

The Single-Agent Routing and Scheduling Problem with Schedule Deviation Penalties

Prunelle Vogler Frédéric Meunier Michel Bierlaire

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

VeroLog, Bath, July 07, 2026.



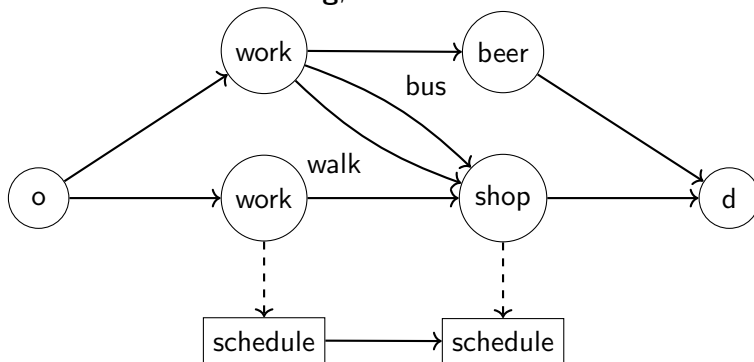
Outline

- Context and motivation
- Problem statement
- Complexity results
- Labeling algorithm
- Experimental results
- Conclusion

Single Agent Routing and Scheduling Problem

One agent. One time horizon T . Activities A .

Choose **what** to do, **when** to do it,
for how long, and **how to travel**.



Why study this problem?



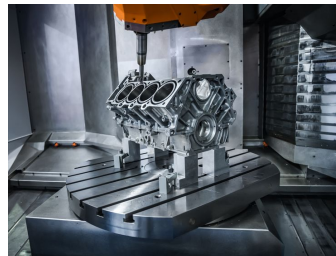
Activity-based models

How people plan their activities → transport demand
[Pougala et al., 2022]

SIMULATION

General structure

The SARSP contains classical problems.



Other applications

Machine scheduling
[Shabtay and Steiner, 2007],
home health care problem
[Di Mascolo et al., 2021].

Outline

- Context and motivation
- **Problem statement**
- Complexity results
- Labeling algorithm
- Experimental results
- Conclusion

Problem definition

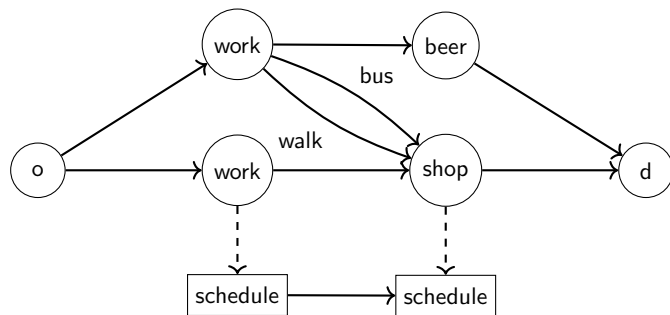
Input: a directed multigraph $G = (V, \vec{E})$ where $V = A \cup \{o, d\}$ + scheduling parameters + utility parameters.

Problem: “Find an $o - d$ path AND a **feasible** schedule maximizing the **utility** of the agent.”

Problem definition

Input: a directed multigraph $G = (V, \vec{E})$ where $V = A \cup \{o, d\}$ + scheduling parameters + utility parameters.

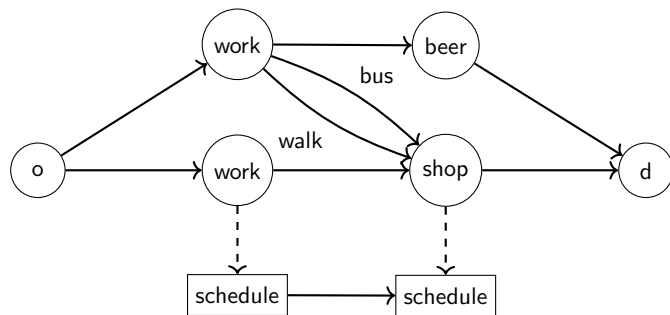
Problem: “Find an $o - d$ path AND a **feasible** schedule maximizing the **utility** of the agent.”



Problem definition

Input: a directed multigraph $G = (V, \vec{E})$ where $V = A \cup \{o, d\}$ + scheduling parameters + utility parameters.

