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$^{(3)}$, ...

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$^{(3)}$, ...

Why would this give an approximation of the global optimum?

- There is a group $\mathcal{F}^{(l^*)}$ s.t. OPT$^{(l^*)} =$ 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:
  $\min_l$ APP$^{(l)} \leq$ APP$^{(l^*)} \leq (1+\varepsilon)$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 $\mathsf{APP}^{(1)}, \mathsf{APP}^{(2)}, \mathsf{APP}^{(3)}, \ldots$.

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

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 1: makespan on two identical machines

Define problem | Define groups | Define single-group approx. algo.

Given:

- two identical machines
- n jobs $J_j (j = 1, ..., n)$ with processing times p_j

Wanted:

- minimal makespan = time when the last job is done

p_{sum} :=total amount of work $\Sigma_{j=1}^{n} p_j$

p_{max} :=time needed for longest job $\max_{j=1}^{n} p_j$

$L := \max\{\frac{1}{2} p_{\text{sum}}, p_{\text{max}}\}$

Example 1: makespan on two identical machines

Define problem | Define groups | Define single-group approx. algo.

Given:

- two identical machines
- n jobs $J_j (j = 1, ..., n)$ with processing times p_j

Wanted:

- minimal makespan = time when the last job is done

p_{sum} :=total amount of work $\Sigma_{j=1}^{n} p_j$

p_{max} :=time needed for longest job $\max_{j=1}^{n} p_j$

$L := \max\{\frac{1}{2}p_{\mathsf{sum}}, p_{\mathsf{max}}\}$

$$\left.\begin{array}{r} \frac{1}{2}p_{\mathsf{sum}} \leq \\ p_{\mathsf{max}} \leq \end{array}\right\} L$$

Example 1: makespan on two identical machines

Define problem | Define groups | Define single-group approx. algo.

Given:

- two identical machines
- n jobs $J_j (j = 1, ..., n)$ with processing times p_j

Wanted:

- minimal makespan = time when the last job is done

p_{sum} :=total amount of work $\Sigma_{j=1}^{n} p_j$

p_{max} :=time needed for longest job $\max_{j=1}^{n} p_j$

$L := \max\{\frac{1}{2}p_{\text{sum}}, p_{\text{max}}\}$

$$\left.\begin{array}{r} \frac{1}{2}p_{\text{sum}} \leq \\ p_{\text{max}} \leq \end{array}\right\} L \leq \text{OPT}$$

Example 1: makespan on two identical machines

Define problem | Define groups | Define single-group approx. algo.

Given:

- two identical machines
- n jobs $J_j (j = 1, ..., n)$ with processing times p_j

Wanted:

- minimal makespan = time when the last job is done

p_{sum} :=total amount of work $\Sigma_{j=1}^{n} p_j$

p_{max} :=time needed for longest job $\max_{j=1}^{n} p_j$

$L := \max\{\frac{1}{2}p_{\text{sum}}, p_{\text{max}}\}$

$$\left.\begin{array}{r} \frac{1}{2}p_{\text{sum}} \leq \\ p_{\text{max}} \leq \end{array}\right\} L \leq \text{OPT} \leq p_{\text{sum}}$$

Example 1: makespan on two identical machines

Define problem | Define groups | Define single-group approx. algo.

Given:

- two identical machines
- n jobs $J_j (j = 1, ..., n)$ with processing times p_j

Wanted:

- minimal makespan $=$ time when the last job is done

p_{sum} :=total amount of work $\Sigma_{j=1}^{n} p_j$

p_{max} :=time needed for longest job $\max_{j=1}^{n} p_j$

$L := \max\{\frac{1}{2}p_{\text{sum}}, p_{\text{max}}\}$

$$\left.\begin{array}{r}\frac{1}{2}p_{\text{sum}} \leq \\ p_{\text{max}} \leq\end{array}\right\} L \leq \text{OPT} \leq p_{\text{sum}} \leq 2L$$

Example 1: makespan on two identical machines

Define problem | **Define groups** | Define single-group approx. algo.

Small jobs: jobs J_j with $p_j \leq \varepsilon L$

Big jobs: jobs J_j with $p_j > \varepsilon L$

Groups: solution σ_1 and σ_2 lie in the same group if and only if σ_1 puts every *big* job on the same machine as σ_2.

There are less than $p_{\text{sum}}/(\varepsilon L) \leq \frac{2L}{\varepsilon L} = 2/\varepsilon$ big jobs

$\rightarrow$ less than $2^{2/\varepsilon}$ ways to divide them among two machines

$\rightarrow$ number of groups is polynomial (in fact: constant) in n.

$$\left.\begin{array}{r} \frac{1}{2}p_{\text{sum}} \leq \\ p_{\text{max}} \leq \end{array}\right\} L \leq \text{OPT} \leq p_{\text{sum}} \leq 2L$$

Example 1: makespan on two identical machines

Define problem | Define groups | Define single-group approx. algo.

Algorithm for a single group $\mathcal{F}^{(l)}$, i.e. a given allocation of big jobs:

1. determine each machine's workload of big jobs;

2. for $j \leftarrow 1$ to n: if $p_j \leq \varepsilon L$, add it to machine with smaller workload

$$\left.\begin{array}{r}\frac{1}{2}p_{\text{sum}} \leq \\ p_{\text{max}} \leq\end{array}\right\} L \leq \text{OPT} \leq p_{\text{sum}} \leq 2L$$

Example 1: makespan on two identical machines

Define problem | Define groups | **Define single-group approx. algo.**

Algorithm for a single group $\mathcal{F}^{(l)}$, i.e. a given allocation of big jobs:

1. determine each machine's workload of big jobs;

2. for $j \leftarrow 1$ to n: if $p_j \leq \varepsilon L$, add it to machine with smaller workload

Approximation ratio: consider machine M with higher load.

