Vehicle Routing for a Complex Waste Collection Problem

Iliya Markov[1,2], Sacha Varone[2], Michel Bierlaire[1]

[1]Transport and Mobility Laboratory
School of Architecture, Civil and Environmental Engineering
Ecole Polytechnique Fédérale de Lausanne

[2]Haute Ecole de Gestion de Genève
University of Applied Sciences Western Switzerland (HES-SO)

14th Swiss Transport Research Conference (STRC)
Monte Verità / Ascona, May 14-16, 2014

Overview

1. Introduction
2. Literature
3. Formulation
4. Solution Approach
5. Case Study
6. Conclusion
7. References

Overview

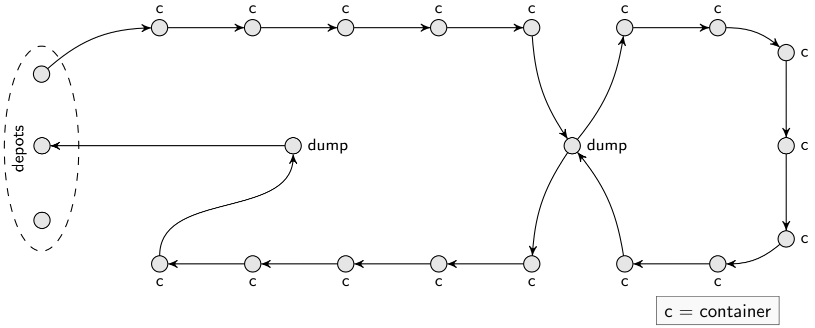
1. Introduction
2. Literature
3. Formulation
4. Solution Approach
5. Case Study
6. Conclusion
7. References

- A heterogeneous fixed fleet with different:
  - volume capacities
  - weight capacities
  - fixed costs
  - unit distance running costs
  - hourly driver wage rates
  - speeds
  - site dependencies (accessibility constraints)

- A heterogeneous fixed fleet with different:
  - volume capacities
  - weight capacities
  - fixed costs
  - unit distance running costs
  - hourly driver wage rates
  - speeds
  - site dependencies (accessibility constraints)
- A set of depots
- A set of containers placed at collection points with time windows
- A set of dumps (recycling plants) with time windows

- A heterogeneous fixed fleet with different:
    - volume capacities
    - weight capacities
    - fixed costs
    - unit distance running costs
    - hourly driver wage rates
    - speeds
    - site dependencies (accessibility constraints)
- A set of depots
- A set of containers placed at collection points with time windows
- A set of dumps (recycling plants) with time windows
- Maximum tour duration, interrupted by a break
- A tour is a sequence of collections and disposals at the available dumps, with a mandatory disposal before the tour end
- A tour need not finish at the depot it started from

Figure 1: Tour illustration



Overview

1 Introduction

2 Literature

3 Formulation

4 Solution Approach

5 Case Study

6 Conclusion

7 References

- VRP with intermediate facilities:
  - VRP with satellite facilities (Bard et al., 1998)
    - no time windows, no driver break, homogeneous fleet
    - branch-and-cut
  - Waste collection VRP (Kim et al., 2006)
    - time windows, driver break, homogeneous fleet
    - simulated annealing
    - Ombuki-Berman et al. (2007) (GA), Benjamin (2011) (VNTS), Buhrkal et al. (2012) (ALNS) improve results by 15-16%
  - MDVRP with inter-depot routes (Crevier et al., 2007)
    - no time windows, no driver break, homogeneous fleet at single depot
    - SP on a pool of single-depot, multi-depot and inter-depot routes
    - Tarantilis et al. (2008) (h-GLS), Hemmelmayr et al. (2013) (VNS) improve results by 1-3%

- VRP with intermediate facilities:
  - VRP with satellite facilities (Bard et al., 1998)
    - no time windows, no driver break, homogeneous fleet
    - branch-and-cut
  - Waste collection VRP (Kim et al., 2006)
    - time windows, driver break, homogeneous fleet
    - simulated annealing
    - Ombuki-Berman et al. (2007) (GA), Benjamin (2011) (VNTS), Buhrkal et al. (2012) (ALNS) improve results by 15-16%
  - MDVRP with inter-depot routes (Crevier et al., 2007)
    - no time windows, no driver break, homogeneous fleet at single depot
    - SP on a pool of single-depot, multi-depot and inter-depot routes
    - Tarantilis et al. (2008) (h-GLS), Hemmelmayr et al. (2013) (VNS) improve results by 1-3%
- Heterogeneous fixed fleet VRP:
  - Proposed by Taillard (1996)
  - Best exact solutions by Baldacci and Mingozzi (2009)
  - Best heuristic solutions by Subramanian et al. (2012) and Penna et al. (2013)

- Contribution:
  - Multiple depots
  - Multiple capacities
  - Realistic cost-based objective function
  - Simplification in the modeling of the dump visits
  - Non-time window constrained break
  - Incentive, rather than enforcement, to go back to the origin depot

