Online scheduling with machine cost and a quadratic objective function

J. Csirik[1], Gy. Dósa[2], and D. Kószó[1]

[1] Department of Informatics, University of Szeged, Árpád tér 2, H-6720 Szeged, Hungary
[2] Department of Mathematics, University of Pannonia, Egyetem u. 10, H-8200 Veszprém, Hungary

Abstract. We will consider a quadratic variant of online scheduling with machine cost. Here, we have a sequence of independent jobs with positive sizes. Jobs come one by one and we have to assign them irrevocably to a machine without any knowledge about additional jobs that may follow later on. Owing to this, the algorithm has no machine at first. When a job arrives, we have the option to purchase a new machine and the cost of purchasing a machine is a fixed constant. In previous studies, the objective was to minimize the sum of the makespan and the cost of the purchased machines. Now, we minimize the sum of squares of loads of the machines and the cost paid to purchase them and we will prove that 4/3 is a general lower bound. After this, we will present a 4/3-competitive algorithm with a detailed competitive analysis.

Keywords: Scheduling · Online algorithms · Analysis of algorithms

1 Introduction

Online scheduling with machine cost is a kind of decision making. In some disciplines it plays a key role. Decision making requires allocating resources to activities which can appear in various forms. To make the best decision, we will optimize one or more performance measures.

In this paper, we will consider a quadratic variant. The model we use was first mentioned in [8]. Let us suppose we have a sequence of independent jobs. They come one by one and each of them has a positive size. We will assign them irrevocably to a machine without prior knowledge about other jobs that may come later on. We have no machine at first. Then, when a job arrives, we have the option to buy a new machine. The cost of purchasing a machine is a fixed constant. In previous studies, the objective was to minimize the sum of the makespan and the cost of purchased machines. Now, in our case, the objective is to minimize the sum of squares of loads of the machines and the total sum needed to purchase all the machines. Formally: Let A be an algorithm and J be an input. Then the *machines of* A *with respect to* J, denoted by $M_{A,J}$, is a linearly ordered set of machines, used by the algorithm to schedule input J. The *total cost of A on J* is defined by

$$A(J) = \sum_{m \in M_{A,J}} \mathrm{ld}(m)^2 + |M_{A,J}|,$$

where $|B|$ denotes the cardinality of a finite set B, i.e., the number of elements of B and $\mathrm{ld}(m)$ is the sum of job sizes of machine m.

In this way, we would like to achieve a uniform loading of the machines. The quadratic cost function was first introduced for single machine problems in [11] and [10].

We will evaluate the quality of an online algorithm by using a competitive analysis. Here, the standard is the *optimal offline algorithm*. In our case the the value of the optimum is well defined but the optimal solution is not unique so we may have different optimal solutions with different number of machines. Now, let us introduce some notations. We will denote an *online algorithm* by A, and one of the *optimal offline* by OPT. Let J be a sequence of jobs. Next, let $A(J)$ be *the total cost of an online algorithm A on a given sequence J*. Similarly, let $OPT(J)$ denote the *optimal offline cost*. We will call *an A online algorithm C-competitive* if $A(J) \leq C \cdot OPT(J)$ for all J.

In [8], it was proved that the competitiveness of each online algorithm is at least 4/3 with the original objective function. Moreover, a $(1 + \sqrt{5}/2) \approx 1.618$-competitive algorithm is given. In [1], an improved algorithm was presented with a competitive ratio of $(2\sqrt{6}+3)/5 \approx 1.5798$. In [4], it was shown that $\sqrt{2}-\varepsilon$ is a lower bound of the problem. A $(2 + \sqrt{7}/3) \approx 1.5486$-competitive algorithm was also introduced. In addition, it was shown that by applying the lower bounds on the optimal objective value introduced earlier, no algorithm can be proven to be C-competitive with any $C \leq 1.5$. Also, some other variants of the problem were studied in [2, 9, 6, 5]. In [7], the original model was extended with a more general machine cost function. In [3], another possible modification of the model was considered.

The structure of the paper is as follows. In Section 2, we will introduce the notations used in this study. In Section 3, we will present a general lower bound of 4/3. Lastly, in Section 4, we will give a 4/3-competitive algorithm and we will also prove its competitiveness.

2 Preliminaries

Here, we will use the following notations. We shall denote the i^{th} job by j_i and its size by p_i (also known as the processing time). And here when we speak about a job size, we will use size and processing time interchangeably.