Problem: “Find an $o - d$ path AND a **feasible** schedule maximizing the **utility** of the agent.”



Schedule modeling

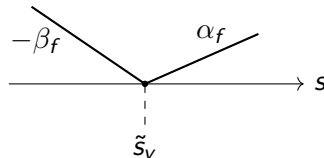
- scheduling variables (continuous) for all $v \in V$:
 - s_v (start),
 - τ_v (duration, redundant).

Scheduling constraints:

- minimum and maximum duration,
- time windows,
- time consistency: a task starts exactly after the end of the previous one and the travel time.

Utility function

Accumulated utility u_e for each arc e (activity participation, activity cost, travel time, travel cost).



Temporal utility for each vertex v :

- $\tilde{s}_v$ and $\tilde{\tau}_v$, exogenous preferred starting time and duration,
- $f_v(s) = \alpha_f(s - \tilde{s}_v)^+ + \beta_f(\tilde{s}_v - s)^+$, with $\alpha_f, \beta_f > 0$ given, (penalty deviation from $\tilde{s}_v$)
- $g_v(\tau) = \alpha_h(\tau - \tilde{\tau}_v)^+ + \beta_h(\tilde{\tau}_v - \tau)^+$, with $\alpha_g, \beta_g > 0$ given. (deviation from $\tilde{\tau}_v$)

$$\max \underbrace{\sum_{e \in \vec{E}} u_e z_e}_{\text{accumulated utility}} - \underbrace{\sum_{v \in V} f_v(s_v) y_v - \sum_{v \in V} g_v(\tau_v) y_v}_{\text{temporal utility}}$$

MILP formulation

Decision variables

$z_e \in \{0, 1\}$ arc selection, $y_v \in \{0, 1\}$ activity selection,
 $s_v \geq 0$ starting time, $\tau_v \geq 0$ duration.

$$\max \sum_{e \in \vec{E}} u_e z_e - \sum_{v \in V} f_v(s_v) y_v - \sum_{v \in V} g_v(\tau_v) y_v \quad (1a)$$

$$\text{s.t. } o\text{-}d \text{ path definition (flow constraints),} \quad (1b)$$

$$s_v \geq s_u + \tau_u + d_e z_e - b_u(1 - z_e), \quad \forall e = (u, v) \in \vec{E}, \quad (1c)$$

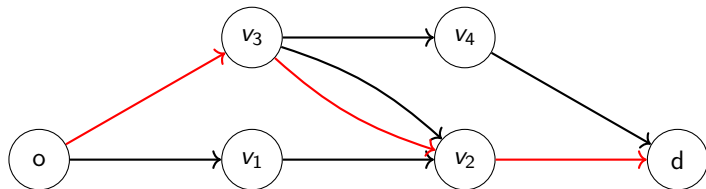
$$s_v \leq s_u + \tau_u + d_e z_e + (b_v - \tau_u)(1 - z_e), \quad \forall e = (u, v) \in \vec{E}, \quad (1d)$$

$$\text{time windows,} \quad \forall v \in V, \quad (1e)$$

$$\text{min and max durations,} \quad \forall v \in V, \quad (1f)$$

$$z_e \in \{0, 1\}, y_v \in \{0, 1\}, s_v \geq 0, \tau_v \geq 0, \quad \forall e \in \vec{E}, \forall v \in V. \quad (1g)$$

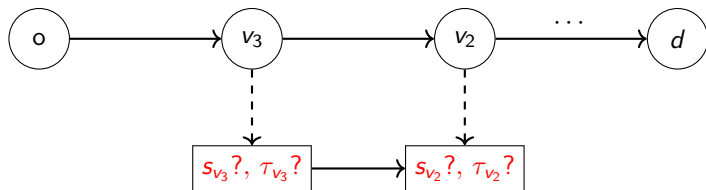
Decomposition in two subproblems



Orienteering subproblem

Longest path under time horizon and time windows.

→ heuristic, DP and labeling
[Shen et al., 2025].



Timing subproblem

From the **single machine scheduling problem with earliness and tardiness costs**. [Lakshminarayan et al., 1978]

Complexity: $\mathcal{O}(n \log n)$

[Chrétienne and Sourd, 2003],
[Sourd, 2005].

