or user trust
- CrewAid lets supervisors explore, override, compare, and repair optimisation outputs

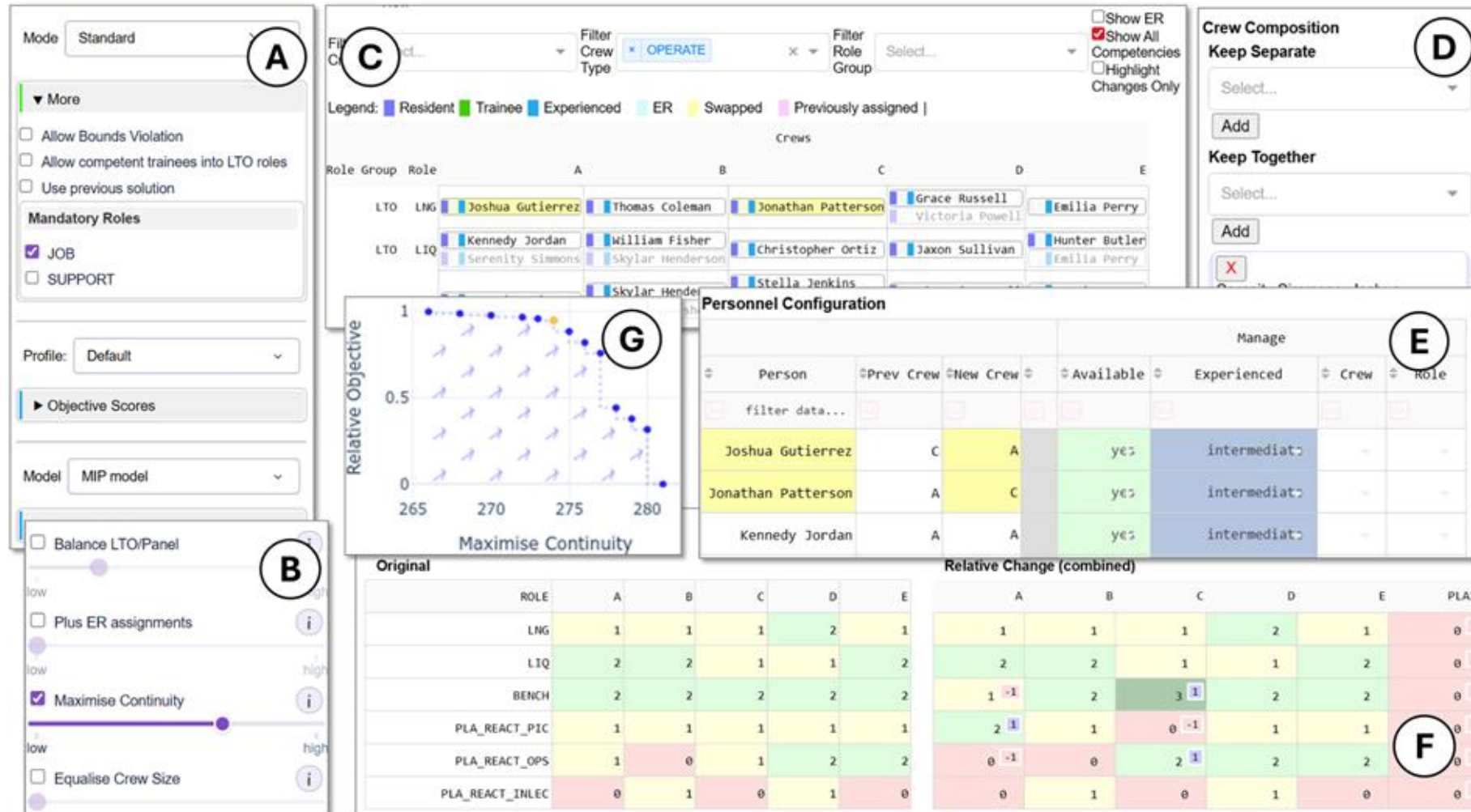
Objectives

1. Distribute space competencies across crews
(one objective per role type)
2. Maximise continuity compared with a previous roster
3. Equalise crew sizes
4. Equalise personnel experience levels across crews
5. Distribute trainees across crews
6. Assign personnel to roles close to their residential area.