Overview

1. Introduction
2. Literature
3. Formulation
4. Solution Approach
5. Case Study
6. Conclusion
7. References

Sets and Parameters:

O' = set of origins

D = set of dumps

$N = O' \cup O'' \cup D \cup P$

K = set of vehicles

O'' = set of destinations

P = set of containers

Sets and Parameters:

O' = set of origins

O'' = set of destinations

D = set of dumps

P = set of containers

$N = O' \cup O'' \cup D \cup P$

K = set of vehicles

π_{ij} = length of edge (i, j)

α_{ijk} = 1 if edge (i, j) is accessible for vehicle k, 0 otherwise

Sets and Parameters:

O' = set of origins
O'' = set of destinations
D = set of dumps
P = set of containers
$N = O' \cup O'' \cup D \cup P$
K = set of vehicles

π_{ij} = length of edge (i,j)
α_{ijk} = 1 if edge (i,j) is accessible for vehicle k, 0 otherwise
τ_{ijk} = travel time of vehicle k on edge (i,j)
ϵ_i = service duration at point i
$[\lambda_i, \mu_i]$ = time window lower and upper bound at point i

Sets and Parameters:

O' = set of origins
O'' = set of destinations
D = set of dumps
P = set of containers
$N = O' \cup O'' \cup D \cup P$
K = set of vehicles

π_{ij} = length of edge (i, j)
α_{ijk} = 1 if edge (i, j) is accessible for vehicle k, 0 otherwise
τ_{ijk} = travel time of vehicle k on edge (i, j)
ϵ_i = service duration at point i
$[\lambda_i, \mu_i]$ = time window lower and upper bound at point i
H = maximum tour duration
η = maximum continuous work limit after which a break is due
δ = break duration

Sets and Parameters:

$$
\begin{aligned}
O' &= \text{set of origins} & \qquad O'' &= \text{set of destinations}\\
D &= \text{set of dumps} & \qquad P &= \text{set of containers}\\
N &= O' \cup O'' \cup D \cup P\\
K &= \text{set of vehicles}
\end{aligned}
$$

$$
\begin{aligned}
\pi_{ij} &= \text{length of edge } (i,j)\\
\alpha_{ijk} &= 1 \text{ if edge } (i,j) \text{ is accessible for vehicle } k\text{, 0 otherwise}\\
\tau_{ijk} &= \text{travel time of vehicle } k \text{ on edge } (i,j)\\
\epsilon_i &= \text{service duration at point } i\\
[\lambda_i, \mu_i] &= \text{time window lower and upper bound at point } i\\
\mathrm{H} &= \text{maximum tour duration}\\
\eta &= \text{maximum continuous work limit after which a break is due}\\
\delta &= \text{break duration}\\
\rho_i^v, \rho_i^w &= \text{volume and weight pickup quantity at point } i\\
\Omega_k^v, \Omega_k^w &= \text{volume and weight capacity of vehicle } k
\end{aligned}
$$

Sets and Parameters:

O'	$=$ set of origins	O''	$=$ set of destinations
D	$=$ set of dumps	P	$=$ set of containers
N	$= O' \cup O'' \cup D \cup P$		
K	$=$ set of vehicles		

π_{ij}	$=$ length of edge (i, j)
α_{ijk}	$=$ 1 if edge (i, j) is accessible for vehicle k, 0 otherwise
τ_{ijk}	$=$ travel time of vehicle k on edge (i, j)
ϵ_i	$=$ service duration at point i
$[\lambda_i, \mu_i]$	$=$ time window lower and upper bound at point i
H	$=$ maximum tour duration
η	$=$ maximum continuous work limit after which a break is due
δ	$=$ break duration
ρ_i^v, ρ_i^w	$=$ volume and weight pickup quantity at point i
Ω_k^v, Ω_k^w	$=$ volume and weight capacity of vehicle k
ϕ_k	$=$ fixed cost of vehicle k
β_k	$=$ unit distance running cost of vehicle k
θ_k	$=$ hourly wage rate of vehicle k

Decision Variables:

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ traverses edge } (i,j) \\ 0 & \text{otherwise} \end{cases}$$

$$b_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ takes a break on edge } (i,j) \\ 0 & \text{otherwise} \end{cases}$$

$$y_k = \begin{cases} 1 & \text{if vehicle } k \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

Decision Variables:

$$x_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ traverses edge } (i,j) \\ 0 & \text{otherwise} \end{cases}$$

$$b_{ijk} = \begin{cases} 1 & \text{if vehicle } k \text{ takes a break on edge } (i,j) \\ 0 & \text{otherwise} \end{cases}$$

$$y_k = \begin{cases} 1 & \text{if vehicle } k \text{ is used} \\ 0 & \text{otherwise} \end{cases}$$

