STRUCTURING THE OUTPUT

section 0.4 from

Approximation Schemes - A Tutorial

by Petra Schuurman & Gerhard J. Woeginger

Structuring the output

- 1. Partition possibe solutions into groups of similar solutions.
- 2. For each group $\mathcal{F}^{(l)}$: find the value APP^(l) of an approximately optimal solution within that group (APP^(l) $\leq (1 + \varepsilon)$ OPT^(l))
- 3. Take the best of $APP^{(1)}, APP^{(2)}, APP^{(3)}, \dots$

Structuring the output

- 1. Partition possibe solutions into groups of similar solutions.
- 2. For each group $\mathcal{F}^{(l)}$: find the value $\mathsf{APP}^{(l)}$ of an approximately optimal solution within that group $(\mathsf{APP}^{(l)} \leq (1 + \varepsilon)\mathsf{OPT}^{(l)})$
- 3. Take the best of $APP^{(1)}, APP^{(2)}, APP^{(3)}, \dots$

Why would this give an approximation of the global optimum?

- There is a group $\mathcal{F}^{(l^*)}$ s.t. $\mathsf{OPT}^{(l^*)} = \mathsf{OPT}$.
- The solution vaue $APP^{(l^*)}$ found for that group is a good approximation of that optimum: $APP^{(l^*)} \leq (1 + \varepsilon)OPT^{(l^*)} = (1 + \varepsilon)OPT.$
- The best solution of all districts is at least as good: $\min_l \mathsf{APP}^{(l)} \leq \mathsf{APP}^{(l^*)} \leq (1 + \varepsilon)\mathsf{OPT}.$

Structuring the output

- 1. Partition possibe solutions into groups of similar solutions.
- 2. For each group $\mathcal{F}^{(l)}$: find the value $\mathsf{APP}^{(l)}$ of an approximately optimal solution within that group $(\mathsf{APP}^{(l)} \leq (1 + \varepsilon)\mathsf{OPT}^{(l)})$
- 3. Take the best of $APP^{(1)}, APP^{(2)}, APP^{(3)}, \dots$

Why would this give an approximation of the global optimum? $\exists l^*: \mathsf{OPT}^{(l^*)} = \mathsf{OPT} \land \min_l \mathsf{APP}^{(l)} \leq \mathsf{APP}^{(l^*)} \leq (1 + \varepsilon)\mathsf{OPT}^{(l^*)}.$

How to make it work?

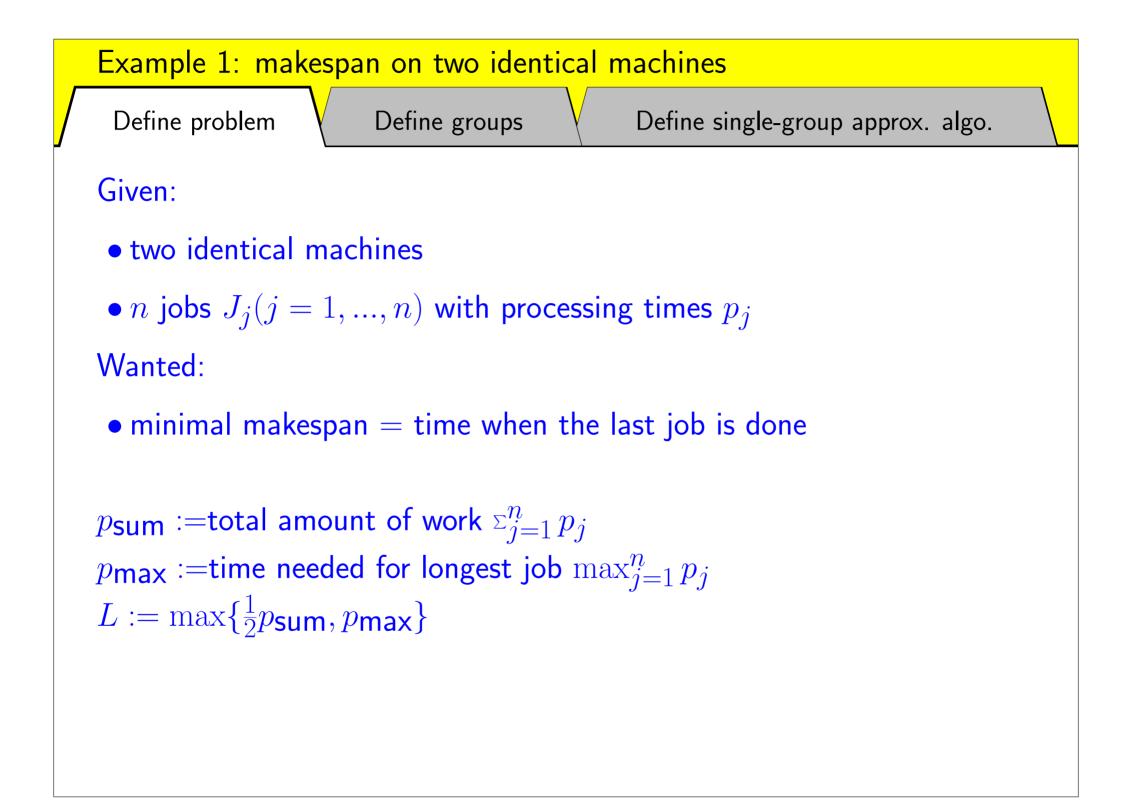
Polynomial number of groups (otherwise step 2 too slow), but not too few (otherwise step 2 too difficult).

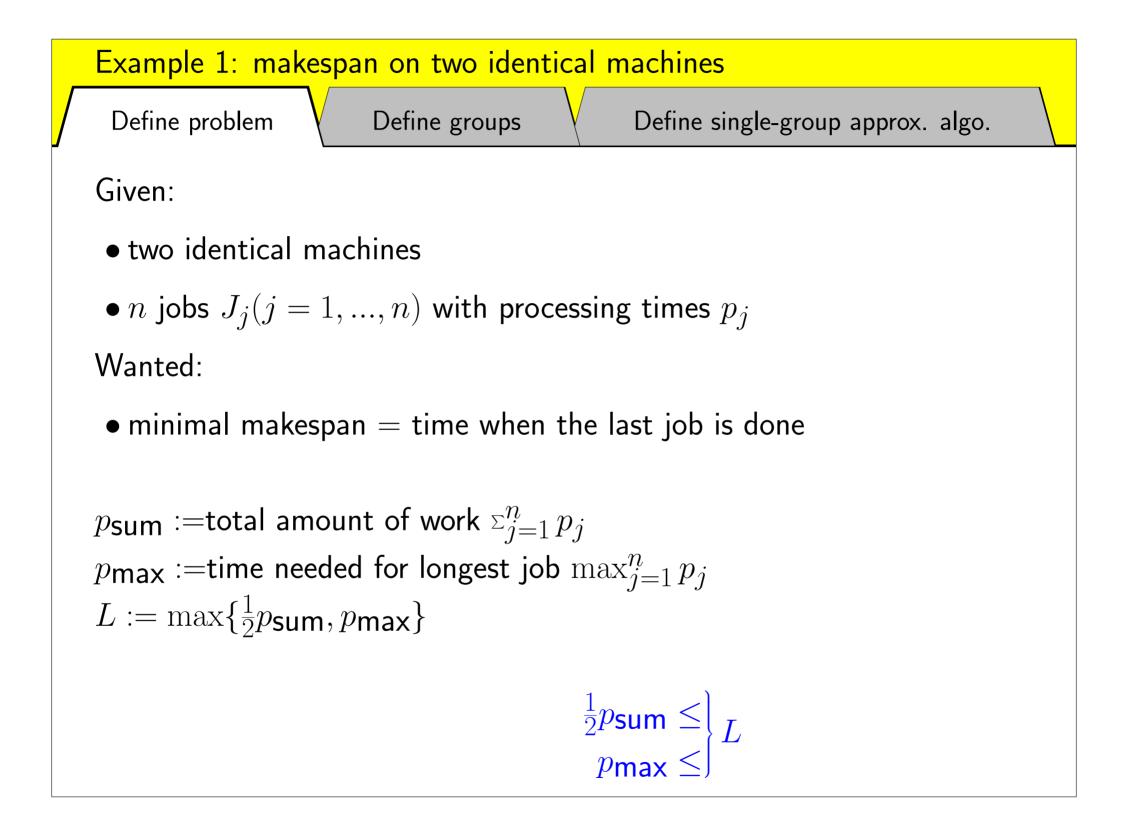
Example 1: makespan on two identical machines