Maximising Crew Robustness

- Strictly satisfying the minimum leaves crews vulnerable to disruptions from personnel absence
- Two stage model formulation minimises chance that a single-person absence will incur an external replacement
- First stage finds crew with maximum personnel and skill redundancy
- Second stage object object measures crew robustness, testing feasibility when each crew member is reassigned

Rostering User Interface

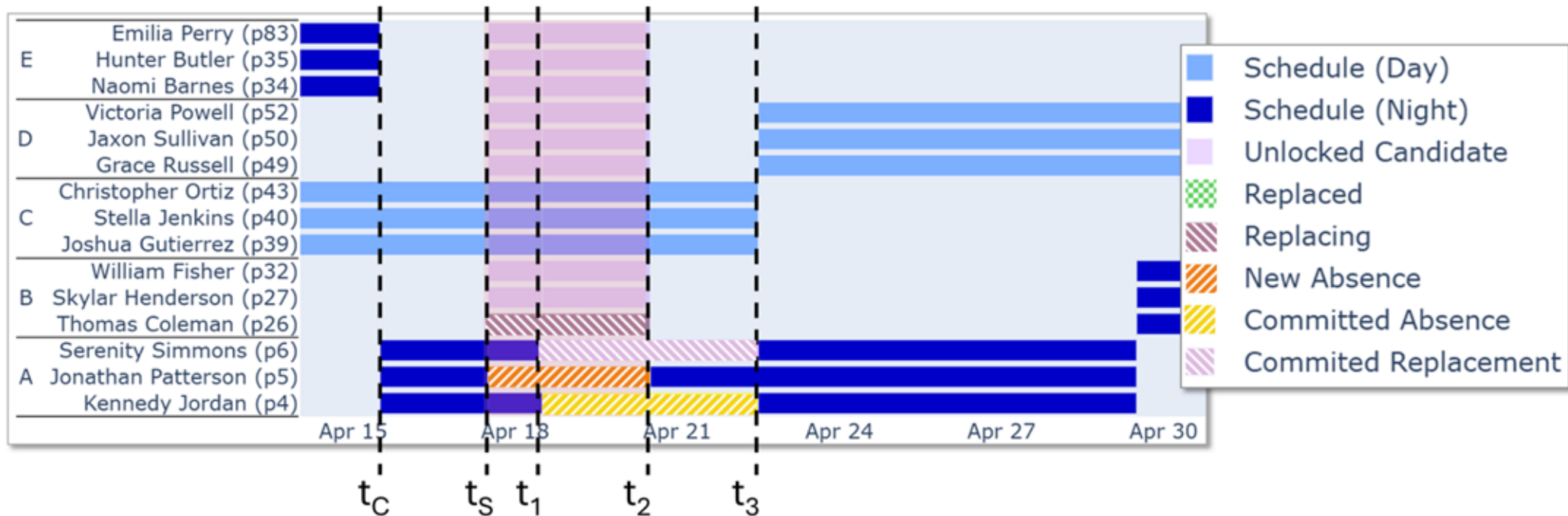


Interactive Crew Rerostering

Problem we try and solve

- Rostering is assignment of personnel without considering time, rostering works with an actual schedule
- Absence requests arrive one at a time
- Previous solutions become commitments
- A new request may overlap with earlier absences or replacements
- Replacement choice depends on skills, availability, restedness, and shift history
- The model solves the relevant timeline window, grouped into tiers where context changes

Rerostering over time



Approach

- Wider context window, typically 35 days before and after
- We group together days with same context into tiers
 - New tiers when 1) active crew changes, 2) new absence, 3) personnel become absent
- Crew allocation done for active crews within tiers
- Off-swing personnel don't need to be allocated.
- Constraint to stop 24-hour shifts
- Four scenarios, using tailored objective functions

