Scheduling on uniform nonsimultaneous parallel machines

Liliana Grigoriu[(a)1], Donald K. Friesen[(b)]

(a)*Department of Control and Computers, Politehnica Universiy Bucharest, Splaiul Independentei 313, Bucharest, 060042, Romania*
(b)*Department of Computer Science, Texas A&M University, College Station, TX, 77843-3112, USA*

Abstract

We consider the problem of scheduling on uniform processors which may not start processing at the same time with the purpose of minimizing the maximum completion time. We give a variant of the Multifit algorithm which generates schedules which end within 1.382 times the optimal maximum completion time for the general problem, and within $\sqrt{6}/2$ times the optimal maximum completion time for problem instances with at most two processors. This results from properties of a variant of the Multifit algorithm for scheduling on uniform processors with simultaneous start times. We also show that if a better approximation bound of the Multifit variant for scheduling on uniform processors will be found in the future, this bound will also apply to our Multifit variant for scheduling on nonsimultaneous uniform processors.

Key words: multiprocessor scheduling, uniform processors, Multifit, nonsimultaneous starttimes, worst-case bounds, makespan

---

1. Introduction

We consider the problem of non-preemptively scheduling a given set of tasks on m uniformly related processors with nonsimultaneous machine available times in order to minimize the maximum completion time. With other

[1]Corresponding author; email addresses: liliana.grigoriu@gmail.com, friesen@cse.tamu.edu

words, the machines process the tasks at different speeds, and they may start processing at at different times, that is, non-simultaneously. After starting to process the machines are available as much as necessary.

This problem is strongly NP-hard since it is a generalization of the multiprocessor scheduling problem. For scheduling on parallel machines in order to minimize the maximum completion time, the algorithms LPT of Graham [5] and Multifit of Coffman, Garey and Johnson [6] are among the most studied.

For same-speed processors which do not start simultaneously, Lee [8] and Chang and Hwang [2] give worst-case analyses for scheduling on non-simultaneous parallel machines when using LPT and Multifit respectively. There, it was shown that schedules produced by a modified version of LPT (MLPT) and by Multifit are bounded by 4/3 and respectively 9/7 (about 1.286) times the optimal maximum completion time. In [9], Wang provided an improvement of the bound from [2] to 1.275.

For uniform processors that start simultaneously, worst-case approximation bounds of 1.4 and respectively 1.382 for a Multifit variant were obtained in Friesen and Langston [4] and Chen [3]. For two uniform processors, Burkard and He [1] derive a tight worst-case bound of $\sqrt{6}/2 + (1/2)^k$ (about $1.2247 + (1/2)^k$) for scheduling using Multifit with k calls of FFD within Multifit. When Multifit is combined with LPT as an incumbent algorithm, they show that the worst case bound decreases to $(\sqrt{2}+1)/2+(1/2)^k$ (about $1.2071 + (1/2)^k$).

Approximation for scheduling on uniform non-simultaneous machines was previously considered in Yong [10], where the performance of LPT was shown to within 5/3 times the optimal schedule's maximum completion time, and that the bound is better when there are only two machines.

In this paper we give a variant of Multifit for scheduling on uniform non-simultaneous parallel machines, and show that its worst-case approximation factor is the same as that of the Multifit variant considered in [4] and [3], namely 1.382.Also, when our Multifit variant is used for two uniform nonsimultaneous parallel machines the bound of $\sqrt{6}/2$ applies as a consequence of the result from [1].

2. A Multifit Variant

We consider the problem of scheduling on uniform processors with non-simultaneous machine available times. In the following we describe a variant of the Multifit algorithm for this problem which approximates the maximum completion time of the optimal schedule with an approximation factor of 1.382 and prove this bound.

We assume that the worst-case approximation bound for the maximum completion time of Multifit for the problem of scheduling on simultaneous uniform parallel machines as it was described in [4] and [3] is q. We show that our Multifit variant for scheduling on uniform nonsimultaneous parallel machines also has a worst-case approximation bound of q when minimizing the maximum completion time. We also assume that Multifit for scheduling on uniform processors first orders the time slots available for processing in nondecreasing order of their capacity to fit jobs, or equivalently, orders the processors on which they are in nondecreasing order of their speed factors, as was done in [4] and [3]. We next define a problem instance and more formally describe the Multifit algorithm for scheduling on nonsimultaneous uniform processors.

