Here we study the role of the migration for the online restricted assignment scheduling problem. In this model the j-th job has a processing time p_j but we cannoit schedule it on every machine only on a subset of machines. In the case of the online problem the best known algorithm is Greedy which is $O(\log m)$-competitive, and it is proved that no $o(\log m)$-competitive algorithm exists. This lower bound can be extended easily to the migration model where in each step it is allowed to migrate jobs depending on the size of the aczual jobs. Therefore we consider the amortized model here, which means that an algorithm has a migration factor λ if in each step the total processing time of the migrated jobs is at most λ times the total processing time of the arrived jobs.

We will suppose that the optimal makespan is known in advance. As it often happens in makespan scheduling we just loose a constant factor in the competitive ratio by this assumption. This can be proved by the standard doubling technique.

Let $\mathcal{J} = \{J_1, \dots, J_n\}$ the set of jobs, $\mathcal{M} = \{M_1, \dots, M_m\}$ the set of machines and $E_i \subseteq \mathcal{M}$ for $1 \leq i \leq n$ the set of machines to which the job J_i is eligible. Let p_i be the size of job J_i and $\ell(i)$ the load of machine M_i at the currently considered point in time, which is the sum of loads of the active jobs assigned to i. Let $\varepsilon > 0$ be a constant. Let OPT be the (guessed) optimum makespan, which is fixed in our calculations.

If there is an incoming job J_i

1. assign J_i to a machine M_j, such that $j = \{\ell(j) \mid M_j \in E_i\}$
2. while there is a job J_t residing on a machine M_j and a machine $M_{j'} \in E_t$ such that $\ell(j) \geq \ell(j') + (1+\varepsilon)OPT$
   reassign J_t from M_j to $M_{j'}$.

lemma Migration is bounded by $3\lambda/\varepsilon$, where λ is the makespan algorithm X achieves.

For a fixed point in time define $\Phi := \sum_{j:M_j \in \mathcal{M}} \ell(j)^2$. Clearly $\Phi \geq 0$. By the invariant that all machines have load at most $\ell(j) \leq \lambda OPT$ before a new job arrives, the increase in potential when a new job J_t arrives is bounded by
$(\ell(j) + p_t)^2 - \ell(j)^2$
$= 2p_t\ell(j) + p_t^2$
$\leq 2p_t\lambda OPT + p_t^2$
$\leq 3p_t\lambda OPT$, where the last inequality is because of $\lambda \geq 1$, $p_t \leq OPT$. Thus, the total potential is bounded by $3\lambda OPT \sum_{1\leq t\leq n} p_t$.

Consider the potential change when a job J_t is migrated from a machine M_j to a machine $M_{j'}$. Clearly, only the load of the two machines M_j and $M_{j'}$ changes and we can bound the change in potential as follows. Let $\ell(j)$ and $\ell(j')$ denote the load of the machines *before* job J_t is migrated. Then we have that the change in potential is
$\ell(j)^2 + \ell(j')^2 - \big((\ell(j) - p_t)^2 + (\ell(j') + p_t)^2\big)$
$= \ell(j)^2 + \ell(j')^2 - \big((\ell(j)^2 - 2p_t\ell(j) + p_t^2 + \ell(j')^2 + 2p_t\ell(j') + p_t^2\big)$

$= 2p_t(\ell(j) - \ell(j')) - 2p_t^2$
$\geq 2p_t(1+\varepsilon)OPT - 2p_t^2$
$\geq 2p_t(1+\varepsilon)OPT - 2p_tOPT$
$= \varepsilon p_tOPT$ when a job J_t with size p_t is migrated. In order to bound the total volume of migration we enumerate the migrated jobs in the ordering they were migrated. Let $J'_1, \ldots, J'_k$ be these jobs. Note that it is possible that a job was migrated multiple times and thus $J'_i = J'_l$ is possible for $i \neq l$. Let $p'_1, \ldots, p'_k$ the corresponding processing times.

As the potential is non-negative and the loss in potential, when job J'_t is migrated, is at least εp_tOPT, we can bound the migrated volume by
$\sum_{1\leq t\leq k} \varepsilon p'_tOPT \leq 3\lambda OPT \sum_{1\leq t\leq n} p_t$
$\iff \sum_{1\leq t\leq k} p'_t \leq 3\lambda/\varepsilon \sum_{1\leq t\leq n} p_t,$
which yields a factor of migration of $3\lambda/\varepsilon$.
Lemma The algorithm is $O(\log(m)/\log\log(m))$-competitive.

Suppose that after the migration step the maximal load is at least $C \cdot OPT$, denote the machine with this load by v. Define the following sets of the machines. Let $S_0 = v$ and S_1 be the set of machines where some of the jobs assigned to v can be executed. For any $i \geq 2$ let S_i contain the machines which are not contained in $S_0 \cup S_1 \ldots \cup S_{i-1}$ and can execute some jobs assigned to one of the machines in $S_0 \cup S_1 \ldots \cup S_{i-1}$. Denote the size of S_i by n_i.

First observe that the load is at least $(C - i(1+\varepsilon))OPT$ on the machines of S_i. We can prove this statement by induction. For $i = 0$ it is obviously true. Suppose that the statement is true for a given i and consider a machine u from set S_{i+1}. Then there exists a machine w in set $S_0 \cup S_1 \ldots \cup S_i$ which has a job which can be executed on u. On the other hand the load of w is at least $(C - i(1+\varepsilon))OPT$ by the induction hypothesis. If u had smaller load than $(C - (i+1)(1+\varepsilon))OPT$ then one job should migrate from w to u which is a contradiction.

Now we prove that $n_i \geq \prod_{j=1}^{i}(C - j(1+\varepsilon))$ for $C - 1 \geq i \geq 1$. We prove that by induction. First consider S_1. The jobs which are assigned to v must be scheduled in the optimal schedule as well. The total load of these jobs is at least $C \cdot OPT$ and the load of a machine in the optimal schedule is at most OPT, therefore these jobs must be distributed to at least C machines in the optimal offline schedule. This also yields that $n_1 \geq C - 1 \geq C - (1+\varepsilon)$. We can prove the inductive step by a similar argument. Suppose that the statement is valid for an $1 \leq i < C - 1$. Consider the machines in $S_0 \cup S_1 \ldots \cup S_i$. The total load on these machines is at least $\sum_{j=0}^{i} n_i \cdot (C - i(1+\varepsilon))OPT$. This yields that at least $\sum_{j=0}^{i} n_i \cdot (C - i(1+\varepsilon))$ machines are used to schedule these jobs in an optimal schedule, therefore

$$n_{i+1} \geq \sum_{j=0}^{i} n_i \cdot (C - i(1+\varepsilon)) - \sum_{j=0}^{i} n_i = \sum_{j=0}^{i} n_i \cdot (C - i(1+\varepsilon) - 1).$$

Thus we obtained that $n_{i+1} \geq n_i(C - 1 - i(1+\varepsilon)) \geq \prod_{j=1}^{i+1}(C - j(1+\varepsilon))$.

On the other hand we have that $\sum_{i=1}^{C-1} n_i \leq m$ thus we obtain that $\prod_{j=1}^{C-1}(C-j(1+\varepsilon)) \leq m$ and this proves that $C = O(\log(m)/\log\log(m))$.