

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

Swiss OR Days, Bern.

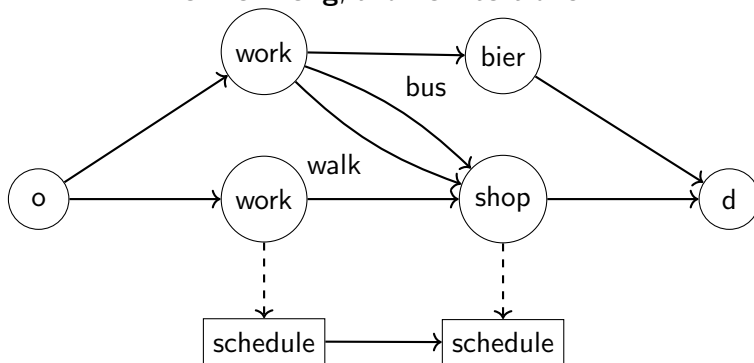


Outline

- 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?



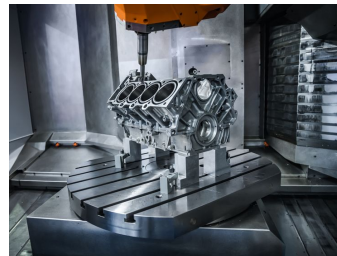
Activity-based models

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



General structure

The SARSP contains several classical routing and scheduling problems as special cases or subproblems.



Other applications

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

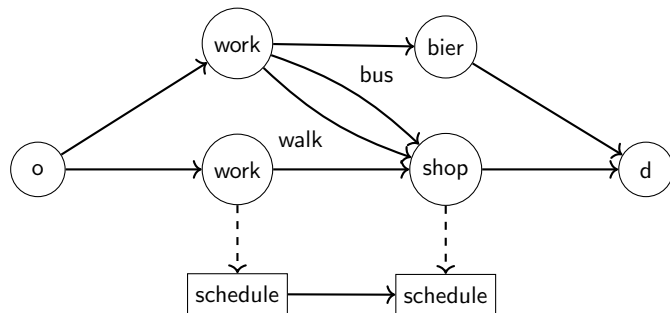
Outline

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

Problem definition

Input: one agent + a directed multigraph ($o, d \in V$) + scheduling parameters + utility parameters.

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



Problem definition

Input: one agent + a directed multigraph ($o, d \in V$) + scheduling parameters + utility parameters.

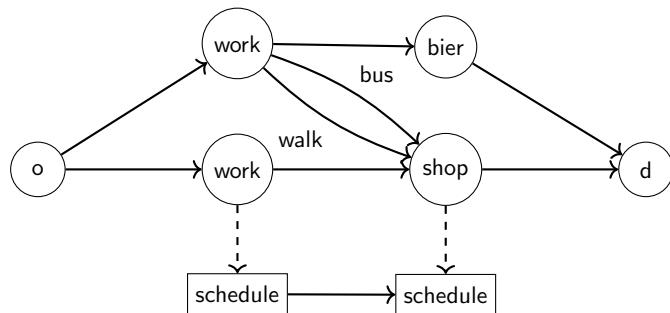
Problem: “Find an $o - d$ path AND a feasible schedule maximizing the utility of the agent.”

Variables:

- path variables (binary),
- scheduling variables (continuous):
 x_v (start), τ_v (duration) for all
 $v \in V$.

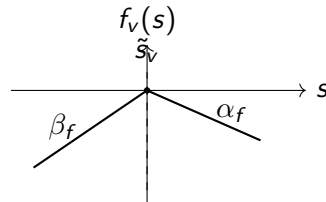
Scheduling constraints:

- minimum and maximum duration,
- opening hours,
- time consistency: a task starts 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, (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 $\tilde{\tau}_v$)

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

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.

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

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

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

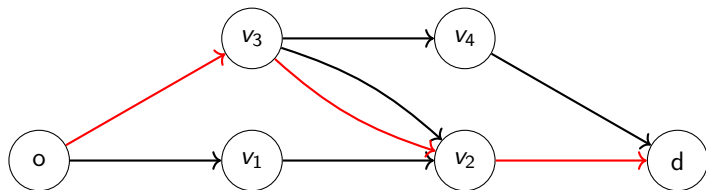
$$s_v \leq s_u + \tau_u + d_e z_e + (b_v - \underline{\tau}_u)(1 - z_e), \quad \forall e = (u, v) \in A, \quad (1d)$$