S_{ik} = start-of-service time of vehicle k at point i

Q^v_{ik} = cumulative volume on vehicle k at point i

Q^w_{ik} = cumulative weight on vehicle k at point i

$$\text{Min} \quad f = \sum_{k \in K} \left(\phi_k y_k + \beta_k \sum_{i \in N} \sum_{j \in N} \pi_{ij} x_{ijk} + \theta_k \left(\sum_{j \in O''} S_{jk} - \sum_{i \in O'} S_{ik} \right) \right) \tag{1}$$

(8)

$$\text{Min} \quad f = \sum_{k \in K} \left(\phi_k y_k + \beta_k \sum_{i \in N} \sum_{j \in N} \pi_{ij} x_{ijk} + \theta_k \left(\sum_{j \in O''} S_{jk} - \sum_{i \in O'} S_{ik} \right) \right) \tag{1}$$

$$\text{s.t.} \quad \sum_{k \in K} \sum_{j \in D \cup P} x_{ijk} = 1, \qquad \forall i \in P \tag{2}$$

(8)

$$\text{Min} \quad f = \sum_{k \in K} \left(\phi_k y_k + \beta_k \sum_{i \in N} \sum_{j \in N} \pi_{ij} x_{ijk} + \theta_k \left(\sum_{j \in O''} S_{jk} - \sum_{i \in O'} S_{ik} \right) \right) \tag{1}$$

$$\text{s.t.} \quad \sum_{k \in K} \sum_{j \in D \cup P} x_{ijk} = 1, \qquad \forall i \in P \tag{2}$$

$$\sum_{i \in O'} \sum_{j \in N} x_{ijk} = y_k, \qquad \forall k \in K \tag{3}$$

$$\sum_{i \in D} \sum_{j \in O''} x_{ijk} = y_k, \qquad \forall k \in K \tag{4}$$

(8)

$$\text{Min} \quad f = \sum_{k \in K} \left(\phi_k y_k + \beta_k \sum_{i \in N} \sum_{j \in N} \pi_{ij} x_{ijk} + \theta_k \left(\sum_{j \in O''} S_{jk} - \sum_{i \in O'} S_{ik} \right) \right) \tag{1}$$

$$\text{s.t.} \quad \sum_{k \in K} \sum_{j \in D \cup P} x_{ijk} = 1, \qquad \forall i \in P \tag{2}$$

$$\sum_{i \in O'} \sum_{j \in N} x_{ijk} = y_k, \qquad \forall k \in K \tag{3}$$

$$\sum_{i \in D} \sum_{j \in O''} x_{ijk} = y_k, \qquad \forall k \in K \tag{4}$$

$$\sum_{i \in N} x_{ijk} = 0, \qquad \forall k \in K, j \in O' \tag{5}$$

$$\sum_{j \in N} x_{ijk} = 0, \qquad \forall k \in K, i \in O'' \tag{6}$$

(8)

$$\text{Min} \quad f = \sum_{k \in K} \left(\phi_k y_k + \beta_k \sum_{i \in N} \sum_{j \in N} \pi_{ij} x_{ijk} + \theta_k \left(\sum_{j \in O''} S_{jk} - \sum_{i \in O'} S_{ik} \right) \right) \tag{1}$$

$$\text{s.t.} \quad \sum_{k \in K} \sum_{j \in D \cup P} x_{ijk} = 1, \qquad \forall i \in P \tag{2}$$

$$\sum_{i \in O'} \sum_{j \in N} x_{ijk} = y_k, \qquad \forall k \in K \tag{3}$$

$$\sum_{i \in D} \sum_{j \in O''} x_{ijk} = y_k, \qquad \forall k \in K \tag{4}$$

$$\sum_{i \in N} x_{ijk} = 0, \qquad \forall k \in K, j \in O' \tag{5}$$

$$\sum_{j \in N} x_{ijk} = 0, \qquad \forall k \in K, i \in O'' \tag{6}$$

$$\sum_{i \in N \setminus O''} x_{ijk} = \sum_{i \in N \setminus O'} x_{jik}, \qquad \forall k \in K, j \in D \cup P \tag{7}$$

$$\tag{8}$$

$$\text{Min} \quad f = \sum_{k \in K} \left(\phi_k y_k + \beta_k \sum_{i \in N} \sum_{j \in N} \pi_{ij} x_{ijk} + \theta_k \left(\sum_{j \in O''} S_{jk} - \sum_{i \in O'} S_{ik} \right) \right) \tag{1}$$

$$\text{s.t.} \quad \sum_{k \in K} \sum_{j \in D \cup P} x_{ijk} = 1, \qquad \forall i \in P \tag{2}$$

$$\sum_{i \in O'} \sum_{j \in N} x_{ijk} = y_k, \qquad \forall k \in K \tag{3}$$

$$\sum_{i \in D} \sum_{j \in O''} x_{ijk} = y_k, \qquad \forall k \in K \tag{4}$$

