Identifying Suitable CG Formulations of Some Fundamental Optimization Problems

OeGOR Summer-School 2024, Krems

Timo Gschwind
Lehrstuhl für BWL, insb. Logistik

Rheinland-Pfälzische Technische Universität Kaiserslautern Landau

July 22 – 26, 2024

Recall: Vehicle Routing Problem with Time Windows

Master program:

$$\min \sum_{r \in \Omega} c_r \lambda_r$$

$$\text{s.t.} \sum_{r \in \Omega} a_{ir} \lambda_r = 1 \quad \forall\, i \in N$$

$$\sum_{r \in \Omega} \lambda_r \leq |K|$$

$$\lambda_r \in \{0, 1\} \quad \forall\, r \in \Omega$$

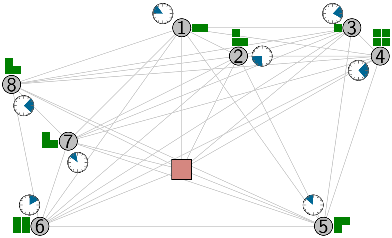
- Variables λ_r correspond with feasible routes $r \in \Omega$

  c_r cost of route r
  a_{ir} indicator if route r services customer i

- Select the best routes such that
  - → total routing costs are minimum
  - → each customer serviced exactly once
  - → at most $|K|$ routes selected

Subproblem:

- Identify missing routes
  - → elementary $o-d$-path with negative reduced cost that respects vehicle capacity and time windows
- ESPPRC



Agenda

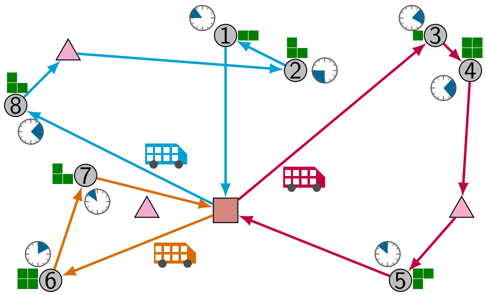
***Straightforward*:**

- Electric Vehicle Routing Problem With Time Windows →
- Pickup and Delivery Problem with Time Windows →
- Bin Packing and Cutting Stock Problems →
- Vertex Coloring Problem →
- Some takeaways so far →

A bit different:

- Heterogeneous Fleet Vehicle Routing Problem With Time Windows →
- Electric Vehicle Routing Problem With Time Windows and Maximum Number of Recharges per Station →
- Temporal Knapsack Problem →

Electric Vehicle Routing Problem with Time Windows



Given

- Depot o, d
- Customers N with
  - demands q_i
  - time windows $[a_i, b_i]$
- Vehicles K (homogeneous)
  - load capacity Q
  - battery capacity B
- Recharging stations R
  - recharging function

Task Find a cost-minimal set of vehicle routes, such that

- each customer is visited exactly once,
- each route starts and ends at the depot,
- the vehicle capacity is respected on all routes,
- all customers are serviced within their time windows, and
- the battery's state of charge is never empty on all routes, where recharging en route is possible.

Master program:

$$\min \sum_{r\in\Omega} c_r \lambda_r$$

$$\text{s.t.} \sum_{r\in\Omega} a_{ir}\lambda_r = 1 \quad \forall\, i \in N$$

$$\sum_{r\in\Omega} \lambda_r \leq |K|$$

$$\lambda_r \in \{0,1\} \quad \forall\, r \in \Omega$$

- Variables λ_r correspond with feasible routes $r \in \Omega$

  c_r cost of route r

  a_{ir} indicator if route r services customer i

- Select the best routes such that

  → total routing costs are minimum

  → each customer serviced exactly once

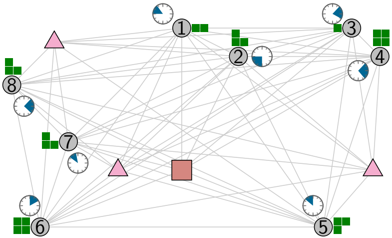
  → at most $|K|$ routes selected

Subproblem:

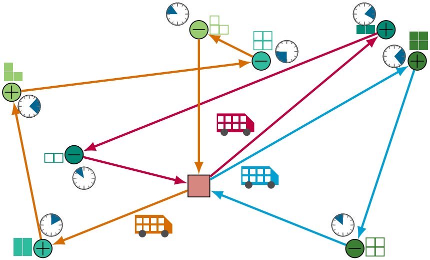
- Identify missing routes

  → elementary $o-d$-path with negative reduced cost that respects load and battery capacities, and time windows

- ESPPRC



Pickup and Delivery Problem with Time Windows



Given

- Depot o, d
- n Transportation requests P
  - $+$ pickup location $i \in P$
  - $-$ delivery location $i+n \in D$
  - transport quantities $q_i = -q_{i+n}$
  - time windows $[a_i, b_i]$
- Vehicles K (homogeneous) with capacity Q

Task Find a cost-minimal set of vehicle routes, such that

- each transportation request is performed exactly once
- each route starts and ends at the depot
- the vehicle capacity is respected on all routes
- all customer locations are serviced within their time windows, and
- each transportation request is performed by a single vehicle and in the correct order (pickup before delivery).

Master program:

$$\min \sum_{r \in \Omega} c_r \lambda_r$$

$$\text{s.t.} \sum_{r \in \Omega} a_{ir} \lambda_r = 1 \quad \forall\, i \in P$$

$$\sum_{r \in \Omega} \lambda_r \leq |K|$$

$$\lambda_r \in \{0, 1\} \quad \forall\, r \in \Omega$$

- Variables λ_r correspond with feasible routes $r \in \Omega$

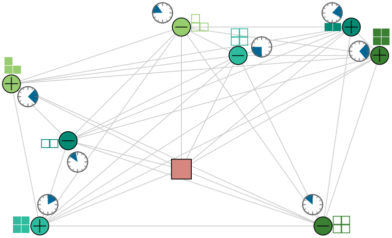
  c_r cost of route r

  a_{ir} indicator if route r performs request i

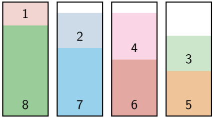
- Select the best routes such that
  - → total routing costs are minimum
  - → each request performed exactly once
  - → at most $|K|$ routes selected

Subproblem:

- Identify missing routes
  - → elementary $o - d$-path with negative reduced cost that respects pairing, precedence, capacity, and time windows
- ESPPRC



Bin Packing Problem



Given

- Set of items N
  - $\rightarrow$ w_i size of item i
- Bins (homogeneous)
  - $\rightarrow$ C bin capacity
  - $\rightarrow$ unlimited number

Task Find a grouping of the items into bins, such that

- each item is packed into exactly one bin,
- the capacity is respected for all bins, and
- the total number of bins used is minimium.

Bin Packing Problem

Master program:

$$\min \sum_{r\in\Omega} \lambda_r$$

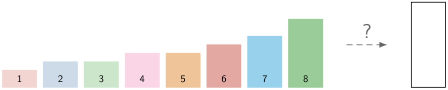
$$\text{s.t.} \sum_{r\in\Omega} a_{ir}\lambda_r = 1 \quad \forall\, i \in N$$

$$\lambda_r \in \{0,1\} \quad \forall\, r \in \Omega$$

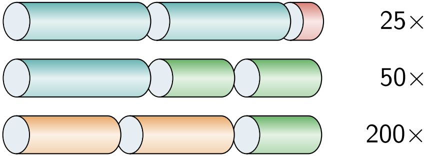
- Variables λ_r correspond with feasible bins $r \in \Omega$ (= subsets of items)
  a_{ir} indicator if item i is in bin r
- Select the best bins such that
  → number of bins is minimum
  → each item packed exactly once