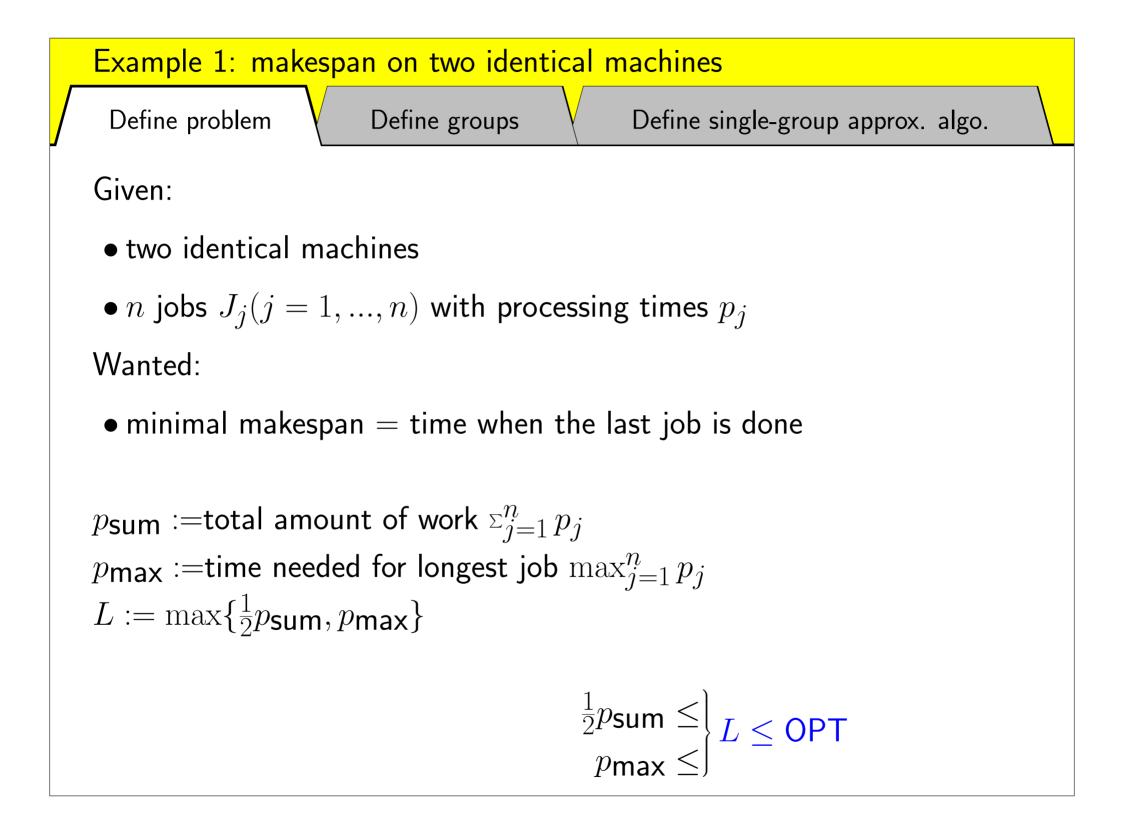
Define problem

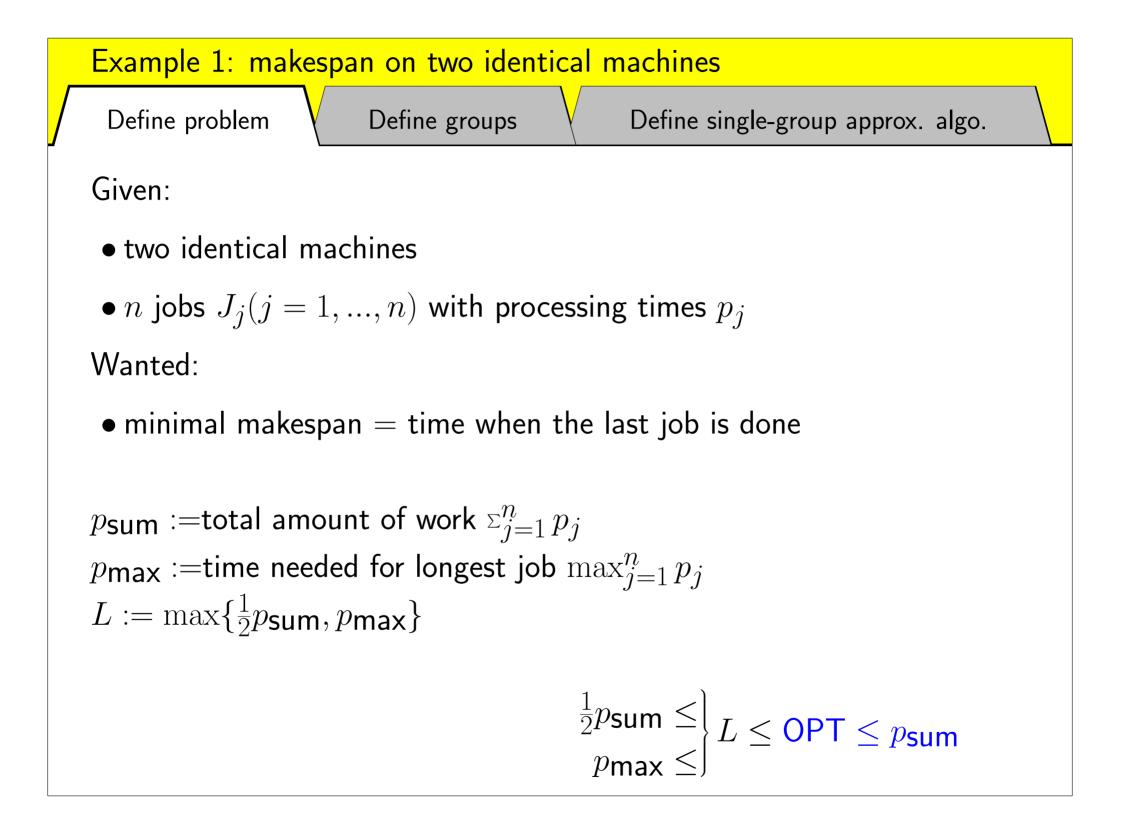
Define groups

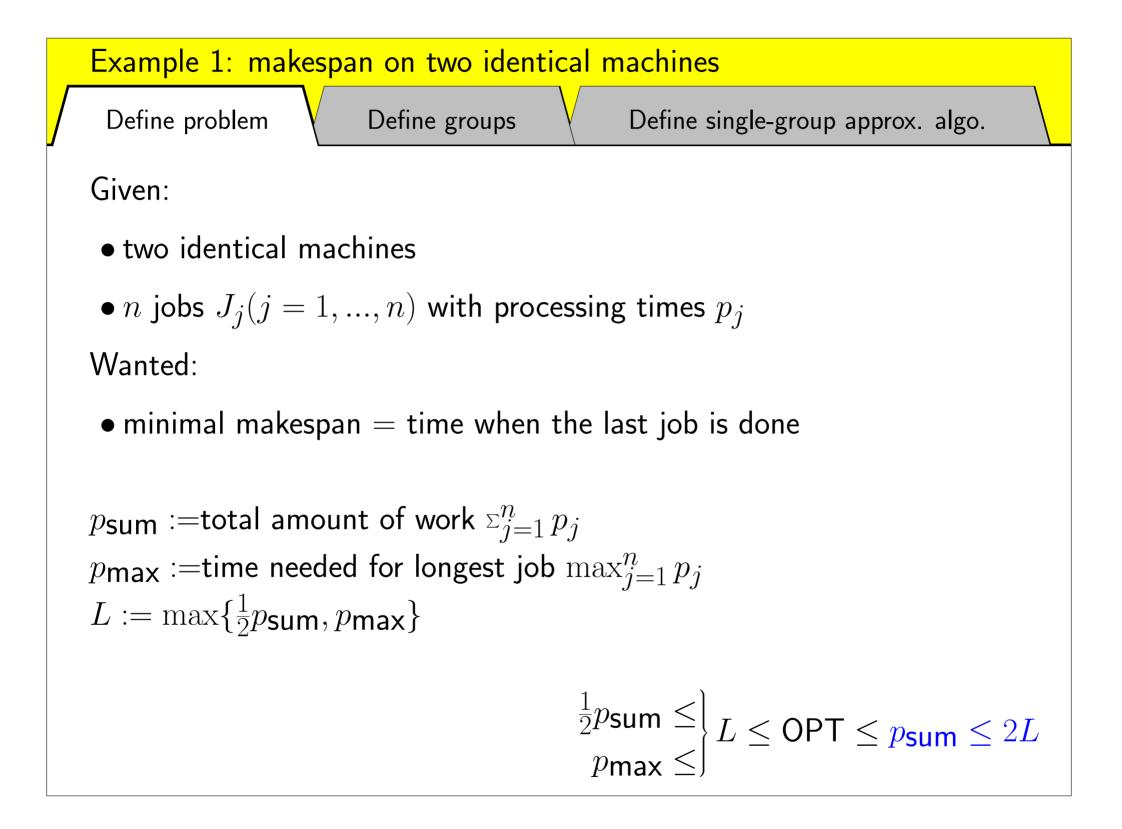
Define single-group approx. algo.