$$\sum_{i \in N} x_{ijk} = 0, \qquad \forall k \in K, j \in O' \tag{5}$$

$$\sum_{j \in N} x_{ijk} = 0, \qquad \forall k \in K, i \in O'' \tag{6}$$

$$\sum_{i \in N \setminus O''} x_{ijk} = \sum_{i \in N \setminus O'} x_{jik}, \qquad \forall k \in K, j \in D \cup P \tag{7}$$

$$x_{ijk} \leqslant \alpha_{ijk}, \qquad \forall k \in K, i \in N \setminus O'', j \in N \setminus O' \tag{8}$$

$$
\text{s.t.} \quad Q_{ik}^{v} \leqslant \Omega_{k}^{v}, \qquad \forall k \in K, i \in P \tag{9}
$$

$$
Q_{ik}^{w} \leqslant \Omega_{k}^{w}, \qquad \forall k \in K, i \in P \tag{10}
$$

(19)

$$\text{s.t.} \quad Q^v_{ik} \leqslant \Omega^v_k, \qquad \forall k \in K, i \in P \tag{9}$$

$$Q^w_{ik} \leqslant \Omega^w_k, \qquad \forall k \in K, i \in P \tag{10}$$

$$Q^v_{ik} = 0, \qquad \forall k \in K, i \in N \setminus P \tag{11}$$

$$Q^w_{ik} = 0, \qquad \forall k \in K, i \in N \setminus P \tag{12}$$

(19)

$$
\begin{aligned}
\text{s.t.} \quad & Q^v_{ik} \leqslant \Omega^v_k, && \forall k \in K, i \in P && (9)\\
& Q^w_{ik} \leqslant \Omega^w_k, && \forall k \in K, i \in P && (10)\\
& Q^v_{ik} = 0, && \forall k \in K, i \in N \setminus P && (11)\\
& Q^w_{ik} = 0, && \forall k \in K, i \in N \setminus P && (12)\\
& Q^v_{ik} + \rho^v_j \leqslant Q^v_{jk} + \left(1 - x_{ijk}\right) M, && \forall k \in K, i \in N \setminus O'', j \in P && (13)\\
& Q^w_{ik} + \rho^w_j \leqslant Q^w_{jk} + \left(1 - x_{ijk}\right) M, && \forall k \in K, i \in N \setminus O'', j \in P && (14)
\end{aligned}
$$

$$(19)$$

$$
\begin{aligned}
\text{s.t.}\quad & Q_{ik}^{v} \leqslant \Omega_{k}^{v}, && \forall k \in K, i \in P && (9)\\
& Q_{ik}^{w} \leqslant \Omega_{k}^{w}, && \forall k \in K, i \in P && (10)\\
& Q_{ik}^{v} = 0, && \forall k \in K, i \in N \setminus P && (11)\\
& Q_{ik}^{w} = 0, && \forall k \in K, i \in N \setminus P && (12)\\
& Q_{ik}^{v} + \rho_{j}^{v} \leqslant Q_{jk}^{v} + \left(1 - x_{ijk}\right) M, && \forall k \in K, i \in N \setminus O'', j \in P && (13)\\
& Q_{ik}^{w} + \rho_{j}^{w} \leqslant Q_{jk}^{w} + \left(1 - x_{ijk}\right) M, && \forall k \in K, i \in N \setminus O'', j \in P && (14)\\
& S_{ik} + \epsilon_{i} + \delta b_{ijk} + \tau_{ijk} \leqslant S_{jk} + \left(1 - x_{ijk}\right) M, && \forall k \in K, i \in N \setminus O'', j \in N \setminus O' && (15)
\end{aligned}
$$

(19)

$$
\begin{aligned}
\text{s.t.}\quad & Q^{v}_{ik} \leqslant \Omega^{v}_{k}, && \forall k \in K, i \in P && (9)\\
& Q^{w}_{ik} \leqslant \Omega^{w}_{k}, && \forall k \in K, i \in P && (10)\\
& Q^{v}_{ik} = 0, && \forall k \in K, i \in N \setminus P && (11)\\
& Q^{w}_{ik} = 0, && \forall k \in K, i \in N \setminus P && (12)\\
& Q^{v}_{ik} + \rho^{v}_{j} \leqslant Q^{v}_{jk} + \left(1 - x_{ijk}\right) M, && \forall k \in K, i \in N \setminus O'', j \in P && (13)\\
& Q^{w}_{ik} + \rho^{w}_{j} \leqslant Q^{w}_{jk} + \left(1 - x_{ijk}\right) M, && \forall k \in K, i \in N \setminus O'', j \in P && (14)\\
& S_{ik} + \epsilon_i + \delta b_{ijk} + \tau_{ijk} \leqslant S_{jk} + \left(1 - x_{ijk}\right) M, && \forall k \in K, i \in N \setminus O'', j \in N \setminus O' && (15)\\
& \left(S_{ik} - \sum_{m \in O'} S_{mk} \right) + \epsilon_i - \eta \leqslant \left(1 - b_{ijk}\right) M, && \forall k \in K, i \in N \setminus O'', j \in N \setminus O' && (16)\\
& \eta - \left(S_{jk} - \sum_{m \in O'} S_{mk} \right) \leqslant \left(1 - b_{ijk}\right) M, && \forall k \in K, i \in N \setminus O'', j \in N \setminus O' && (17)\\
& b_{ijk} \leqslant x_{ijk}, && \forall k \in K, i, j \in N && (18)\\
& \left(\sum_{j \in O''} S_{jk} - \sum_{i \in O'} S_{ik} \right) - \eta \leqslant \left(\sum_{\substack{i \in N \setminus O'' \\ j \in N \setminus O'}} b_{ijk} \right) M, && \forall k \in K && (19)
\end{aligned}
$$