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

$$\text{bounds on activity durations } \tau_v, \quad \forall v \in V, \quad (1f)$$

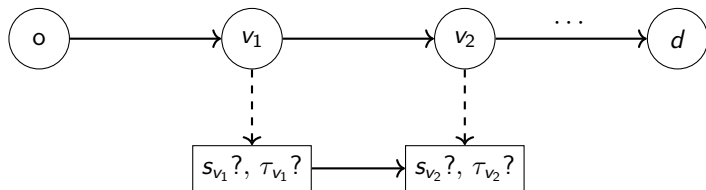
$$z_e \in \{0, 1\}, y_v \in \{0, 1\}, s_v \geq 0, \tau_v \geq 0, \quad \forall e \in A, \forall v \in V. \quad (1g)$$

Decomposition in two subproblems



Orienteering subproblem

Longest path under resource constraints.
Introduced by [Golden et al., 1987].
Survey [Shen et al., 2025].



Timing subproblem

From the **single machine scheduling problem with earliness and tardiness costs**. [Lakshminarayan et al., 1978]
Timing subproblem
[Chrétienne and Sourd, 2003],
[Sourd, 2005]

Contributions: 2 complexity results AND a labeling algorithm to solve SARSP solving the timing subproblem (LP) at each iteration.

Outline

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

Complexity results

SARSP NP-hard

Proof by reduction from the knapsack problem.

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

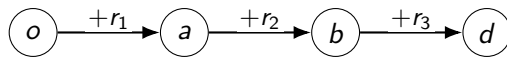
Proof inspired by [Tseng and Luo, 1996]: serial compositions of sums and inf-convolutions of simple piecewise-linear functions.

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

Labeling algorithm for Resource Constrained Shortest Path Problem



Along the path: $R = r_1 + r_2 + r_3$

Easy forward label extension and dominance rules, when resources are:

- **monotone** along the path,
- accumulated additively along the path, **no need to evaluate a function that depends on the whole path.**

Why the temporal utility is not additive?

In our case:

$$\min \underbrace{\sum_{e \in A} c_e z_e}_{\text{accumulated utility}} + \underbrace{\sum_{v \in V} f_v(s_v) y_v + \sum_{v \in V} g_v(\tau_v) y_v}_{\text{temporal utility}}$$

- accumulated utility: accumulated (additive) but not monotone
- temporal utility: monotone but not accumulated

Solution: **augment the label state**, carrying upper bound and lower bound on the utility of the whole path.

Labels definition

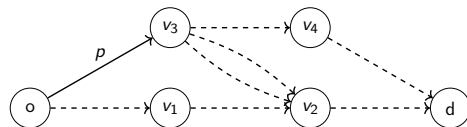
A label $\mathcal{L}$ is a tuple of:

- v ,
 - $p = e_1, \dots, e_{k-1}$, $k - 1$ arcs from o to v ,
 - u^a : accumulated utility of p ,
 - $t^{\min}$: earliest feasible arrival time,
 - $t^{\max}$: latest feasible arrival time,
- (Accumulated resources)
-
- u^t : **sofar optimal temporal utility of the subpath** (solving an LP, the timing subproblem),
 - $\text{LB}(\mathcal{L})$: **lower bound** on an optimal $o - d$ path starting with p ,
 - $\text{UB}(\mathcal{L})$: **upper bound** on an optimal $o - d$ path starting with p .
- (Computed resources)

Bounds: 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
- **Solve the LP** $\rightarrow$ temporal utility
- Return: accumulative utility + temporal utility



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

$u^t(p)$: full problem optimal

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

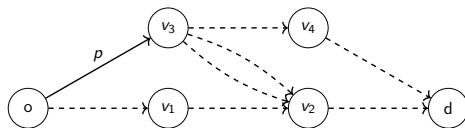
$u_d^t(p)$: w/o initial constraint $s_1 = 0$

Consider an extension $p + q$ to d ,

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

Proposition. For subpaths p and q sharing vertex v :

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



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

$$UB = \max_q \sum_{e \in q} u_e \quad (2)$$

$$\text{s.t.} \quad \sum_{e \in q} t_e \leq T - t_v \text{ early}, \quad (3)$$

$$V(q) \cap V(p) = \emptyset, \quad (4)$$

$$q \in \mathcal{P}_{vd}. \quad (5)$$

- Option 1: greedy algorithm to solve the **relaxation of a knapsack problem**
- Option 2: LP to solve the relaxation of the actual problem $\rightarrow$ **flow problem**
- Option 3: **precomputed MILP** to solve the exact bound for different time horizon

Label update rule

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

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

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 $z^* \leftarrow \text{LB}(\mathcal{L}_{\text{new}})$,
 - (Dominance) Remove all labels $\mathcal{L}_j$ such that $\text{UB}(\mathcal{L}_j) < z^*$,
 - Add $\mathcal{L}_{\text{new}}$ to Q .

