Vehicle Scheduling for Accessible Transportation Systems

Etienne Beauchamp[1] Claudio Contardo[2]
Furkan Enderer[3] Mounira Groiez[1] Nadia Lahrichi[1]
Mehdi Mahnam[1] Odile Marcotte[4] Sofiane Soualah[5]
Patrick St-Louis[6]

[1]École Polytechnique
[2]UQAM
[3]Concordia University
[4]CRM and UQAM
[5]Université de Montréal
[6]GIRO

August 23, 2013

Outline

- The scheduling problem
- The mandate:
  1. Prove that Dumas' algorithm is still optimal with the wait time constraint
  2. Add the Pick-up & Delivery Span constraint
  3. Add the route span contraint
- Many ideas
- Further steps

The team

Etienne Beauchamp, Claudio Contardo , Furkan Enderer, Mounira Groiez, Nadia Lahrichi, Mehdi Mahnam, Odile Marcotte, Sofiane Soualah, Patrick St-Louis

A first version of the scheduling problem

Etienne Beauchamp, Claudio Contardo , Furkan Enderer, Mounira Groiez, Nadia Lahrichi, Mehdi Mahnam, Odile Marcotte, Sofiane Soualah, Patrick St-Louis

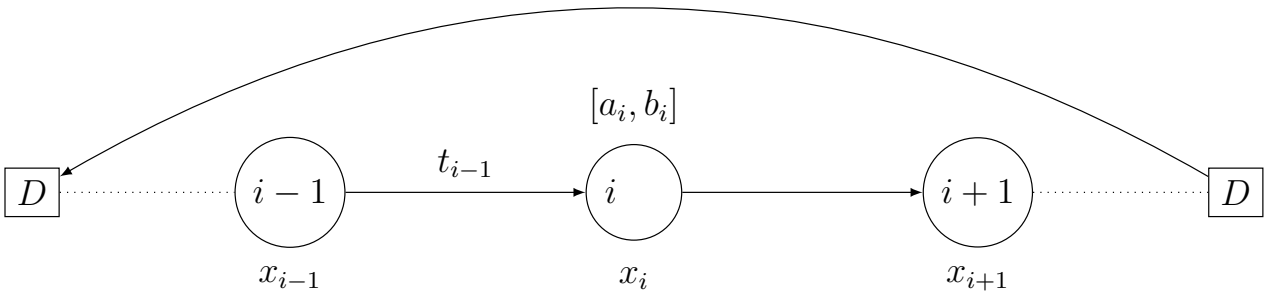
Given a sequence of n "stops" that must be serviced by a vehicle, we wish to find the best times $(x_1, x_2, \ldots, x_n)$ at which the vehicle must arrive at those stops.

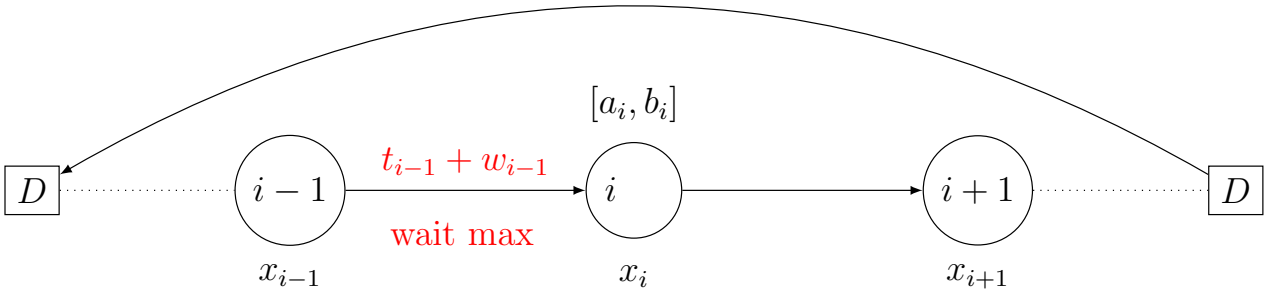
- Each stop (or node) corresponds to the pickup of a client or the delivery of a client.
- Each client has a convex utility function, denoted by f_i, describing his or her preference with regard to the pickup (or delivery) time. The value $\overline{x_i}$ at which f_i attains its minimum is the client's preferred time.
- For each i the time x_i must belong to the *window* $[a_i, b_i]$.
- The x_i must satisfy the so-called *travel constraints*, i.e., the constraints $x_i + T_i \leq x_{i+1}$ for $1 \leq i \leq n$.