Let q be $\sqrt{2}/2 \approx 0.707$. We will consider three different types of jobs. We will call a job *small* if $p_i \leq q$, *medium* if $q < p_i \leq 2q$, and *big* if $2q < p_i$. We will denote the total load by $P(= P(J)) = \sum_{j_i \in J} p_i$ and the total load of all small jobs by $P_s(= P_s(J)) = \sum_{p_i \text{ is small}} p_i$. Note that $P \geq P_s$.

In our algorithm, we will use two types of machines. The first type is called SM, which can receive only small and medium jobs, and its maximum possible load is $2q$. The second type is called B, which can process only big jobs (*big machines*). In the proof we further divide the SM machines into those that process only small jobs called *small machines* (S) and the remaining machines from SM are called *medium machines* (M).

3 Lower Bound

Proposition 1. *Consider two machines with loads $l_1 \geq l_2$. If we reschedule any job with size $x < l_2$ from the second machine to the first one, then the cost will grow.*

Proof. Evidently, $l_1^2 + l_2^2 < (l_1 + x)^2 + (l_2 - x)^2 = l_1^2 + l_2^2 + 2x(l_1 - l_2 + x)$, as $2x(l_1 - l_2 + x) > 0$ since $x > 0$ and $l_1 \geq l_2$. □

Proposition 2. *$2P$ is a lower bound of the cost of the optimal schedule.*

Proof. We will suppose that OPT can distribute P equally among the m machines. In this case $f(m) = m \cdot (P/m)^2 + m$ gives the total cost of OPT. This function has its minimum at $m = P$. So, if we replace m by P, then we will get the optimal value of $2P$. If we cannot distribute the loads equally on the machines, because of the previous proposition the cost will be larger. □

Lemma 1 *An online algorithm which never purchases a second machine is not constant competitive.*

Proof. We prove our statement by contradiction. Let A be a C-competitive online algorithm such that it uses one machine for each input I. Let J be an input having k jobs, each of size 1. Then the optimum will use one machine for each job and so it has a cost of $2k$. Algorithm A will use only one machine and so it will have a cost of $1 + k^2$. But then $A(J)/OPT(J) = k/2 + 1/(2k)$, which is larger than C if k is large enough. This leads to contradiction. □

Proposition 3. *Let J be a finite sequence of arbitrarily small ε jobs, having an even k number of jobs. Then OPT purchases at least two machines if $\sqrt{2} \leq P$.*

Proof. To prove our statement, we have to check whether the cost of having two machines is smaller than having only one. If k is even, this means that

$$2 \cdot (P/2)^2 + 2 \leq P^2 + 1, \tag{1}$$

which is exactly valid if $\sqrt{2} \leq P$. □

Theorem 2 *No online algorithm has a competitive ratio smaller than $4/3$.*

Proof. Let A be an online algorithm and J be a finite sequence of arbitrarily small ε jobs. The sequence terminates depending on the situation where A purchases the second machine. If at the moment of puchasing the second machine the number of jobs in J is even then we stop. If the number of jobs is odd at this moment then the input will get one more small job. The best possible algorithm A will schedule this last job to the second machine. (Clearly, A will purchase a second machine since A is constant competitive and because of Lemma 1.) We now have an even number of jobs in our input so by Proposition 3, OPT purchases at least two machines if $\sqrt{2} \leq P$. We will only describe in detail the case where the second machine of A has one small job - the other case can be handled similarly.

Thus, we consider the following two cases with respect to P.

1. If $P \leq \sqrt{2}$, then we have

$$\lim_{\varepsilon \to 0^+} \frac{A(J)}{OPT(J)} = \lim_{\varepsilon \to 0^+} \frac{(P-\varepsilon)^2 + \varepsilon^2 + 2}{P^2+1} = \frac{P^2+2}{P^2+1} \geq 4/3 \ .$$

2. If $\sqrt{2} < P$, then we have

$$\lim_{\varepsilon \to 0^+} \frac{A(J)}{OPT(J)} \geq \lim_{\varepsilon \to 0^+} \frac{(P-\varepsilon)^2 + \varepsilon^2 + 2}{2 \cdot (\frac{P}{2})^2 + 2} = \frac{2 \cdot P^2 + 4}{P^2+4} \geq 4/3 \ .$$

