A Constraint Programming Model for the Electric Bus Assignment Problem with Parking Constraints

Mathis Azéma, Guy Desaulniers, Jorge Mendoza, Gilles Pesant

CPAIOR 2024, Uppsala, Sweden

POLYTECHNIQUE MONTRÉAL
WORLD-CLASS ENGINEERING

1. Problem statement
2. Literature
3. MILP formulation
4. Constraint Programming Formulation

Introduction

- The transportation sector accounted for 20.2% of the global CO_2 emissions in 2021 → development of electric buses.

Introduction

- The transportation sector accounted for 20.2% of the global CO_2 emissions in 2021 → development of electric buses.

	diesel bus	electric bus
Refuel	Few minutes	Several hours
Autonomy	High	Low

⟹ Need to take into account charging events in schedules.

Introduction

- The transportation sector accounted for 20.2% of the global CO_2 emissions in 2021 → development of electric buses.

	diesel bus	electric bus
Refuel	Few minutes	Several hours
Autonomy	High	Low

⟹ Need to take into account charging events in schedules.

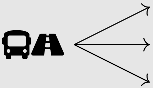
Problem specificity

→ In the Nordic countries, buses are parked in closed depots between trips.

⟹ **Existence of parking constraints.**

Problem description

Trip = sequence of bus routes



starting time

ending time

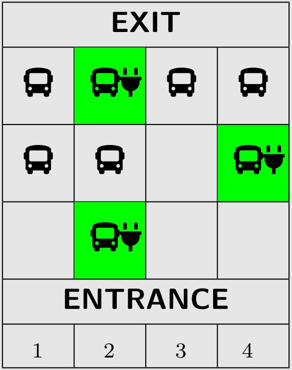
energy consumption

Problem description

Trip = sequence of bus routes

- starting time
- ending time
- energy consumption

Depot



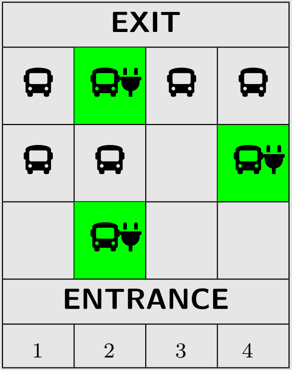
Problem description

Trip = sequence of bus routes

→ starting time
→ ending time
→ energy consumption

Constraints

Depot



Problem description

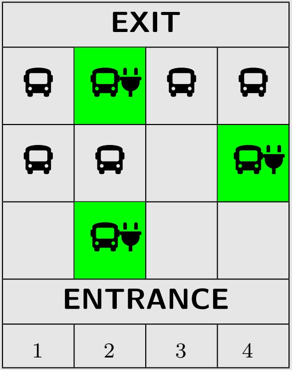
Trip = sequence of bus routes

- starting time
- ending time
- energy consumption

Constraints

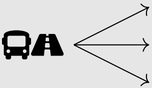
→ Trip assignment

Depot



Problem description

Trip = sequence of bus routes



starting time

ending time

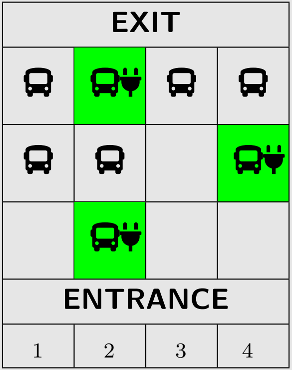
energy consumption

Constraints

→ Trip assignment

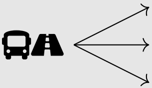
→ Which parking lane after each trip?

Depot



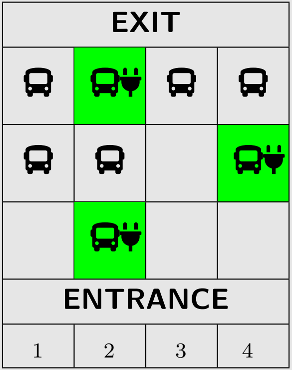
Problem description

Trip = sequence of bus routes



- starting time
- ending time
- energy consumption

Depot



Constraints

→ Trip assignment

→ Which parking lane after each trip?

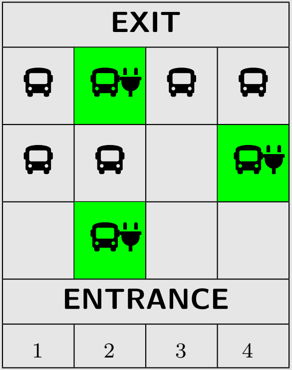
→ FIFO policy in each lane

Problem description

Trip = sequence of bus routes

→ starting time
→ ending time
→ energy consumption

Depot

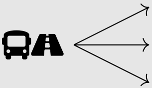


Constraints

- → Trip assignment
- → Which parking lane after each trip?
- → FIFO policy in each lane
- → Battery limits

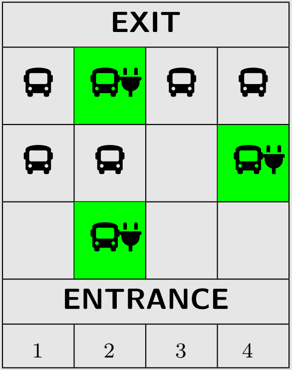
Problem description

Trip = sequence of bus routes



- starting time
- ending time
- energy consumption

Depot



Constraints

→ Trip assignment

→ Which parking lane after each trip?