Definition 2.1 (Problem Instance)
A problem instance is given by a tuple $(P, T, \alpha : P \to \mathbb{Q}, \gamma : P \to \mathbb{N}, l : T \to \mathbb{N})$, where $\mathbb{N}$ represents the set of natural numbers and $\mathbb{Q}$ is the set of rational numbers. Here,

- P is a set of processors,
- T is a set of tasks,
- $l(X)$ denotes the length of a task X, that is, the time the task needs to execute on the slowest processor
- $\gamma(p)$ denotes the moment at which processor p is able to start processing tasks, its starting time,
- $\alpha(p)$ is the speed factor of the processor p, meaning that the time a task X takes to execute on p is $\frac{l(X)}{\alpha(p)}$.

We call *length* of a time slot the time that would be necessary on the slowest processor to process tasks that would fill that time slot, and use the term *duration* for the actual time passed from the beginning until the end of that time slot. For example, a time slot on a processor p starting at $\gamma(p)$ and ending at a deadline b assigned by Multifit has a duration $b - \gamma(p)$ and a length $\alpha(p)(b - \gamma(p))$.

A pseudocode for the Multifit variant we propose for uniform processors with nonsimultaneous machine available times is given as follows:

Multifit(ϵ, upper bound, lower bound)
(1) Order all tasks in nonincreasing order of their length ;
(2) Set deadline at $b = \frac{\text{upper bound + lower bound}}{2}$;
(3) Order all time slots ts_p with $p \in P$ in nondecreasing order of their length and initialize all time slot schedules as the empty list;
(4) Assign tasks in the given order to the first time slot in which they fit while considering the time slots in the determined order;
(5) If all tasks were successfully assigned
(5.1) decrease the upper bound (upper bound=b) and
(5.2) save the schedule;
Else increase the lower bound (lower bound =b);
(6) If $b-$ lower bound $\geq \epsilon$ loop back to step (2);
(7) return the saved schedule;

This algorithm builds a schedule the maximum completion time of which is within ϵ accuracy of the earliest time t^* with the property that FFD within the Multifit loop returns a feasible schedule for any deadline $t \geq t^*$. With other words, if Multifit would be allowed to run indefinitely, it would return t^*, unless it would stumble by accident upon a deadline $t < t^*$ for which a feasible schedule is obtained by FFD.

3. Approximation bound

Next, we prove a statement that is necessary for showing the main result of this work. Also, it holds only in case the upper and lower bound are not wired into the Multifit algorithm for uniform processors, i.e. they are given as input to the algorithm, as in the variant presented above and as was done in [3].

Lemma 3.1 (Property of approximation bound)
If the Multifit algorithm for simultaneous uniform processors (as described in [3]) has a worst-case approximation bound q, then for any problem instance PI (with simultaneous machine available times), FFD always returns a feasible schedule if the Multifit deadline is set at $b \geq q * opt$. Here *opt* denotes the end of an optimal schedule for PI.

Proof: Suppose this is not the case for a problem instance PI for a bound $b \geq q * opt$.

Let $FFD(b')$ be the schedule returned by Multifit when $2b$ and 0 are assigned as the initial upper and lower bounds respectively. Let b' be the deadline considered by Multifit when it constructed $FFD(b')$. Let $FFD(b)$ be the schedule constucted by FFD for PI when the Multifit deadline is set at b. By assumption, $FFD(b)$ does not contain all tasks, since FFD does not produce a feasible schedule when the deadline is b. Also, the search for deadlines is continued by Multifit between b and $2b$ after the deadline b is considered, thus $b' > b$.

Since PI is an instance with simultaneous machine available times, the length of the time slot on each processor is equal to the considered deadline times the speed factor of the processor, and thus FFD considers the time slots in nondecreasing order of the speed factors of the processors on which they are. Thus, when assigning tasks to processors, FFD considers these in an order which is the same for any deadline.