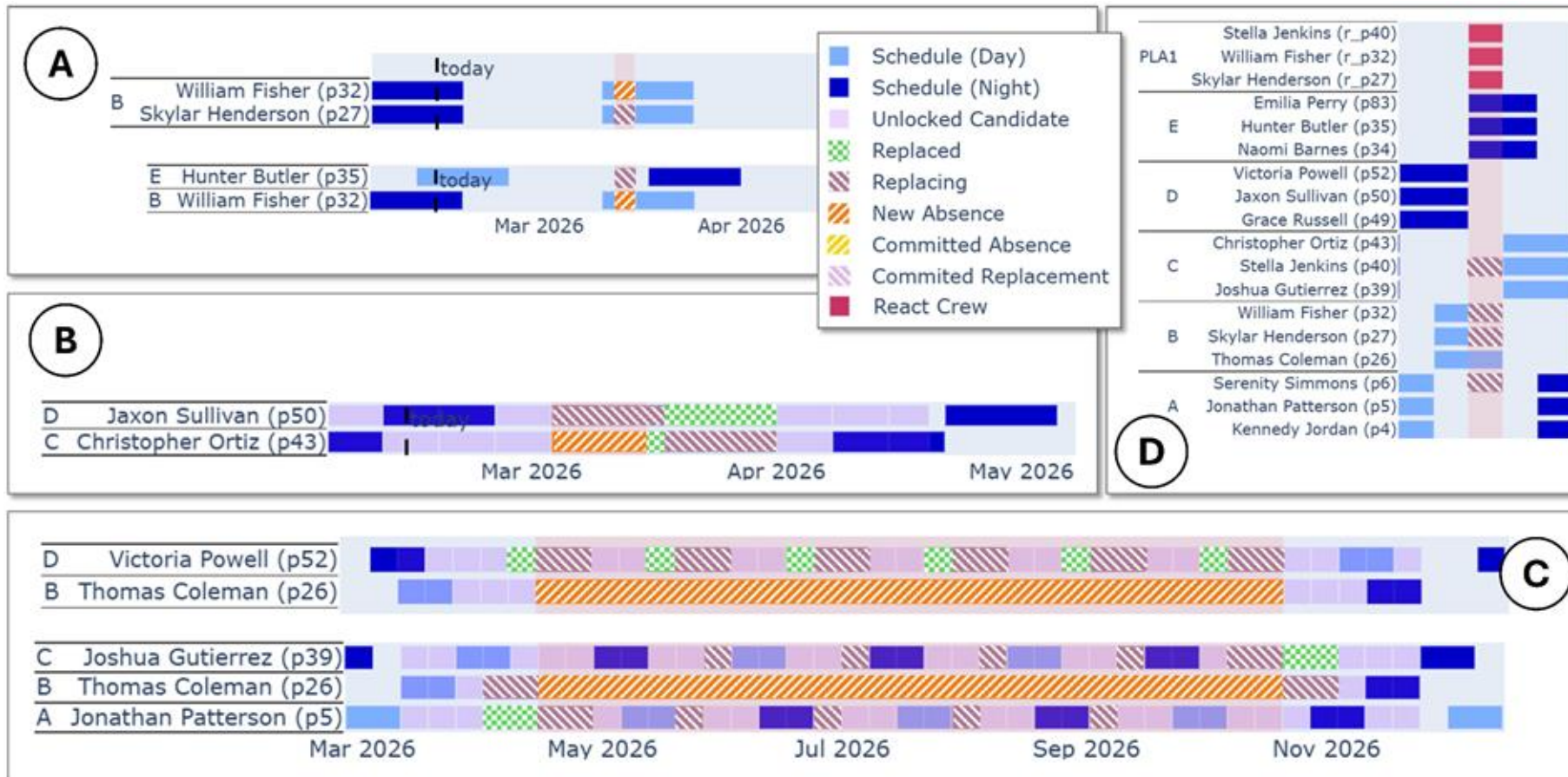
Objective components

1. Crew and role changes compared to the original roster
2. Changes to the original shift plan in terms of working and off days
3. Overcycled days (excess working days beyond the planned roster)
4. The number of overcyclers (external personnel brought in)
5. The number of person-days with short breaks (e.g., less than 14 days between shift cycles)
6. The number of person-days exceeding a normal shift length (e.g., more than 21 days)
7. The number of personnel whose assignments differ from the original roster
8. The number of times personnel change from day to night shift in consecutive tiers
9. The number of assignment changes for each person (promoting roster smoothness)
10. Deviations in total working days compared to the original roster
11. Cost of forming specialised maintenance crew

