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

STRC, Ascona.

EPFL

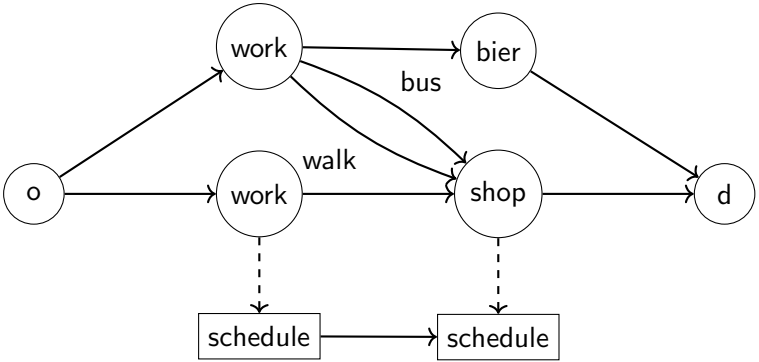
Outline

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**.

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**.



Temporal constraints

Penalties for the deviation from the **preferred schedule**

Looking for an **o-d path** and a **feasible schedule** (start time and duration)

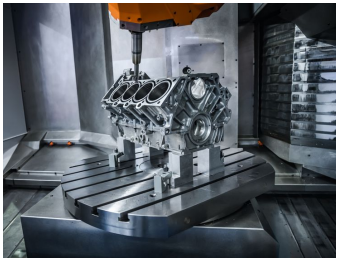
Why study this problem?

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].

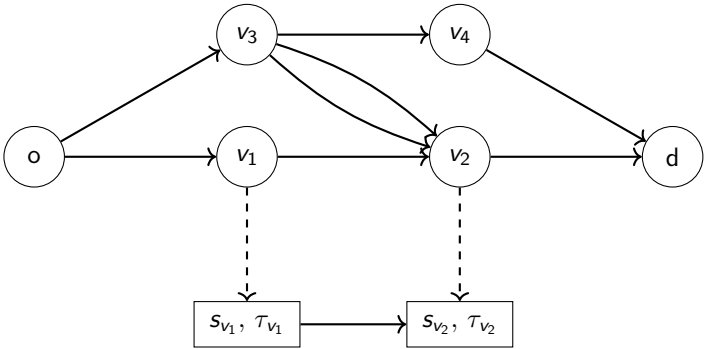
Outline

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."

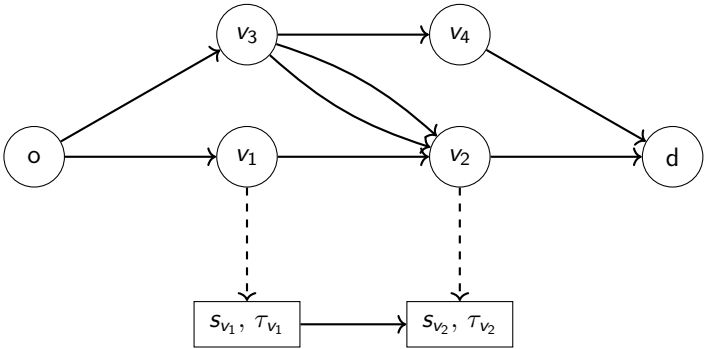
Problem definition

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."



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:

- Utility for each activated vertex (activity participation, activity cost)
- Utility for each crossed arc (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.

$$\max \quad \underbrace{\sum_{e \in A} u_e z_e}_{\text{accumulated utility}} \quad \underbrace{- \sum_{i \in V} f_i(s_i) y_i - \sum_{i \in V} g_i(\tau_i) y_i}_{\text{temporal utility}}$$

MILP formulation

Decision variables

$$z_e \in \{0,1\} \quad \text{arc selection}, \qquad y_i \in \{0,1\} \quad \text{activity selection},$$
$$s_i \geq 0 \quad \text{starting time}, \qquad \tau_i \geq 0 \quad \text{duration}.$$

$$\max \quad \sum_{e \in A} u_e z_e - \sum_{i \in V} f_i(s_i) y_i - \sum_{i \in V} g_i(\tau_i) y_i \tag{1a}$$

s.t. o-d path definition, (1b)