→ generalization of the $\mathcal{O}(n \log n)$. → labeling algorithm to solve SARSP in 2 cases.

Outline

- Context and motivation
- Problem statement
- Complexity results
- Labeling algorithm
- Experimental results
- Conclusion

Complexity results

Timing subproblem of SARSP $\mathcal{O}(n \log n)$

Proof inspired by [Tseng and Luo, 1996]:
 serial compositions of sums and
 inf-convolutions of simple
 piecewise-linear functions.
 Extension in $\mathcal{O}(\log n)$.

Given a path p :

$$\min \sum_{v \in V(p)} f_v(s_v) + \sum_{v \in V(p)} g_v(\tau_v)$$

$$\begin{aligned} \text{s.t. } s_v &= s_u + \tau_u + d_e, & \forall e = (u, v) \in p, \\ &\text{opening and closing time,} & \forall v \in V(p), \\ &\text{min and max durations,} & \forall v \in V(p), \\ s_v &\geq 0, \tau_v \geq 0, & \forall v \in V(p). \end{aligned}$$

where n is the number of breakpoints in the temporal utility function.

Outline

- Context and motivation
- Problem statement
- Complexity results
- **Labeling algorithm**
- Experimental results
- Conclusion

Two important warnings

- **No direct comparison of subpaths.**

Future decisions can improve or worsen the final objective (because not monotone).



⇒ Use LB and UB on complete $o-d$ paths.

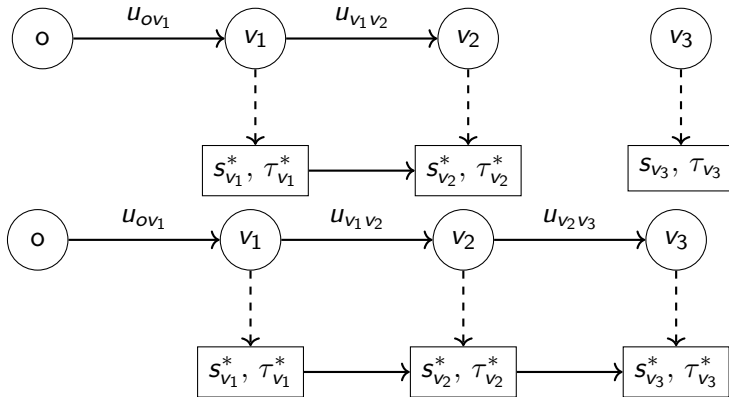
- **Temporal utility is not straightforward to extend.**

In theory: $\mathcal{O}(\log(n))$.

⇒ Focus on two easily extendable special cases.

- Earliest best (EB): preferred starting time $\tilde{s}_v = a_v$ opening time a_v , and no penalty for τ
- Same penalties (SP): coefficients are equals across the vertices and no penalty for τ

Why the temporal utility is not additive?



Labels definition

A label $\mathcal{L}$ is a tuple of:

- v ,
- $p = (e_1, \dots, e_{k-1})$, a partial path from o to v ,
- u^a : accumulated utility of p ,
- $t^{\min}$: earliest feasible arrival time,
- $t^{\max}$: latest feasible arrival time,
- u^t : temporal utility of the subpath,
- $s_1^*, \dots, s_k^*$: best starting times so far,
- $\text{LB}(\mathcal{L})$: lower bound on any feasible extension of p toward d ,
- $\text{UB}(\mathcal{L})$: upper bound on any feasible extension of p toward d .

Accumulated
resources $\mathcal{O}(1)$

Computed
resources:
 $\mathcal{O}(\log k)$

$\mathcal{O}(n^2)$
 $\mathcal{O}(k)$

Label update rule

Let Q be the current set of labels and z^* the objective of the best solution found so far, i.e.,

$$z^* = \max_{\mathcal{L} \in Q} \text{LB}(\mathcal{L}).$$

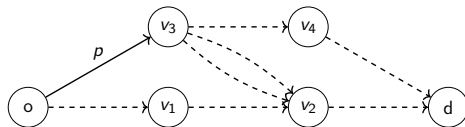
Given a new label $\mathcal{L}_{\text{new}}$:

