The Parameterized Complexity of Scheduling with Precedence Delays: Shuffle Product and Directed Bandwidth

Hans L. Bodlaender[1], Maher Mallem[2]

[1]Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands

[2]Université Grenoble Alpes, CNRS, Grenoble INP, G-SCOP, 38000 Grenoble, France

June 9, 2026

Plan

1. Motivation
   - Scheduling with precedence delays
   - The XNLP class
2. Contribution
   - Summary
   - Shuffle Product
   - Directed Bandwidth
3. Discussion
4. References

Outline

Cooking is scheduling

Cooking is scheduling



Cooking = sequence of tasks ruled by a partial order and waiting time constraints.

Precedence delays

- If τ is a feasible schedule and $(i \xrightarrow{\ell_{i,j}} j)$:
  - Exact delay: $\boxed{i} \xrightarrow{=\ell_{i,j}} \boxed{j}$
  - Minimum delay: $\boxed{i} \xrightarrow{\geq \ell_{i,j}} \boxed{j}$
  - Maximum delay: $\boxed{i} \xrightarrow{\leq \ell_{i,j}} \boxed{j}$

Why precedence delays?

Powerful modeling tool

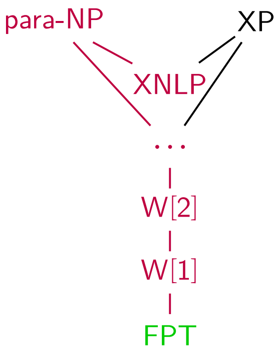
- Cooking.
- Setup times.
- Coupled tasks (e.g., loading / unloading operations).

Many related problems outside scheduling

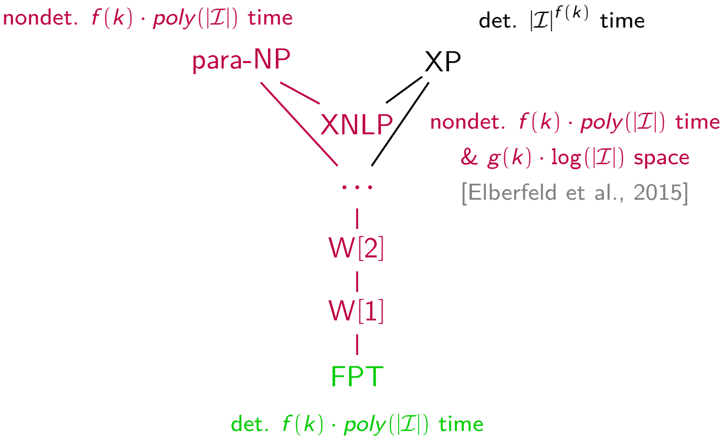
- Minimum: SHUFFLE PRODUCT.
  - W[2]-hard(#source words) [van Bevern et al., 2016].
- Exact: UNARY BIN PACKING.
  - W[1]-hard(#bins) [Jansen et al., 2013].
- Maximum: DIRECTED BANDWIDTH.
  - W[t]-hard(target value) for all t [Bodlaender, 2021].

Outline

Parameterized complexity classes



Parameterized complexity classes



Parameterized complexity classes

nondet. $f(k) \cdot poly(|\mathcal{I}|)$ time

para-NP

$\searrow$

XNLP — nondet. $f(k) \cdot poly(|\mathcal{I}|)$ time & $g(k) \cdot \log(|\mathcal{I}|)$ space [Elberfeld et al., 2015]

$\downarrow$

$\cdots$

$\downarrow$

W[2]

$\downarrow$

W[1]

Parameterized complexity classes

nondet. $f(k) \cdot poly(|\mathcal{I}|)$ time

para-NP

$\downarrow$

XNLP

nondet. $f(k) \cdot poly(|\mathcal{I}|)$ time
& $g(k) \cdot \log(|\mathcal{I}|)$ space
[Elberfeld et al., 2015]

$\downarrow$

$\cdots$

$\downarrow$

W[2]

$\downarrow$

W[1]

Recent use of XNLP in scheduling

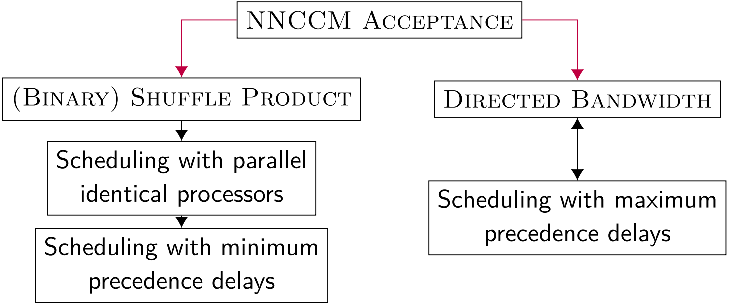
- [Bodlaender et al., 2022] Parallel identical machines, no precedence delays.
  - Width, #machines.
- [Mallem et al., 2026] Single machine with precedence delays.
  - Width, maximum delay value.

