IEOR 8100 Scheduling Algorithms
Fall 2016 Lecture 9

Instructor: Clifford Stein

Scriber: Mingxian Zhong

Oct. 7, 2016

1 α-Point Algorithm

1.1 Review and Overview

We are discussing the problem $1/r_i/\sum C_i$. Specifically, the problem of scheduling jobs with release date r_i on a single machine subject to minimizing the sum of completion time, $\sum C_i$. Note that this objective is equivalent to minimize the average completion time and the preemption is not allowed.

The problem is easy if we allow preemption, since we can simply schedule the jobs with shortest remaining process time first. In the last lecture, we first discussed an algorithm which use the ordering of jobs by their completion time from optimal solution in preemptive case to create a non-preemptive solution and proved that it provides a 2-approximation. For $0 < \alpha \leq 1$, we denoted the α-point of a job j with processing time p_j in a schedule by its first time at which αp_j of the job has complete. Then we considered a generalization of this algorithm by using the ordering of the α *points* instead of the completion time[1]. In last lecture, we showed that this algorithm can only reach $1 + \dfrac{1}{\alpha} \geq 2$ in the worst case. However we will show that this algorithm can actually reach a $\dfrac{e}{e-1} \approx 1.58$-approximation by choosing the value of α wisely, which is the best we can hope in the algorithms that starts with the optimal preemptive schedule[3]. In the remainder of this lecture, we will discuss a $PTAS$[2] of our problem.

1.2 Analysis of the α-Point Algorithm

Here is an outline of the α-point algorithm[1]:

- Pick $\alpha \in (0, 1]$ from some p.d.f. $f(\alpha)$;
- Find the optimal solution of our problem with the allowing of preemption;
- Non-preemptively schedule the jobs in order of their α-points in the scheduling of preemptive solution.

We begin with some notations. We use C_i^p and C_i^α to denote the completion time of job j in the preemptive solution and the solution of the α-point algorithm respectively. In the preemptive solution, we use T_i to denote the idle time before C_i^p and $S_i(\beta)$ to denote the set of jobs that complete exactly a β fraction before C_i^p. By definition $C_i^p = T_i + \sum_{0<\beta\leq 1} \beta p(S_i(\beta))$. Here we use $p(X)$ to denote the sum of processing times of jobs in set X. Note that there are only finitely many value such that $\beta \neq 0$, so we could use the summation here.

Hence by the definition of α-point we have,

$$
\begin{aligned}
C_i^\alpha &\leq T_i + \sum_{\beta<\alpha} \beta p(S_i(\beta)) + (1+\alpha) \sum_{\beta\geq\alpha} p(S_i(\beta)) \\
E[C_i^\alpha] = \int_0^1 f(\alpha) C_i^\alpha d\alpha &\leq \int_0^1 f(\alpha)[T_i + \sum_{\beta<\alpha} \beta p(S_i(\beta)) + (1+\alpha) \sum_{\beta\geq\alpha} p(S_i(\beta))] d\alpha \\
&= T_i + \int_0^1 f(\alpha) \sum_{\beta<\alpha} \beta p(S_i(\beta)) d\alpha + \int_0^1 f(\alpha)(1+\alpha) \sum_{\beta\geq\alpha} p(S_i(\beta)) d\alpha \\
&= T_i + \int_\beta^1 f(\alpha) \sum_{0<\beta\leq 1} \beta p(S_i(\beta)) d\alpha + \int_0^\beta f(\alpha)(1+\alpha) \sum_{0<\beta\leq 1} p(S_i(\beta)) d\alpha \\
&= T_i + \sum_{0<\beta\leq 1} \beta p(S_i(\beta)) [\int_\beta^1 f(\alpha) d\alpha + \int_0^\beta \frac{f(\alpha)(1+\alpha)}{\beta} d\alpha] \\
&\leq C_i^p \max_{0<\beta\leq 1} [\int_\beta^1 f(\alpha) d\alpha + \int_0^\beta \frac{f(\alpha)(1+\alpha)}{\beta} d\alpha] \\
&= C_i^p (1 + \max_{0<\beta\leq 1} \int_0^\beta f(\alpha) \frac{1+\alpha-\beta}{\beta} d\alpha)
\end{aligned}
$$

So now the approximation ratio only depends on $\max_{0<\beta\leq 1} \int_0^\beta f(\alpha) \frac{1+\alpha-\beta}{\beta} d\alpha$, which depends on the p.d.f. we use. One example is by setting $f(\alpha) = 1$, then $\max_{0<\beta\leq 1} \int_0^\beta f(\alpha) \frac{1+\alpha-\beta}{\beta} d\alpha = \max_{0<\beta\leq 1} (1 - \beta + \beta/2) = 1$. This means that if we uniformly choose the value of α, then we will reach a 2-approximation, which is the same approximation ratio as setting α to be a constant 1.