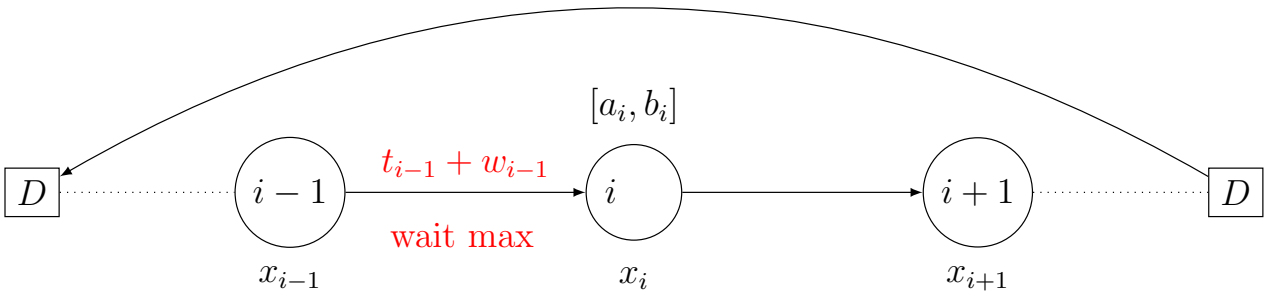
A more complete version of the scheduling problem

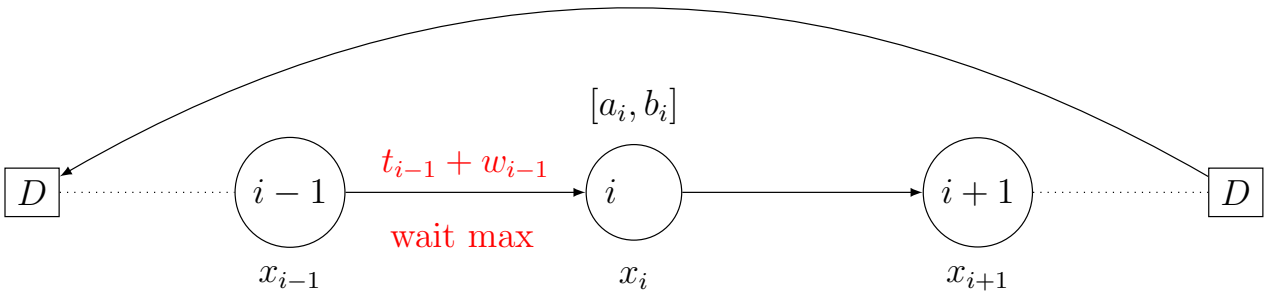
In real life we need to consider further constraints.

- The x_i must satisfy the *maximum wait constraints* $x_i + T_i + W_i \geq x_{i+1}$ for $1 \leq i \leq n$.
- Each client has a *maximum ride time*. Thus if the client is picked up at Stop i and delivered at Stop j, $x_j - x_i$ must not exceed a given upper bound. If the maximum span were exceeded, a penalty could be imposed.
- The *route span*, which is equal to $x_n - x_1$, should be minimized.


$[a_i, b_i]$
D
$i-1$
t_{i-1}
i
$i+1$
D
x_{i-1}
x_i
x_{i+1}


$[a_i, b_i]$
D
$i-1$
$t_{i-1} + w_{i-1}$
wait max
i
$i+1$
D
x_{i-1}
x_i
x_{i+1}


$[a_i, b_i]$
D
$i-1$
$t_{i-1} + w_{i-1}$
wait max
i
$i+1$
D
x_{i-1}
x_i
x_{i+1}
+
Pickup and Delivery Span
−



$+$ Pickup and Delivery Span $-$

Route Span

Goals of the workshop

There was a theoretical goal as well as "algorithmic goals".