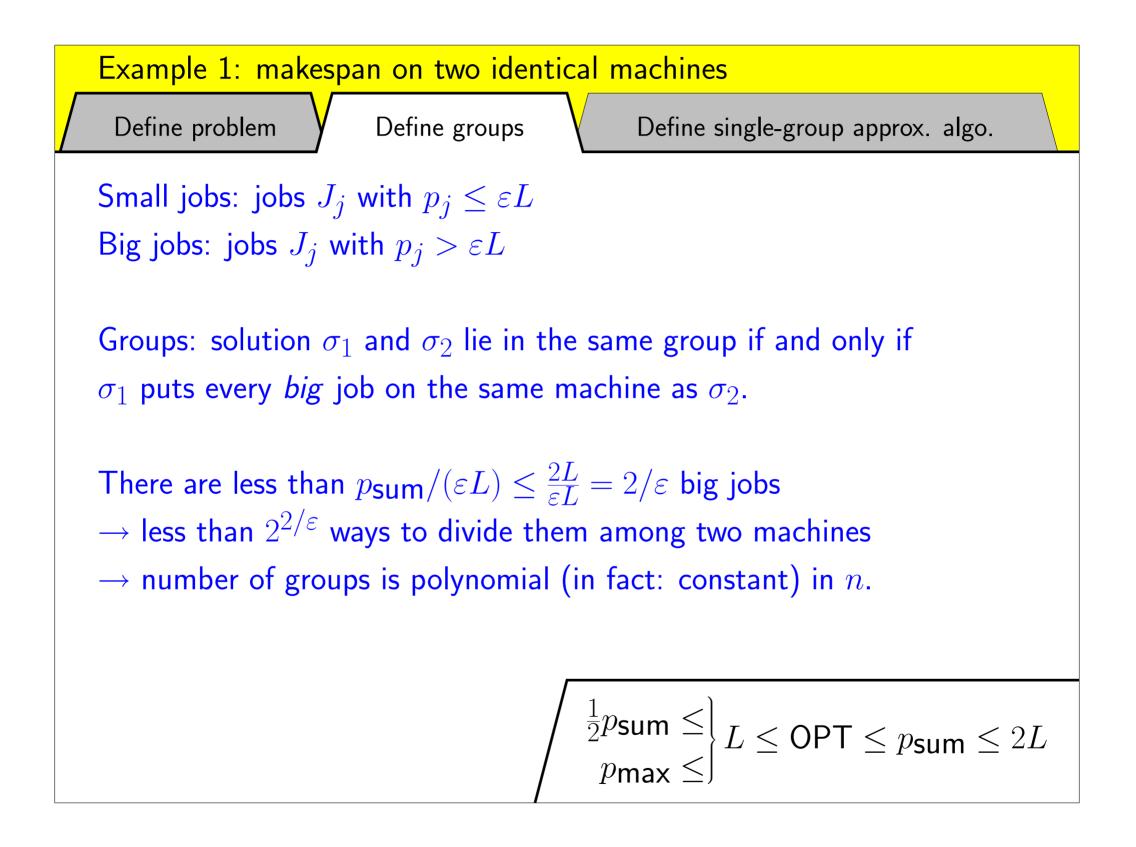


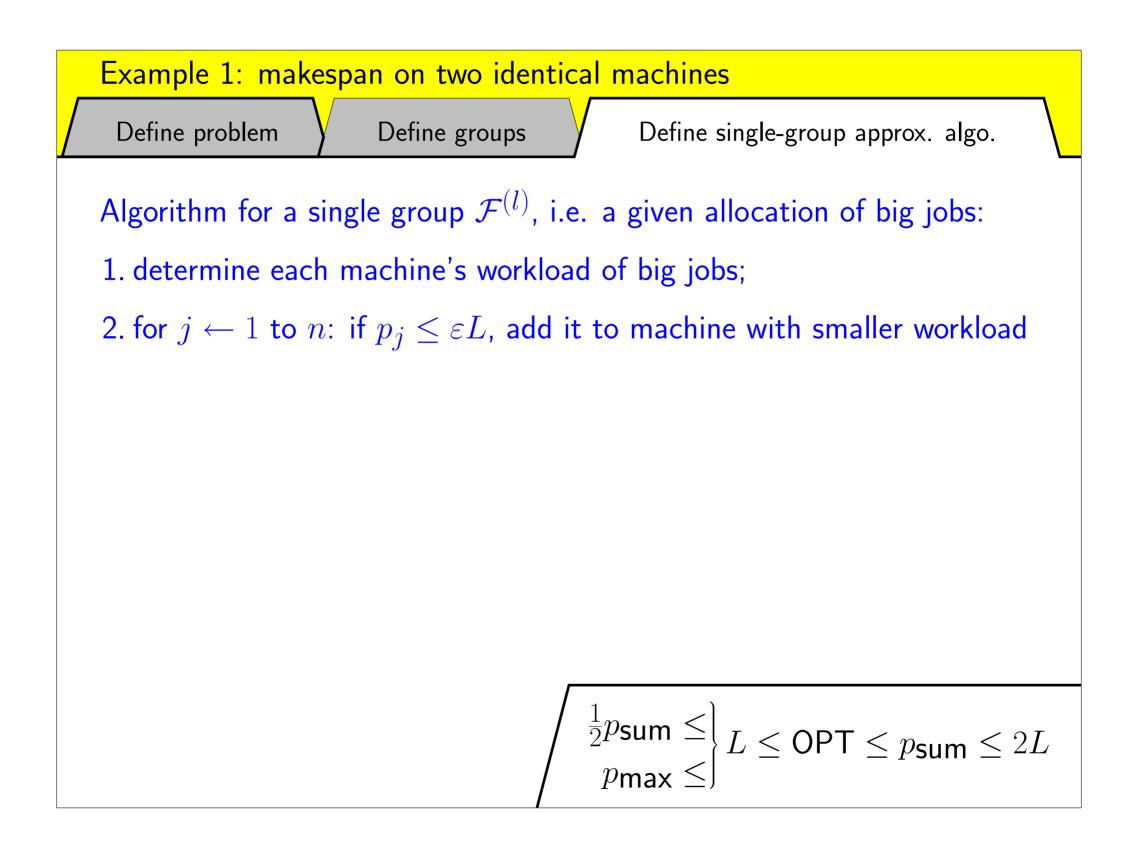


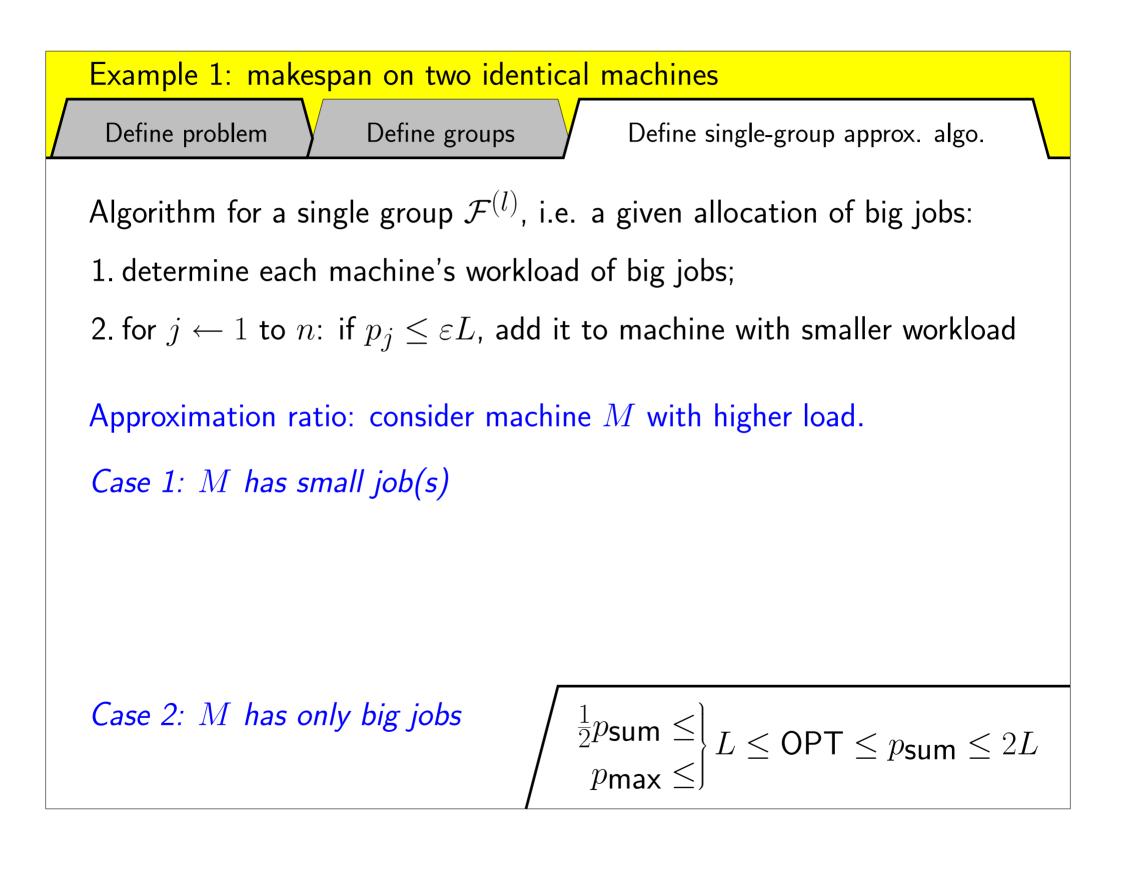


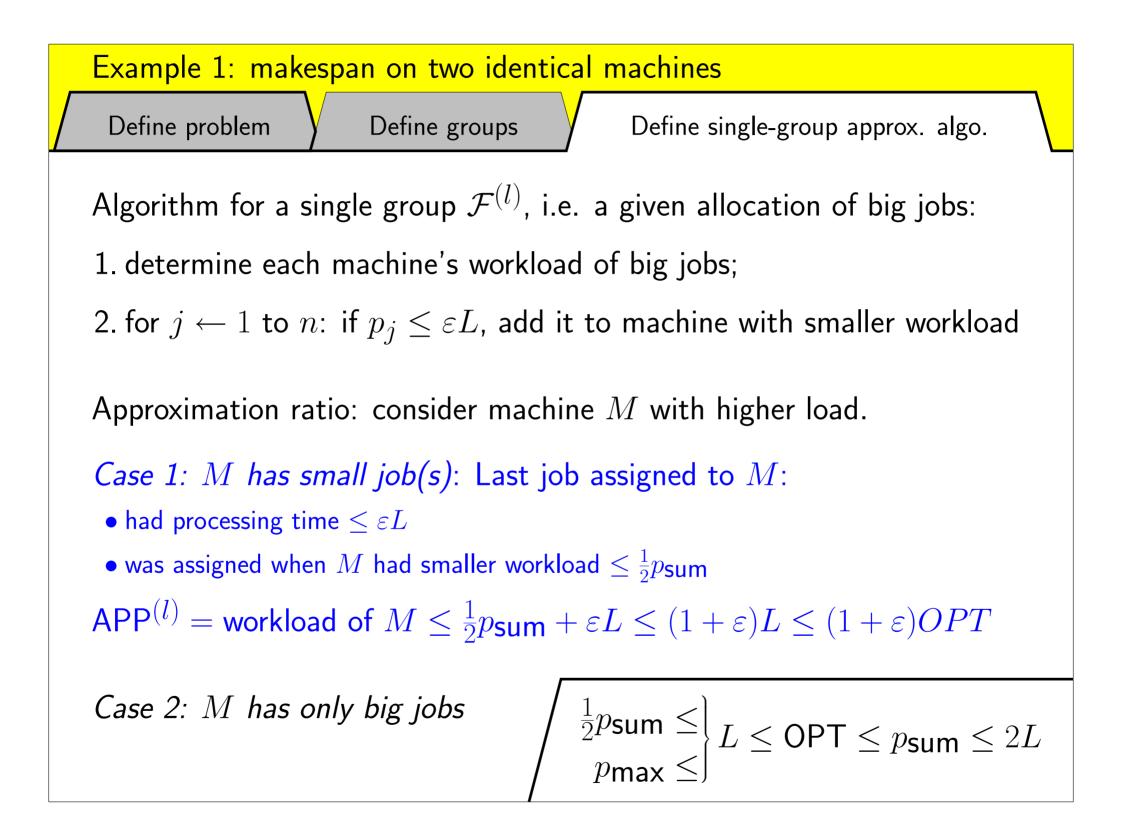


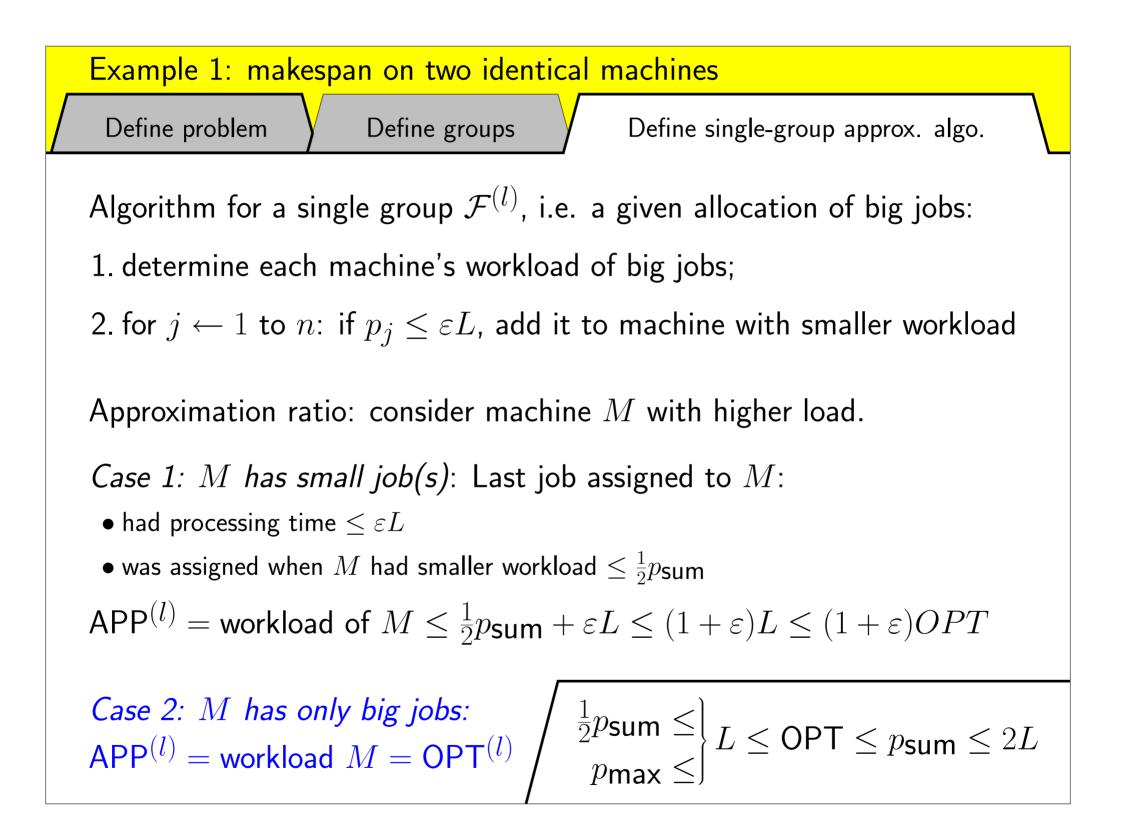


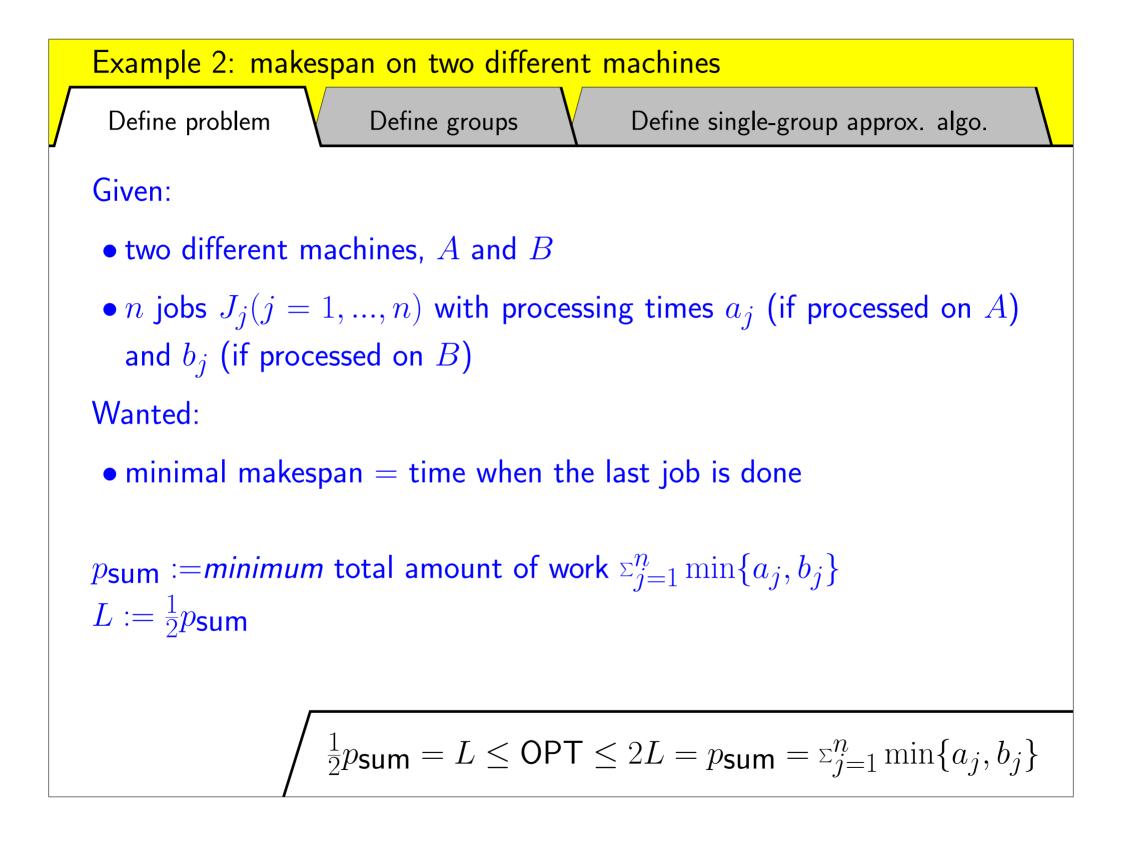


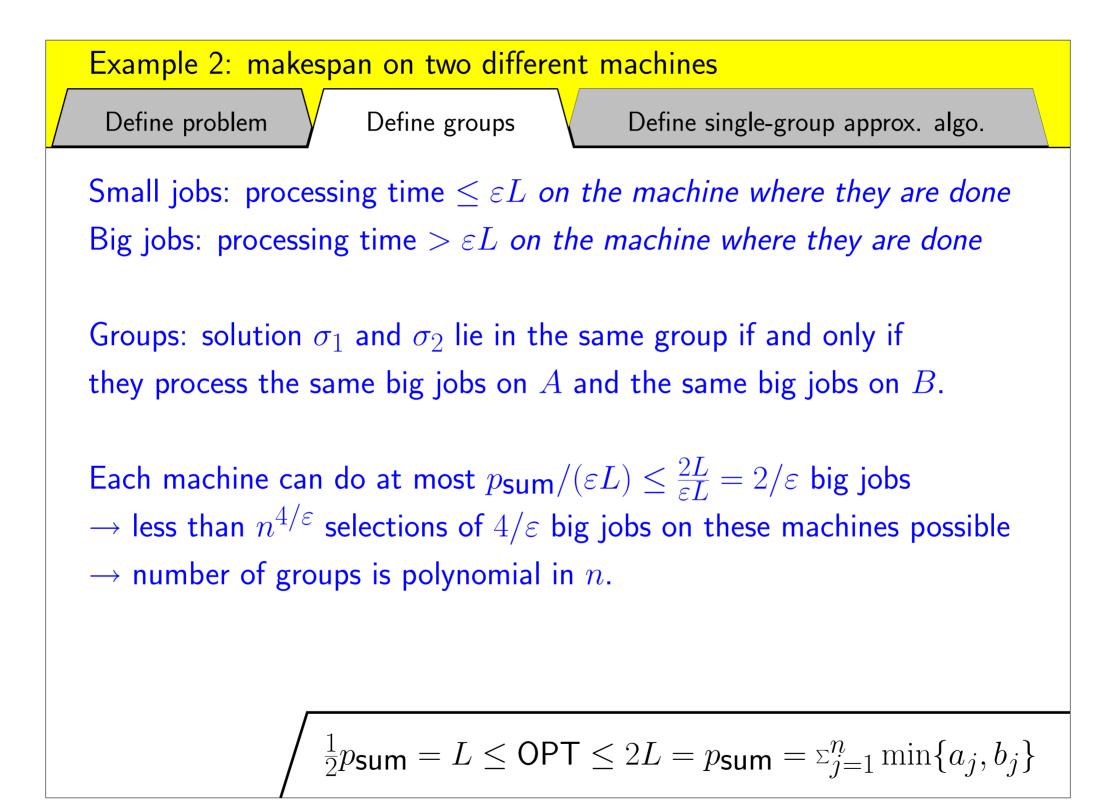


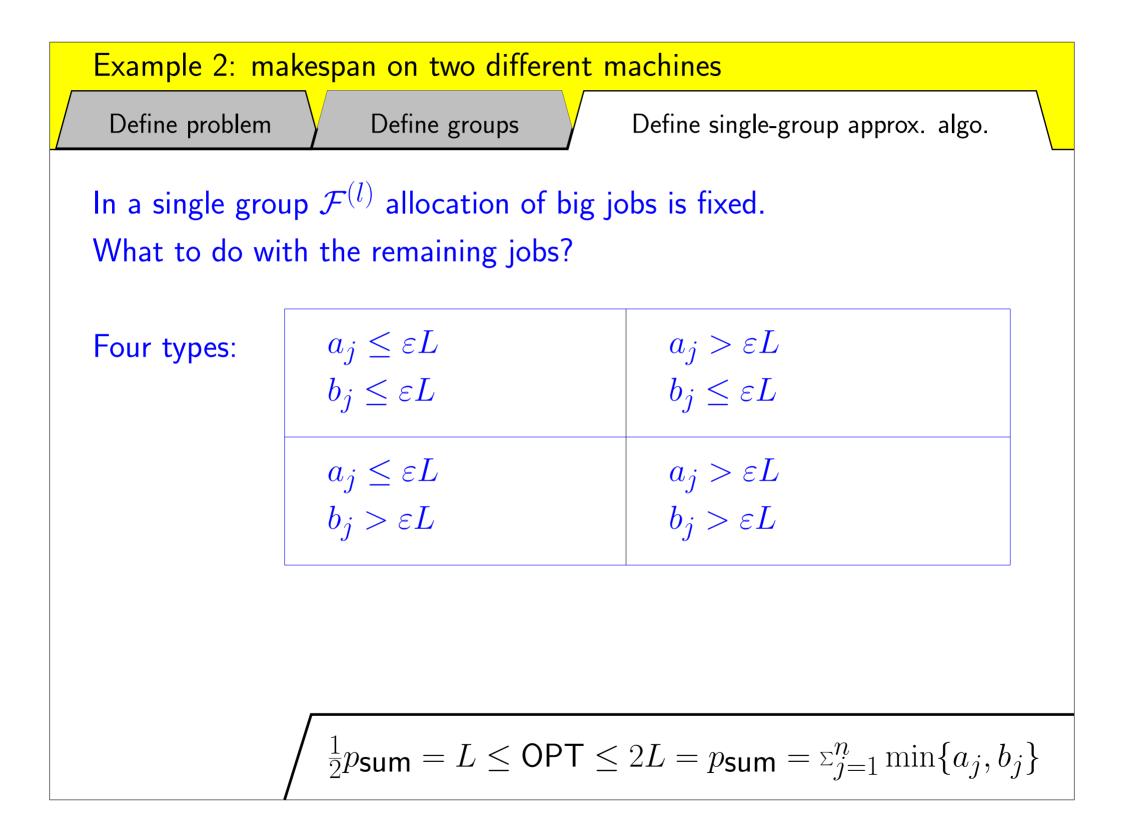


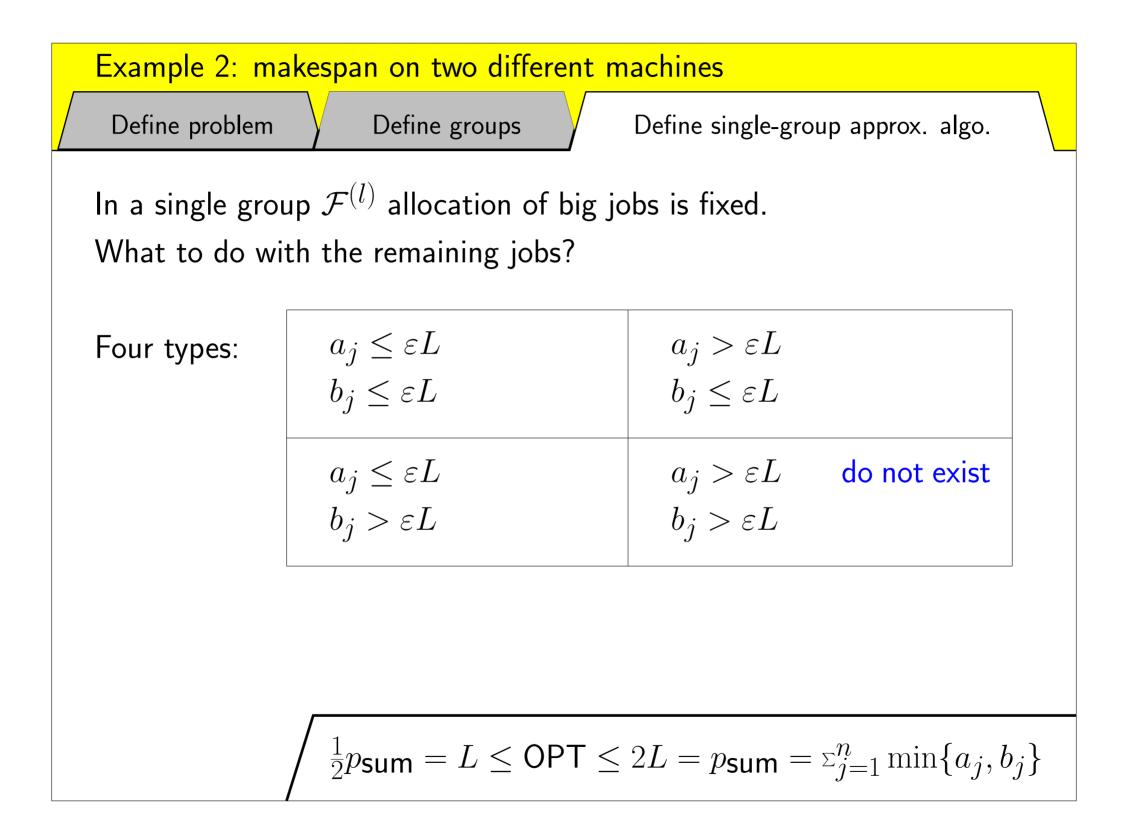