→ FIFO policy in each lane

→ Battery limits

→ depot charging capacity (= Number of chargers: they are available from any slot)

Problem description

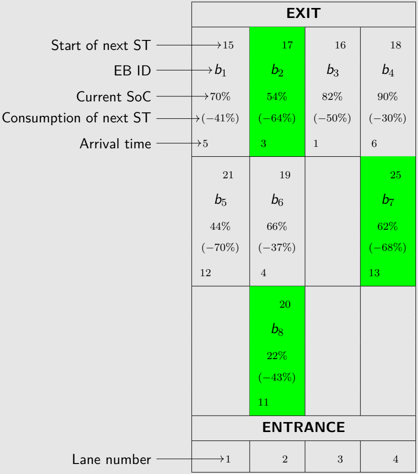


Fig. 1: Example of a depot at time $t = 14$. EBs in green slots are charging.

1 Problem statement

2 Literature

3 MILP formulation

4 Constraint Programming Formulation

Literature

electric Vehicle Scheduling Problem

Some features of eBAP-PC are already studied:

- linear charging time
- Partial charging
- Charging capacity constraints

Literature

electric Vehicle Scheduling Problem

Some features of eBAP-PC are already studied:

- linear charging time
- Partial charging
- Charging capacity constraints

Parking constraints

Few authors dealt with them, but none responded to the eBAP-PC

Literature

electric Vehicle Scheduling Problem

Some features of eBAP-PC are already studied:

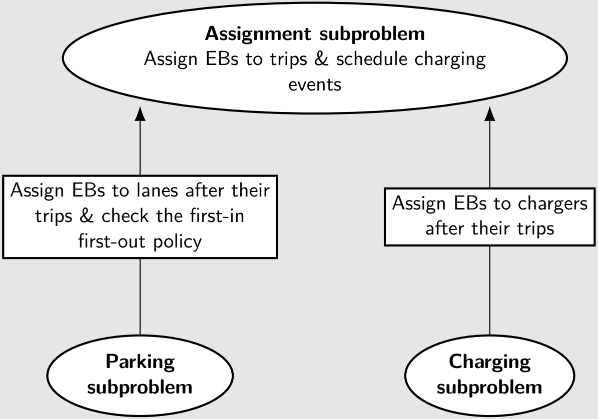
- linear charging time
- Partial charging
- Charging capacity constraints

Parking constraints

Few authors dealt with them, but none responded to the eBAP-PC

The literature contains many MILP models but very few in Constraint Programming

A three-graph representation



One graph for each subproblem. Nodes are the trips and a path represents the trips assigned to either a bus or a lane or a charger.

$\Longrightarrow$ Three flow models with additional constraints linking them.

Acceleration strategies

- **Graph reduction**
  → Initial idea: A bus does not stay in the depot for 24 consecutive hours

Acceleration strategies

- **Graph reduction**
  - → Initial idea: A bus does not stay in the depot for 24 consecutive hours
- **Decomposition in two steps**

Step 1	Step 2
eBAP-PC without the constraint on the number of chargers	Find a feasible schedule

Acceleration strategies

- **Graph reduction**
  - → Initial idea: A bus does not stay in the depot for 24 consecutive hours
- **Decomposition in two steps**

Step 1	Step 2
eBAP-PC without the constraint on the number of chargers	Find a feasible schedule

- **Objective function**: penalization of some schedule features

Assignment graph	Parking graph
deviation to a target parking time: $4h$	deviation to a target waiting time between two arrivals in a lane: $1h$

1 Problem statement

2 Literature

3 MILP formulation
- A three-graph representation
- Acceleration strategies
- Results

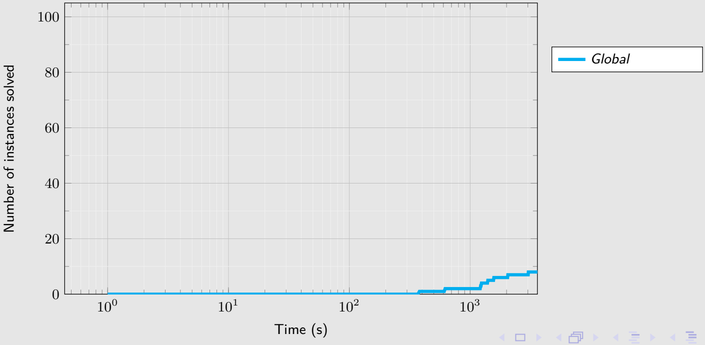
4 Constraint Programming Formulation

Data

- Two instances provided by our industrial partner GIRO Inc., a world-leading developer of optimization solutions for public transit
  - → 30 bus, 98 trips/day
  - → 42 bus, 125 trips/day
- Generating random instances from these 223 trips to assess the robustness of solution methods with respect to the number of buses, number of trips/bus/day, number of days and size of the depot

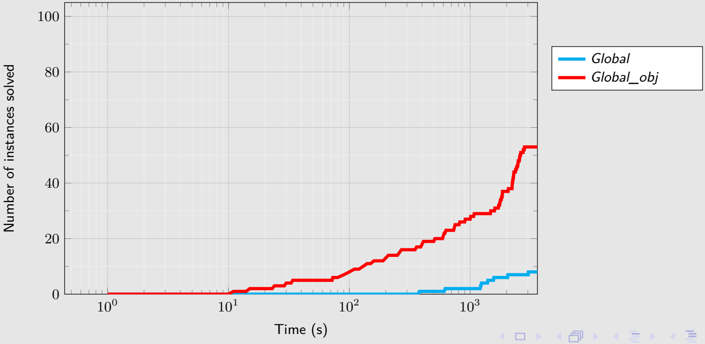
Results: 30-40 bus, 2-3 days, 2-3 trips/bus/day