Outline

1. Motivation
   - Scheduling with precedence delays
   - The XNLP class
2. Contribution
   - Summary
   - Shuffle Product
   - Directed Bandwidth
3. Discussion
4. References

Summary of contributions

Problem	*Parameter*	*Result*
(Binary) Shuffle Product	#source words	XNLP-complete
Directed Bandwidth	target value plus width	XNLP-complete for directed acyclic graphs
	target value	XNLP-complete for directed trees



Nondeterministic Nonincreasing Counter Check Machine (NNCCM)

NNCCM Acceptance

Given: the description of a NNCCM:

- number of counters k, max. counter value n, number of checks r,
- a sequence of r checks in $([1, k] \times [0, n] \times [1, k] \times [0, n])^r$.

Parameter: number of counters k.

Question: is there an accepting run of the NNCCM?

Example: $k = 3$, $n = 2$, $r = 2$ checks: $((1, 0, 2, 0), (1, 1, 3, 0))$.

		r_1	r_2
c_1	0	1	1
c_2	0	0	0
c_3	0	0	1

		r_1	r_2
c_1	0	2	2
c_2	0	0	0
c_3	0	0	0

Outline

(Binary) Shuffle Product

(Binary) Shuffle Product

Given: source words $s_1, \ldots, s_k$ and target word t over a (binary) alphabet Σ.

Parameter: number of source words k.

Question: is t in the shuffle product of $s_1, \cdots, s_k$?

$$\begin{array}{rcccccccccc} s_1 & = & c & b & a & a \\ s_2 & = & a & b & c \\ s_3 & = & c & a \\ t & = & a & c & b & b & c & a & a & a & c \end{array}$$

(Binary) Shuffle Product

(BINARY) SHUFFLE PRODUCT

Given: source words $s_1, \ldots, s_k$ and target word t over a (binary) alphabet Σ.

Parameter: number of source words k.

Question: is t in the shuffle product of $s_1, \cdots, s_k$?

$$\begin{array}{rcccccccccc}
s_1 = & & c & b & & & a & a & & \\
s_2 = & a & & & b & & & & & c \\
s_3 = & & & & & c & & & a & \\
t = & a & c & b & b & c & a & a & a & c
\end{array}$$

(Binary) Shuffle Product

(Binary) Shuffle Product

Given: source words $s_1, \ldots, s_k$ and target word t over a (binary) alphabet Σ.

Parameter: number of source words k.

Question: is t in the shuffle product of $s_1, \cdots, s_k$?

$$
\begin{array}{rcccccccccc}
s_1 = & & c & & b & & a & a & & \\
s_2 = & a & & b & & & & & & c \\
s_3 = & & & & & c & & & a & \\
t = & a & c & b & b & c & a & a & a & c
\end{array}
$$

(Binary) Shuffle Product (cont.)

Example: $k = 3$, $n = 2$, $r = 2$ checks: $((1, 0, 2, 0), (1, 1, 3, 0))$.

		r_1	r_2
c_1	0	1	1
c_2	0	0	0
c_3	0	0	1

$$s_1 = a\ b\ \ \ a\ b\ a\ bb \quad a\ b\ a\ b\ a\ bba\ b \quad bba\ b$$
$$s_2 = a\ b\ a\ b\ a\ bba\ b \ (\dots)\ a\ b\ a\ b\ a\ b\ a\ b \ (\dots)\ b$$
$$s_3 = a\ b\ a\ b\ a\ b\ a\ b \quad a\ \ \ b\ a\ b\ a\ bb \quad b\ a\ bb$$
$$t = a^3\, b^5\, a^3\, b^5\, a^3\, b^4\, a^3\, b^5 \quad a^3\, b^5\, a^3\, b^5\, a^3\, b^4\, a^3\, b^5 \quad b^4\, a^3\, b^5\, a^3\, b^5$$

(Binary) Shuffle Product (cont.)

Example: $k = 3,\ n = 2,\ r = 2$ checks: $((1, 0, 2, 0),\ (1, 1, 3, 0))$.

		r_1	r_2
c_1	0	1	1
c_2	0	0	0
c_3	0	0	1

$$
\begin{array}{rcccccccccccccccccccccc}
s_1 = & a & b & & & a & \boxed{1} & a & bb & & a & b & a & b & a & \boxed{1} & a & b & & bb & a & b & & \\
s_2 = & a & b & a & b & a & \boxed{0} & a & b & (\ldots) & a & b & a & b & a & \boxed{0} & a & b & (\ldots) & b & & & & \\
s_3 = & a & b & a & b & a & \boxed{0} & a & b & & a & & & b & a & \boxed{1} & a & bb & & b & a & bb & & \\
t = & a^3 & b^5 & a^3 & b^5 & a^3 & b^4 & a^3 & b^5 & & a^3 & b^5 & a^3 & b^5 & a^3 & b^4 & a^3 & b^5 & & b^4 & a^3 & b^5 & a^3 & b^5
\end{array}
$$