□

Lemma 1. *Let r, s, t_1 and t_2 be positive values with $4/3 < r/s$, $t_1 \leq (4/3)t_2 < r$ and $s > t_2$. Then $4/3 < (r-t_1)/(s-t_2)$.*

Proof. We will prove this by contradiction. Let us suppose that

$$(r-t_1)/(s-t_2) \leq 4/3.$$

Then from the conditions

$$(r-(4/3)t_2)/(s-t_2) \leq (r-t_1)/(s-t_2)$$

or equivalently $3r - 4t_2 \leq 4s - 4t_2$, i.e. $r/s \leq 4/3$, we arrive at a contradiction.

□

4 Algorithm

4.1 Description

We will denote the following algorithm by ALG. Algorithm ALG applies the bin packing algorithm First Fit (FF for short) as a slave algorithm. In the bin packing problem we are given items with positive sizes and unit capacity bins; and we would like to pack the items into as few bins as possible, but the bin capacity cannot be exceeded. Hence, in any bin the total size of items is at most one unit. The FF algorithm packs the items one by one, and the next item is always packed into the first bin it fits. If it does not fit into any bin, we open a new bin for it and pack the item into this new bin.

Algorithm ALG

1. If a small or a medium job arrives, then we will use an SM machine (its maximum possible load is $2q$). We will apply the FirstFit algorithm to decide which machine gets the job. (If there is no SM machine that has enough free space, then we will purchase a new one.)
2. If a big job comes, then we will schedule this to a B machine. We will not schedule any other job to this machine.

We will suppose that ALG uses a small, b medium and c big machines.

In the rest of the paper, we will use the following relaxed problem to estimate $OPT(J)$. We will permit preemption for every small job, but not for any medium or big job. Preemption means that the execution of a job can be divided into non-overlapping time slots, and these parts can be executed by different machines.

We will denote the optimal solution of the relaxed problem by $OPT_R(J)$ for every J and we will call it the *relaxed optimum*. We know that $OPT_R(J) \leq OPT(J)$.

Theorem 1. *$ALG(J)/OPT(J) \leq 4/3$ for every input J.*

Proof. It is enough to prove that $ALG(J)/OPT_R(J) \leq 4/3$ for every input J. First, let us suppose that the opposite is true. Take the case $ALG(J)/OPT_R(J) > 4/3$ for an input J. Input J may contain small, medium and big jobs. If input J contains a big job, and we can leave out one of the big jobs so that for the remaining J $ALG(J)/OPT_R(J) > 4/3$ is still valid, then we will leave out this job and we will repeat this step. It is possible that not all big jobs can be removed. Now, we may suppose that J is the minimal counterexample in the sense of not having a removable big job.

The general flow of our proof will be the following. In the next subsection we will prove some properties of the relaxed optimum and the ALG algorithm. Next, we will give some reduction steps which can be used to modify the minimal counterexample so that its structure is simpler, but it remains a counterexample. In the last subsection we will show that the final reduced example cannot be a counterexample and so the proof is completed.

4.2 Properties of the relaxed optimum and algorithm ALG

First we will prove some properties of the relaxed optimum.

Lemma 2. *Consider the (relaxed) optimal scheduling of a minimal counterexample J. In this case, there is no big job which uses a machine on its own.*

Proof. Suppose a big job X with size x uses a machine alone. This job is also alone in ALG. Let $J' = J \setminus \{X\}$. It follows from Lemma 1 that

$$\frac{ALG(J')}{OPT(J')} = \frac{ALG(J) - 1 - x^2}{OPT(J) - 1 - x^2} > 4/3,$$

which contradicts the fact that J is a minimal counterexample. □

Lemma 3. *Consider the (relaxed) optimal schedule of J. In this case, there is no machine whose jobs can be distributed into two sets and the load of each set is greater than q.*

Proof. Suppose the load of a machine can be distributed into sets S_1 and S_2 with loads $x_1, x_2 > q$. The total cost of this machine is $1 + (x_1 + x_2)^2$. If we schedule S_1 to a new machine and S_2 to a second new machine, then the total cost of the two machines is $2 + x_1^2 + x_2^2$, which is obviously less than $1 + (x_1 + x_2)^2$, and hence it is a contradiction. □

The following corollary is a consequence of Lemma 3.