Method	2-step	Objective	Graph reduction
Global			X



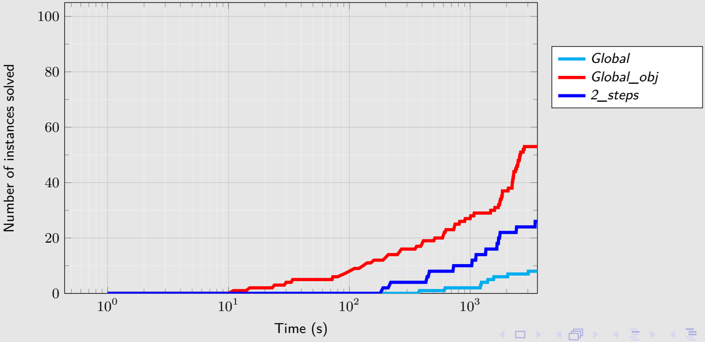
Results: 30-40 bus, 2-3 days, 2-3 trips/bus/day

Method	2-step	Objective	Graph reduction
Global			X
Global_obj		X	X



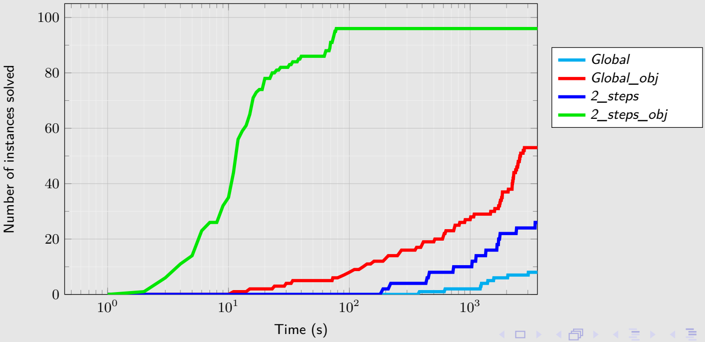
Results: 30-40 bus, 2-3 days, 2-3 trips/bus/day

Method	**2-step**	**Objective**	**Graph reduction**
Global			X
Global_obj		X	X
2_steps	X		X



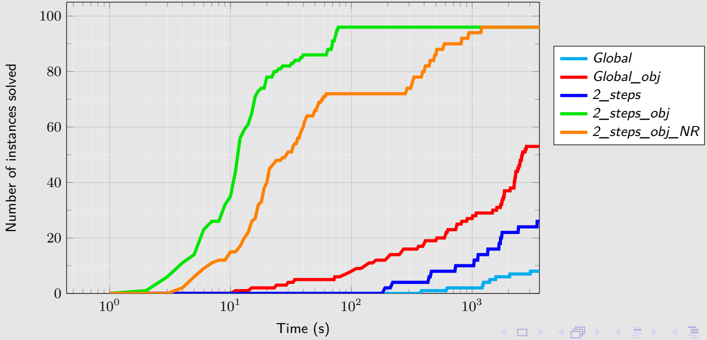
Results: 30-40 bus, 2-3 days, 2-3 trips/bus/day

Method	2-step	Objective	Graph reduction
Global			X
Global_obj		X	X
2_steps	X		X
2_steps_obj	X	X	X



Results: 30-40 bus, 2-3 days, 2-3 trips/bus/day

Method	2-step	Objective	Graph reduction
Global			X
Global_obj		X	X
2_steps	X		X
2_steps_obj	X	X	X
2_steps_obj_NR	X	X	



Constraint Programming scheduling model

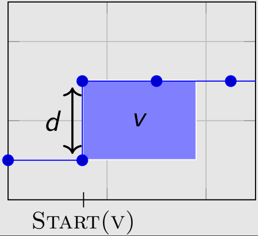
- *Interval variables v* representing each task (trip, parking, charging) are defined by :
  - $\textsc{Pres}(v)$: boolean
  - $\textsc{Start}(v)$: integer
  - $\textsc{Length}(v)$: integer

Constraint Programming scheduling model

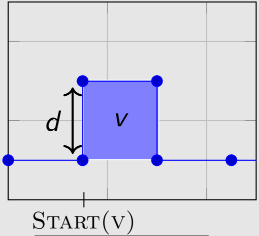
- *Interval variables v* representing each task (trip, parking, charging) are defined by :
  - $\textsc{Pres}(v)$: boolean
  - $\textsc{Start}(v)$: integer
  - $\textsc{Length}(v)$: integer
- *Sequence variables* :
  - Set of interval variables
  - Specific constraints : $\textsc{NoOverlap}$, $\textsc{Previous}$, ...

Constraint Programming scheduling model

- *Interval variables* v representing each task (trip, parking, charging) are defined by :
  - $\textsc{Pres}(v)$: boolean
  - $\textsc{Start}(v)$: integer
  - $\textsc{Length}(v)$: integer
- *Sequence variables* :
  - Set of interval variables
  - Specific constraints : NoOverlap, Previous, ...
- *Cumulative functions* :



$\textsc{StepAtStart}(v, d)$



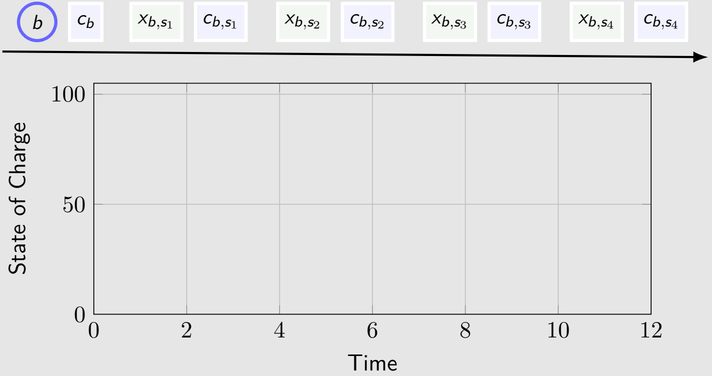
$\textsc{Pulse}(v, d)$

CP formulation: two resources to model

- Bus batteries :
  - $\rightarrow$ Consumed by trips.
  - $\rightarrow$ Produced by charging events in the depot.