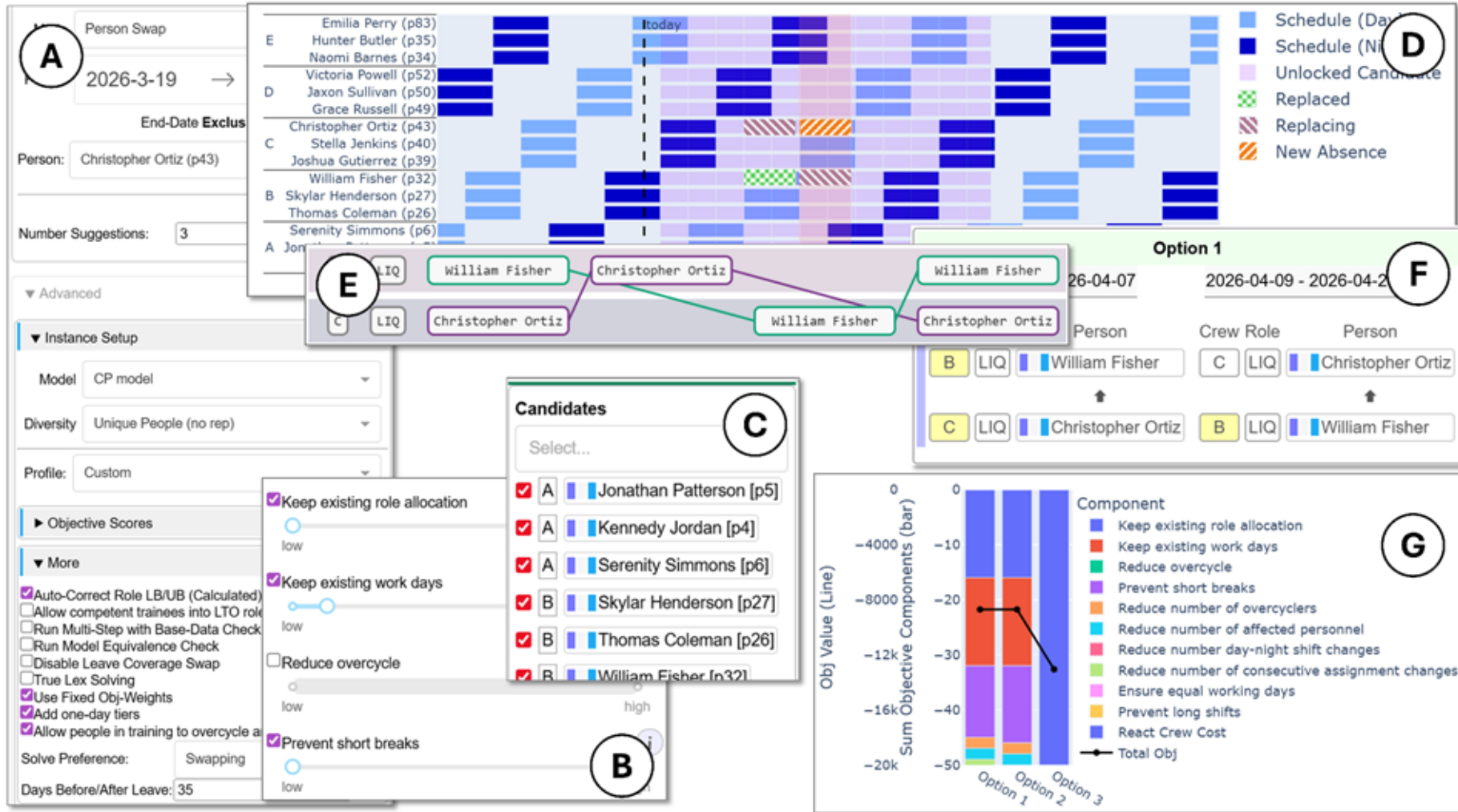
Four scenarios (modes)

- Short absence
 - Unplanned sickness or planned training
 - Prefer internal swaps; overcycle only if needed
- Swing trade
 - Absence from usual swing
 - Exchange work fairly across nearby swings
- Long absence
 - Medical leave, Parental leave
 - Temporarily move or repeatedly backfill a role
- Maintenance crew
 - Form a temporary skilled team at minimum disruption

