THREE-MACHINE UNIT-JOB SCHEDULING IN TIME $O((L+2)^{276})$

DAVID NIEDBALA GIRAUDIN

Abstract. OpenAI recently gave a deterministic polynomial-time algorithm for $P3 \mid \mathrm{prec}, p_j = 1 \mid C_{\max}$ (problem OPEN8 of Garey and Johnson), with running time $O((L+2)^{150020})$ on a multitape Turing machine, L being the input length; the same exponent appears in the accompanying Lean statement. The exponent comes from enumerating all Boolean formulas with at most 10000 leaves, each leaf choosing its own triple of jobs. We show that every description actually used by the structural proof involves at most 21 distinct job triples and at most 3 distinct rank thresholds. Restricting the enumerated family accordingly, and using the length bound on accepting paths already proved there to limit the number of reachability passes, the same algorithm runs in time $O((L+2)^{276})$.

hexagon:2611.00060v1

1. Statement

The problem $P3 \mid \mathrm{prec}, p_j = 1 \mid C_{\max}$ is OPEN8 in [1]. We use the notation and numbering of [2]. The input is an explicitly listed directed acyclic graph on n unit jobs and a deadline T; after padding there are $N = 3T \le 3n$ jobs, and $L \ge n$ is the binary input length. The algorithm of [2, §3] is a dynamic program over a family $\mathcal{F}$ of subsets of J described by Boolean formulas with at most $K = 10000$ leaves, the leaves being: constants; rank thresholds $\lambda \le k$, $\lambda \ge k$ $(0 \le k \le N+1)$ in either frame; and membership in a triple of jobs or in one of its four cones $P, D, \widehat{P}, \widehat{D}$, in either frame. Counting formula instances leaf by leaf gives $|\mathcal{F}| = O(N^{3K})$ and the running time $O_K(N^{15K+13})$, whence $O((L+2)^{150020})$ [2, Lemma 2.3 and §9].

Theorem 1. *The minimum-makespan schedule and the deadline decision of* [2, Theorem 1.1] *can be computed in $O((L+2)^{276})$ steps on a deterministic multitape Turing machine.*

The algorithm is the one of [2], run with a smaller family $\mathcal{F}^* \subseteq \mathcal{F}$ (Definition 5) and with fewer reachability passes. Correctness is not re-proved: soundness of the dynamic program holds for every family [2, proof of Proposition 3.1], and completeness only requires that the node sets and split sets of a bounded-description hierarchy [2, Definition 3.2, Lemma 3.3] belong to the family. We check this in Section 2.

2. Distinct triples and thresholds in the required descriptions

For a formula, let τ be the number of distinct job triples appearing in its cone or membership leaves (a triple is counted once, whatever the cones and frames used), and θ the number of distinct threshold leaves (a threshold leaf is a frame, a direction and a value). Sentinel tests are constants and are not counted. An intersection of two rank suffixes $\{\lambda \ge r\} \cap \{\lambda \ge r'\}$ in the same frame is the single suffix $\{\lambda \ge \max(r, r')\}$, as noted in [2, §6]; we always use this merged form.

Lemma 2 (only the first entry is trimmed)**.** *In every list and every carry of the construction of* [2, §§5–6]*, every entry other than the first one carries no outer rank trim.*

Proof. The root list (J) and a new switch pair $(G \cap \{\lambda \ge h\}, D_a)$ [2, (6.4)] have this property. A carry [2, (6.1)] trims only K_d, which becomes its first entry, and discards $K_1, \dots, K_{d-1}$; the

Date: 3 November 2026.

appended injection $D_a \cap (J \setminus P_b)$ [2, (6.2)] is untrimmed. Pruning only deletes entries; if it deletes the first entry, the entry moving to the first place is untrimmed. A switch discards the old list entirely. Repeated trims of the first entry merge into one. □

We bound (τ, θ) for the entries. Let U be the parent that produced the node W where a switch occurs, and use the current frame, in which $U = (a_-, b_-)$ with centre v_- (see [2, §6, before (6.6)]). The right boundary of U in the old frame is the triple a_-, and its predecessor cone there is D_{a_-} in the current frame. In the old frame (the tag frame of W) we write $U = (a_U, b_U)$, so that $b_U = a_-$ and $a_U = b_-$ as triples. "Switch case" and "same case" below refer to the edge $U \to W$, compared with the tag of U, as in [2, Lemma 6.6].