We next show that $FFD(b')$ ends after $q * opt$, thus contradicting the Therorem's hypothesis. Let X be the first task which is not scheduled on the same processor by FFD when the deadline is b' and when the deadline is b. There must be such a task since $FFD(b')$ contains all tasks while $FFD(b)$ does not. At the time at which X is assigned the two FFD schedules are thus identical. Let p' be the processor on which FFD schedules X when the deadline is b'. Such a processor must exist since $FFD(b')$ contains all tasks. When the deadline is b, FFD can not schedule X on any processor that is considered before p', since else X would also be assigned to that processor when the deadline is b'. When the deadline is b, FFD also does not assign X to p' (by definition of X), and thus X does not fit on p' when the deadline is b but does so when the deadline is b'. Thus the maximum completion time of $FFD(b')$ is greater than b, and thus also greater than $q * opt$. △

The rest of this work is mostly dedicated to proving the following Theorem.

Theorem 3.2 (Approximation bound)
We assume that for each instance of the problem of scheduling on simultaneous uniform processors any schedule generated by Multifit ends within $q * opt$, where opt is the maximum completion time of an optimal schedule. Then, for the problem of scheduling on nonsimultaneous uniform processors

the worst-case approximation bound of our Multifit variant for the maximum completion time is also q.

The statement of Theorem 3.2 is proved by contradiction. For this, we assume that there is a *counterexample*, that is, a problem instance and a Multifit deadline $b \geq q * opt$ for which FFD within the Multifit loop from our algorithm does not produce a feasible schedule. We define *minimal counterexample* to be a counterexample with a minimal number of processors.

Obviously, if there is a counterexample, there also is a minimal counterexample. Let $PI = (P, T, \alpha : P \to \mathbb{Q}, \gamma : P \to \mathbb{Q}, l : T \to \mathbb{N})$ be a minimal counterexample, and $b \geq q * opt$ be a deadline for which FFD within the Multifit loop does not generate a feasible schedule.

We next show that in a minimal counterexample the optimal schedule ends after all start times, a statement which we afterward use to prove that there is no minimal counterexample, and thereby show the statement of Theorem 3.2.

Lemma 3.3 **(**$opt > \max_{p \in P}(\gamma(p))$**)**
The optimal schedule of PI ends after all start times.

Proof: Suppose there is a processor p such that $\gamma(p) \geq opt$. In that case, the optimal schedule has no task on p. Removing p and any tasks FFD put on p when the deadline is b we obtain a lesser counterexample, which contradicts the assumption that PI is minimal. This is because after removing p and the mentioned tasks the optimal schedule stays the same or gets better whereas the FFD schedule stays unchanged. △

Now we are ready to proceed with the proof of Theorem 3.2.

Completion of proof of Theorem 3.2:
We next construct a new instance $PI' = (P, T, \beta, \gamma', l)$ of the problem of scheduling on uniform processors with simultaneous machine available times, such that the lengths of the time slots available for processing on processors, ts'_p, are the same as in PI when the deadline assigned by Multifit is b. This can be done by assigning for each processor p a new speed factor $\beta(p)$, such that $ts_p = ts'_p = \beta(p) * b$, and by setting $\gamma'(p) = 0$. Thus

$$\beta(p) = \frac{ts_p}{b} = \frac{\alpha(p)(b - \gamma(p))}{b}. \quad (1)$$

We will denote the maximum completion time of an optimal schedule for PI' with opt'.

For the new instance PI' we have

$$C_{max}(Multifit(PI')) \leq q * opt', \tag{2}$$

according to the bound for scheduling on uniform processors. Here, $C_{max}(Multifit(PI'))$ is the maximum completion time of the schedule produced by Multifit for PI'. We consider the schedule produced by FFD for the problem instance PI' when the Multifit deadline is set at b, $FFD(PI', b)$. For a time slot ts we denote with $FFD(ts)$ the FFD schedule produced by FFD in the time slot ts when the Multifit assigned deadline is b. We have $FFD(ts_p) = FFD(ts'_p)$, since for all $p \in P$ we have $ts'_p = ts_p$ and since Multifit orders the time slots in the same way in both cases. Then FFD assigns in each time slot ts'_p when considering problem instance PI' the same tasks it assigns in ts_p when considering the problem PI. Concluding, for problem instance PI' and Multifit deadline b, FFD also fails to successfully schedule all tasks. Therefore, $b < q * opt'$ by Lemma 3.1.