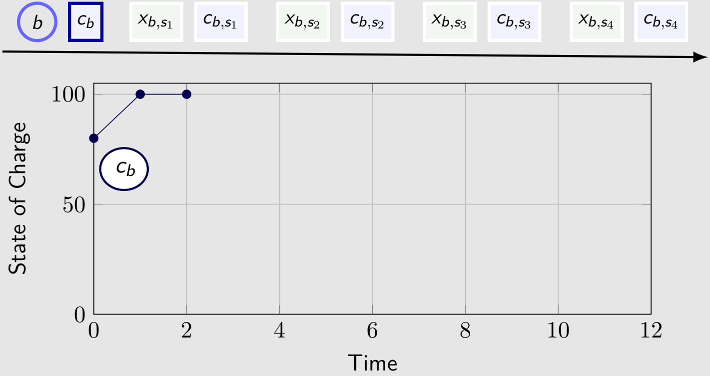
CP formulation: two resources to model

- Bus batteries :
  - → Consumed by trips.
  - → Produced by charging events in the depot.



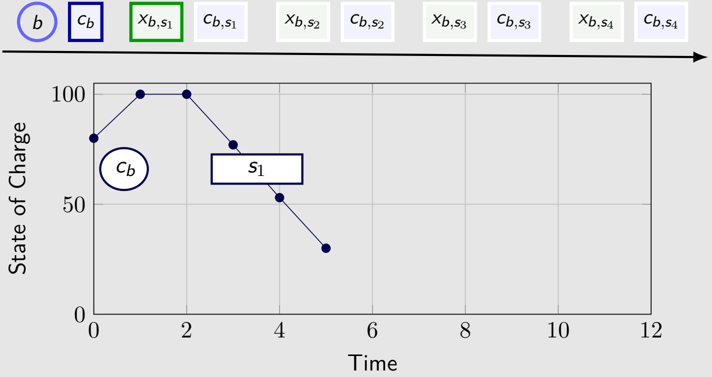
CP formulation: two resources to model

- Bus batteries :
  - → Consumed by trips.
  - → Produced by charging events in the depot.



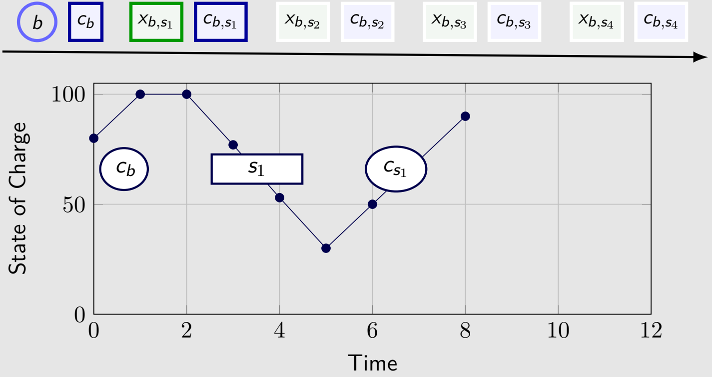
CP formulation: two resources to model

- Bus batteries :
  - → Consumed by trips.
  - → Produced by charging events in the depot.



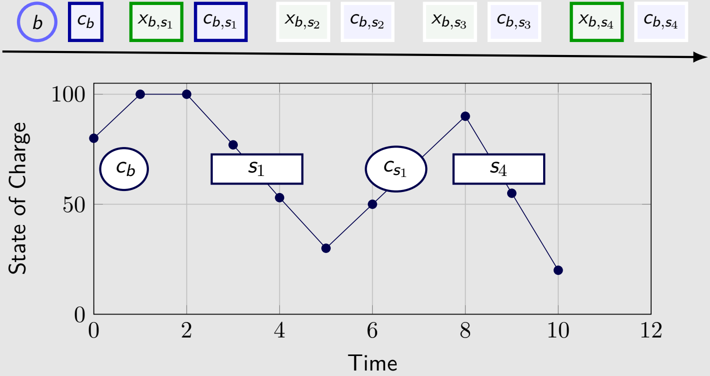
CP formulation: two resources to model

- Bus batteries :
  - → Consumed by trips.
  - → Produced by charging events in the depot.



CP formulation: two resources to model

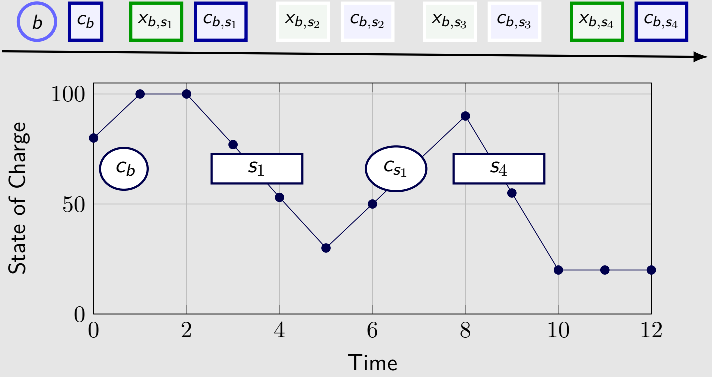
- Bus batteries :
  - → Consumed by trips.
  - → Produced by charging events in the depot.



Mathis Azéma, Guy Desaulniers, Jorge Mendoza, Gilles Pesant

CP formulation: two resources to model

- Bus batteries :
  - → Consumed by trips.
  - → Produced by charging events in the depot.