Corollary 1. *In the case of a relaxed optimal schedule:*

(i) Any two big or medium jobs are scheduled to two different machines;
(ii) If a machine processes a big or a medium job, then the rest load is at most q, which can come from only small jobs;
(iii) If a machine processes only small jobs, then its total load is at most $2q$.

Proof. Item (i) and item (ii) are immediate consequences of Lemma 3. To prove item (iii) we can use the preemption possibilities of small jobs: if the total load is larger then $2q$ then we can split this up into two parts where each of them is larger than q and then we can use Lemma 3. □

Proposition 4. *In the case of a relaxed optimal schedule there are no two machines, each with a load less than q.*

Proof. Suppose we have two machines with loads $x_1, x_2 < q$. Then the cost of these two machines is $2 + x_1^2 + x_2^2$, which is obviously more than $1 + (x_1 + x_2)^2$ and hence there is a contradiction. □

Next we will prove some properties of *ALG*.

Lemma 4. *Consider any three small machines of ALG, each having at least two jobs. In this case, the total load of the three small machines is greater than or equal to $4q$ and at most one of the machines can have a load $< (4/3)q$.*

Proof. Consider the last machine of the three. We will suppose that this machine has a load of $< (4/3)q$, and let this load be $(4/3)q - 2x$ for some $0 < x < (2/3)q$. Then, the size of the smallest job is at most $(2/3)q - x$ on this machine. Next, the load of each of the first two machines is at least $(4/3)q + x$ since the smallest job of the third machine does not fit into the first two machines as we apply First Fit packing and the load of these machines cannot be larger then $2q$. So in this case the total load is bigger than $4q$ and only the third machine has a load of $< (4/3)q$.

After this we assume that the load of the third machine is at least $(4/3)q$.

Now we suppose that the load of the second machine is $< (4/3)q$. Let this load be $(4/3)q - 2x$ for some $0 < x < (2/3)q$. Then, the size of the smallest job is at most $(2/3)q - x$ on this machine and this does not fit into the first machine, so the first machine has a load of at least $(4/3)q + x$. The load of the third machine is at least $(4/3)q + 4x$, because there are two jobs and they do not fit into the second machine. So the total load of the three machines is $> 4q$ and only the second machine has a load of $< (4/3)q$.

After this we assume that the load of the second and the third machine is at least $(4/3)q$.

If the load of the first machine is less than $(4/3)q$, then let this load be $(4/3)q - 2x$ for some $0 < x < (2/3)q$. Then the load of the second and the third machine is at least $(4/3)q + 4x$, because there are two jobs and they do not fit into the second machine. So the total load of the three machines is $> 4q$ and only the first machine has a load of $< (4/3)q$. □

Lemma 5. *Take the schedule of ALG. If $a \geq 2$, then $P_s \geq 2q + (4q/3) \cdot (a-2)$.*

Proof. If $a = 2$, then the Lemma is clearly true as we start the second small machine when the load of the first small machine and the next small job together is larger than $2q$.

Let us suppose that $a = 3$. If any of the three small machines has at least two jobs, the assertion follows from Lemma 4. Suppose there is a machine with only one job. There can be only one such machine as the total load of any two small machines is bigger than $2q$, but the size of any small job is at most q. This is clearly the third (latest) machine. Now let us take the first two machines of three. If one of them has a load $\geq (4/3)q$, then we are done because the other and the third machine have altogether a load of $> 2q$. If the first two machines have a load $< (4/3)q$ then the second machine has a load of $< (4/3)q$, and let this load be $(4/3)q - 2x$ for some $0 < x < (2/3)q$. Then, the size of the smallest job is at most $(2/3)q - x$ on this machine. Next, the load of the first machine is at least $(4/3)q + x$ since the smallest job of the second machine does not fit into the first machine as we use First Fit packing and the schedule of these machines cannot be larger then $2q$. So the first machine has a load of $> (4/3)q$ and we are done.

Now we will suppose that $a \geq 4$. If every small machine has at least two jobs then the total load is at least $(4/3)q \cdot a$, which is more than what we need in the lemma. Otherwise there is a small machine with one job, but the load of any other small machine is greater than $(4/3)q$. Now the total load of the small machine with one job and any other small machine is bigger than $2q$, and the load of any other small machine is bigger than $(4/3)q$. Hence, we are done. □

4.3 Modifying the two schedules

We will suppose that $ALG(J)/OPT_R(J) > 4/3$ for a fixed minimal counterexample J. We will reduce and modify some jobs in J. We shall rename ALG to A_0 and OPT_R to O_0. Clearly, $A_0/O_0 > 4/3$. We also know that every machine has at most one medium job or a big job: in the relaxed optimal packing because of Corollary 1; and in the ALG because of the scheduling rule.