We next show that $opt' \leq opt$. We denote the set of tasks assigned by the optimal schedule OPT of PI to p with $OPT(ts_p)$. With opt^*_p we denote the end of the processing time of this set of tasks in ts'_p. We have:

$$opt^*_p = \frac{1}{\beta(p)} \sum_{X \in OPT(ts_p)} l(X) = \frac{\alpha(p)}{\beta(p)} \sum_{X \in OPT(ts_p)} \frac{l(X)}{\alpha(p)} \tag{3}$$

Let opt_p denote end of the optimal schedule OPT on p. Since $opt_p - \gamma(p)$ is greater than or equal to the duration of the time interval in which the tasks $X \in OPT(ts_p)$ were scheduled for problem instance PI, we have $\sum_{X \in OPT(ts_p)} \frac{l(X)}{\alpha(p)} \leq opt_p - \gamma(p)$. Together with (3) this implies

$$opt^*_p \leq \frac{\alpha(p)}{\beta(p)}(opt_p - \gamma(p)), \tag{4}$$

Because of (4) and (1) we have:

$$\frac{opt^*_p}{opt_p - \gamma(p)} \leq \frac{\alpha(p)}{\beta(p)} = \frac{b}{b - \gamma(p)}. \tag{5}$$

Next, we derive the inequality

$$\frac{opt_p}{opt_p - \gamma(p)} \geq \frac{b}{b - \gamma(p)}, \tag{6}$$

which together with (5) implies $opt^*_p \leq opt_p$ since by Lemma 3.3 $opt_p > \gamma(p)$ and since thus $b \geq q * opt > \gamma(p)$.

Inequality (6) can be derived as follows. Let $\epsilon = b - opt_p$. We have $\epsilon > 0$ by Lemma 3.3. Inequality (6) becomes $\frac{opt_p}{opt_p - \gamma(p)} \geq \frac{opt_p + \epsilon}{opt_p + \epsilon - \gamma(p)}$, which is equivalent to $opt_p(opt_p + \epsilon - \gamma(p)) \geq (opt_p - \gamma(p))(opt_p + \epsilon)$, and thus to $opt_p^2 + opt_p(\epsilon - \gamma(p)) \geq opt_p^2 - opt_p\gamma(p) + opt_p\epsilon - \gamma(p)\epsilon$ and to $0 \geq -\gamma(p)\epsilon$, which holds.

Let opt^* be the end of the schedule OPT when used as a solution for PI', i.e. $opt^* = \max_{p \in P}(opt^*_p)$. An optimal schedule for PI' must end at time opt^* or before that. For at least one processor p we have $opt^* = opt^*_p$. Then $opt' \leq opt^* = opt^*_p \leq opt_p \leq opt$, and thus $opt' \leq opt$, which together with $b < q * opt'$ implies that $b < q * opt$, contradiction. △

By Theorem 3.2 the described Multifit variant generates schedules which end within $q * opt$ when Multifit schedules for scheduling on simultaneous uniform processors always end within q times the end of the optimal schedule. In particular, according to the bound in [3], schedules produced by the presented Multifit variant end within 1.382 times the optimal maximum completion time when scheduling on uniform processors with nonsimultaneous machine available times.

The conclusion of Lemma 3.1 for the bound of $q = 1.382$ was also shown in [3], in the process of proving this bound for simultaneous uniform processors. In fact, proofs of Multifit bounds usually also show that a feasible schedule is returned by FFD for any deadline that is later than the bound to prove.

Even for same-speed processors, the statement similar to that of of Lemma 3.1, that if a feasible schedule is found for a bound b than for any bound $b' > b$ FFD also returns a feasible schedule, does not hold. This was shown in [6].