Subproblem:

- Identify missing bins
  → selection of items with negative reduced cost that respects the bin capacity
- Binary knapsack problem



Cutting Stock Problem



<u>Given</u>

- Set of pieces N
  - $\rightarrow$ ℓ_i length of piece i
  - $\rightarrow$ b_i demand of piece i
- Rolls (homogeneous)
  - $\rightarrow$ L roll length
  - $\rightarrow$ unlimited number

Task Cut rolls into pieces, such that

- the demand for each piece is satisfied,
- the length is respected for all rolls, and
- the total number of rolls used is minimium.

Remark: Bin Packing Problem = Binary Cutting Stock Problem

Master program:

$$\min \sum_{r \in \Omega} \lambda_r$$

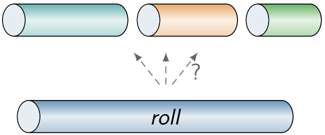
$$\text{s.t.} \sum_{r \in \Omega} a_{ir} \lambda_r = b_i \quad \forall\, i \in N$$

$$\lambda_r \in \mathbb{Z}^+ \quad \forall\, r \in \Omega$$

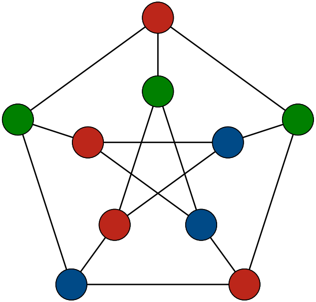
- Variables λ_r correspond with feasible cutting patterns $r \in \Omega$
  - a_{ir} number of times piece i is cut in pattern r
- Select the best patterns such that
  - → number of rolls is minimum
  - → demand of each piece is met

Subproblem:

- Identify missing patterns
  - → selection of pieces and multiplicities with negative reduced cost that respects the roll length
- Integer knapsack problem



Vertex Coloring Problem



Given

- Graph G
  - $\rightarrow$ N vertex set (*objects*)
  - $\rightarrow$ E edge set (*conflicts*)
- Colors
  - $\rightarrow$ unlimited number

Task Find a coloring (grouping) of the vertices, such that

- each vertex is assigned to exactly one color,
- adjacent vertices are assigned to different colors, and
- the total number of colors used is minimium.

Master program:

$$\min \sum_{r \in \Omega} \lambda_r$$

$$\text{s.t.} \sum_{r \in \Omega} a_{ir} \lambda_r = 1 \quad \forall\, i \in N$$

$$\lambda_r \in \{0, 1\} \quad \forall\, r \in \Omega$$

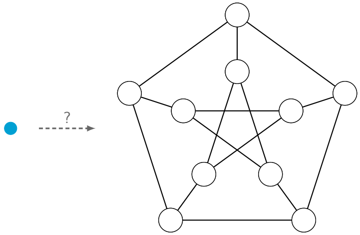
- Variables λ_r correspond with feasible colors $r \in \Omega$ ($=$ subsets of vertices)

  a_{ir} indicator if vertex i is in color r

- Select the best colors such that
  - → number of colors is minimum
  - → exactly one color is assigned to each vertex

Subproblem:

- Identify missing colors
  - → subset of pairwise non-adjacent vertices with negative reduced cost
- Maximum weight independent set problem

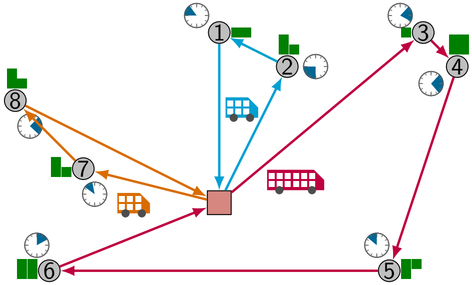


Some Takeaways

Many fundamental OR problems have set-partitioning/set-covering (sub)structure:

- CG-based approaches often promising
  - → master: (extended) set-partitioning/set-covering formulation
  - → variables: feasible subsets (routes, bins, independent sets, ...)
  - → subproblem: identify feasible sets with negative reduced cost
- Approach immediately applicable to many problem variants
  - → VRPs with any(!) intra-route constraint
  - → 2D/3D bin packing and cutting stock
  - → clustering/graph partitioning with other structures (cliques, relaxed cliques, ...)
- Many more problem classes:
  - → machine/vehicle/crew/shift/... scheduling
  - → ...

