The analysis of algorithm for three machines scheduling with general eligibility

임 경 국, 박 종 호, 장 수 영*
포항공과대학교 산업경영공학부
(*syc@postech.ac.kr)

Abstract

Online parallel machine scheduling problems have been studied by many researchers and enormous results are appeared in the last 40 years. With the development of scheduling theory and application, new online scheduling problems where the partial information is known in advance, that is, semi-online, gained much interest due to their increased application in practice. So we consider the online scheduling of three machines with general eligibility and its semi-online variant where the total processing time is known in advance. For the online and semi-online problems, we develop algorithms with competitive ratio of 5/2 which are shown to be optimal.

1. Introduction

스케줄링 문제는 작업(job) 정보의 유무, 작업의 선점(preemption) 허용 유무, 작업의 재배치 가능성과 주어진 기계들의 동급 유무 등에 따라 다양하게 분류된다. 이 논문은 주어진 작업들을 세 대의 동급 기계들에 가장 효율적으로 배치하여 스케줄의 완료 시간(makespan)을 최소화하는 것을 목표로 하였다.

작업의 스케줄을 정할 때 작업의 정보가 없는 상태에서 일정을 정하고 한번 정해진 일정에 대해서 수정이 불가능한 경우를 온라인(on-line)이라 한다. 이와 대조적으로 작업 정보가 미리 알려져 있으며 한번 정해진 일정에 대해서 수정이 가능한 경우를 오프라인(off-line)이라 한다. 그러나 실제 상황에서는 작업 정보의 일부분이 주어진 경우가 많은데 이런 경우를 세미 온라인(semi-online)이라고 한다. 이런 세미 온라인은 활용할 수 있는 정보에 따라 몇 가지로 분류할 수 있다[2]. 이 중에서 우리는 총 작업 시간이 알려진 경우와 가장 긴 작업 시간이 알려진 경우에 대해서 연구하였다.

스케줄링 문제가 NP-complete라 증명이 된 이후로 많은 논문을 통해 근사 알고리즘(approximation algorithm)이 개발되어 사용되었었다. 1985년 Sleator와 Tarjan은 근사 알고리즘들의 성능 평가(performance of algorithms)를 위한 경쟁 비율(competitive ratio)을 제안하였다[4]. 알고리즘 A로 생성된 작업 완료 시간을 $A(I)$이라 하고 오프라인 알고리즘으로 생성된 최적의 작업 완료 시간을 $OPT(I)$이라 하자. 이 두 비율인 $\sigma(A) := A(I)/OPT(I)$의 상한(maximum)을 알고리즘 A의 경쟁 비율이라 하고 $\sigma(A) \leq c$일 경우에는 알고리즘 A가 c-competitive를 가진다고 말한다. 특히 알고리즘 A가 c-competitive를 갖고 있을 때, c 를 그 문제의 하한선(low bound)이라 부른다. 더욱이 하한선이 c인 문제에 대해서 알고리즘 A가 c-경쟁 비율을 갖는다면 그 알고리즘을 최적(optimal)이라고 부른다.

처음 온라인 스케줄링에 대한 분석은 Y.Azar에 의해 분석이 되었다. 그는 m 대의 기계에 대한 온라인 스케줄링 문제을 분석하기 위해서 이분 그래프(bipartite graph)를 이용하였고 그 경쟁 비율이 $\lceil log_2 m \rceil + 1$임이 밝

혔다[3]. 우리 문제를 그의 결과에 대입하면 경쟁 비율은 3이 됨을 알 수 있다. 하지만 이번 연구에 의하면 세 대의 기계의 경쟁 비율은 5/2임을 알 수 있었다.

2장에서는 경쟁 비율의 하한선이 5/2임을 밝히고, 3장에서는 근사 알고리즘 GE를 제안한 후 그것이 최적임을 보이겠다. 마지막 4장에서는 우리 연구 결과를 요약 정리하겠다.