Now we will modify A_0 and O_0 in several simple steps. After each step their ratio will remain larger than 4/3.

Step 1: We apply this reduction step only if there is at least one big job, otherwise we go directly to Step 2. Therefore suppose there is a big job. We will reduce the size of every big job to $2q = \sqrt{2}$ in both A_0 and O_0. We will denote the new (reduced) schedules by A_1 and O_1. We know that $A_1 \leq A_0$ and $O_1 \leq O_0$. We also note that $A_0 - A_1 \leq O_0 - O_1$. To show this, let us consider a big job with size $2q+x$ with $x > 0$. In O_0, the machine processing this particular job has a load $2q+x+y$ with $y > 0$ because a big machine of O_0 contains small job(s)s as well. After reducing the job size the load of this particular machine will be $2q+y$. In A_1, the new load is $2q$ instead of $2q+x$. The difference of squares in A_1 is $4qx+x^2$. In O_1, $(2q+x+y)^2$ becomes $(2q+y)^2$ with difference $2(2q+y)x+x^2$. Then we get from Lemma 1 that $A_1/O_1 > 4/3$.

Step 2: We will decrease the total cost of O_1 by using $2P$ instead of the actual cost of O_1. Since $2P \leq O_1$ by Proposition 2, clearly $A_1/(2P) > 4/3$. Thus let $O_2 = 2P$. We do not change in this step A_1, hence we let $A_2 = A_1$.

Step 3: In A_2, we will reduce the load of every medium machine to q by decreasing the size of the medium job to exactly q and deleting the small job(s) here if they exist. We will call it A_3. Due to Lemma 1, $A_3/O_3 > 4/3$ where $O_3 = 2P$. To show this, let us consider a medium machine. Its total load is $(q+x)$ with $0 < x \leq q$. After the reduction, it is only q. So the difference in cost is $2qx + x^2$. $2P$ is then decreased by $2x$. To use Lemma 1 we need $2qx + x^2 \leq (4/3) \cdot 2x$, i.e. $2q + x \leq 8/3$, which is true, because $x \leq q$. After this step on all medium machines in A_3, we will have one medium size job of size q.

Step 4: Now, we will change the loads of small machines of A_3. Let the new schedule be A_4. Next, $A_4 \geq A_3$, but P is not changed. Let us consider two small machines of A_3. We will move some loads from one to the other, until one machine has a load $2q$ or the other one has a load of 0. We notice that here we can use a relaxation regarding small jobs, as it will only increase the cost of ALG. The total cost will then increase. Now, in A_4, except at most one, every small machine has a load of exactly $2q$ or 0. If we have the small machine with a load different from 0 and $2q$, then we will denote its load by x. Note that we keep the machines with load 0, because the cost of purchasing these machines is included in the total cost. According to the lower bound of Lemma 5,

- if $a = 3k+2$, then we have at least $2k+1$ machines with load of $2q$,

- if $a = 3k+3$, then we have at least $2k+1$ machines with load of $2q$. If there are exactly $2k+1$ machines with this load, then at least one further machine has load of at least $(4/3)q$,

- if $a = 3k+1, k \geq 1$, then we have at least $2k$ machines with load of $2q$. If there are exactly $2k$ machines with this load, then at least one other machine has a load of at least $(2/3)q$.

Step 5: In the last step, we will change the load of those small machines whose load is greater than the lower bound of Lemma 5. From A_4, we will get A_5 and from O_4 we will get O_5.

This means that

- if $a = 3k+2$ then we keep $2k+1$ machines with load of $2q$ and we set the size of all other jobs to zero;

- if $a = 3k+3$ then we keep $2k+1$ machines with load of $2q$ and one machine with load of q and we set all other sizes to zero;

- if $a = 3k+1, k \geq 1$ then we keep $2k$ machines with load of $2q$ and one machine with load of $(2/3)q$ and we set all other sizes to zero.