The optimal choice of $f(\alpha)$ is by setting $f(\alpha) = \frac{e^\alpha}{e-1}$. Then we have

$$
\begin{aligned}
\max_{0<\beta\leq 1} \int_0^\beta f(\alpha) \frac{1+\alpha-\beta}{\beta} d\alpha &= \max_{0<\beta\leq 1} \int_0^\beta \frac{e^\alpha}{e-1} \frac{1+\alpha-\beta}{\beta} d\alpha \\
&= \frac{1}{e-1}
\end{aligned}
$$

This implies that $E[C_i^\alpha] \leq C_i^p(1 + \frac{1}{e-1})$ and moreover $E[\sum C_i^\alpha] \leq \sum C_i^p(1 + \frac{1}{e-1})$, which means that our algorithm is a $1 + \frac{1}{e-1} = \frac{e}{e-1} \approx 1.58$-approximation.

1.3 Observations

1. In [3], Torng and Uthaisombut proved that no ordering rules that starts with the optimal preemptive schedule provides an approximation ratio better than $\frac{e}{e-1}$.

2. We can always find a deterministic algorithms which provides a $\frac{e}{e-1}$-approximation ratio because by the definition of the expectation we know that for any input there exists an α such that $\sum C_i^{\alpha} \leq \sum C_i^{p}(1+\frac{1}{e-1})$. Note that the preemptive schedule switches jobs only when a new job released or a job finished. Hence there are at most $2n$ combinatorially distinct α (which means that the different α that actually imply to different orders). We can simply try all of them.

2 PTAS

In this section we will describe a PTAS[2] for the problem $1/r_j/\sum C_j$. The rough idea of this algorithm is that starting from the input, we will do several different steps $T_1, T_2, \ldots, T_k$, where k is a constant, such that each T_i increase op by at most $1+\epsilon$ factor. Since $(1+\epsilon)^k \approx 1+\epsilon k$, the algorithm is still a PTAS. Now we will briefly describe the steps and leave the rest to the next lecture.

Define $R_x = (1+\epsilon)^x$ and $I_x = [R_x, R_{x+1}]$. Note that now $|I_x| = R_{x+1} - R_x = (1+\epsilon)^x\epsilon = \epsilon R_x$.

1. Round r_j, p_j to the power of $(1+\epsilon)$ for each job j.

2. Charge the end of the interval as the completion time.

3. Set $r_j = \max(r_j, \epsilon p_j)$.

4. We claim each job runs at most $S = log_{1+\epsilon}(1+\frac{1}{\epsilon})$ intervals:
   $p_i \leq \frac{1}{\epsilon} r_i = \frac{1}{\epsilon} R_x = \frac{1}{\epsilon^2}|I_x|$(Let x be such that $R_x = r_j$)
   and the size of the next S interval is $\sum_{j=0}^{S-1} |I_{x+j}| = \sum_{j=0}^{S-1} |I_x|(1+\epsilon)^j = \frac{1}{\epsilon^2}|I_x|$

References

[1] C. Chekuri, R. Motwani, B. Natarajan, and C. Stein. Approximation techniques for average completion time scheduling. In Proceedings of the Eighth Annual ACM-SIAM Symposium on Discrete Algorithms (SODA), pages 609-618, 1997.

[2] F. Afrati, E. Bampis, C. Chekuri, D. Karger, C. Kenyon, S. Khanna, I. Milis, M. Queyranne, M. Skutella, C. Stein, and M. Sviridenko. Approximation schemes for minimizing average weighted completion time with release dates. In Proceedings of the 40th Annual IEEE Symposium on Foundations of Computer Science, pages 32-43, 1999.

[3] E. Torng and P. Uthaisombut. Lower bounds for SRPT subsequence algorithms for non-preemptive scheduling. In Proceedings of the 10th ACM-SIAM Symposium on Discrete Algorithms, pages 973-974, 1999.