Heterogeneous Fleet VRPTW



Given

- Depot o, d
- Customers N with
  - demands q_i
  - time windows $[a_i, b_i]$
- Vehicles K_t (heterogeneous) of type $t \in T$ with capacity

Task Find a cost-minimal set of vehicle routes, such that

- each customer is visited exactly once,
- each route starts and ends at the depot,
- each route is assigned to one vehicle type,
- the heterogeneous vehicle capacity is respected on all routes,
- and all customers are serviced within their time window.

Heterogeneous Fleet VRPTW

Master program:

$$\min \sum_{t \in T} \sum_{r \in \Omega_t} c_r \lambda_r^t$$

$$\text{s.t.} \sum_{t \in T} \sum_{r \in \Omega_t} a_{ir} \lambda_r^t = 1 \quad \forall\, i \in N$$

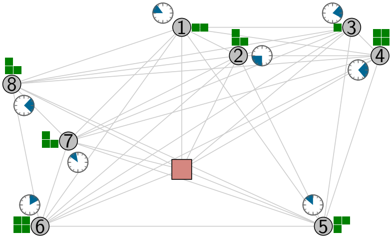
$$\sum_{t \in T} \sum_{r \in \Omega_t} \lambda_r^t \leq |K_t|$$

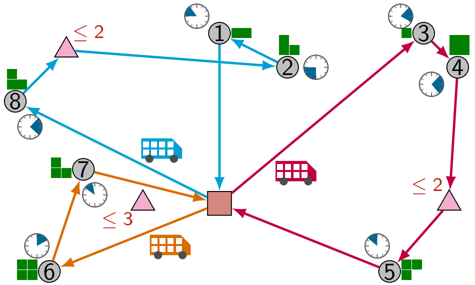
$$\lambda_r^t \in \{0, 1\} \quad \forall\, r \in \Omega_t$$

- Variables λ_r^t correspond with feasible routes $r \in \Omega_t$ of type $t \in T$
  - c_r cost of route r
  - a_{ir} indicator if route r services customer i
- Select the best routes such that
  - → total routing costs are minimum
  - → each customer serviced exactly once
  - → at most $|K|$ routes of type $t \in T$

Subproblems:

- Identify missing routes of type $t \in T$
  - → elementary $o-d$-path with negative reduced cost that respects capacity of type $t \in T$ and time windows
- ESPPRC for each type $t \in T$





Given

- Depot o, d
- Customers N with
  - demands q_i
  - time windows $[a_i, b_i]$
- Vehicles K (homogeneous)
  - load capacity Q
  - battery capacity B
- Recharging stations R
  - recharging function
  - maximum # recharges m_i

Task Find a cost-minimal set of vehicle routes, such that

- each customer is visited exactly once,
- each route starts and ends at the depot,
- the vehicle capacity is respected on all routes,
- all customers are serviced within their time windows,
- the battery's state of charge is never empty on all routes, where recharging en route is possible, and
- the maximum number of recharges at each station is respected.