2. Lower bound

이 논문에서는 기계에 배치되는 작업들은 들어오는 순서에 따르고 먼저 들어온 작업이 끝나야 스케줄이 될 수 있다. 또한 스케줄링에 사용될 작업의 총 개수는 모르지만 편의상 작업의 지수 집합(index set)를 $J=\{1,2,\cdots,n\}$이라 하자. 각 작업 j에 대해서 작업 시간(processing time)을 p_j 라 하고 작업 가능한 기계들의 지수 집합을 자격(eligibility) $e_j \subseteq \{1,2,3\}$이라 하자. 만약 자격 $e_j=\{k,l\}$인 작업이 있다면 그 작업은 k 번째 기계와 l 번째 기계에서 작업 가능한 것이다. 물론, 작업 j가 들어오기 전까지는 작업 시간 p_j와 자격 e_j은 알 수 없다.

S_i를 i 번째 기계에서 작업하는 작업 지수들의 집합이라 하고, J의 임의의 부분집합 S에 대해서 $t(S)=\sum_{j\in S} p_j$ 라 정의한다. 마지막으로 z^A 를 알고리즘 A에 의한 완료 시간, z^* 를 최적의 오프라인 알고리즘으로 생성된 작업 완료 시간이라고 하면 z^A/z^*는 알고리즘 A의 경쟁 비율이 된다.

이제 우리 문제에 대한 경쟁 비율의 하한선이 5/2임을 보이도록 하겠다.

Theorem 1. 스케줄링 문제에 대한 임의의 알고리즘 A의 하한선은 5/2이다.

proof. 첫 번째 작업의 작업 시간을 $p_1=1$, 작업의 자격을 $e_1=\{1,2,3\}$이라고 하자. 첫 번째 작업이 첫 번째 기계에 배치되었다면 작업 시간이 1이고 자격 $e_2=\{2,3\}$인 두 번째 작업을 생성한다. 그 작업이 첫 번째 기계에 배치되면 작업 시간이 2이고 자격이 $e_3=\{1,2\}$인 세 번째 작업을 생성한다. 세 번째 작업이 첫 번째 작업에 배치되었다면 작업 시간이 2이고 자격 $e_4=\{1\}$인 네 번째 작업을 생성하면 경쟁 비율이 5/2을 만들 수 있다. 만약 세 번째 작업이 두 번째 기계에 배치된 경우에는 작업 시간이 2이고 자격이 $e_4=\{2\}$인 네 번째 작업을 생성하면 경쟁 비율이 5/2을 만들 수 있다.
만약 두 번째 작업이 세 번째 기계에 배치되었다고 가정하면 네 번째 작업을 생성하는데 그 작업 시간을 2이고 자격 $e_3=\{1,3\}$이라 하자. 세 번째 작업이 첫 번째 기계에 배치되었다면 작업 시간이 2이며 자격이 $e_4=\{1\}$인 네 번째 작업을 생성하여 경쟁 비율이 5/2을 만들 수 있다. 만약 세 번째 작업이 세 번째 기계에 배치가 되었을 경우는 작업 시간이 2이고 자격이 $e_4=\{3\}$인 네 번째 작업이 생성하여 경쟁 비율을 5/2로 만들 수 있다.

만약 첫 번째 작업이 두 번째 기계에 배치되었다고 가정하면, 작업 시간이 1이고 자격이 $e_2=\{1,3\}$인 두 번째 작업을 생성한다. 두 번째 작업이 두 번째 기계에 배치되었다면 첫 번째 작업과 두 번째 작업을 각각 첫 번째 기계와 두 번째 기계에 배치한 것과 같다. 이제 두 번째 작업이 세 번째 기계에 배치되었다고 가정하자. 이 때, 작업의 작업 시간은 2이며 자격이 $e_3=\{2,3\}$인 세 번째 작업을 생성한다. 이 세 번째 작업이 두 번째 기계에 배치되었다면 작업 시간이 2이고 자격이 $e_4=\{2\}$인 네 번째 작업을 생성한다면 이 때의 경쟁 비율은 5/2이가 된다. 만약 세 번째 작업이 세 번째 기계에 배치된 경우라면 작업 시간이 2이고 자격이 $e_4=\{3\}$인 네 번째 작업을 생성하여 경쟁 비율이 5/2가 된다.