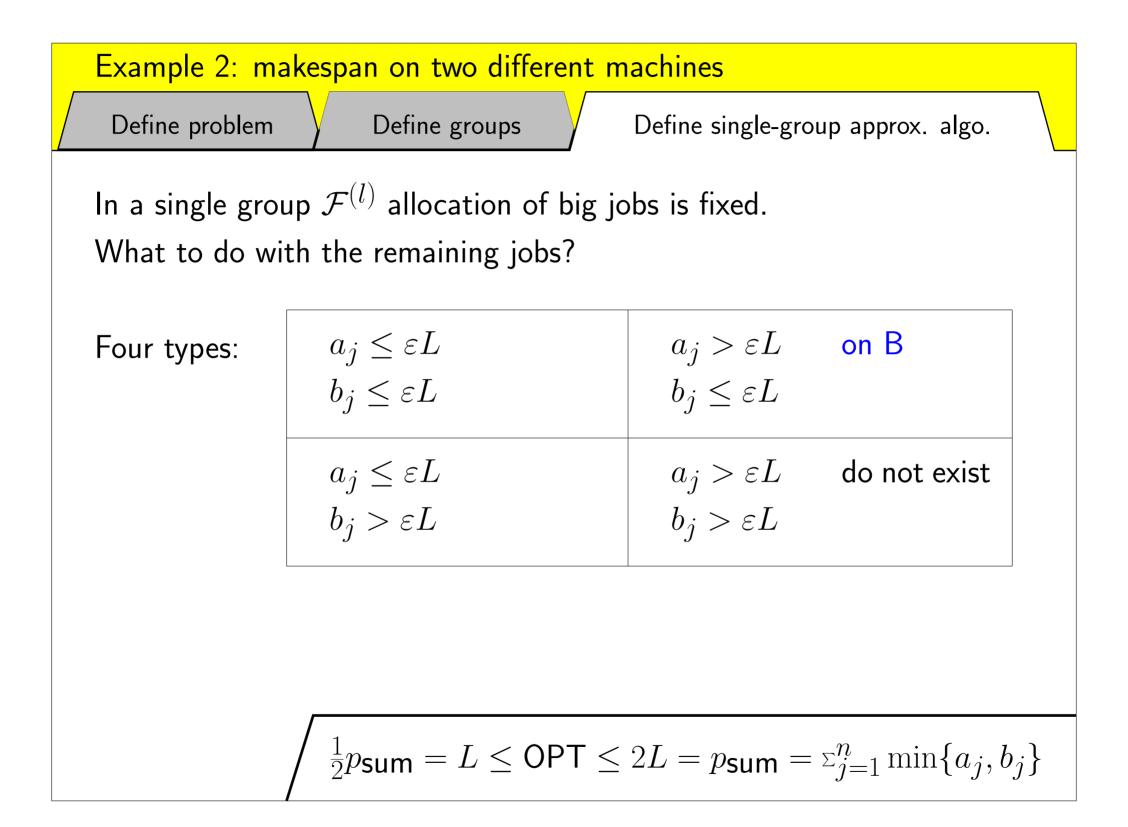


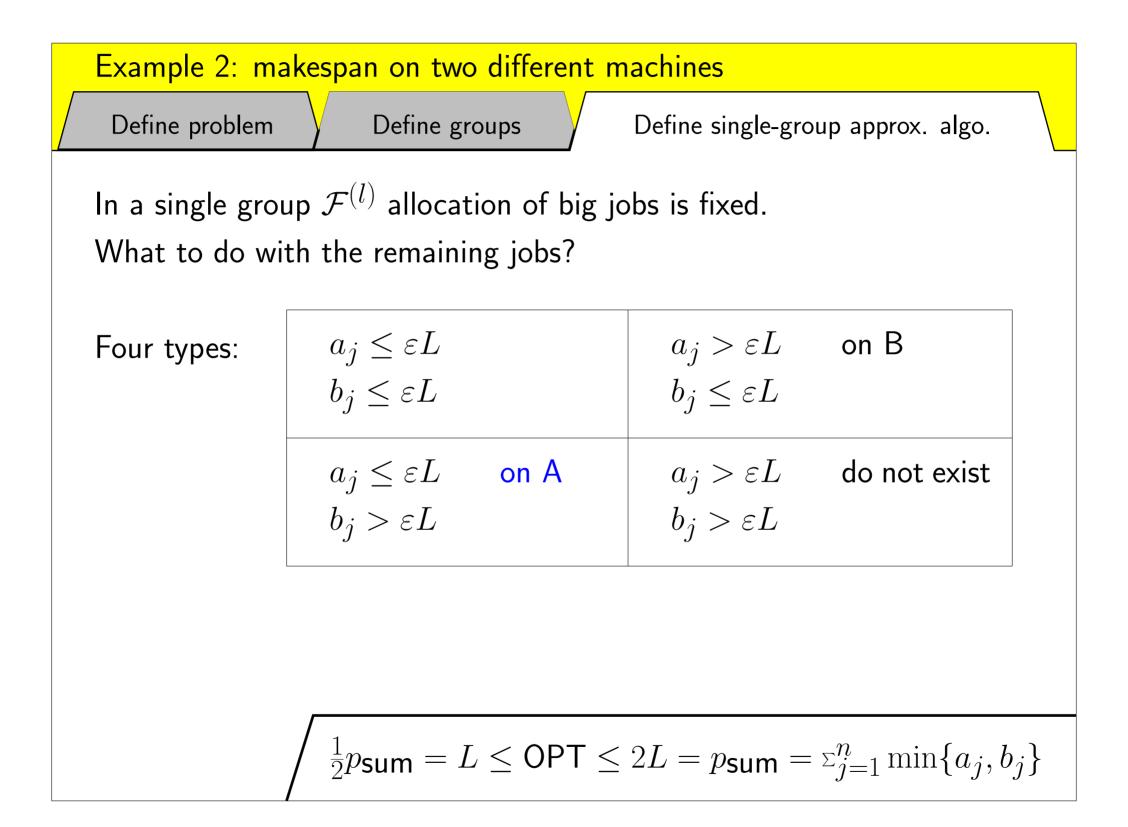


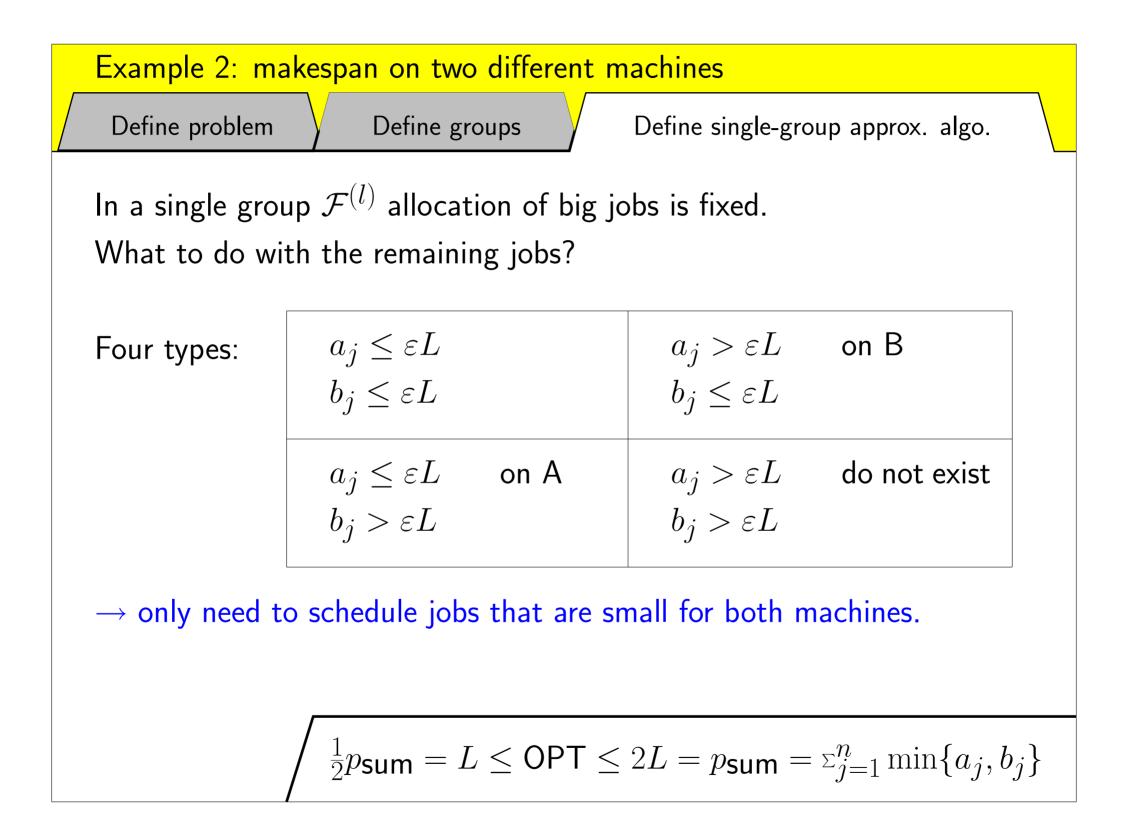


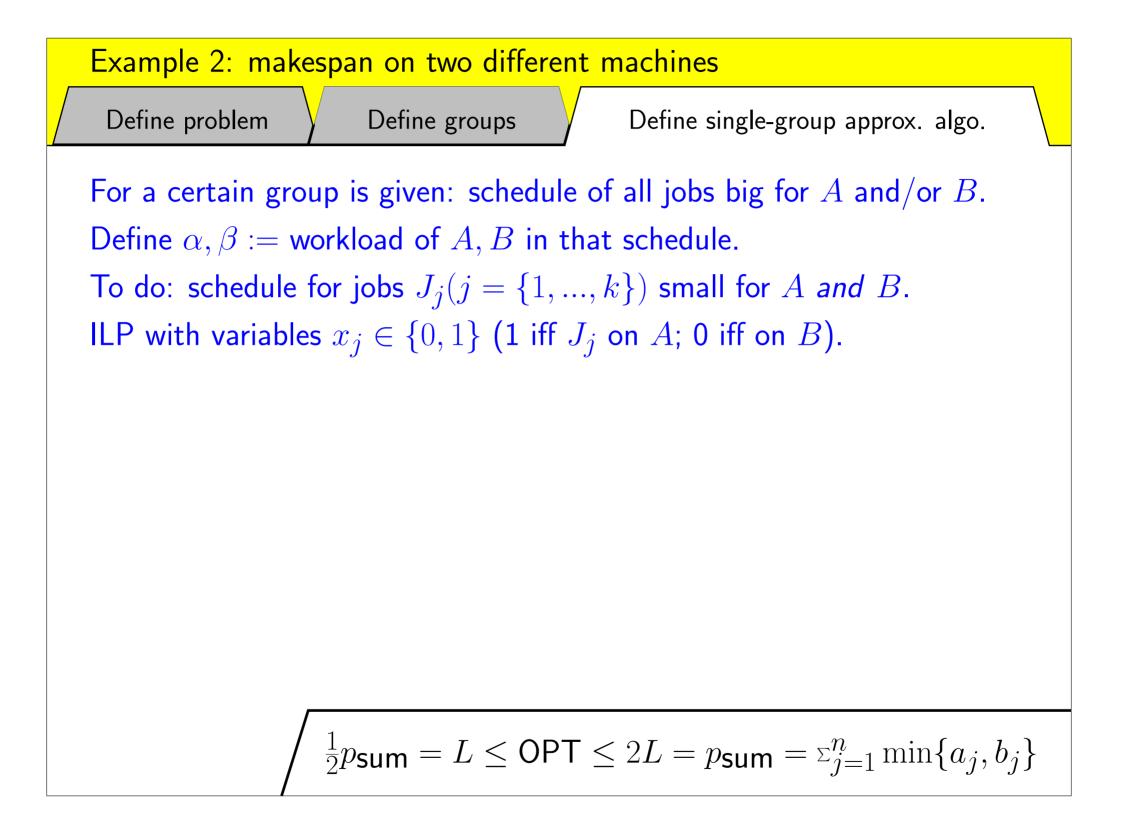


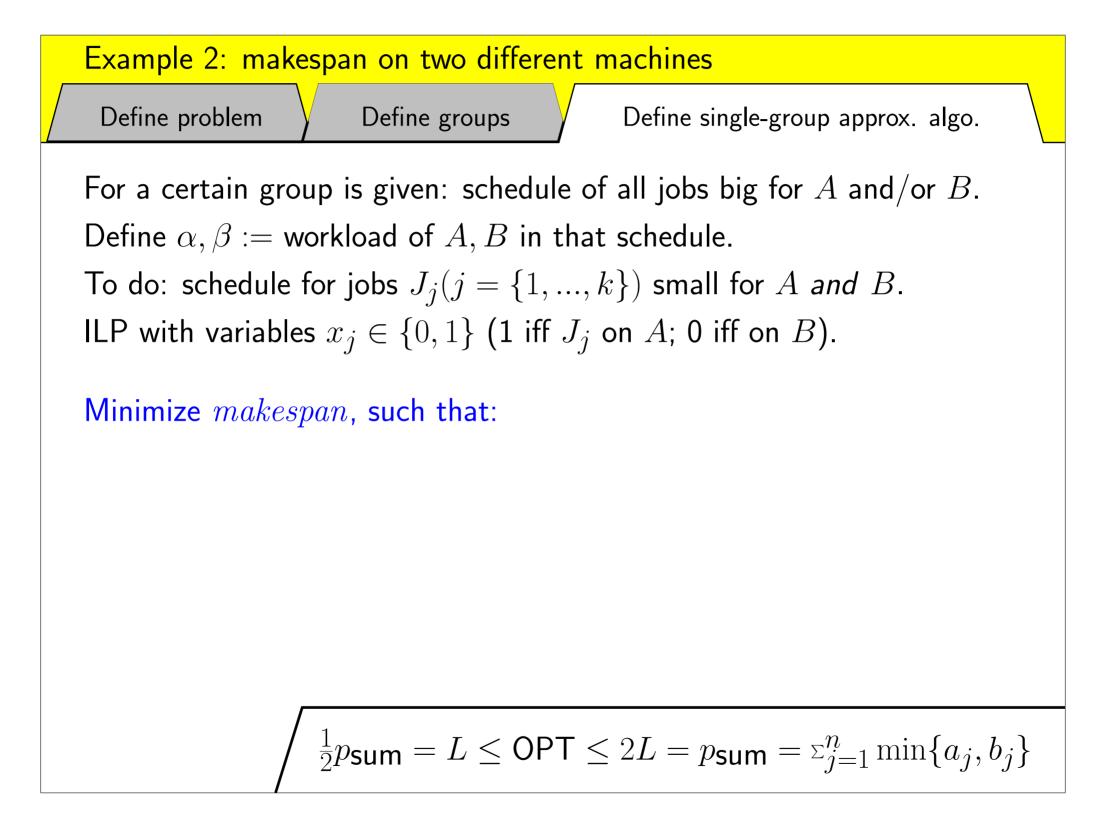


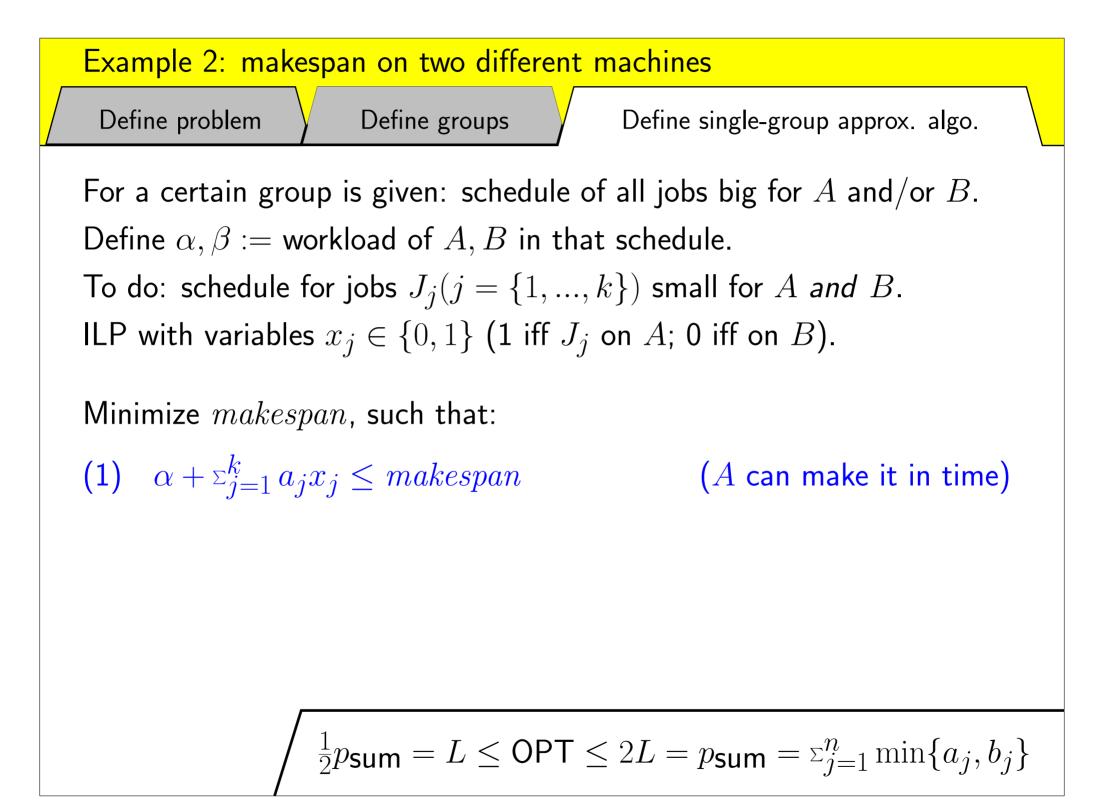


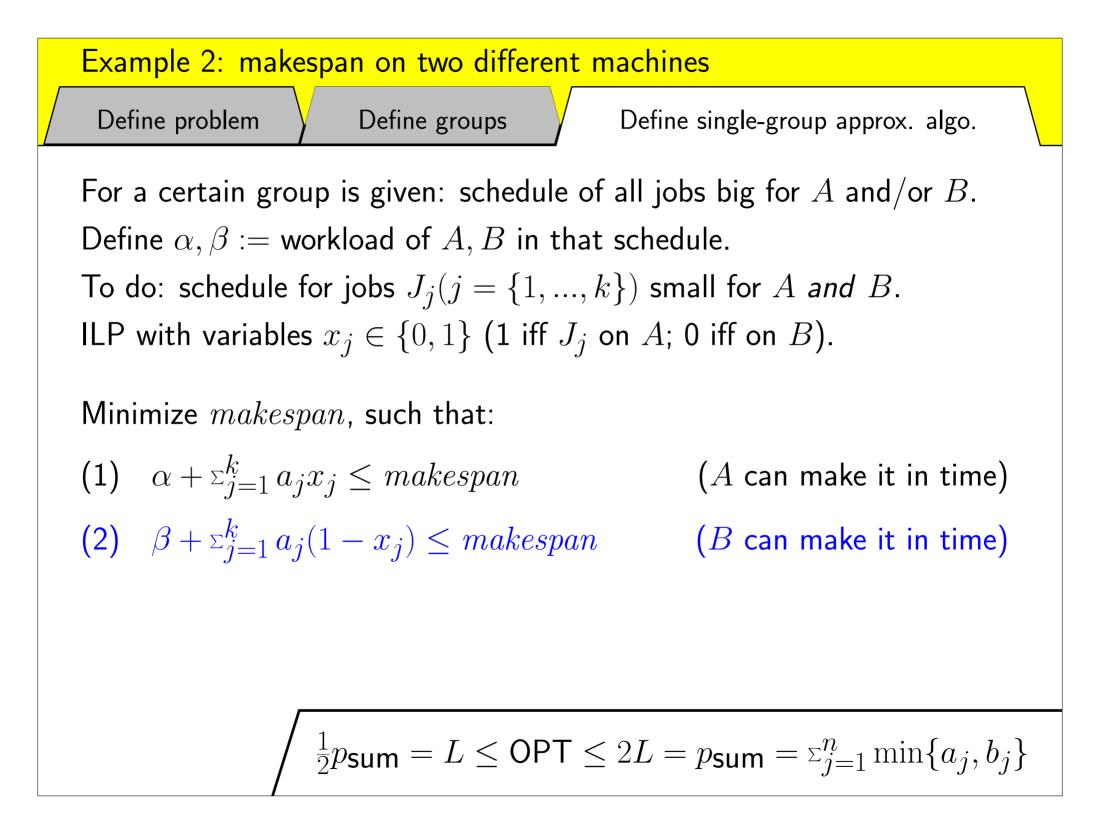


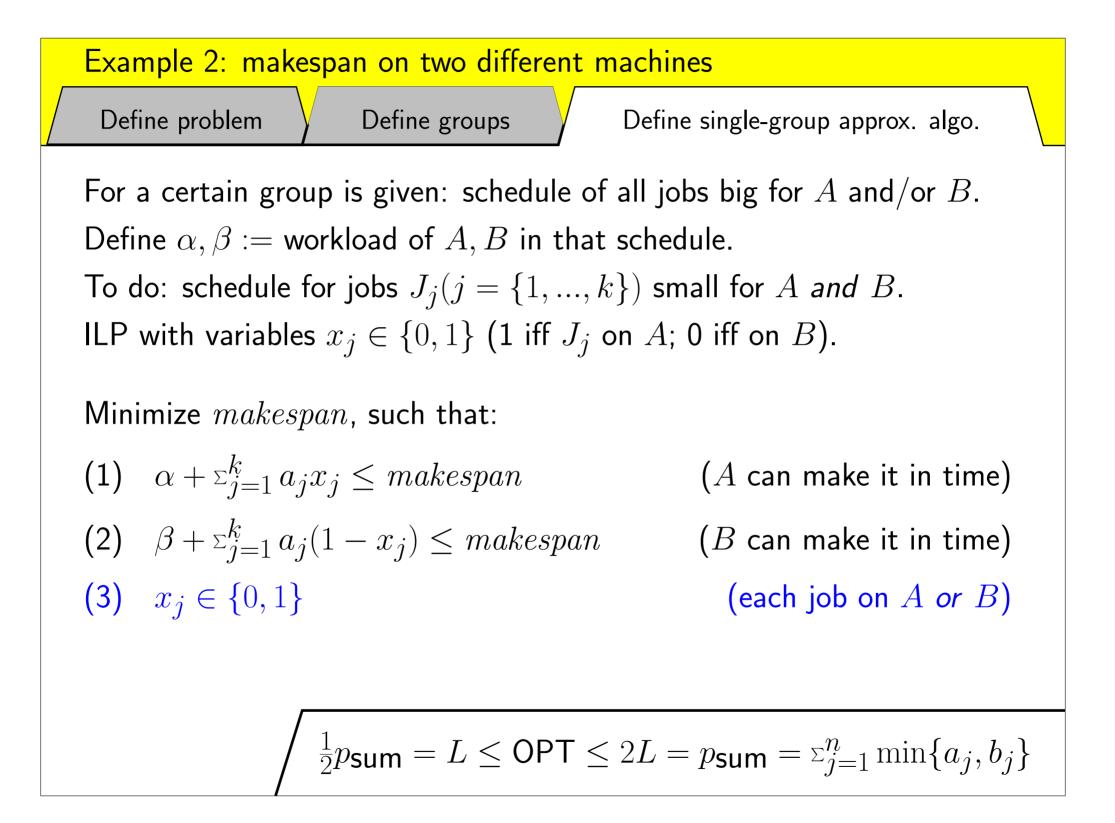