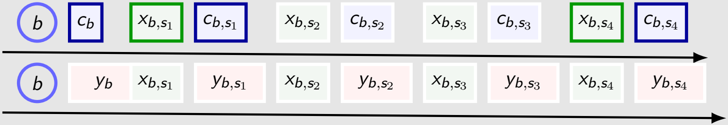


Formulation: two resources to model

- Lanes in the depot :
  - → Buses consume one unit during their parking events.

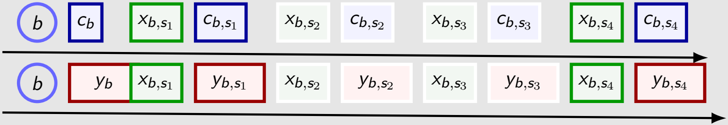
Formulation: two resources to model

- Lanes in the depot :
  - → Buses consume one unit during their parking events.



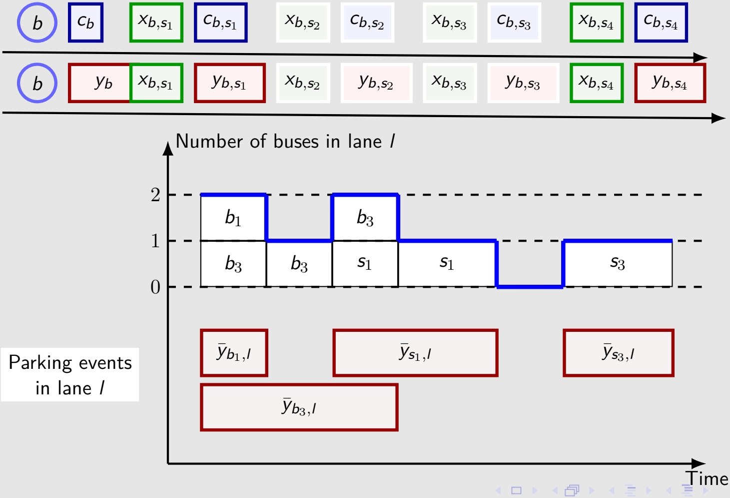
Formulation: two resources to model

- Lanes in the depot :
  - → Buses consume one unit during their parking events.



Formulation: two resources to model

- Lanes in the depot :
  - → Buses consume one unit during their parking events.



Results

Two solution methods

→ *CP-searchPhase*: search phase with all the variables except the charging variables

→ *CP*: No search phase

Results

Two solution methods

→ *CP-searchPhase*: search phase with all the variables except the charging variables

→ *CP*: No search phase

Instances with two different horizons

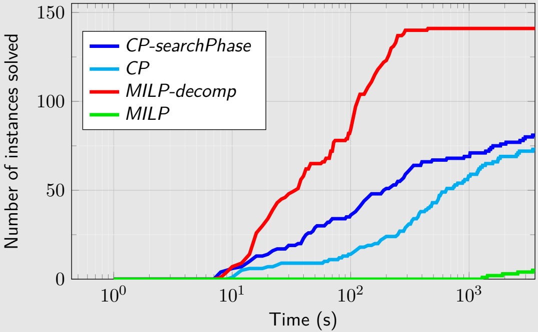
→ Three-day horizon

→ One-day horizon with minimum end-of-day charge level constraints

  → Trips are the same every day. The schedule is repeated from one day to the next, ensuring that the buses are sufficiently recharged at the end of the day.

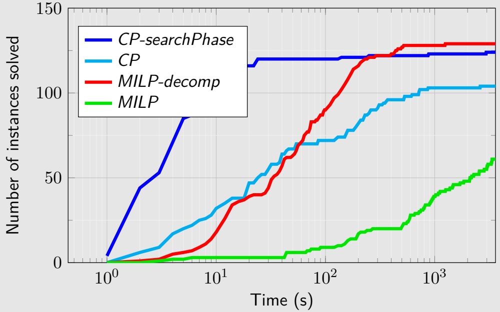
Results: Three-day horizon

Instance	*CP*	*CP-searchPhase*	*MILP-decomp*	*MILP*
real_30/5/3/0.5	643	396	13	X
real_30/6/3/0.5	287	X	41	X
real_30/7/3/0.5	X	2047	8	X
real_42/5/3/0.5	286	X	447	X
real_42/6/3/0.5	X	X	176	X
real_42/7/3/0.5	X	2972	212	X



Results: One-day horizon

Instance	*CP*	*CP-searchPhase*	*MILP-decomp*	*MILP*
real_30/6/1/0.5	3	1	4	2839
real_42/6/1/0.5	8	3	41	X



Note: In the future, there may be instances with 150 buses. Solving the problem in a matter of seconds or minutes is significant.

Conclusion

- MILP formulation :
  - A three-graph representation.
  - Development of three acceleration strategies.
  - Find a schedule of three days in less than 5 minutes for 50 buses.

Conclusion

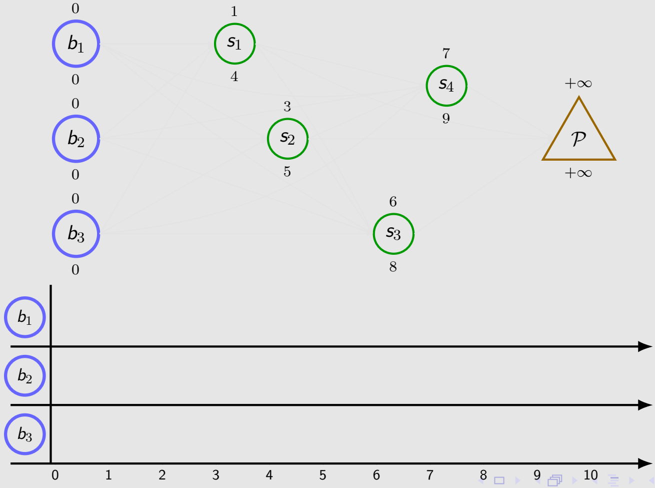
- MILP formulation :
  - A three-graph representation.
  - Development of three acceleration strategies.
  - Find a schedule of three days in less than 5 minutes for 50 buses.
- CP formulation :
  - Based on interval variables consuming/producing 2 resources: batteries and lanes.
  - Less effective for multi-day horizon.
  - Very effective for repeated one-day horizon with minimum end-of-day charge level constraints.