First we note that it may happen that more machines will have a full load of $2q$ than we need in the proof (whose total load is provided by Lemma 5). In this case we will delete the jobs of these small machines as follows.

- Some load of $0 \leq x \leq 2q$ is reduced to zero. In this case the cost of A_4 will decrease by x^2 and the lower bound of the optimal algorithm will decrease by $2x$.

- The load is $(4/3)q \leq x \leq 2q$ and it is decreased to q, or the load is $(2/3)q \leq x \leq 2q$ and it is decreased to $(2/3)q$.

It is easy to see that in both cases Lemma 1 can be applied and $A_5/O_5 > 4/3$ still holds.

4.4 Competitivness

So we reduced and modified our minimal counterexample. We proved that the input after Step 5 is still a counterexample. Now we will prove that it cannot be a counterexample.

After the reduction we have $a \geq 0$ small machines. Among them, there a_1 such machines where the load of a machine is exactly $2q$, and a_2 machines with load 0. Moreover, we have at most one additional small machine; and if it exists then its load is denoted by x ($2q \geq x \geq 0$), and according to the subcase above, here $x = q$ or $x = (2/3)q$.

We also know that in the A_5 schedule the load of the medium machines is exactly q, and the load of the big machines is exactly $2q$ (where $q = 1/\sqrt{2}$). To get the contradiction, we have to prove the following inequality:

$$a + b + c + a_1 \cdot (2q)^2 + x^2 + b \cdot q^2 + c \cdot (2q)^2 \leq \frac{4}{3} \cdot 2(a_1 \cdot 2q + x + b \cdot q + c \cdot 2q),$$

where on the right hand side we used the $2P$ lower bound of the optimum value from Proposition 2, which is actually the same as O_5. This inequality (using $q^2 = 1/2$) leads after simplification to

$$a + 2a_1 + x^2 + \frac{3}{2}(b + 2c) \leq \frac{4\sqrt{2}}{3}(2a_1) + \frac{8}{3} \cdot x + \frac{4\sqrt{2}}{3}(b + 2c). \qquad (2)$$

Here the coefficient of $(b + 2c)$ is $\frac{4\sqrt{2}}{3} \approx 1.8856$ on the right hand side, while it is (only) 1.5 on the left hand side. This means that if the inequality is valid for an input with certain $(b + 2c)$ value, then it is also valid for the modified input where the small machines are the same (after the reduction) but the value of $(b+2c)$ is bigger. Hence we will consider our main inequality (2) only if $b+2c \leq 1$. If the inequality is valid even for $b + 2c = 0$, then we are done (as it is also valid for bigger values of $b+2c$). Otherwise we will consider the case where $b+2c = 1$.

Now we will consider three cases according to the remainder of a divided by three. Several small cases will remain, and these remaining cases will be considered so that instead of the lower bound of the optimum value (i.e. $2P$) sometimes it will be easier to compare the objective value of the algorithm with the objective value of the optimum.

Case 1, $a = 3k + 2$; moreover $k \geq 1$ or $b + 2c \geq 1$. In this case according to the reduction Step 5 we get that $a_1 = 2k + 1$ so among the small machines there are $2k+1$ machines with load $2q$ and there are $k+1$ machines with 0 load (here $x = 0$). The total load of the small machines (after the reduction) is $(2k+1) \cdot 2q$. In the case $b = c = 0$, (2) looks like

$$3k + 2 + 2(2k + 1) \leq \frac{8\sqrt{2}}{3} \cdot (2k + 1)$$

$$7k + 4 \leq \frac{16\sqrt{2}}{3} k + \frac{8\sqrt{2}}{3}.$$

If $k = 1$, the inequality looks like $11 \leq 8\sqrt{2} \approx 11.314$, and since the coefficient of k on the right hand side is $\frac{16\sqrt{2}}{3} \approx 7.5425 > 7$, the inequality holds for any $k \geq 1$. If $k = 0$ then we may assume that $b + 2c = 1$, so (as we saw above) it suffices to show that $4 + 3/2 \leq \frac{8\sqrt{2}}{3} + \frac{4\sqrt{2}}{3} = 5.6569$. As this holds once again, we are done.