(1) *Root base* J: $(0, 0)$, plus a trim if first.
(2) *Injection* $D_a \cap (J \setminus P_b)$: $(2, 0)$, plus a trim if first.
(3) *Second switch entry* D_a: one triple, which is also used in G.
(4) *The extension* H^*_- *of* [2, Lemma 6.6]. In the switch case it is $(J \setminus D_{a_U}) \cap \{\lambda \le r\}$: $(1, 1)$. In the same case it is P_{b_U} minus the union of a carry of the list of U, where the possibly complicated first switch entry is replaced by the flattened pair $D_{a_0} \cup \{\lambda \ge k\}$ of [2, Lemma 6.5]. That carry has at most five entries, at most three of them injections [2, Lemma 6.4], and at most one trim (Lemma 2); the initial part contributes the root base, or one triple a_0 (flattened pair or second switch entry). Since P_{b_U} is the cone of a_-, H^*_- uses a_- and at most $1 + 6 = 7$ further triples, and at most one threshold.
(5) *First switch entry* $G \cap \{\lambda \ge h\}$ with $G = D_{a_-} \cap (J \setminus \widehat{P}_a) \cap (J \setminus \widehat{P}_{v_-}) \cap H^*_-$ [2, (6.7)]: triples a_-, a, v_- and at most 7 more, so $\tau \le 10$; thresholds: one inside H^*_- and the outer one, so $\theta \le 2$.

By [2, Lemma 6.4 and the remark following it] a pruned list, or a carry taken from it, has at most two initial entries and at most three injections. Hence, using Lemma 2:

Lemma 3. *The union of all entries of a pruned list or of a carry uses* $\tau \le 16$ *and* $\theta \le 2$.

Proof. In a run started by a switch: 10 triples for the first switch entry, none new for the second, 2 for each of at most three injections; thresholds: at most two, both in the first entry. In the root run: at most 6 triples and one threshold. □

Proposition 4. *Every global node description of* [2, Lemma 8.1] *can be written with* $\tau \le 21$, $\theta \le 3$, *and every relative split predicate of* [2, Lemma 8.2] *with* $\tau \le 19$, $\theta \le 2$*; all these formulas have fewer than* 500 *leaves.*

Proof. Same edge. The description is the union of [2, (8.4)] and [2, (8.5)] with the cutoff predicate $\widetilde{S} = P_b \setminus \mathrm{Qual}(\mathbf{K}_{\mathrm{carry}})$ [2, (8.2)]. The new list $\mathbf{K}'$ is the pruned concatenation of $\mathbf{K}_{\mathrm{carry}}$ and the injection on (a, b). The triples are those of $\mathbf{K}_{\mathrm{carry}}$ (at most 16, Lemma 3), the parent boundaries a, b, and a', b', v: at most 21. The thresholds are those of $\mathbf{K}_{\mathrm{carry}}$, reused in $\mathbf{K}'$: at most 2.

Switch edge. Here $\mathbf{K}'$ is the pruned switch pair and $\widetilde{S} = (J \setminus D_a) \cap \{\lambda \le r\}$ [2, (8.3)]. Triples: those of G (at most 10, including a) and a', b', b, v: at most 14. Thresholds: two in the first switch entry and r: at most 3.

Relative splits. The centre predicate $B_v(H_*)$, with $B_z(H) = P_z \cup (H \setminus \widehat{D}_z)$ [2, (4.1)] and $H_* \in \{P_b, J \setminus D_a\}$, uses two triples. An internal split $B_v(H_*) \cap B_z(\widetilde{S})$ [2, (8.6)] uses v, z, the boundary in H_* and the triples of $\widetilde{S}$: at most $3 + 16 = 19$ triples and 2 thresholds in the same case, and 3 triples and one threshold in the switch case. The conversion [2, (8.7)] adds a membership leaf for the triple z, already counted.