- ① (Pruning) If $\text{UB}(\mathcal{L}_{\text{new}}) \leq z^*$, discard $\mathcal{L}_{\text{new}}$.
- ② If $\text{LB}(\mathcal{L}_{\text{new}}) \leq z^*$, add $\mathcal{L}_{\text{new}}$ to Q .
- ③ If $\text{LB}(\mathcal{L}_{\text{new}}) > z^*$:
 - Update $z^* \leftarrow \text{LB}(\mathcal{L}_{\text{new}})$,
 - (Pruning) Remove all labels $\mathcal{L}_j$ such that $\text{UB}(\mathcal{L}_j) < z^*$,
 - Add $\mathcal{L}_{\text{new}}$ to Q .

Lower bound: given a label $\mathcal{L}$ with a subpath p

Lower bound on the objective of **any extension of p**

- Construct a **feasible solution** up to d by adding the arc e with the best u_e while it's possible
- Return the utility of the path



Upper bound: given a label $\mathcal{L}$ with a subpath p

Proposition. For subpaths p and q sharing vertex v :

$$u^t(p + q) \leq u_o^t(p) + u_d^t(q)$$

$u^t(p)$: full problem temporal utility

$u_o^t(p)$: w/o final constraint $s_N + \tau_N = T$

$u_d^t(p)$: w/o initial constraint $s_1 = 0$

Upper bound: given a label $\mathcal{L}$ with a subpath p

Proposition. For subpaths p and q sharing vertex v :

$$u^t(p + q) \leq u_o^t(p) + u_d^t(q)$$

Consider an extension $p + q$ to d ,

$$\begin{aligned} u_{p+q} &= u_p^a + u_q^a + u^t(p + q) \\ &\leq u_p^a + u_q^a + u_o^t(p) + u_d^t(q) \quad (\text{Proposition}) \\ &\leq u_p^a + u_q^a + u_o^t(p) \quad (u_d^t(q) \leq 0) \\ &\leq u_p^a + \text{UB} + u_o^t(p). \quad u_q^a \leq \text{UB} \quad \forall q \end{aligned}$$

$u^t(p)$: full problem temporal utility

$u_o^t(p)$: w/o final constraint $s_N + \tau_N = T$

$u_d^t(p)$: w/o initial constraint $s_1 = 0$

Upper bound: given a label $\mathcal{L}$ with a subpath p

Proposition. For subpaths p and q sharing vertex v :

$$u^t(p+q) \leq u_o^t(p) + u_d^t(q)$$

Consider an extension $p+q$ to d ,

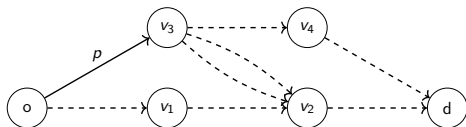
$$\begin{aligned} u_{p+q} &= u_p^a + u_q^a + u^t(p+q) \\ &\leq u_p^a + u_q^a + u_o^t(p) + u_d^t(q) \quad (\text{Proposition}) \\ &\leq u_p^a + u_q^a + u_o^t(p) \quad (u_d^t(q) \leq 0) \\ &\leq u_p^a + \text{UB} + u_o^t(p). \quad u_q^a \leq \text{UB} \quad \forall q \end{aligned}$$

$\Rightarrow$ UB: greedy algorithm to solve the **relaxation of a knapsack problem** where items=arcs, weight=time and gain=utility.

$u^t(p)$: full problem temporal utility

$u_o^t(p)$: w/o final constraint $s_N + \tau_N = T$

$u_d^t(p)$: w/o initial constraint $s_1 = 0$



Outline

- Context and motivation
- Problem statement
- Complexity results
- Labeling algorithm
- **Experimental results**
- Conclusion

Data and computational settings

Instances

- Based on the classical Solomon instances [Solomon, 1987]
- First 20 (and first 50) customers

Random utility parameters based on 3 activity categories [Pougala et al., 2023]

Activity	Prob.	γ_i	EB: α_i^f	SP: β_i^f	SP: α_i^f
work	40%	8	2.4	2.4	0.61
shopping	30%	5	0.61	2.4	0.61
leisure	30%	2	0.61	2.4	0.61

Computational settings

- Planning horizon: 24 hours
- MILP solved with GUROBI v12.0.3
- Time limit: 1800 seconds

Table 1: Computational results for the Earliest Best case.