$$\text{s.t.} \quad \lambda_i \sum_{j \in N \setminus O'} x_{ijk} \leqslant S_{ik} \leqslant \mu_i \sum_{j \in N \setminus O'} x_{ijk}, \qquad \forall k \in K, i \in N \setminus O'' \tag{20}$$

(23)

$$\text{s.t.} \quad \lambda_i \sum_{j \in N \setminus O'} x_{ijk} \leqslant S_{ik} \leqslant \mu_i \sum_{j \in N \setminus O'} x_{ijk}, \qquad \forall k \in K, i \in N \setminus O'' \tag{20}$$

$$\sum_{j \in O''} S_{jk} - \sum_{i \in O'} S_{ik} \leqslant \mathrm{H}, \qquad \forall k \in K \tag{21}$$

$$\tag{23}$$

$$\text{s.t.} \quad \lambda_i \sum_{j \in N \setminus O'} x_{ijk} \leqslant S_{ik} \leqslant \mu_i \sum_{j \in N \setminus O'} x_{ijk}, \qquad \forall k \in K, i \in N \setminus O'' \tag{20}$$

$$\sum_{j \in O''} S_{jk} - \sum_{i \in O'} S_{ik} \leqslant \mathrm{H}, \qquad \forall k \in K \tag{21}$$

$$x_{ijk}, y_k, b_{ijk} \in \{0, 1\}, \qquad \forall k \in K, i, j \in N \tag{22}$$

$$Q^v_{ik}, Q^w_{ik}, S_{ik} \geqslant 0, \qquad \forall k \in K, i \in N \tag{23}$$

Extension:

$$z_{ijk} = \begin{cases} 1 & \text{if } i \text{ is the origin and } j \text{ the destination of vehicle } k \\ 0 & \text{otherwise} \end{cases}$$

Ψ = weight of relocation term

$$\text{Min} \quad f = \text{Objective (1)} + \Psi \sum_{k \in K} \sum_{i \in O'} \sum_{j \in O''} \left(\beta_k \pi_{ji} + \theta_k \tau_{jik}\right) z_{ijk} \tag{24}$$

s.t. Constraints (2) to (23)

$$\sum_{m \in P} x_{imk} + \sum_{m \in D} x_{mjk} - 1 \leqslant z_{ijk}, \qquad \forall k \in K, i \in O', j \in O'' \tag{25}$$

$$z_{ijk} = \{0, 1\}, \qquad \forall k \in K, i \in O', j \in O'' \tag{26}$$

Overview

1 Introduction

2 Literature

3 Formulation

4 Solution Approach

5 Case Study

6 Conclusion

7 References

- For small instances, common solver for the MILP formulation enhanced by valid inequalities and elimination rules, including:
  - Impossible traversals
  - Time window infeasible traversals
  - Latest start/earliest finish
  - Minimum tour duration
  - Symmetry breaking for subsets of identical vehicles
  - Minimum/maximum number of dump visits

- For small instances, common solver for the MILP formulation enhanced by valid inequalities and elimination rules, including:
    - Impossible traversals
    - Time window infeasible traversals
    - Latest start/earliest finish
    - Minimum tour duration
    - Symmetry breaking for subsets of identical vehicles
    - Minimum/maximum number of dump visits
- For realistic-size instances, a feasibility preserving local search heuristic
- A feasible tour satisfies three criteria:
    - Time-window feasibility
    - Duration feasibility
    - Capacity feasibility

- For small instances, common solver for the MILP formulation enhanced by valid inequalities and elimination rules, including:
  - Impossible traversals
  - Time window infeasible traversals
  - Latest start/earliest finish
  - Minimum tour duration
  - Symmetry breaking for subsets of identical vehicles
  - Minimum/maximum number of dump visits
- For realistic-size instances, a feasibility preserving local search heuristic
- A feasible tour satisfies three criteria:
  - Time-window feasibility
  - Duration feasibility
  - Capacity feasibility
- The quality of the heuristic is assessed by benchmarking its results to the optimal ones obtained with the MILP model on small instances.

