Fast bounds in Hexaly based on single-machine scheduling problems

Léa Petit-Jean Genat
Hexaly, 251 Boulevard Pereire, Paris, France
e-mail: lpetitjean@hexaly.com

1 Introduction

Given a scheduling problem expressed in the Hexaly framework, our goal is to *quickly* calculate a reasonable lower bound on the objective at the beginning of the solution process, supplementing more computationally expensive bounds computed during the search. By *quickly*, we mean within a very short computation time relative to the overall time allocated for solving the problem. Typically under one second for problems involving several thousand activities. To achieve this, we propose using classical algorithms with a complexity of at most $O(n\log(n))$ on single-machine subproblems.

2 Notation et considered polynomial algorithms

Let $\mathcal{A}$ represent the set of interval variables (activities) in the model defined for Hexaly. For $x \in \mathcal{A}$, $C(x)$ denotes the end time of x and $d(x)$ its minimum duration. Let $\mathcal{P}(x) \subset \mathcal{A}$ denote the set of intervals that must end before $C(x)$ due to precedence constraints, and $\mathcal{M} \subset 2^{\mathcal{A}}$ the set of disjunctive constraints of the problem (machines).

The algorithms considered are Moore-Hodgson [2] and Potts-Wassenhove [1] for problems $(1||U_i)$ and $(1||w_iU_i)$, Smith [3] for $(1||\sum w_iC_i)$, and Jackson [4] for $(1||L_{\max})$. These algorithms have a complexity of $O(n\log(n))$ and yield optimal solutions for their specific problem, except for Potts-Wassenhove's, which is a relaxation.

In the following sections, we illustrate our work using Smith's algorithm. The other algorithms are approached in a similar manner.

3 Example of Smith's algorithm

To recall, Smith's algorithm for solving the problem $(1||\sum w_iC_i)$ involves ordering tasks by increasing d_i/w_i ratios.

All model expressions that can be represented as a weighted sum of interval variable end times are identified. Let $r = \sum w_iC(x_i)$ be one such expression. We compute a lower bound for r as described bellow. It is worth noting that this approach is not limited to expressions r that participate in the objective function; any expression of this form is subject to a lower bound calculation that propagates throughout the rest of the problem.

Given a machine $m \in \mathcal{M}$, we can calculate a bound on $r = \sum w_i C(x_i)$ using Smith's algorithm by distributing the weights w_i of intervals x_i in r across the intervals on m. For example, consider a model expression $r = w_3 C(x_3) + w_4 C(x_4) + w_5 C(x_5)$ and a machine $m = \{x_1, x_2, x_3\}$. Suppose $\{x_1, x_2, x_3\} \subset \mathcal{P}(x_4)$ and $\{x_1, x_2\} \subset \mathcal{P}(x_5)$. We can select weights to use in Smith's algorithm on machine m that yield a valid bound on r, for instance, by evenly distributing the w_i based on precedences: $\{\frac{w_4}{3} + \frac{w_5}{2}, \frac{w_4}{3} + \frac{w_5}{2}, w_3 + \frac{w_4}{3}\}$. Other distributions can also produce valid bounds, with the highest of these bounds being retained.

4 Results

The approach was extended to other algorithms and objective types mentioned above and tested on classic benchmarks (e.g., job shop), either used directly or modified according to the objective. In the table below, the *Gap* columns indicate the average gap between the lower and best upper bounds, before and after the process described here.

Objective	#	Improvement	Gap Before	Gap After	Difference
U_i	640	84%	66%	48%	-18%
$w_i U_i$	760	77%	63%	54%	-9%
$w_i C_i$	652	95%	55%	24%	-31%
$L_{\max}$	700	69%	-	-	-35%

Table 1: Average Lower Bound Improvement by Objective Type.

5 Conclusions and Future Work

The identification of single-machine subproblems suitable for polynomial algorithms helps to improve bounds from Hexaly 12.5. Further research is needed to examine different ways to distribute the weights w_i (and, depending on the objectives, due dates) of expressions r among the machine intervals to achieve even better bounds.

References

[1] C. Potts et L. Van Wassenhove. Algorithms for scheduling a single machine to minimize the weighted number of late jobs. *Management Science 1988.*

[2] J. M. Moore. An n job, one machine sequencing algorithm for minimizing the number of late jobs. *Management Science 1968.*

[3] W. Smith. Various s for single-stage production. *Naval Research Logistics Quarterly 1956.*

[4] J.R. Jackson. Scheduling a production line to minimize maximum tardiness. *Office of Technical Services 1955.*