Thank you !

A three-graph representation

1^{st} graph: Assign buses to trips

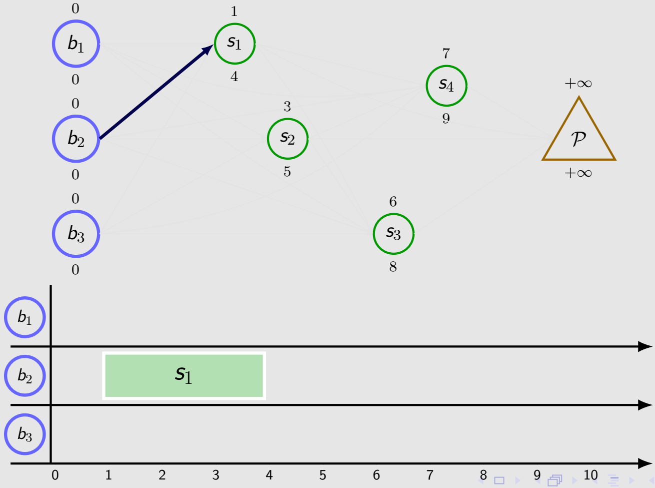
A path is the trips performed by a bus



A three-graph representation

1^{st} graph: Assign buses to trips

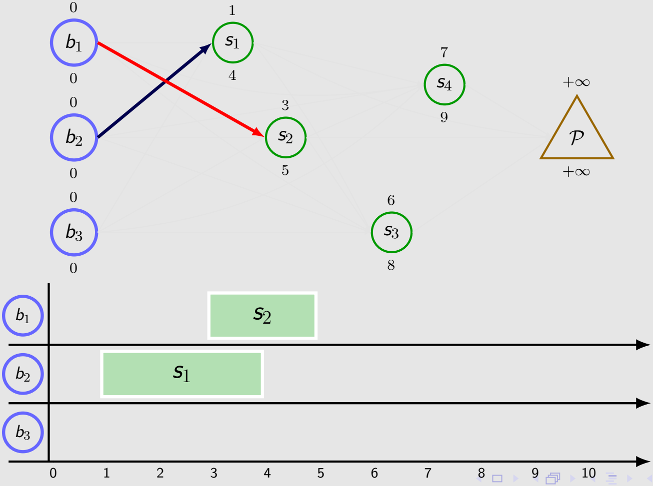
A path is the trips performed by a bus



A three-graph representation

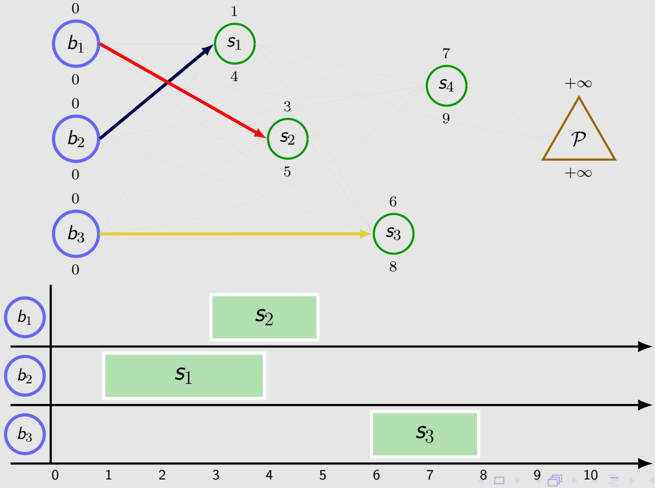
1^{st} graph: Assign buses to trips

A path is the trips performed by a bus



A three-graph representation

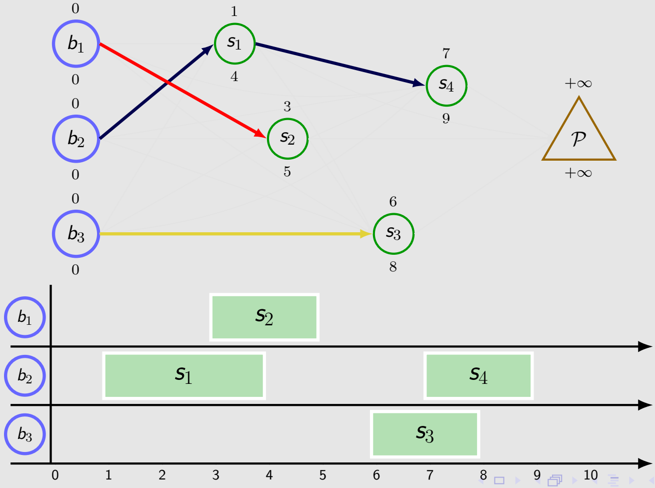
1^{st} graph: Assign buses to trips
A path is the trips performed by a bus