Case 2, $a = 3k + 3$, where $k \geq 0$. We suppose that $b = c = 0$. In this case after the reduction we have $a_1 = 2k + 1$ and $x = q$. Then (2) looks like the following:

$$(3k + 3) + 2(2k + 1) + \frac{1}{2} \leq \frac{4\sqrt{2}}{3}(4k + 2) + (4/3)(\sqrt{2}),$$

which is

$$7k + 5.5 \leq \frac{16\sqrt{2}}{3} k + 4\sqrt{2}$$

and for $k = 0$ it means $5.5 < 4\sqrt{2} = 5.656$, which is true.

Case 3, $a = 3k + 1$, where $k \geq 1$. We suppose that $b = c = 0$. After the reduction Step 5 we have $2k$ machines with load $2q$, one machine with load $\frac{2}{3}q$, and k machines with load 0. We need to show that

$$(3k + 1) + 2 \cdot (2k) + \frac{2}{9} \leq \frac{4\sqrt{2}}{3}(4k) + \frac{8}{9} \cdot \sqrt{2}$$

i.e.

$$7k + 11/9 \leq \frac{16\sqrt{2}}{3} k + \frac{8\sqrt{2}}{9},$$

where it is enough to examine the case $k = 1$. Here, the inequality looks like $8.222 \approx 7 + 11/9 \leq \frac{16\sqrt{2}}{3} + \frac{8\sqrt{2}}{9} = 8.799$, which is true.

At this point we have seen that the algorithm is at most (4/3)-competitive in the above cases. Below, we will continue with the cases not yet covered. First, let us see what cases have already been investigated, and what remain:

	covered	remain
Case 1:	$a = 3k + 2$; where $k \geq 1$ or $b + 2c > 0$	$k = 0$ and $b = c = 0$
Case 2:	$a = 3k + 3$, where $k \geq 0$.	-
Case 3:	$a = 3k + 1$, where $k \geq 1$.	$k = 0$

We realize that two cases remain. One possibility is that there are two small machines, and no other machine (first row in the table, Case R1 below). The only other possibility is that there is exactly one small machine, and there are possibly several medium and/or big machines (last row in the table, Case R2 below). If there is no medium and no big machine the schedule is optimal; so we can assume that there is also at least one machine which is not small.

Case R1, there are two small machines and no other machine. Let us consider the moment when the first job is assigned to the second (small) machine by the algorithm. At that moment the sum of the loads of the machines is bigger than $2q$. After applying the reduction let the loads of the two machines be $2q$ and x, respectively. Here $0 < x \leq 2q = \sqrt{2}$. The objective value of the algorithm is $2 + x^2 + (2q)^2 = 4 + x^2$.

Suppose that in the optimal solution the jobs are assigned to one machine. Then $OPT = 1 + (x + 2q)^2 \geq 1 + 2 + x^2 = 3 + x^2$, and we are done. If they are assigned to two machines then $OPT \geq 2 + \frac{(2q+x)^2}{2} = 2 + \frac{2+4qx+x^2}{2} = 3 + x^2/2 + 2qx \geq 3 + x^2$ and we are done again. If they are assigned to three machines, then $OPT \geq 3 + \frac{(2q+x)^2}{3}$, thus $\frac{4}{3}OPT \geq 4 + \frac{4}{9}(2+4qx+x^2) > 4 + \frac{4}{9}(2x \cdot x + x^2) > 4 + x^2$. It is easy to see that the optimal solution will not use four or more machines.

Case R2, $a = 1$ and $b + c > 0$. After simple calculation we get that our main inequality (2) is valid if $b + 2c \geq 3$. Thus it remains for us to consider the case where $b + 2c \leq 2$. Within this we will distinguish three subcases, and compare the objective value of the algorithm to the optimum value (instead of its lower bound) as follows.

Subcase R2.1. $a = 1$, $b = 0, c = 1$. Because the reduction, the size of the big job is $y = \sqrt{2}$. The total size of the small jobs is x for some $0 < x \leq 2q = \sqrt{2}$. The objective value of the algorithm is $2 + x^2 + y^2 = 4 + x^2$. Suppose that in the optimal solution the jobs are assigned to one machine. Then $OPT = 1 + (x + y)^2 \geq 1 + x^2 + y^2 = 3 + x^2$, and we are done. If they are assigned to two machines then the schedule of the algorithm is optimal.