Merging thresholds only decreases the leaf counts of [2, Lemma 8.3], which are below 500. No other predicate is used: the boundary invariants of [2, §7] are proved about these lists and add no description. □

3. The restricted family and the running time

Definition 5. Let $\mathcal{F}^*$ be the family of subsets of J described by formulas in the atomic language of [2, §2.3] with at most 500 leaves, at most 21 distinct triples and at most 3 distinct threshold leaves.

By Proposition 4, every node set and every relative split set of the bounded-description hierarchy of [2, Theorem 3.4] belongs to $\mathcal{F}^*$, and $J \in \mathcal{F}^*$ (formula "true"). Hence [2, Lemma 3.3] holds with $\mathcal{F}^*$ in place of $\mathcal{F}$.

Lemma 6. *$|\mathcal{F}^*| \le C\,N^{66}$ for a constant C, and $\mathcal{F}^*$ can be listed deterministically in polynomial time.*

Proof. Choose a sequence of 21 triples or blanks ($\le (N^3+1)^{21}$ choices), a sequence of three threshold leaves or blanks ($\le (4(N+2)+1)^3$ choices) and a formula template with at most 500 leaves over the resulting fixed finite set of atoms (a number of templates independent of N). Since $N \ge 3$, $(N^3+1)^{21}(4N+9)^3 \le C' N^{63} N^3$. Evaluation and deduplication proceed as in [2, Lemma 2.3]. □

Proof of Theorem 1. We follow the accounting of [2, §9] verbatim, with $B = CN^{66}$ in place of $C_K N^{3K}$. There are at most B sets and at most $S = BN^3$ states per set, and one ordered state-pair test with its reachability update costs $O(BN^4)$ steps. The only change concerns the number of passes. Along every transition $|L|$ increases by at least three [2, proof of Proposition 3.1], so every path in the state graph of a set has at most $N/3$ edges. After p passes over all ordered pairs, every state reachable from an initial state by a path of at most p edges is marked. Hence $\lceil N/3 \rceil + 1$ passes mark every reachable state, instead of S passes. For all sets together the cost is

$$O\big(B \cdot N \cdot S^2 \cdot BN^4\big) = O(B^4 N^{11}) = O(N^{4\cdot 66+11}) = O(N^{275}).$$

Generating and deduplicating $\mathcal{F}^*$ costs $O(B^2 N^4) = O(N^{136})$; reachability preprocessing is polynomial of small degree, and state construction and table initialisation cost at most $O(B \cdot S \cdot BN^4) = O(N^{3\cdot 66+7})$, all far below N^{275}. With $N \le 3n \le 3L$ (the vertices are explicitly listed, so $L \ge n$), the deadline decision costs $O((L+2)^{275})$, and the $n \le L$ deadline trials for makespan minimisation give $O((L+2)^{276})$. □

Remark 7. Since feasibility is monotone in the deadline, a binary search over T reduces the makespan computation to $O(\log L)$ deadline trials, giving $O((L+2)^{275} \log(L+2))$.

Remark 8. The improvement does not alter the structural proof of [2]: it uses its descriptions as they stand, and only counts their parameters instead of their leaves. The accounting of the dynamic program itself is the conservative scan-based one of [2, §9]; a finer implementation of the table lookups would lower the exponent further, which we do not pursue.

Acknowledgement. Technical assistance: Claude (Anthropic).

References

[1] M. R. Garey and D. S. Johnson, *Computers and Intractability*, Freeman, 1979.

[2] OpenAI, *A polynomial-time algorithm for three-machine unit-job scheduling*, OpenAI Math Release preprint, September 24, 2026; Lean statement `ComparatorChallenges/ThreeMachine.lean` in the same release.

ORCID: 0009-0009-1526-1178