A three-graph representation

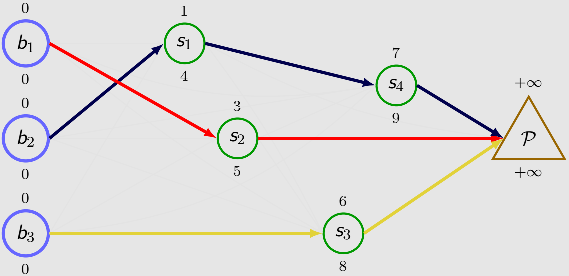
1^{st} graph: Assign buses to trips

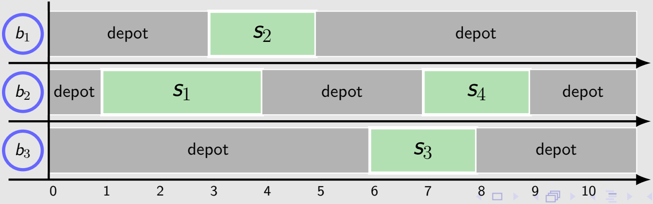
A path is the trips performed by a bus



A three-graph representation

1^{st} graph: Assign buses to trips
A path is the trips performed by a bus

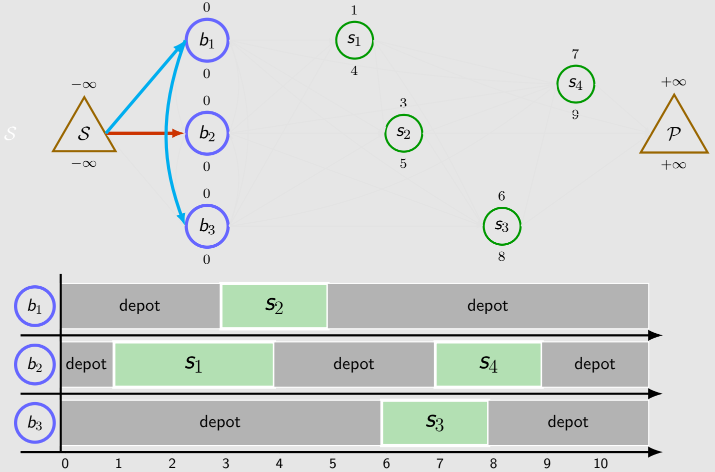




A three-graph representation

2^{nd} graph: Assign lanes to trips

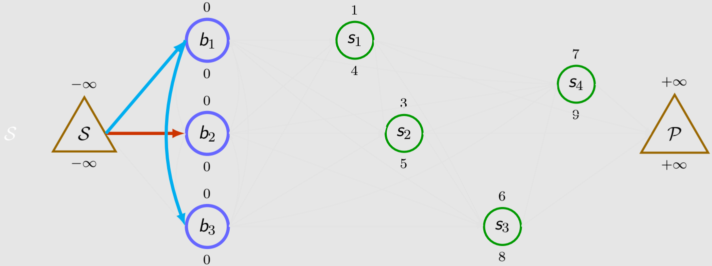
A path is the "trips" parked in a same lane

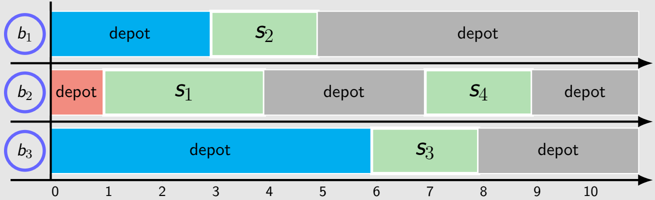


A three-graph representation

2^{nd} graph: Assign lanes to trips

A path is the "trips" parked in a same lane

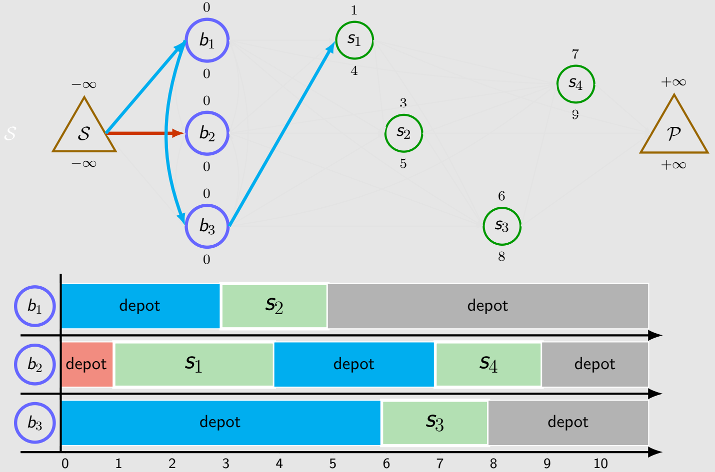




A three-graph representation

2^{nd} graph: Assign lanes to trips

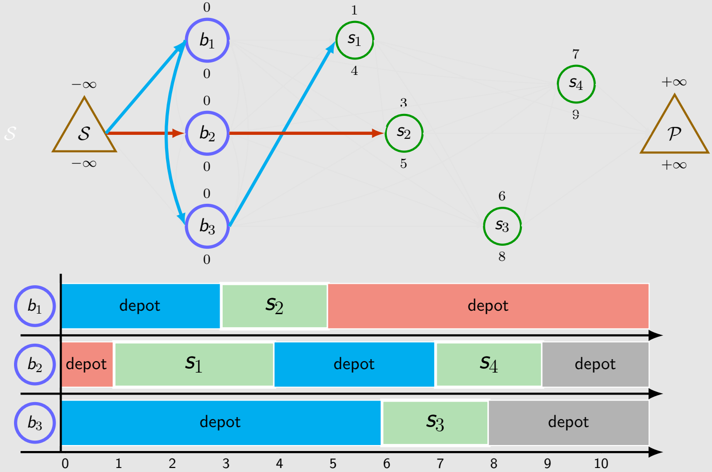
A path is the "trips" parked in a same lane