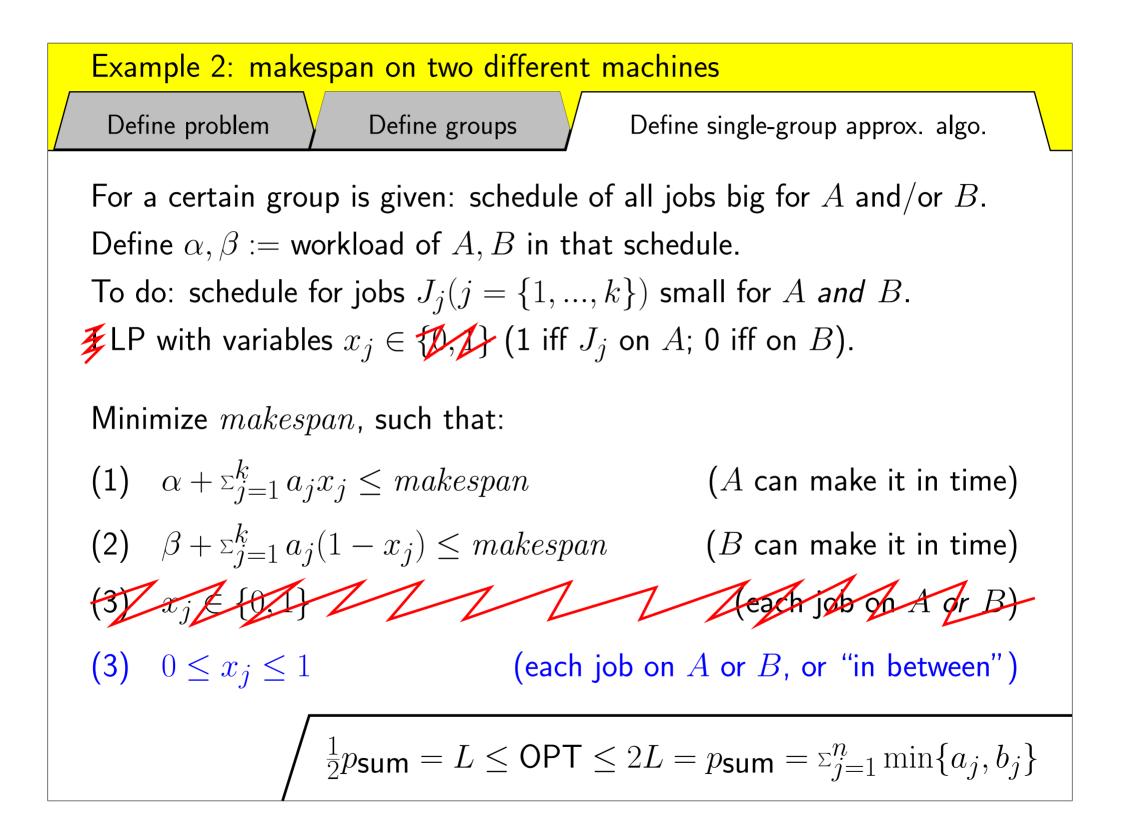












Example 2: makespan on two different machines Define problem Define groups Define single-group approx. algo. Solution space is (k + 1)-dimensional convex polyhedron. Optimal solution in vertex, i.e. $\geq k+1$ constraints fulfilled with equality. E.g. (1), and (2), and at least k - 1 of (3) \rightarrow leaves at most one x_i that isn't 