It is not possible to adapt our proof of Theorem 3.2 to show that the approximation bound of Multifit for scheduling on simultaneous same-speed processors also applies to scheduling on nonsimultaneous same-speed processors. This is mainly because the assumption that the approximation bound q applies to all instances of scheduling on simultaneous uniform processors is necessary even when considering instances with nonsimultaneous same-speed processors in order to derive equation (2) from our proof. In fact, Kellerer [7] gives a problem instance for which the approximation factor of Multifit for non-simultaneous same-speed processors is 24/19, which is greater than the approximation bound proved for simultaneous same-speed processors.

Next, we consider the situation when there are at most two processors in the considered problem instance. We show that a worst-case bound of $\sqrt{6}/2$ applies in this case.

Proposition 3.4 (Bound for instances with 2 processors)
For problem instances with at most 2 processors, our Multifit variant produces schedules which end within q times the maximum completion time of an optimal schedule, assuming that q is a worst-case approximation bound for scheduling on two simultaneous uniform processors for our Multifit variant.

Proof: Let PI be a problem instance for which the schedule generated by our Multifit variant ends later than q times the end of an optimal schedule, *opt*. If the optimal schedule only uses one processor, say p_1, Multifit will find this schedule, since any subset of tasks assigned to p_1 ends before or at the optimal maximum completion time *opt*, and thus, any schedule generated when the Multifit deadline is $b \geq opt$ contains all tasks. Thus, PI must have two processors, and *opt* is greater than the latest processing starttime.

The proof of Lemma 3.1 can also be used to show that if for the Multifit algorithm for instances with m simultaneous uniform processors a worst-case bound q applies, than for such problem instances for any deadline $b \geq q * opt$ $FFD(b)$ contains all tasks.

Having thus shown statements corresponding to those of Lemma 3.1 and of Lemma 3.3 for our bound q for all problem instances with at most 2 processors that may violate this bound, we can use the proof of Theorem 3.2 as it is given to show that this bound also holds for problem instances with non-simultaneous machine available times with at most 2 processors. From [1] we know that $q = \sqrt{6}/2$ if the Multifit loop is repeated enough times. △

4. Conclusion

We described a variant of the Multifit algorithm which applies to non-preemptive scheduling on uniform processors with nonsimultaneous machine available times. We have then shown that the worst-case approximation factor when minimizing the maximum completion time is 1.382 for the general problem, and $\sqrt{6}/2$ for problem instances with at most two processors. We also showed that in certain situations Multifit bounds for scheduling on simultaneous uniform machines also hold for our Multifit variant for scheduling on nonsimultaneous uniform machines. As a consequence, tightness results for Multifit scheduling on uniform processors can now be shown by using instances with nonsimultaneous uniform processors.

References

[1] R. E. Burkard and Y. He. A note on multifit scheduling for uniform machines. *Computing*, 61:277 – 283, March 1998.

[2] Soo Y. Chang and Hark-Chin Hwang. The worst-case analysis of the multifit algorithm for scheduling nonsimultaneous parallel machines. *Discrete Applied Mathematics*, 92:135 – 147, June 1999.

[3] B. Chen. Tighter bound for multifit scheduling on uniform processors. *Discrete Applied Mathematics*, 31:227–260, May 1991.

[4] D.K. Friesen and M.A. Langston. Bounds for multifit scheduling on uniform processors. *SIAM Journal on Computing*, 12:60 – 69, February 1983.

[5] R.L. Graham. Bounds on multiprocessing timing anomalies. *SIAM J. of Applied Mathematics*, 17:416 – 429, March 1969.

[6] E. G. Coffman Jr., M. R. Garey, and D.S.Johnson. An application of bin-packing to multiprocessor scheduling. *SIAM J. on Computing*, 7:1–17, February 1978.

[7] H. Kellerer. Algorithms for multiprocessor scheduling with machine release times. *IIE Transactions*, 30:991 – 999, 1998.

[8] C.Y. Lee. Parallel machine scheduling with nonsimultaneous machine available time. *Discrete Applied Mathematics*, 30:53 – 61, January 1991.

[9] H.M. Wang. The multifit algorithm for a class Pm||Cmax problem. *Applied Mathematics - A Journal of Chinese University*, 12:361 – 368, 1998.

[10] H. Yong. Uniform machine scheduling with machine available constraints. *Acta Mathematicae Applicatae Sinica (English Series)*, 16:122 – 129, April 2000.