Electric Vehicle Routing Problem with Time Windows

Master program:

$$\min \sum_{r \in \Omega} c_r \lambda_r$$

$$\text{s.t.} \sum_{r \in \Omega} a_{ir} \lambda_r = 1 \quad \forall\, i \in N$$

$$\sum_{r \in \Omega} \lambda_r \le |K|$$

$$\sum_{r \in \Omega} b_{ir} \lambda_r \le m_i \quad \forall\, i \in R$$

$$\lambda_r \in \{0, 1\} \quad \forall\, r \in \Omega$$

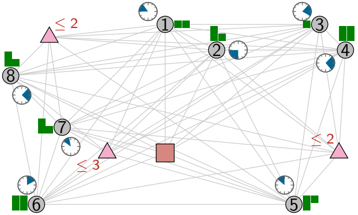
Subproblem:

- Identify missing routes
  - → elementary $o - d$-path with negative reduced cost that respects load and battery capacities, and time windows
- ESPPRC

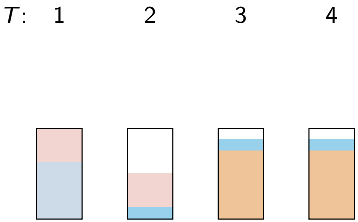
- Variables λ_r correspond with feasible routes $r \in \Omega$

  c_r cost of route r
  a_{ir} indicator if route r services customer i
  b_{ir} number of times route r recharges at station i

- Select the best routes such that
  - → total routing costs are minimum
  - → each customer serviced exactly once
  - → at most $|K|$ routes selected
  - → at most m_i recharges at station i



Temporal Knapsack Problem



Given

- Discrete time horizon T
  - $\rightarrow$ N_t active items at time t
- Set of items N
  - $\rightarrow$ w_i size of item i
  - $\rightarrow$ p_i profit of item i
  - $\rightarrow$ T_i active times of item i
- Single knapsack
  - $\rightarrow$ C capacity
  - $\rightarrow$ available at all $t \in T$

Task Find a selection of the items, such that

- the capacity is respected at any time, and
- the total profit of the selected items is maximum.

Temporal Knapsack Problem

Compact formulation:

$$\max \sum_{i \in N} p_i x_i$$
$$\text{s.t.} \sum_{i \in N_t} w_i x_i \leq C \quad \forall\, t \in T$$
$$x_i \in \{0, 1\} \quad \forall\, i \in N$$

- Binary selection variables x_i for each item i
- Maximize total profit
- Respect capacity at all t

CG formulations:

- Intuitive:
  - → Variables correspond with feasible knapsack packings of a time t
  - → Master Program: select compatible packings
  - → Subproblem: binary knapsack problem
- Uncommon (but worked much better): [Caprara et al., 2013]
  - → Partition time horizon into smaller blocks of times
  - → Variables correspond with feasible packings for a block
  - → Master Program: select compatible packings
  - → Subproblem: temporal knapsack problem

Temporal Knapsack Problem

Master program:[1]

$$
\begin{aligned}
\max \; & \sum_{i \in N} p_i x_i \\
\text{s.t. } & x_i = \sum_{r \in \Omega^q} a_{ir} \lambda_r^q \quad \forall\, q \in Q, i \in N_q \\
& \sum_{r \in \Omega^q} \lambda_r^q = 1 \quad \forall\, q \in Q \\
& \lambda_r^q \in \{0, 1\} \quad \forall\, q \in Q, r \in \Omega^q
\end{aligned}
$$

- Q set of time blocks (w. items N_q)
- Variables λ_r^q correspond with feasible solutions (= packings) $r \in \Omega^q$ of block $q \in Q$
  - a_{ir} indicator if item i is selected in r
- Keep original x_i variables
- Select the best packings such that
  - → total profit is maximum
  - → packings are compatible across blocks
  - → one packing per block

Subproblems:

- Identify missing packings of block q
- Temporal knapsack problem (over items/times relevant for q)

[1] *Here:* discretization; *Originally* [Caprara et al., 2013]: convexification

Alberto Caprara, Fabio Furini, and Enrico Malaguti. Uncommon dantzig-wolfe reformulation for the temporal knapsack problem. *INFORMS Journal on Computing*, 25(3):560–571, 2013.