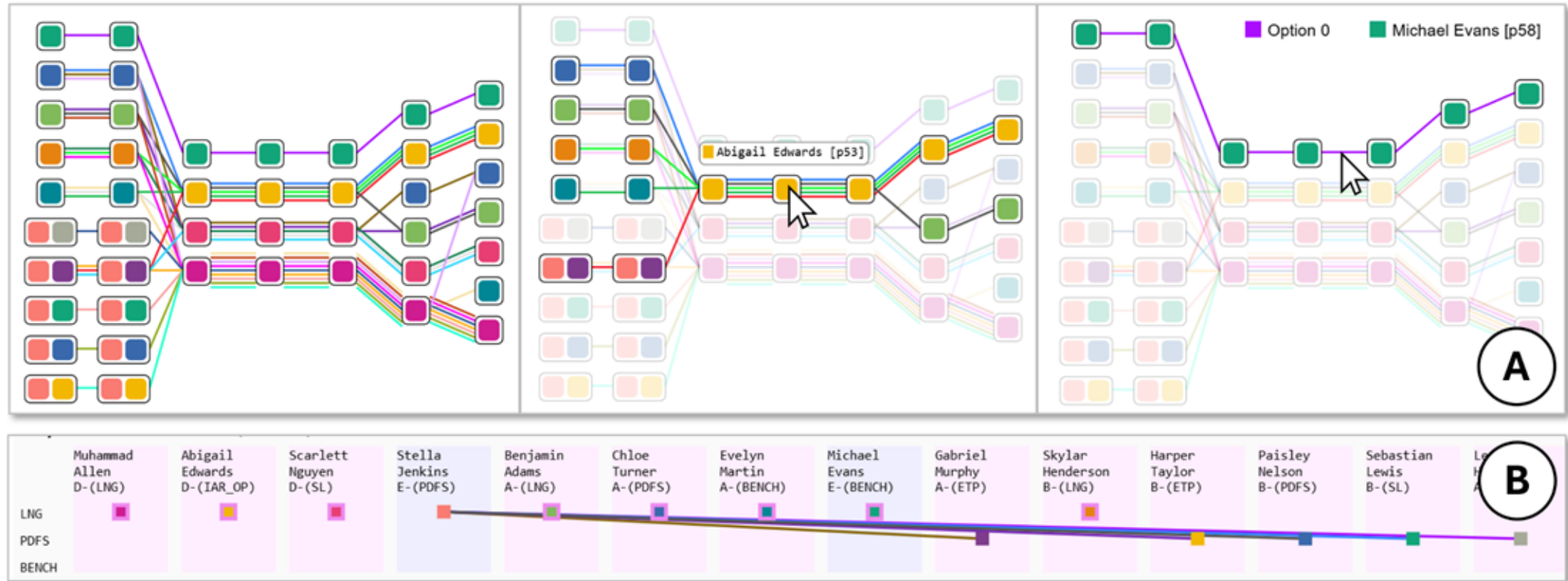
Solution examples for Rerostering Modes



Rerostering User Interface



Visualising the Solution Space



Repairing Infeasible Instances using Human-Centric Restoration

Infeasibility as a decision

- Solvers can report infeasible; users need to see which assumptions can move
- CrewAld supports slack minimisation and MUS enumeration through MiniZinc
 - Based on previous work: Ilankaikone Senthooran, Matthias Klapperstueck, Gleb Belov, Tobias Czauderna, Kevin Leo, Mark Wallace, Michael Wybrow, and Maria Garcia de la Banda. **Human-centred feasibility restoration in practice**. Constraints, 28:203–243, 2023.
- The UI lets users choose violable constraints, inspect conflicts, and solve again

Infeasibility is part of the workflow

- Infeasible does not mean “no useful answer”
- It means the current assumptions conflict
- Domain users know which assumptions may be relaxed
- The system must expose conflicts, not hide them
- Restoration turns solver failure into a guided decision

Restoration Interface

A

Step 1

The current solution is infeasible. Use the controls to start the restoration process.

Restoration Mode findMUS

Allowed Violations

- ☒ Allow not to assign mandatory roles to a person
- ☒ Allow Lower Bounds Violation
- ☒ Allow Upper Bounds Violation
- ☒ Allow Inclusions Violation
- ☒ Allow Exclusions Violation
- ☒ Allow Keep Crew Violation
- ☒ Allow Keep Role Violation
- ☒ Allow competent trainees into LTO roles

Findmus Settings

Num Muses

Repair - Step1

Step 2

Run Soft Optimisation with selected MCS.

Repair - Step2

Final Step

Applying correction values shown in the Slack Table to input and rerun.