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]
- Two sets: first 20 and first 50 customers

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

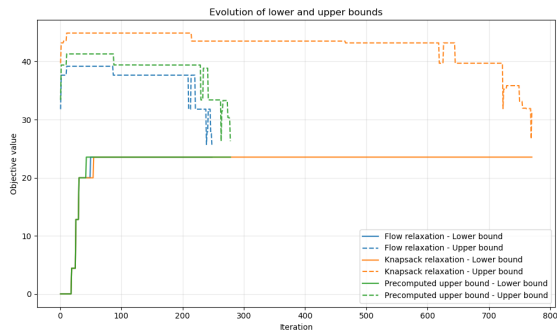
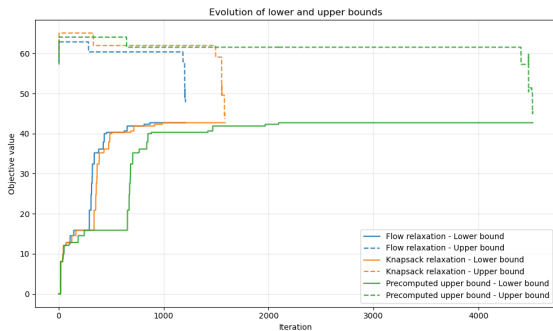
Activity	Prob.	γ	early/late	short/long
work	40%	8	-2.4	-9.6
shopping	30%	5	-0.61	-2.4
leisure	30%	2	-0.61	-2.4

Computational settings

- Planning horizon: 24 hours
- MILP and LPs in the labeling algorithm solved with GUROBI v12.0.3
- Time limit: 600 seconds (first set) 3600 seconds (second set)

Instance	V	Precomp			Knapsack	Flow	Gurobi
		total (s)	presol(s)	lab (s)	lab(s)	lab(s)	MILP(s)
from_c107.csv	20	44.79	16.80	27.99	18.81	42.79	2.55
from_c108.csv	20	120.50	18.75	101.74	60.84	130.97	1.68
from_c207.csv	20	46.30	31.31	14.99	9.31	12.37	3.62
from_c208.csv	20	87.88	31.56	56.32	32.57	52.83	2.64
from_r102.csv	20	27.87	22.74	5.13	5.27	8.66	2.13
from_r103.csv	20	124.62	27.47	97.15	94.37	141.71	5.41
from_r104.csv	20	634.74	34.72	600.01	600.01	600.02	44.12
from_r105.csv	20	27.92	23.45	4.47	7.17	8.38	0.91
from_r202.csv	20	60.48	22.85	37.63	41.28	62.24	6.83
from_r203.csv	20	312.51	29.01	283.50	284.70	465.79	9.45
from_r204.csv	20	632.30	32.29	600.01	600.02	600.02	44.49
from_r205.csv	20	285.51	20.72	264.79	329.46	352.58	13.97
from_rc101.csv	20	29.81	21.97	7.85	8.87	14.48	1.53
from_rc102.csv	20	116.62	23.24	93.39	65.26	87.87	4.30
from_rc201.csv	20	31.51	22.96	8.55	11.92	19.84	0.74
from_rc202.csv	20	116.10	27.86	88.24	61.19	78.11	7.09

Lower and upper bound analysis



Outline

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

Further questions

Improve the understanding of the problem

- why is the labeling algorithm not competitive?
- what is the closest classical problem?

Experiments on realistic instances

- ABM instances
- Home care scheduling instances
- Comparison with existing home care scheduling methods

Future

- Use it in decomposition methods: Multi-agent scheduling problem
- ABM with households or social networks

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.



Golden, B. L., Levy, L., and Vohra, R. (1987).
The orienteering problem.
Naval Research Logistics, 34(3):307–318.



Bibliography II



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.



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.



Bibliography III



Shabtay, D. and Steiner, G. (2007).

A survey of scheduling with controllable processing times.

Discrete Applied Mathematics, 155(13):1643–1666.



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.



Bibliography IV



Sourd, F. (2005).

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

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



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.