- Solve the problem with three additional constraints:
  1. wait max at each stop,
  2. ride time max for each customer,
  3. route span max.
- Either:
  1. Extend the algorithm of Dumas et al. (for instances with window and travel constraints) to instances with window, travel, and maximum wait constraints.
  2. Design an exact algorithm (not necessarily a fast one) for the full version of the problem.
  3. Design a fast heuristic returning a good solution for the full version of the problem.

Reference: Dumas, Soumis, and Desrosiers have presented a linear-time algorithm for solving the simple version of the scheduling problem (*Transportation Science*, Vol. 24, No. 2, May 1990, pp. 145–152).

Part I - Theoretical properties

The simple version of the problem, denoted by (P), can be formulated as follows.

$$\min \sum_{i=1}^{n} f_i(x_i)$$

subject to

$$a_i \leq x_i \leq b_i \text{ for } i = 1, 2, \ldots, n-1$$

$$x_i + T_i \leq x_{i+1} \text{ for } i = 1, 2, \ldots, n$$

Observe that if we know that some optimal solution satisfies the constraint $x_i + T_i \leq x_{i+1}$ at equality, then we can eliminate a variable from the problem and solve a reduced problem that is equivalent to the original one.

The basic lemma

The *relaxed problem* is the problem described on the previous slide minus the travel constraints. Dumas et al. prove the following lemma.

Lemma

Let $\overline{X}$ denote an optimal solution of the relaxed problem. If $\overline{x}_k + T_k > \overline{x}_{k+1}$ holds for some k, then there is an optimal solution X^* of (P) satisfying $x^*_k + T_k = x^*_{k+1}$.

This lemma implies that (P) can be solved in linear time (assuming that the minimization of a convex function $f(x)$ takes a constant time). Note that solving the relaxation of the reduced problem only requires one function minimization.

The simple version plus the maximum wait constraints

We now include into (P) the maximum wait constraints.

$$\min \sum_{i=1}^{n} f_i(x_i)$$

subject to

$$a_i \leq x_i \leq b_i \text{ for } i = 1, 2, \ldots, n-1$$

$$x_i + T_i \leq x_{i+1} \text{ for } i = 1, 2, \ldots, n$$

$$x_i + T_i + W_i \geq x_{i+1} \text{ for } i = 1, 2, \ldots, n$$

To extend the algorithm by Dumas et al., we need a more general version of their lemma.

Two new lemmas

Now the relaxed problem is the problem (P) without the travel and maximum wait constraints. We first consider the case where the solution of the relaxed problem violates one of the maximum wait constraints.

Lemma 1

Let $\overline{X}$ denote an optimal solution of the relaxed problem. If $\overline{x}_k + T_k + W_k < \overline{x}_{k+1}$ holds for some k, then there is an optimal solution X^* of (P) satisfying one of the following constraints:

1. $x^*_k + T_k + W_k = x^*_{k+1}$,
2. $x^*_{k+1} + T_{k+1} = x^*_{k+2}$, and
3. $x^*_{k-1} + T_{k-1} = x^*_k$.

Etienne Beauchamp, Claudio Contardo , Furkan Enderer, Mounira Groiez, Nadia Lahrichi, Mehdi Mahnam, Odile Marcotte, Sofiane Soualah, Patrick St-Louis

Two new lemmas (continued)

There is a similar lemma for the case where the solution of the relaxed problem violates one of the travel constraints.

Lemma 2

Let $\overline{X}$ denote an optimal solution of the relaxed problem. If $\overline{x}_k + T_k > \overline{x}_{k+1}$ holds for some k, then there is an optimal solution X^* of (P) satisfying one of the following constraints:

1. $x^*_k + T_k = x^*_{k+1}$,
2. $x^*_{k+1} + T_{k+1} + W_{i+1} = x^*_{k+2}$, and
3. $x^*_{k-1} + T_{k-1} + W_{k-1} = x^*_k$.

We don't know yet how to use these lemmas for designing fast algorithms.