Restore Infeasible Allocation

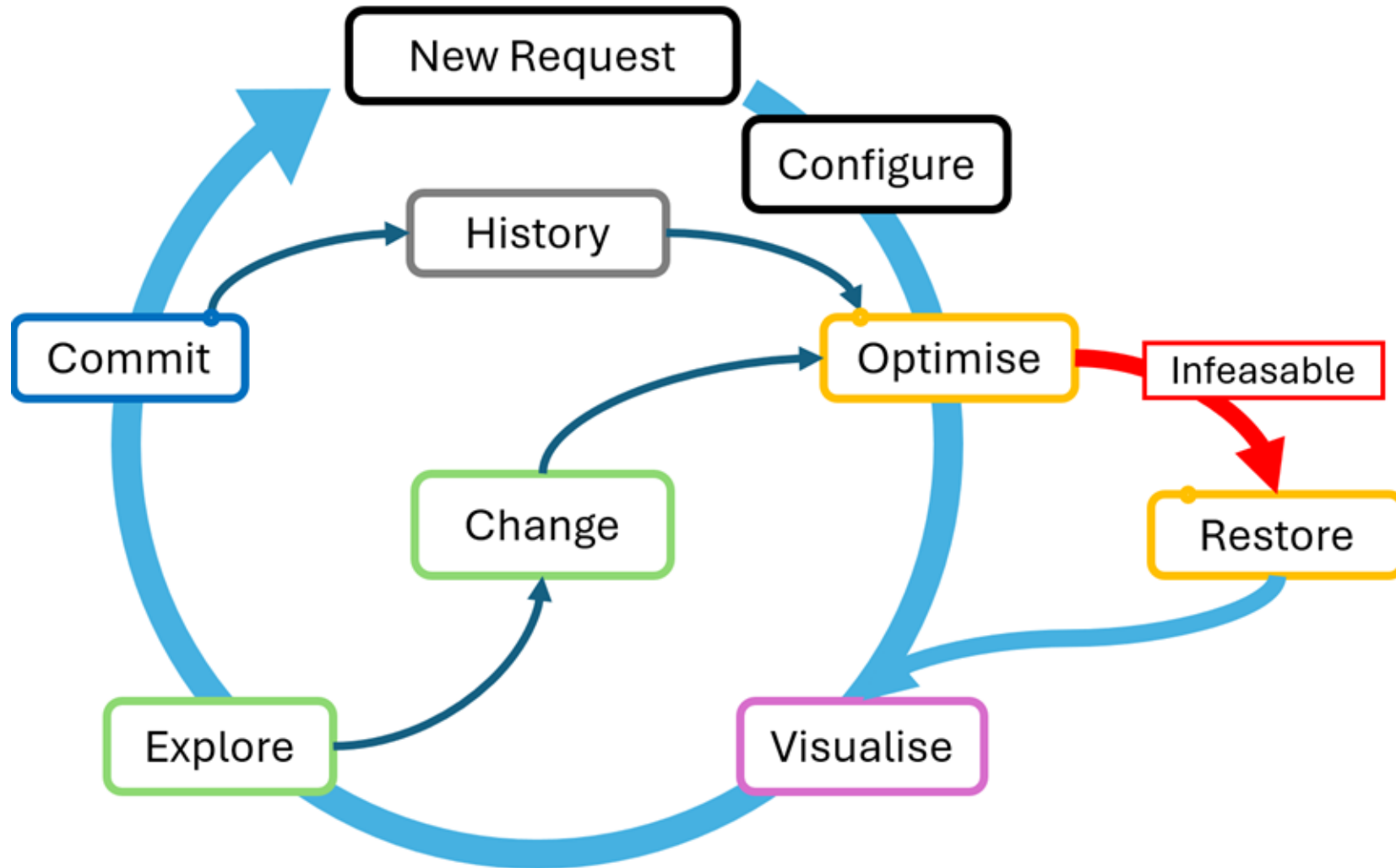
MUS	Conflict type	Members	Description	Select for repair
MUS 1				
MUS 2	INCLUSION	Kennedy Jordan,Zoey Hughes	Cannot put 'p1, p23' together in one crew.	☐
MUS 3	ROLE_LB	ROLE: PANEL_FOCAL	Role 'PANEL_FOCAL' cannot find enough personnel.	☐

B

Type	Description	Correction Value
INCLUSION	Inclusion {"person_inclusion {"p23", 'p1'"}, Action is to Remove entry	1

D

Workflow



Conclusions

- Deployable CP applications needs combination of the model, the workflow, and the explanation interface
- Partner-scale instances solve fast enough for interactive decision support
- The interface keeps domain expertise in control while making decisions consistent
- Industry partner is looking to build on the work for operational use

Thank You
—
Questions?

Performance

- Our data-driven system scales to our industry partner's real-world data
 - 100 personnel with individual sets of competencies and varying availability
 - Operating within 12 crews, having 3 crew types and 3 role types.
- Instances of crew rostering typically solve in under 10 seconds using MiniZinc with Gurobi or OR Tools CP-SAT