Implication to scheduling problems

(BINARY) SHUFFLE PRODUCT

↓ [van Bevern et al., 2016]

Scheduling with parallel identical processors

↓ [Mallem et al., 2026]

Scheduling with minimum precedence delays

- Number of source words $\leftrightarrow$ width of precedence graph.
- Alphabet size $\leftrightarrow$ number of processors $\leftrightarrow$ max. delay value.

Outline

1. Motivation
   - Scheduling with precedence delays
   - The XNLP class
2. Contribution
   - Summary
   - Shuffle Product
   - Directed Bandwidth
3. Discussion
4. References

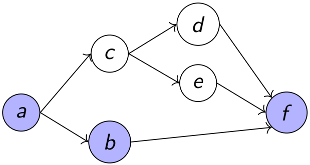
Directed Bandwidth

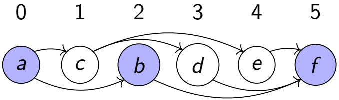
DIRECTED BANDWIDTH

Given: a directed acyclic graph $G = (V, A)$.

Parameter: target value B.

Question: is there $f : V \rightarrow \{0, \cdots, |V| - 1\}$ a bijective mapping such that, for each arc (i, j) in A, $f(j) - f(i) \leq B$?





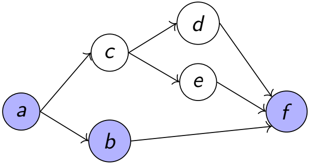
Directed Bandwidth

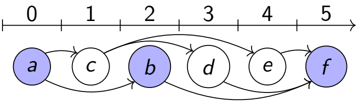
DIRECTED BANDWIDTH

Given: a directed acyclic graph $G = (V, A)$.

Parameter: target value B.

Question: is there $f : V \to \{0, \cdots, |V| - 1\}$ a bijective mapping such that, for each arc (i, j) in A, $f(j) - f(i) \leq B$?





- $\Leftrightarrow$ single processor scheduling of unit-time tasks with equal-length **maximum precedence delays**.

Reduction from NNCCM Acceptance: directed acyclic graphs

Given: the description of a NNCCM:

- number of counters k, max. counter value n, number of checks r,
- a sequence of r checks in $([1,k] \times [0,n] \times [1,k] \times [0,n])^r$.

Parameter: number of counters k.

Key gadgets

- Floor path (length L).
  - Delay values forced to their maximum.
  - Other vertices come in "batches".
- For each of the k counters: a path of length $L + n$.
  - Two vertices in the same batch $\leftrightarrow$ counter increment.
- Check gadgets along the floor path and counter paths.
  - "Critical" batches at key steps of the floor path.
  - Cannot have two counter check gadgets in a critical batch.

(Binary) Shuffle Product (recall)

Example: $k = 3,\ n = 2,\ r = 2$ checks: $((1, 0, 2, 0),\ (1, 1, 3, 0))$.

$$
\begin{array}{c|c|c|c|}
 & & r_1 & r_2 \\
\hline
c_1 & 0 & 1 & 1 \\
\hline
c_2 & 0 & 0 & 0 \\
\hline
c_3 & 0 & 0 & 1 \\
\hline
\end{array}
$$

$$
\begin{array}{rcccccccccccccccccccccccc}
s_1 = & a & b & & & a & \boxed{b} & a & bb & & a & b & a & b & a & \boxed{bb} & a & b & & bb & a & b & & \\
s_2 = & a & b & a & b & a & \boxed{bb} & a & b & (\dots) & a & b & a & b & a & \boxed{b} & a & b & (\dots) & b & & & & \\
s_3 = & a & b & a & b & a & \boxed{b} & a & b & & a & & & b & a & \boxed{b} & a & bb & & b & a & bb & & \\
t = & a^3 & b^5 & a^3 & b^5 & a^3 & \boxed{b^4} & a^3 & b^5 & & a^3 & b^5 & a^3 & b^5 & a^3 & \boxed{b^4} & a^3 & b^5 & & b^4 & a^3 & b^5 & a^3 & b^5
\end{array}
$$

(Binary) Shuffle Product (recall)

Example: $k = 3,\ n = 2,\ r = 2$ checks: $((1, 0, 2, 0),\ (1, 1, 3, 0))$.