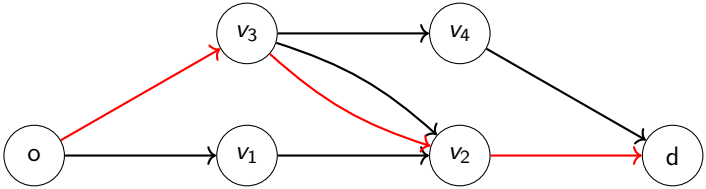
$$s_v \geq s_u + \tau_u + d_e z_e - b_u(1 - z_e), \qquad \forall e = (u,v) \in A, \tag{1c}$$
$$s_v \leq s_u + \tau_u + d_e z_e + (b_v - \underline{\tau}_u)(1 - z_e), \qquad \forall e = (u,v) \in A, \tag{1d}$$

opening and closing time, $\forall i \in V$, (1e)

bounds on activity durations τ_i, $\forall i \in V$, (1f)

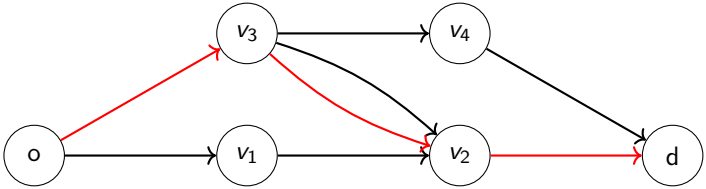
$$z_e \in \{0,1\},\ y_i \in \{0,1\},\ s_i \geq 0,\ \tau_i \geq 0, \qquad \forall e \in A,\ \forall i \in V. \tag{1g}$$

Decomposition in two subproblems

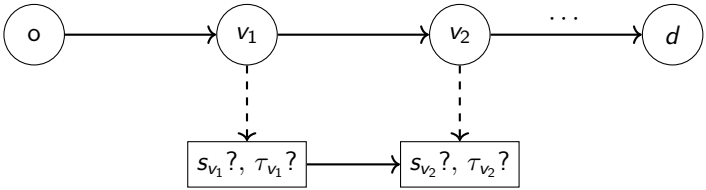


Orienteering subproblem
Longest path under resource constraints.
Introduced by [Golden et al., 1987].
Survey [Shen et al., 2025].

Decomposition in two subproblems

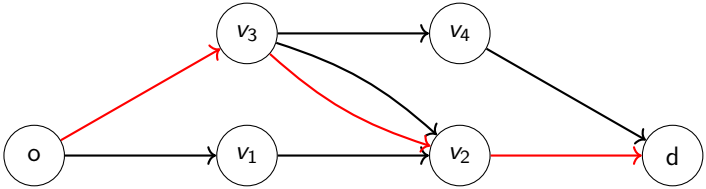


Orienteering subproblem
Longest path under resource constraints.
Introduced by [Golden et al., 1987].
Survey [Shen et al., 2025].

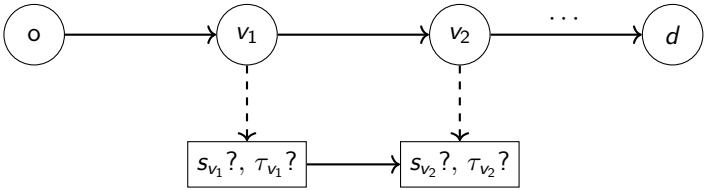


Timing subproblem
From the **single machine scheduling problem with earliness and tardiness costs**. [Lakshminarayan et al., 1978]
Timing subproblem
[Chrétienne and Sourd, 2003],
[Sourd, 2005]

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

Complexity results

SARSP

NP-hard

Proof by reduction from the knapsack problem.

Timing subproblem of SARSP

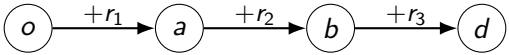
$\mathcal{O}(n \log n)$

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

Labeling algorithm for Resource Constrained Shortest Path Problem

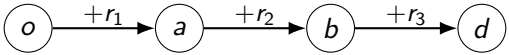


Along the path: $R = r_1 + r_2 + r_3$

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**.