Figure 2: Temporal feasibility algorithm

Data: tour k as a sequence of points $1, \ldots, n$ after a change
Result: start-of-service times, waiting times and temporal feasibility of tour k

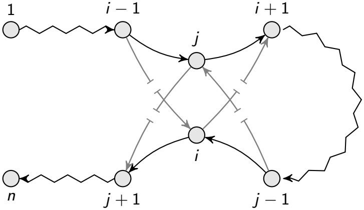
```
set S_1k to earliest possible;
for i = 2...n in tour k do
    // Calculate tentative start-of-service times
    S_ik = S_(i-1)k + ε_(i-1) + τ_(i-1)ik;
    // Insert break
    if S_(i-1)k + ε_(i-1) ≤ S_1k + η and S_ik + ε_i > S_1k + η then
        S_ik = S_ik + δ;
    end
    // Calculate waiting times
    if S_ik < λ_i then
        w_ik = λ_i − S_ik;
        S_ik = λ_i;
    else
        w_ik = 0;
    end
end
```

Figure 2: Temporal feasibility algorithm, cont'd

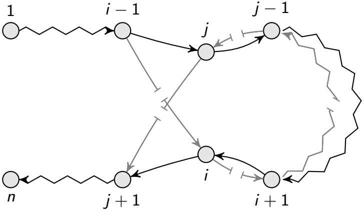
```
// Check time window feasibility
if S_ik ⩽ μ_i, ∀i then
    // Forward time slack reduction
    for i = n...2 in tour k do
        S'_(i−1)k = S_(i−1)k;
        S_(i−1)k = min (S_(i−1)k + w_ik, μ_(i−1));
        w_(i−1)k = w_(i−1)k + (S_(i−1)k − S'_(i−1)k);
        w_ik = w_ik − (S_(i−1)k − S'_(i−1)k);
    end
    w_1k = 0;
    // Check duration feasibility
    if S_nk − S_1k ⩽ H then
        tour k is temporally feasible;
    else
        discard tour k as duration infeasible;
    end
else
    discard tour k as time-window infeasible;
end
```

Figure 3: Neighborhood operators

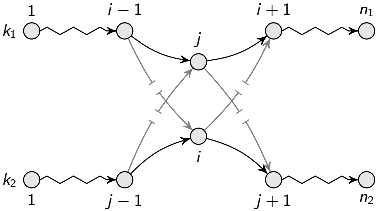
Single-tour 1-1 exchange



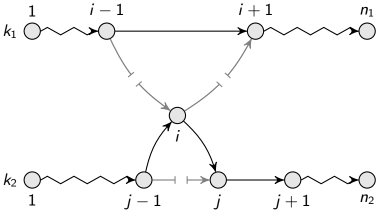
Single-tour 2-opt



Inter-tour 1-1 exchange



Inter-tour reinsert



- Tour construction:
  - Sequential feasibility preserving insertion heuristic
  - At every iteration an unassigned container is inserted at the point that yields the smallest increase in the objective value
  - When container insertions would violate capacity, a dump is inserted using the same logic
  - A dump insertion should allow for at least one subsequent temporally feasible container insertion

- Tour construction:
  - Sequential feasibility preserving insertion heuristic
  - At every iteration an unassigned container is inserted at the point that yields the smallest increase in the objective value
  - When container insertions would violate capacity, a dump is inserted using the same logic
  - A dump insertion should allow for at least one subsequent temporally feasible container insertion
- Tour improvement:
  - Alternation between inter-tour and single-tour improvement
  - The application of an inter-tour operator is followed by single-tour improvement of the affected tours
  - Every operator is applied for *maxOpIter* iterations and *maxOpNonImpIter* non-improving iterations, before changing to the next operator
  - Both single-tour and multi-tour improvement run for *maxIter* iterations and *maxNonImpIter* non-improving iterations
  - The resulting tour schedule is the best found during all iterations

Overview

1. Introduction
2. Literature
3. Formulation
4. Solution Approach
5. Case Study
6. Conclusion
7. References

- 5 instances extracted randomly from real underlying data
- 3 versions of each instance:
  - No time windows (iX)
  - Wide time windows (iX_tw) - randomly assigned
  - Narrow time windows (iX_ntw) - randomly assigned
- 1 depot, 1 dump, 2 identical vehicles

- 5 instances extracted randomly from real underlying data
- 3 versions of each instance:
  - No time windows (iX)
  - Wide time windows (iX_tw) - randomly assigned
  - Narrow time windows (iX_ntw) - randomly assigned
- 1 depot, 1 dump, 2 identical vehicles
- Tests on 2.60 GHz Intel Core i7, 8GB of RAM
  - Local search heuristic coded in Java
  - Model solved on Gurobi 5.6.2 warm-started with the solutions from the local search heuristic
  - Solver time limit set to 1000 sec