Instance	Labeling: Earliest Best			Gurobi
	Created labels	Pruned labels	Time (s)	Time (s)
c107	4 627	15 827	1.59	1.43
c108	19 622	69 555	5.75	3.70
c207	10 610	35 479	3.03	27.22
c208	5 298	19 161	1.61	1.49
r102	2 763	8 820	0.79	1.49
r103	24 059	140 359	11.00	3.01
r104	249 562	1 827 588	146.19	125.13
r105	2 679	5 185	0.57	1.82
r202	21 667	55 394	5.50	2.74
r203	83 307	454 773	38.51	13.02
r204	1 093 451	6 749 418	1293.25	1800.11
r205	45 644	97 688	10.83	13.30
rc101	2 384	4 479	0.54	1.87
rc102	30 506	129 297	11.87	4.45

Table 2: Computational results for the Same Penalties case.

Instance	Labeling: Same Penalties			Gurobi
	Created labels	Pruned labels	Time (s)	Time (s)
c107	3 649	12 390	1.23	2.30
c108	12 684	46 023	4.16	6.52
c207	3 826	8 935	0.95	14.75
c208	5 282	18 402	1.97	3.78
r102	696	3 466	0.30	2.99
r103	22 248	135 776	10.59	209.69
r104	928 513	5 451 106	529.91	1800.26
r105	1 543	3 215	0.46	1.21
r202	3 773	14 638	1.50	9.28
r203	55 181	332 491	30.91	63.85
r204	2 869 414	15 340 764	1802.05	1800.21
r205	40 079	94 672	11.17	29.56
rc101	1 272	2 654	0.30	0.54
rc102	10 107	46 135	4.54	12.38

Outline

- Context and motivation
- Problem statement
- Complexity results
- Labeling algorithm
- Experimental results
- Conclusion

Further questions

Useful method for OP or similar problems

Extend to the general case in terms of temporal utility

Experiments on real instances

- ABM instances
- Home health care instances with preferred service time

Comparison

- Constraint programming approach
- Heuristic

Use to solve the multi-agent problem with synchronization

Thanks for your attention :)

Bibliography I



Chrétienne, P. and Sourd, F. (2003).
PERT scheduling with convex cost functions.
Theoretical Computer Science, 292(1):145–164.



Di Mascolo, M., Martinez, C., and Espinouse, M.-L. (2021).
Routing and scheduling in Home Health Care: A literature survey and bibliometric analysis.
Computers & Industrial Engineering, 158:107255.



Lakshminarayan, S., Lakshmanan, R., Papineau, R. L., and Rochette, R. (1978).
Optimal Single-Machine Scheduling with Earliness and Tardiness Penalties.
Operations Research, 26(6):1079–1082.



Bibliography II



Pougala, J., Hillel, T., and Bierlaire, M. (2022).
Capturing trade-offs between daily scheduling choices.
Journal of Choice Modelling, 43:100354.



Pougala, J., Hillel, T., and Bierlaire, M. (2023).
OASIS: Optimisation-based Activity Scheduling with Integrated Simultaneous choice dimensions.
Transportation Research Part C: Emerging Technologies, 155:104291.



Shabtay, D. and Steiner, G. (2007).
A survey of scheduling with controllable processing times.
Discrete Applied Mathematics, 155(13):1643–1666.



Bibliography III

 Shen, S., Zhou, Y., Lei, Q., and Wu, Z. (2025).

A survey of the orienteering problem: model evolution, algorithmic advances, and future directions.

[arXiv:2512.16865 \[math\]](#).

 Solomon, M. M. (1987).

Algorithms for the Vehicle Routing and Scheduling Problems with Time Window Constraints.

Operations Research, 35(2):254–265.

 Sourd, F. (2005).

Optimal timing of a sequence of tasks with general completion costs.

European Journal of Operational Research, 165(1):82–96.



Bibliography IV



Tseng, P. and Luo, Z.-Q. (1996).

On Computing the Nested Sums and Infimal Convolutions of Convex Piecewise-Linear Functions.

Journal of Algorithms, 21(2):240–266.