Part II - Algorithmic ideas and developments

- Dynamic Programming to solve P with the three additional constraints
- Heuristic Algorithm to optimize the ride time (two steps)
- Heuristic Algorithm to tighten the time windows and generate a (good?) initial solution

A- A dynamic programming algorithm

- A label $L = (i(L), x(L), f(L), (t_k)_{k=1}^{K})$ is a tuple as follows:
  1. $i(L)$: the current stop being visited
  2. $x(L)$: the arrival time to node i, which includes the waiting time
  3. $f(L) = \sum_{i \leq i(L)} f_{i(L)}(x_i(L))$: accumulated cost
  4. $t_k(L)$: accumulated traveling time of customer k
- The beginning of the recursion is represented by a label $L_0 = (0, 0, 0, (0)_{k=1}^{K})$

Etienne Beauchamp, Claudio Contardo , Furkan Enderer, Mounira Groiez, Nadia Lahrichi, Mehdi Mahnam, Odile Marcotte, Sofiane Soualah, Patrick St-Louis

A- A dynamic programming algorithm

- When the route is extended from a label L we do as follows:
  - If $i(L)+1$ is a delivery node:
    - We create a single label L' as follows:
    - $i(L') = i(L)+1$
    - $x(L') = \max\{a_{i+1}, x(L)+T_i\}$
    - $f(L') = f(L) + f_{i(L')}(i(L'))$
    - $(t_k)_{k=1}^{K}(L') = (t_k(L) + x(L') - x(L))_{k=1}^{K}$
  - If $i(L)+1$ is a pickup node:
    - We create $w_{i(L)}$ labels L', each of which corresponds to a different waiting time
    - For a given waiting time $w \in [0, w_{i(L)}]$, L' is as follows:
    - $i(L') = i(L)+1$
    - $x(L') = x(L) + T_i + w$
    - $f(L') = f(L) + f_{i(L')}(i(L'))$
    - $(t_k)_{k=1}^{K}(L') = (t_k(L) + x(L') - x(L))_{k=1}^{K}$
  - An extension is performed only if it respects the time windows and/or ride time constraints

Etienne Beauchamp, Claudio Contardo , Furkan Enderer, Mounira Groiez, Nadia Lahrichi, Mehdi Mahnam, Odile Marcotte, Sofiane Soualah, Patrick St-Louis

A- A dynamic programming algorithm

Etienne Beauchamp, Claudio Contardo , Furkan Enderer, Mounira Groiez, Nadia Lahrichi, Mehdi Mahnam, Odile Marcotte, Sofiane Soualah, Patrick St-Louis

Dominance rule

- A label L dominates another label L' if
  - $i(L) = i(L')$
  - $x(L) = x(L')$
  - $f(L) \leq f(L')$
  - $t_k(L) \leq t_k(L')$ for all k except for current customer if $i(L)$ is a delivery
- In this case, label L' is discarded

A- A dynamic programming algorithm

Etienne Beauchamp, Claudio Contardo , Furkan Enderer, Mounira Groiez, Nadia Lahrichi, Mehdi Mahnam, Odile Marcotte, Sofiane Soualah, Patrick St-Louis

Pseudocode

1. $\mathcal{L} = \{L_0\}$
2. while($\mathcal{L} \neq \emptyset$)
   - 2.1 Pick $L \in \mathcal{L}$ and do $\mathcal{L} \leftarrow \mathcal{L} \setminus \{L\}$
   - 2.2 Extend L to create one or several labels L'
   - 2.3 For each label L' apply dominance rule
   - 2.4 If L' is not discarded, do $\mathcal{L} \leftarrow \mathcal{L} \cup \{L'\}$
3. Return the optimal schedule

A- A dynamic programming algorithm

Acceleration techniques

- Decremental state-space relaxation (DSSR)
  - If for a certain customer the ride time constraint is discarded, the extension rule for its pickup node is as for a delivery one
  - $\Rightarrow$ Discard all ride time constraints and add them as needed
- Completion bounds
  - DSSR provides lower bounds on the optimal solution
  - IDEA: Use the previous iterations to detect and fathom unpromising labels

B- Heuristic method to include the ride time

Let C = Set of customers, $C \subset I$.