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:

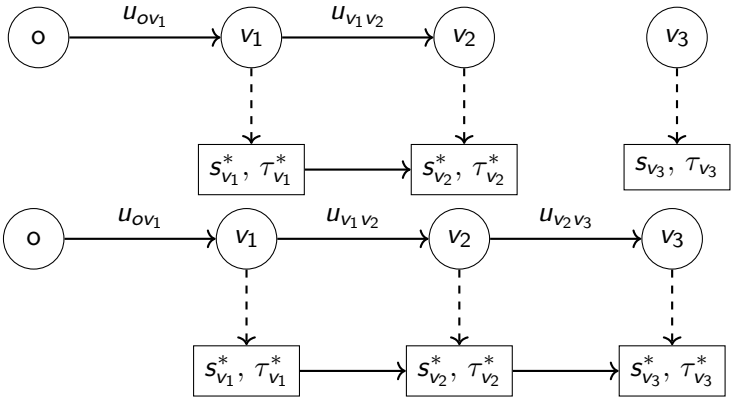
- **monotone** along the path,
- accumulated additively along the path, **no need to evaluate a function that depends on the whole path**.

In our case:

$$\max \quad \underbrace{\sum_{e \in A} u_e z_e}_{\text{accumulated utility}} \quad \underbrace{- \sum_{i \in V} f_i(s_i) y_i - \sum_{i \in V} g_i(\tau_i) y_i}_{\text{temporal utility}}$$

- accumulated utility: accumulated (additive) but not monotone

Why the temporal utility is not additive?



- 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,
- $p = e_1, ...e_{k-1}$, $k-1$ 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)

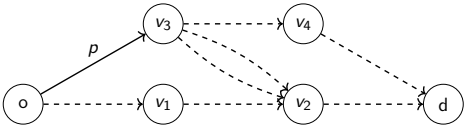
- u^t: **sofar optimal temporal utility of the subpath**,
- $\mathrm{LB}(\mathcal{L})$: **lower bound** on an optimal $o-d$ path starting with p,
- $\mathrm{UB}(\mathcal{L})$: **upper bound** on an optimal $o-d$ path starting with p.

(Computed resources)

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

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

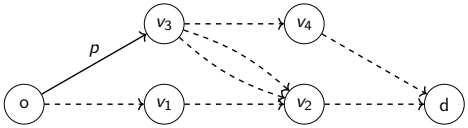
- Construct a **feasible solution** up to d
- **Solve the LP** $\rightarrow$ temporal utility
- Return: accumulative utility + temporal utility



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

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



Upper bound on the objective of any extension of p

Consider an extension $p + q$ to d,

$$
\begin{aligned}
u_{p+q} &= u^a_p + u^a_q + u^t_{p+q} && (u^t_{p+q} \leq u^t_p + u^t_q) \\
&\leq u^a_p + u^a_q + u^t_p + u^t_q && (u^t_q \leq 0) \\
&\leq u^a_p + u^a_q + u^t_p && u^a_q \leq \mathrm{UB}^{\mathrm{knap}} \quad \forall q \\
&\leq u^a_p + \mathrm{UB}^{\mathrm{knap}} + u^t_p.
\end{aligned}
$$

$\mathrm{UB}^{\mathrm{knap}}$: knapsack relaxation with weight = time and gain = utility.

Label update rule

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

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

Label update rule

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

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

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

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

Theorem

At any iteration, for all $\mathcal{L}$ and $\mathcal{L}'$ in Q:

$$\mathrm{UB}(\mathcal{L}) > z^\star \geq \mathrm{LB}(\mathcal{L}').$$

Outline

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) 3000 seconds (second set)