Table 1: Comparison between heuristic and solver on random instances
$maxOpIter = 100$, $maxOpNonImpIter = 13$, $maxIter = 100$, $maxNonImpIter = 1$

	Heuristic		Solver				
Instance	Objective	Runtime (sec.)	Objective	L Bound	MIP gap (%)	Runtime (sec.)	Opt gap (%)
i1	214.849	0.170	214.849	214.837	0.006	375.562	0.000
i1_tw	284.016	0.070	252.825	252.825	0.000	4.038	12.337
i1_ntw	428.539	1.093	394.817	394.817	0.000	0.922	8.541
i2	249.317	0.042	249.317	249.317	0.000	400.032	0.000
i2_tw	257.583	0.050	257.582	257.582	0.000	2.306	0.000
i2_ntw	460.635	0.756	439.769	439.769	0.000	2.420	4.745
i3	240.133	0.051	240.133	76.004	68.349	1000.000	0.000
i3_tw	245.457	0.070	245.457	245.457	0.000	2.894	0.000
i3_ntw	444.589	0.641	444.589	444.589	0.000	2.446	0.000
i4	138.643	0.077	138.643	138.643	0.000	521.509	0.000
i4_tw	140.204	0.030	140.204	140.204	0.000	7.660	0.000
i4_ntw	179.537	0.043	179.537	179.537	0.000	2.849	0.000
i5	220.770	0.070	220.770	129.834	41.190	1000.000	0.000
i5_tw	233.211	0.050	233.211	233.211	0.000	3.501	0.000
i5_ntw	405.622	0.848	405.622	405.622	0.000	3.051	0.000

Table 1: Comparison between heuristic and solver on random instances
$maxOpIter = 100$, $maxOpNonImpIter = 13$, $maxIter = 100$, $maxNonImpIter = 1$

	Heuristic		Solver				
Instance	Objective	Runtime (sec.)	Objective	L Bound	MIP gap (%)	Runtime (sec.)	Opt gap (%)
i1	214.849	0.170	214.849	214.837	0.006	375.562	0.000
i1_tw	284.016	0.070	252.825	252.825	0.000	4.038	**12.337**
i1_ntw	428.539	1.093	394.817	394.817	0.000	0.922	**8.541**
i2	249.317	0.042	249.317	249.317	0.000	400.032	0.000
i2_tw	257.583	0.050	257.582	257.582	0.000	2.306	0.000
i2_ntw	460.635	0.756	439.769	439.769	0.000	2.420	**4.745**
i3	240.133	0.051	240.133	76.004	68.349	1000.000	0.000
i3_tw	245.457	0.070	245.457	245.457	0.000	2.894	0.000
i3_ntw	444.589	0.641	444.589	444.589	0.000	2.446	0.000
i4	138.643	0.077	138.643	138.643	0.000	521.509	0.000
i4_tw	140.204	0.030	140.204	140.204	0.000	7.660	0.000
i4_ntw	179.537	0.043	179.537	179.537	0.000	2.849	0.000
i5	220.770	0.070	220.770	129.834	41.190	1000.000	0.000
i5_tw	233.211	0.050	233.211	233.211	0.000	3.501	0.000
i5_ntw	405.622	0.848	405.622	405.622	0.000	3.051	0.000

Table 1: Comparison between heuristic and solver on random instances
$maxOpIter = 100$, $maxOpNonImpIter = 13$, $maxIter = 100$, $maxNonImpIter = 1$

	Heuristic		Solver				
Instance	Objective	Runtime (sec.)	Objective	L Bound	MIP gap (%)	Runtime (sec.)	Opt gap (%)
i1	214.849	0.170	214.849	214.837	0.006	375.562	0.000
i1_tw	284.016	0.070	252.825	252.825	0.000	4.038	12.337
i1_ntw	428.539	1.093	394.817	394.817	0.000	0.922	8.541
i2	249.317	0.042	249.317	249.317	0.000	400.032	0.000
i2_tw	257.583	0.050	257.582	257.582	0.000	2.306	0.000
i2_ntw	460.635	0.756	439.769	439.769	0.000	2.420	4.745
i3	240.133	0.051	240.133	76.004	**68.349**	**1000.000**	0.000
i3_tw	245.457	0.070	245.457	245.457	0.000	2.894	0.000
i3_ntw	444.589	0.641	444.589	444.589	0.000	2.446	0.000
i4	138.643	0.077	138.643	138.643	0.000	521.509	0.000
i4_tw	140.204	0.030	140.204	140.204	0.000	7.660	0.000
i4_ntw	179.537	0.043	179.537	179.537	0.000	2.849	0.000
i5	220.770	0.070	220.770	129.834	**41.190**	**1000.000**	0.000
i5_tw	233.211	0.050	233.211	233.211	0.000	3.501	0.000
i5_ntw	405.622	0.848	405.622	405.622	0.000	3.051	0.000