만약 첫 번째 작업이 세 번째 기계에 배치되었다고 가정하자. 그리고 두 번째 작업은 작업 시간이 1이고 자격이 $e_2=\{1,2\}$인 작업이라 가정하자. 두 번째 작업이 첫 번째 기계에 배치되었다면, 첫 번째 작업과 두 번째 작

업이 각각 첫 번째 기계와 세 번째 기계에 배치한 경우와 같다. 다음으로 두 번째 작업이 두 번째 기계에 배치되었다고 가정하자. 이번에는 첫 번째 작업과 두 번째 작업이 각각 두 번째 기계와 세 번째 기계에 배치한 경우와 같다. □

위의 증명을 보면 모든 경우의 총 작업 시간이 6이고 가장 긴 작업 시간이 2이다. 이 theorem 1은 총 작업 시간이 알려진 세미 온라인 문제나 가장 긴 작업 시간이 알려진 세미 온라인에 적용할 수 있다.

3. Upper bound

우리가 제안하는 알고리즘 GE는 [그림 1]과 같이 LS(Least Schedule)를 바탕으로 하고 있다. 즉 여러 대의 기계에서 작업 가능한 작업이 들어오면 이 알고리즘에 따라 작업을 빨리 끝내도록 하는 기계에 배치하게 된다.

Algorithm GE

Let $S_1 := \emptyset$, $S_2 := \emptyset$ and $S_3 := \emptyset$.

REPEAT UNTIL(no more job arrives)

Receive job j with p_j and e_j;

IF ($|e_j| = 1$) $S_i := S_i \bigcup \{j\}$

IF ($|e_j| = 2$)

  IF ($t(S_i) := min\{t(S_i), t(S_k)\}$) $S_i := S_i \bigcup \{j\}$

  ELSE $S_k := S_k \bigcup \{j\}$

IF ($|e_j| = 3$)

  IF ($t(S_1) := min\{t(S_1), t(S_2), t(S_3)\}$) $S_1 := S_1 \bigcup \{j\}$

  ELSE IF ($t(S_2) := min\{t(S_1), t(S_2), t(S_3)\}$) $S_2 := S_2 \bigcup \{j\}$

  ELSE $S_3 := S_3 \bigcup \{j\}$

[그림 1] 세 대의 기계에 대한 알고리즘 GE

이 알고리즘을 분석하기 위해서 S_i^j는 j번째 작업을 배치한 후, i 번째 기계에서 작업하는 작업 지수의 집합이라 하자. 그리고 초기 $S_i^0 := \varnothing$이라 하자. 단, i 는 1, 2, 3 중에 하나이다.

Theorem 2.

$$\frac{z^{GE}}{z^*} \le \frac{5}{2}$$

proof. 이 theorem이 틀렸다고 가정하자. 그러면 theorem에 대한 반례(counter example)가 존재한다. 즉

$$z^{GE} = \max(S_1^n, S_2^n, S_3^n) > \frac{5}{2} z^*$$

$$z^{GE} = \max(S_1^{n-1}, S_2^{n-1}, S_3^{n-1}) \le \frac{5}{2} z^*$$

이다.

이제 마지막 작업 n에 대해서 다음 세 가지 경우를 가정하자.