Instance	$\|A\|$	Labels	Dom.	Prun.	Lab (s)	MILP (s)	MILP / Lab
from_c107	20	5515	7	463	8.26	1.43	0.17
from_c108	20	15320	20	2706	22.01	2.60	0.12
from_c207	20	7086	676	3445	7.96	7.23	0.91
from_c208	20	8044	81	4509	9.31	1.08	0.12
from_r102	20	2537	12	1126	2.92	4.09	1.40
from_r103	20	8025	9	3947	9.21	52.98	5.76
from_r104	20	43376	725	24296	52.02	600.45	11.54
from_r105	20	478	1	111	0.55	0.34	0.62
from_r202	20	4797	13	2859	4.75	3.37	0.71
from_r203	20	15997	219	8451	16.35	69.22	4.23
from_r204	20	93982	98	52288	94.16	600.27	6.37
from_r205	20	2098	7	620	2.04	2.37	1.16
from_rc101	20	716	0	100	0.84	0.61	0.72
from_rc102	20	17344	4	6290	21.72	16.22	0.75
from_rc201	20	614	4	106	0.62	0.23	0.37
from_rc202	20	31401	1227	14135	33.39	15.94	0.48

Table 1: Results on the small instances (20 vertices).

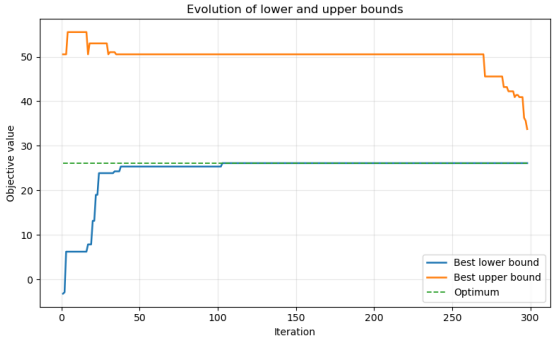
EPFL

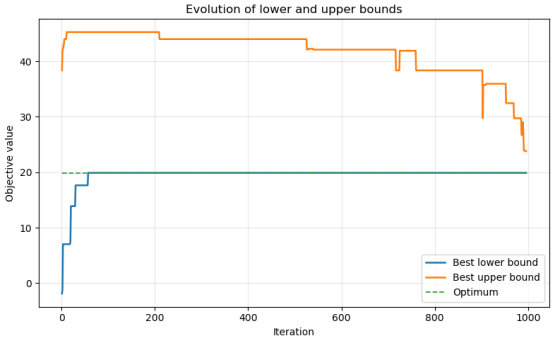
Instance	$\|A\|$	Labels	Dom.	Prun.	Lab (s)	MILP (s)	MILP / Lab
from_c107	50	58027	601	24987	103.50	5.30	0.05
from_c108	50	229866	1558	106903	454.21	37.63	0.08
from_c207	50	1803633	62180	1056102	3003.28	3000.76	1.00
from_c208	50	836801	8125	439774	1268.86	468.15	0.37
from_r102	50	78614	1836	43311	110.31	2957.33	26.81
from_r103	50	1255485	38214	804083	2203.03	3000.21	1.36
from_r104	50	1525185	8719	976177	3003.85	3000.23	0.99
from_r105	50	21732	435	8277	37.70	4.18	0.11
from_r202	50	308490	27555	209913	400.74	3000.34	7.49
from_r203	50	2155119	88236	1533945	3000.91	3000.18	0.99
from_r204	50	1932961	18053	1263880	3001.13	3000.43	0.99
from_r205	50	207175	13575	118414	319.19	15.51	0.05
from_rc101	50	4195	69	1081	6.84	0.78	0.11
from_rc102	50	63981	1373	31664	114.78	93.95	0.82
from_rc201	50	2985	16	894	4.47	0.72	0.16
from_rc202	50	88268	603	52073	142.81	71.13	0.50

Table 2: Results on the medium instances (50 vertices).

EPFL

Lower and upper bound analysis





Outline

Further questions

Upper bound

- Improve the current upper bound
- Current version: greedy knapsack relaxation
- Limitation: selected arcs do not necessarily form a path

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 housholds or social networks

Thanks for your attention :)

EPFL

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.

TRANSP-OR

EPFL

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.

TRANSP-OR

EPFL

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.

TRANSP-OR

EPFL

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.

TRANSP-OR

EPFL