		r_1	r_2
c_1	0	1	1
c_2	0	0	0
c_3	0	0	1

$$s_1 = a\ b\ \ a\ \boxed{1}\ a\ bb \qquad a\ b\ a\ b\ a\ \boxed{1}\ a\ b \qquad bb\,a\ b$$
$$s_2 = a\ b\ a\ b\ a\ \boxed{0}\ a\ b \ (\ldots)\ a\ b\ a\ b\ a\ \boxed{0}\ a\ b \ (\ldots)\ b$$
$$s_3 = a\ b\ a\ b\ a\ \boxed{0}\ a\ b \qquad a\ \ b\ a\ \boxed{1}\ a\ bb \qquad b\ a\ bb$$
$$t = a^3\, b^5\, a^3\, b^5\, a^3\, b^4\, a^3\, b^5 \qquad a^3\, b^5\, a^3\, b^5\, a^3\, b^4\, a^3\, b^5 \qquad b^4\, a^3\, b^5\, a^3\, b^5$$

Reduction from NNCCM Acceptance: directed trees

Given: the description of a NNCCM:

- number of counters k, max. counter value n, number of checks r,
- a sequence of r checks in $([1, k] \times [0, n] \times [1, k] \times [0, n])^r$.

Parameter: number of counters k.

Key ideas

- Start from previous construction.
- Adapt counter gadgets.
  - Width becomes unbounded.
- Deal with added flexibility.
  - "Tail" gadgets to fix the order of path-ending vertices.

Outline

Discussion: scheduling with precedence delays

Problem	*Parameter*	*Result*
(BINARY) SHUFFLE PRODUCT	#source words	XNLP-complete
DIRECTED BANDWIDTH	target value plus width	XNLP-complete for directed acyclic graphs
	target value	XNLP-complete for directed trees

Takeaway

- Managing a restaurant is (essentially) XNLP-complete.
- Scheduling as a complexity hub.

Open questions

- UNARY BIN PACKING with a bounded number of bins.
  - ↔ scheduling unit-time tasks with equal-length **exact delays**.
  - W[1]-hard [Jansen et al., 2013], in XP [Garey and Johnson, 1979].
- Minimum delays, constant max. delay value.
  - ↔ no delays, constant #processors (open since [Ullman, 1975]).
  - XNLP-hard [Bodlaender et al., 2022].

Discussion: scheduling with precedence delays

Problem	*Parameter*	*Result*
(BINARY) SHUFFLE PRODUCT	#source words	XNLP-complete
DIRECTED BANDWIDTH	target value plus width	XNLP-complete for directed acyclic graphs
	target value	XNLP-complete for directed trees

Takeaway

- Managing a restaurant is (essentially) XNLP-complete.
- Scheduling as a complexity hub.

Open questions

- UNARY BIN PACKING with a bounded number of bins.
  - $\leftrightarrow$ scheduling unit-time tasks with equal-length **exact delays**.
  - W[1]-hard [Jansen et al., 2013], in XP [Garey and Johnson, 1979].
- Minimum delays, constant max. delay value.
  - $\leftrightarrow$ no delays, constant #processors (open since [Ullman, 1975]).
  - XNLP-hard [Bodlaender et al., 2022].

References I

Bodlaender, H. L. (2021).
Parameterized complexity of bandwidth of caterpillars and weighted path emulation.
In *Graph-Theoretic Concepts in Computer Science: 47th International Workshop, WG 2021, Warsaw, Poland, June 23–25, 2021, Revised Selected Papers 47*, pages 15–27. Springer.

Bodlaender, H. L., Groenland, C., Nederlof, J., and Swennenhuis, C. M. (2022).
Parameterized problems complete for nondeterministic FPT time and logarithmic space.
In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 193–204. IEEE.

Elberfeld, M., Stockhusen, C., and Tantau, T. (2015).
On the space and circuit complexity of parameterized problems: classes and completeness.
Algorithmica, 71:661–701.

Garey, M. R. and Johnson, D. S. (1979).
Computers and Intractability, volume 174.
Freeman San Francisco.

References II

Jansen, K., Kratsch, S., Marx, D., and Schlotter, I. (2013).
Bin packing with fixed number of bins revisited.
Journal of Computer and System Sciences, 79(1):39–49.

Mallem, M., Hanen, C., and Munier-Kordon, A. (2026).
Single machine scheduling with precedence constraints and bounded maximum delay value.
Journal of Combinatorial Optimization, 51(3):33.

Ullman, J. D. (1975).
NP-complete scheduling problems.
Journal of Computer and System sciences.

van Bevern, R., Bredereck, R., Bulteau, L., Komusiewicz, C., Talmon, N., and Woeginger, G. J. (2016).
Precedence-constrained scheduling problems parameterized by partial order width.
In Kochetov, Y., Khachay, M., Beresnev, V., Nurminski, E., and Pardalos, P., editors, *Discrete Optimization and Operations Research*, pages 105–120, Cham. Springer International Publishing.