[경우 1] $|e_n| = 3$

마지막 작업 n이 S_i에 배치되었다고 가정하면 $t(S_i^n) > \frac{5}{2} z^*$이 성립한다. $p_n \le z^*$인 관계에 의해 $t(S_i^{n-1}) > \frac{3}{2} z^*$를 얻게 된다. 그리고 algorithm GE에 의하여

$\frac{3}{2} z^* < t(S_i^{n-1}) \le t(S_k^{n-1})$와

$\frac{3}{2} z^* < t(S_i^{n-1}) \le t(S_l^{n-1})$이라는 사실을 얻을 수 있다. 이는

$\frac{9}{2} z^* < t(S_i^{n-1}) + t(S_k^{n-1}) + t(S_l^{n-1})$이 되어 모순을 얻게 된다.

[경우 2] $|e_n| = 2$

마지막 작업 n이 S_i에 배치되었다고 가정하면 $t(S_i^{n-1}) > \frac{3}{2} z^*$이 성립한다. 그리고 algorithm GE에 의하여

$\frac{3}{2} z^* < t(S_i^{n-1}) \le t(S_k^{n-1})$의 관계를 얻을 수 있다. 이는

$3z^* < t(S_i^{n-1}) + t(S_k^{n-1}) + t(S_l^{n-1})$이 되어 모순을 얻게 된다.

[경우 3] $|e_n| = 1$

마지막 작업 n이 S_i에 배치되었다면

$t(S_i^n) > \frac{5}{2} z^*$인 관계를 얻을 수 있다. 그리고 이 것은 작업 자격이 i 만 가진 작업의 총 작업 시간은 z^* 보다 작거나 같아야 하므로 S_i에는 다른 기계에 배치될 수 있는 작업들이 존재한다는 말이 된다. 이러한 작업들 중에서 S_i에 가장 늦게 배치된 작업의 지수를 α라고 하자. 그러면 $t(S_i^{\alpha}) > \frac{3}{2} z^*$를 얻게되고, $p_\alpha \le z^*$인 관계에 의하여 $t(S_i^{\alpha-1}) > \frac{1}{2} z^*$ 라 할 수 있다. 그리고

$t(S_k^n) \ge t(S_k^{\alpha-1}) \ge t(S_i^{\alpha-1}) > \frac{1}{2} z^*$

임을 알 수 있다. 그러나 이는

$t(S_i^n) + t(S_k^n) + t(S_l^n) > 3z^*$라는 말이 되어 모순이 된다. □

알고리즘 GE와 theorem 2도 총 작업 시간이 알려진 세미 온라인 문제나 가장 긴 작업 시간이 알려진 세미 온라인에 적용 가능할 수 있다. 그래서 우리가 고려하는 온라인 문제와 세미 온라인 문제의 최적 경쟁 비율은 5/2이라는 것도 알 수 있다.

4. Conclusion

이번 연구를 통해서 일반적인 자격(general eligibility)을 가진 경우의 온라인 스케쥴링 문제와 총 작업 시간의 합 정보나 가장 긴 작업 시간의 정보가 주어진 세미 온라인 스케쥴링을 살펴보았다. 놀랍게도 이 모든 것은 같은 알고리즘과 theorem들로 설명 가능했으며 그 최적 경쟁 비율이 5/2임을 알게 되었다.

논문에는 나타내지 않았지만 Azar의 논문에 의하면 4대의 기계에 대한 최적 경쟁 비율이 3이라는 것을 알 수 있었다. 그래서 우리는 별도로 이것을 검증해 보았고 그 최적 경쟁 비율이 3이 되는 것까지 확인해 보았다.

5. References

[1] J. Park, S. Chang, K. Lee, Online and semi-online scheduling of two machines under a grade of service provision, Oper. Res. Lett., Vol.34, (2006), 692-696.

[2] Y. He and G.C. Zhang, Semi-online scheduling on two identical machines, Computing, Vol. 62, (1999), 179-187.

[3] Y. Azar, J. Naor, R. Rom, The competitiveness of online assignments, J. Algorithms, Vol. 18, (1995), 221-237.

[4] D.D. Sleator and R.E. Tarjan, Amortized efficiency of list update and paging rules, Communications of the ACM 28, Vol. 2, (1985), 202-208.