- Generate the initial solution X^0 with Dumas' algorithm while:
  1. Updating the ride time $r_i, i \in C$ for each customer;
  2. Updating $\sigma_i^+ = b_i - x_i^0$ and $\sigma_i^- = x_i^0 - a_i$ the tolerance at each stop $i \in I$ plus depot ;
- If $r_i > r_i^*, i \in C$: Solve P'.

B- The Model P'

Let:

- $\delta_i^+ =$ additional wait time at stop $i, \forall i \in I$;
- $\delta_i^- =$ wait time reduction at stop $i, \forall i \in I$;
- $y_i \in 0, 1$ ensuring either δ_i^+ or $\delta_i^- \neq 0$;
- $Z_i =$ excessive ride time for i and $Z_i = r_i - r_i^*$;
- $A_i =$ Cumulative variation of wait time at stop i, $A_i = \sum_{k=0}^{i-1}(\delta_k^+ - \delta_k^-)$.

B- The Model P' (continued)

$$Minimize \sum_{i \in C} Z_i^2 \tag{1}$$

subject to:

$$\delta_i^- \leq w_i \qquad \forall i \in I \tag{2}$$

$$\delta_i^- \leq \sigma_{i+1}^- \times y_i \qquad \forall i \in I \tag{3}$$

$$\delta_i^+ \leq \sigma_{i+1}^+ \times (1 - y_i) \qquad \forall i \in I \tag{4}$$

$$A_i + \delta_i^+ \leq \sigma_{i+1}^+ \qquad \forall i \in I \tag{5}$$

$$A_i + \delta_i^+ \geq -\sigma_{i+1}^- \qquad \forall i \in I \tag{6}$$

$$A_i - \delta_i^- \geq -\sigma_{i+1}^- \qquad \forall i \in I \tag{7}$$

$$(x_j^0 + A_j) - (x_i^0 + A_i) - r_i^* \leq Z_i \qquad \forall i \in C \tag{8}$$

$$x_i = A_i + x_i^0 \qquad \forall i \in I \tag{9}$$

$$A_i = \sum_{k=0}^{i-1} (\delta_k^+ - \delta_k^-) \qquad \forall i \in I \tag{10}$$

$$Z_i, \delta_i^+, \delta_i^- \geq 0 \qquad \forall i \in I \tag{11}$$

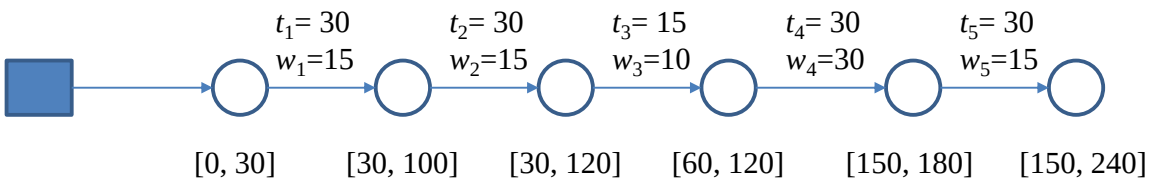
Etienne Beauchamp, Claudio Contardo , Furkan Enderer, Mounira Groiez, Nadia Lahrichi, Mehdi Mahnam, Odile Marcotte, Sofiane Soualah, Patrick St-Louis