A three-graph representation

2^{nd} graph: Assign lanes to trips

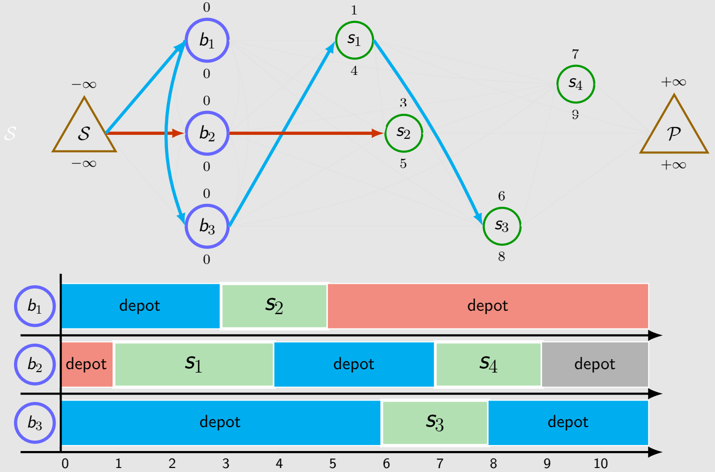
A path is the "trips" parked in a same lane



A three-graph representation

2^{nd} graph: Assign lanes to trips

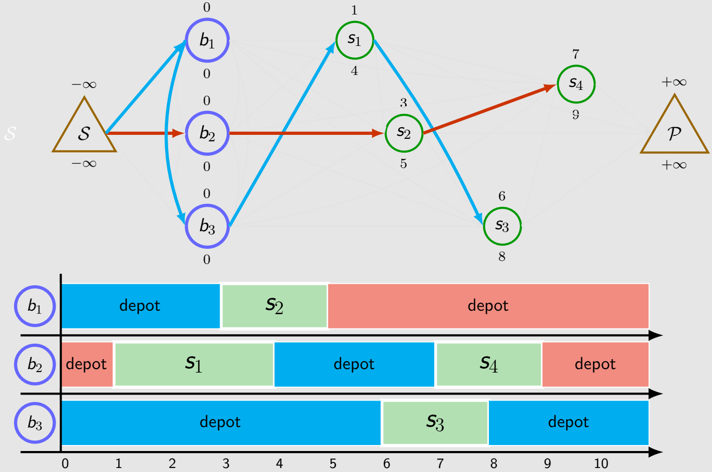
A path is the "trips" parked in a same lane



A three-graph representation

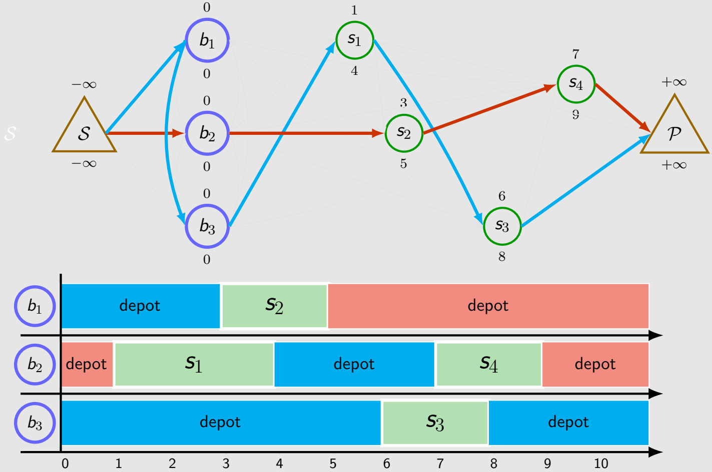
2^{nd} graph: Assign lanes to trips

A path is the "trips" parked in a same lane



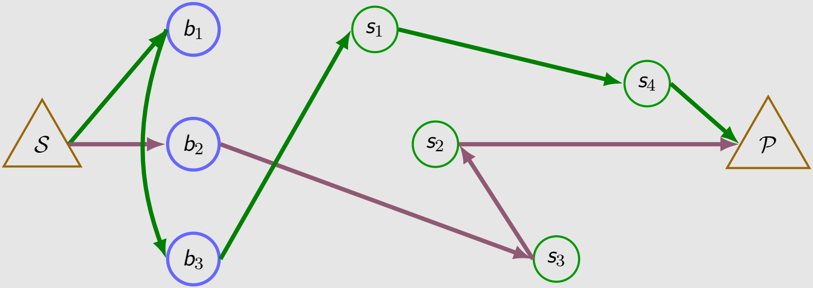
A three-graph representation

2^{nd} graph: Assign lanes to trips
A path is the "trips" parked in a same lane



A three-graph representation

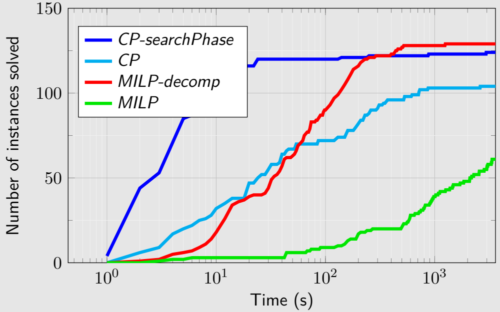
3^{rd} graph: Assign chargers to trips
A path is the "trips" charged by a same charger



Results: One-day horizon

Final fleet energy $\geq \alpha \cdot$ Initial fleet energy / Initial fleet energy $= 80\%$

Instance	α	*CP*	*CP-searchPhase*	*MILP-decomp*	*MILP*
real_30/6/1/0.5	0.8	3	1	4	2839
real_30/6/1/0.5	0.9	3	1	3	X
real_30/6/1/0.5	1	9	3	14	1257
real_42/6/1/0.5	0.8	8	3	41	X
real_42/6/1/0.5	0.9	8	3	62	X
real_42/6/1/0.5	1	7	3	51	X



Note: In the future, there may be instances with 150 buses. Solving the problem in a matter of seconds or minutes is significant.