Table 2: Comparison between heuristic and solver on selected random instances
maxOpIter = 350, *maxOpNonImpIter* = 37, *maxIter* = 100, *maxNonImpIter* = 1

	Heuristic		Solver				
Instance	Objective	Runtime (sec.)	Objective	L Bound	MIP gap (%)	Runtime (sec.)	Opt gap (%)
i1_tw	252.825	0.410	252.825	252.825	0.000	3.487	0.000
i1_ntw	394.817	3.399	394.817	394.817	0.000	0.916	0.000
i2_ntw	439.769	3.080	439.769	439.769	0.000	2.309	0.000

Table 2: Comparison between heuristic and solver on selected random instances
$maxOpIter = 350$, $maxOpNonImpIter = 37$, $maxIter = 100$, $maxNonImpIter = 1$

	Heuristic		Solver				
Instance	Objective	Runtime (sec.)	Objective	L Bound	MIP gap (%)	Runtime (sec.)	Opt gap (%)
i1_tw	252.825	0.410	252.825	252.825	0.000	3.487	**0.000**
i1_ntw	394.817	3.399	394.817	394.817	0.000	0.916	**0.000**
i2_ntw	439.769	3.080	439.769	439.769	0.000	2.309	**0.000**

Overview

1. Introduction
2. Literature
3. Formulation
4. Solution Approach
5. Case Study
6. Conclusion
7. References

- Conclusions:
  - Mathematical model
  - Local search heuristic
  - The heuristic performs favorably with an average optimality gap of less than 2% in short computation times (less than 1 sec)

- Conclusions:
  - Mathematical model
  - Local search heuristic
  - The heuristic performs favorably with an average optimality gap of less than 2% in short computation times (less than 1 sec)
- Future work
  - Mathematical model improvement to solve larger instances
  - Extension of the heuristic to include all features of the mathematical model
  - Development of efficient inter-tour operators to respond to the challenge posed by the heterogeneous fleet
  - Sensitivity analysis of the parameters
  - Benchmarking for realistic-size instances against current state of practice

Thank you for your attention!

Questions?

Overview

1. Introduction
2. Literature
3. Formulation
4. Solution Approach
5. Case Study
6. Conclusion
7. References

Baldacci, R. and Mingozzi, A. (2009). A unified exact method for solving different classes of vehicle routing problems. *Mathematical Programming*, 120(2):347–380.

Bard, J. F., Huang, L., Dror, M., and Jaillet, P. (1998). A branch and cut algorithm for the VRP with satellite facilities. *IIE Transactions*, 30(9):821–834.

Benjamin, A. M. (2011). *Metaheuristics for the Waste Collection Vehicle Routing Problem with Time Windows*. PhD thesis, Department of Mathematical Sciences, Brunel University.

Buhrkal, K., Larsen, A., and Ropke, S. (2012). The waste collection vehicle routing problem with time windows in a city logistics context. *Procedia-Social and Behavioral Sciences*, 39:241–254.

Crevier, B., Cordeau, J.-F., and Laporte, G. (2007). The multi-depot vehicle routing problem with inter-depot routes. *European Journal of Operational Research*, 176(2):756–773.

Hemmelmayr, V., Doerner, K. F., Hartl, R. F., and Rath, S. (2013). A heuristic solution method for node routing based solid waste collection problems. *Journal of Heuristics*, 19(2):129–156.

Kim, B. I., Kim, S., and Sahoo, S. (2006). Waste collection vehicle routing problem with time windows. *Computers & Operations Research*, 33(12):3624–3642.

Ombuki-Berman, B. M., Runka, A., and Hanshar, F. T. (2007). Waste collection vehicle routing problem with time windows using multi-objective genetic algorithms. In *Proceedings of the Third IASTED International Conference on Computational Intelligence*, CI '07, pages 91–97, Anaheim, CA, USA.

Penna, P., Subramanian, A., and Ochi, L. (2013). An iterated local search heuristic for the heterogeneous fleet vehicle routing problem. *Journal of Heuristics*, 19(2):201–232.

Rochat, Y. and Taillard, E. (1995). Probabilistic diversification and intensification in local search for vehicle routing. *Journal of Heuristics*, 1:147–167.

Sahoo, S., Kim, S., Kim, B.-I., Kraas, B., and Popov, A. (2005). Routing optimization for waste management. *Interfaces*, 35(1):24–36.

Subramanian, A., Penna, P. H. V., Uchoa, E., and Ochi, L. S. (2012). A hybrid algorithm for the heterogeneous fleet vehicle routing problem. *European Journal of Operational Research*, 221(2):285–295.

Taillard, E. D. (1996). A heuristic column generation method for heterogeneous fleet VRP. *Publication CRT-96-13*.

Tarantilis, C. D., Zachariadis, E. E., and Kiranoudis, C. T. (2008). A hybrid guided local search for the vehicle-routing problem with intermediate replenishment facilities. *INFORMS Journal on Computing*, 20(1):154–168.