0 or 1 Minimize *makespan*, such that: (1) $\alpha + \sum_{j=1}^{k} a_j x_j \leq makespan$ (A can make it in time)(2) $\beta + \sum_{j=1}^{k} a_j (1 - x_j) \le makespan$ (B can make it in time) L (each job on A or B) (3) $0 \le x_j \le 1$ (each job on A or B, or "in between") $\frac{1}{2}p_{\mathsf{sum}} = L \le \mathsf{OPT} \le 2L = p_{\mathsf{sum}} = \sum_{j=1}^{n} \min\{a_j, b_j\}$ Example 2: makespan on two different machines Define problem Define groups Define single-group approx. algo. Solution space is (k + 1)-dimensional convex polyhedron. Optimal solution in vertex, i.e. $\geq k+1$ constraints fulfilled with equality. E.g. (1), and (2), and at least k - 1 of (3) \rightarrow leaves at most one x_i that isn't 0 or 1

Put that one on any machine; schedule the rest according to LP solution. Approx. factor: only 1 small job may be scheduled on wrong machine:

 $\mathsf{APP}^{(l)} \le makespan + \varepsilon L \le \mathsf{OPT}^{(l)} + \varepsilon L \le (1 + \varepsilon)\mathsf{OPT}^{(l)}$

 $\frac{1}{2}p_{\mathsf{sum}} = L \le \mathsf{OPT} \le 2L = p_{\mathsf{sum}} = \sum_{j=1}^{n} \min\{a_j, b_j\}$