Subcase R2.2. $a = 1$, $b = 2, c = 0$. In this case the reduction should be applied in a different way. Note that the total load of any two machines is more than $2q$ using the algorithmic rule. Hence let us perform the reduction so that we decrease the load of some machine, and at the same time increase the load of one other machine. During this time, the load of some machine will reach $2q$. After this we perform another reduction to make the loads of the two other machines as unbalanced as possible. The next two cases can happen after the reduction.

a, The loads are 0, $2q$ and $q + x$ with some $0 < x \leq q$.

b, The loads are x, $2q$ and $2q$ with some $0 < x \leq 2q$.

Note that in the optimal solution the two medium jobs are assigned to different machines. We make the calculations in these cases one by one.

Case a: The objective value of the algorithm is $3+2+x^2 = 5+x^2 \leq 5.5$. Let us see the optimal value. If the jobs are assigned to two machines then $OPT \geq 2 + \frac{(3q+x)^2}{2} = 2 + \frac{(9/2+6qx+x^2)}{2} = 17/4 + x^2/2 + 3qx$, thus $\frac{4}{3}OPT \geq 17/3 + \frac{2}{3}x^2 + 4qx > 5 + x^2$. If they are assigned to three machines, then $\frac{4}{3}OPT \geq \frac{4}{3}(3 + \frac{9/2}{3}) = 6$. Optimum will certainly not use four or more machines.

Case b: The objective value of the algorithm is $3+2+2+x^2 = 7+x^2 \leq 9$. Let us see the optimal value. If the jobs are assigned to two machines then $OPT \geq 2 + \frac{(4q+x)^2}{2} = 2 + \frac{(8+8qx+x^2)}{2} = 6 + x^2/2 + 4qx$, similarly as before, we are done. If they are assigned to three machines, then $\frac{4}{3}OPT \geq \frac{4}{3}(3 + \frac{8+8qx+x^2}{3}) = \frac{4}{9}x^2 + \frac{32}{9}qx + \frac{68}{9}$, we are done. If they are assigned to four machines, then $\frac{4}{3}OPT \geq \frac{4}{3}(4 + \frac{8+8qx+x^2}{4}) = \frac{1}{3}x^2 + \frac{8}{3}qx + 8$ which is enough. Optimum will certainly not use five or more machines.

Subcase R2.3. $a = 1$, $b = 1, c = 0$. In this case the reduction is similar to that of performed in case R1, namely we decrease the load of the small machine and increase the load of the medium machine. The load of the medium machine will grow to reach $2q$. Let the load of the small machine be x for some $0 < x \leq 2q = \sqrt{2}$. Then we have that the value of the objective is $4 + x^2$, and from this point the proof is the same.

References

1. Gy. Dósa and Y. He. Better online algorithms for scheduling with machine cost. *SIAM Journal on Computing*, 33(5):1035–1051, 2004.
2. Gy. Dósa and Y. He. Scheduling with machine cost and rejection. *Journal of Combinatorial Optimization*, 12(4):337–350, 2006.
3. Gy. Dósa and Cs. Imreh. The generalization of scheduling with machine cost. *Theoretical Computer Science*, 510:102–110, 2013.
4. Gy. Dósa and Z. Tan. New upper and lower bounds for online scheduling with machine cost. *Discrete Optimization*, 7(3):125–135, 2010.
5. S. Han, Y. Jiang, and J. Hu. Online algorithms for scheduling with machine activation cost on two uniform machines. *Journal of Zhejiang University-SCIENCE A*, 8(1):127–133, 2007.
6. Y. He and S. Cai. Semi-online scheduling with machine cost. *Journal of Computer Science and Technology*, 17(6):781–787, 2002.
7. Cs. Imreh. Online scheduling with general machine cost functions. *Discrete Applied Mathematics*, 157(9):2070–2077, 2009.
8. Cs. Imreh and J. Noga. *Scheduling with Machine Cost*, volume 1671 of *Lecture Notes in Computer Science*, pages 168–176. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.
9. J. Nagy-György and Cs. Imreh. Online scheduling with machine cost and rejection. *Discrete Applied Mathematics*, 155(18):2546–2554, 2007.
10. W. Szwarc and S. K. Mukhopadhyay. Minimizing a quadratic cost function of waiting times in single-machine scheduling. *Journal of the Operational Research Society*, 46(6):753–761, 1995.
11. W. Townsend. The single machine problem with quadratic penalty function of completion times: A branch-and-bound solution. *Management Science*, 24(5):530–534, 1978.