C - Heuristic to tighten the time windows

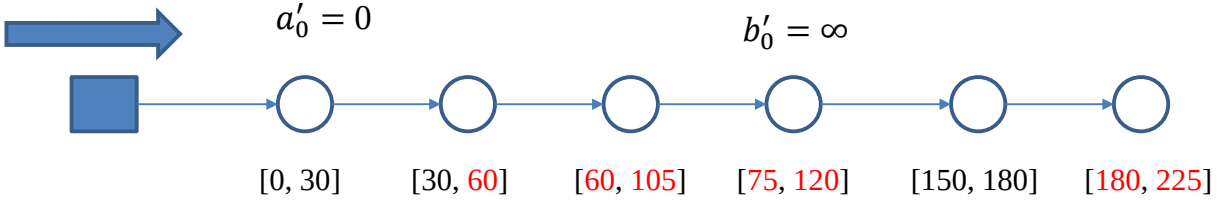
1. Set: $a'_0 = 0,\ b'_0 = \infty,\ t_0 = 0$ and $w^*_0 = 0$
2. For i $= 1$ to n
   - 2.1 $a'_i = \max\{a_i; a'_{i-1} + t_{i-1}\}$
   - 2.2 $b'_i = \min\{b_i; b'_{i-1} + t_{i-1} + w^*_{i-1}\}$
3. Set: $a''_{n+1} = 0$ and $b''_{n+1} = \infty$
4. For i $=$ n to 1
   - 4.1 $a''_i = \max\{a'_i; a''_{i+1} - t_i - w^*_i\}$
   - 4.2 $b''_i = \min\{b'_i; b''_{i+1} - t_i\}$
5. Set: $x_1 \in [a''_1; b''_1]$
6. For i $= 2$ to n
   - 6.1 $x_i = \begin{cases} \min\{b''_i; x_{i-1} + t_{i-1} + w_{i-1}*\} & \text{if } i \in C \\ \max\{a''_i; x_{i-1} + t_{i-1}\} & \text{otherwise} \end{cases}$

Etienne Beauchamp, Claudio Contardo , Furkan Enderer, Mounira Groiez, Nadia Lahrichi, Mehdi Mahnam, Odile Marcotte, Sofiane Soualah, Patrick St-Louis

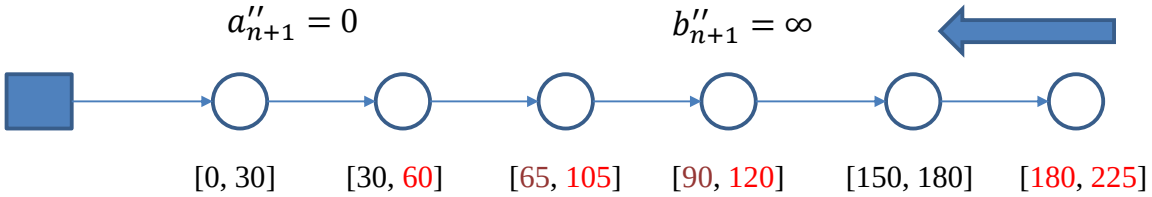
Illustration



First Phase: $a'_i = \max\{a_i, a'_{i-1} + t_{i-1}\}$ $a'_0 = 0$ $b'_i = \min\{b_i, b'_{i-1} + t_{i-1} + w_{i-1}\}$ $b'_0 = \infty$



Second Phase: $a''_i = \max\{a'_i, a''_{i+1} - t_i - w_i\}$ $a''_{n+1} = 0$ $b''_i = \max\{b'_i, b''_{i+1} - t_i\}$ $b''_{n+1} = \infty$



Etienne Beauchamp, Claudio Contardo , Furkan Enderer, Mounira Groiez, Nadia Lahrichi, Mehdi Mahnam, Odile Marcotte, Sofiane Soualah, Patrick St-Louis

Preliminary results

	$\sum_{i \in I} Z_i^2$	$\sum_{i \in I} Z_i$	$\bar{Z}_i$	Route Span
Initial Sol. (Dumas)	2904	298	6.48	594
Heuristic	2147	233	5.07	572
CPLEX with Dumas'	**172.67**	**21**	**0.45**	594
CPLEX version 2	385.25	70	1.52	589

Etienne Beauchamp, Claudio Contardo , Furkan Enderer, Mounira Groiez, Nadia Lahrichi, Mehdi Mahnam, Odile Marcotte, Sofiane Soualah, Patrick St-Louis

Further research

- Exploit the two new lemmas to develop an efficient algorithm
- Test the acceleration techniques for the dynamic programming algorithm
- Analyze and test P' to find better solutions
- Analyze the performance of the heuristic based on the TW tightening