Case 1: M has small job(s)

Case 2: M has only big jobs

$$\left.\begin{array}{r}\frac{1}{2}p_{\text{sum}} \leq \\ p_{\text{max}} \leq\end{array}\right\} L \leq \text{OPT} \leq p_{\text{sum}} \leq 2L$$

Example 1: makespan on two identical machines

Define problem | Define groups | Define single-group approx. algo.

Algorithm for a single group $\mathcal{F}^{(l)}$, i.e. a given allocation of big jobs:

1. determine each machine's workload of big jobs;

2. for $j \leftarrow 1$ to n: if $p_j \leq \varepsilon L$, add it to machine with smaller workload

Approximation ratio: consider machine M with higher load.

Case 1: M has small job(s): Last job assigned to M:

- had processing time $\leq \varepsilon L$
- was assigned when M had smaller workload $\leq \frac{1}{2}p_{\text{sum}}$

$$\text{APP}^{(l)} = \text{workload of } M \leq \frac{1}{2}p_{\text{sum}} + \varepsilon L \leq (1+\varepsilon)L \leq (1+\varepsilon)OPT$$

Case 2: M has only big jobs

$$\left.\begin{array}{r}\frac{1}{2}p_{\text{sum}} \leq \\ p_{\max} \leq\end{array}\right\} L \leq \text{OPT} \leq p_{\text{sum}} \leq 2L$$

Example 1: makespan on two identical machines

Define problem | Define groups | **Define single-group approx. algo.**

Algorithm for a single group $\mathcal{F}^{(l)}$, i.e. a given allocation of big jobs:

1. determine each machine's workload of big jobs;

2. for $j \leftarrow 1$ to n: if $p_j \leq \varepsilon L$, add it to machine with smaller workload

Approximation ratio: consider machine M with higher load.

Case 1: M has small job(s): Last job assigned to M:

- had processing time $\leq \varepsilon L$
- was assigned when M had smaller workload $\leq \frac{1}{2} p_{\text{sum}}$

$\text{APP}^{(l)} = \text{workload of } M \leq \frac{1}{2} p_{\text{sum}} + \varepsilon L \leq (1+\varepsilon)L \leq (1+\varepsilon)OPT$

Case 2: M has only big jobs:
$\text{APP}^{(l)} = \text{workload } M = \text{OPT}^{(l)}$

$$\left.\begin{matrix} \frac{1}{2} p_{\text{sum}} \leq \\ p_{\max} \leq \end{matrix}\right\} L \leq \text{OPT} \leq p_{\text{sum}} \leq 2L$$

Example 2: makespan on two different machines

Define problem | Define groups | Define single-group approx. algo.

Given:

- two different machines, A and B
- n jobs $J_j (j = 1, ..., n)$ with processing times a_j (if processed on A) and b_j (if processed on B)

Wanted:

- minimal makespan $=$ time when the last job is done

p_{sum} :=*minimum* total amount of work $\Sigma_{j=1}^{n} \min\{a_j, b_j\}$
$L := \frac{1}{2} p_{\text{sum}}$

$$\frac{1}{2} p_{\text{sum}} = L \leq \text{OPT} \leq 2L = p_{\text{sum}} = \Sigma_{j=1}^{n} \min\{a_j, b_j\}$$

Example 2: makespan on two different machines

Define problem | **Define groups** | Define single-group approx. algo.

Small jobs: processing time $\leq \varepsilon L$ *on the machine where they are done*

Big jobs: processing time $> \varepsilon L$ *on the machine where they are done*

Groups: solution σ_1 and σ_2 lie in the same group if and only if they process the same big jobs on A and the same big jobs on B.

Each machine can do at most $p_{\mathsf{sum}}/(\varepsilon L) \leq \frac{2L}{\varepsilon L} = 2/\varepsilon$ big jobs

$\rightarrow$ less than $n^{4/\varepsilon}$ selections of $4/\varepsilon$ big jobs on these machines possible

$\rightarrow$ number of groups is polynomial in n.

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

Example 2: makespan on two different machines

Define problem | Define groups | Define single-group approx. algo.

In a single group $\mathcal{F}^{(l)}$ allocation of big jobs is fixed.

What to do with the remaining jobs?

Four types:

$a_j \leq \varepsilon L$ $b_j \leq \varepsilon L$	$a_j > \varepsilon L$ $b_j \leq \varepsilon L$
$a_j \leq \varepsilon L$ $b_j > \varepsilon L$	$a_j > \varepsilon L$ $b_j > \varepsilon L$

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

In a single group $\mathcal{F}^{(l)}$ allocation of big jobs is fixed.

What to do with the remaining jobs?

Four types:

$a_j \le \varepsilon L$ $b_j \le \varepsilon L$	$a_j > \varepsilon L$ $b_j \le \varepsilon L$
$a_j \le \varepsilon L$ $b_j > \varepsilon L$	$a_j > \varepsilon L$ $b_j > \varepsilon L$ do not exist

Example 2: makespan on two different machines

In a single group $\mathcal{F}^{(l)}$ allocation of big jobs is fixed.

What to do with the remaining jobs?

Four types:

$a_j \leq \varepsilon L$ $b_j \leq \varepsilon L$	$a_j > \varepsilon L$ $b_j \leq \varepsilon L$ on B
$a_j \leq \varepsilon L$ $b_j > \varepsilon L$	$a_j > \varepsilon L$ $b_j > \varepsilon L$ do not exist

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

Example 2: makespan on two different machines

Define problem | Define groups | **Define single-group approx. algo.**

In a single group $\mathcal{F}^{(l)}$ allocation of big jobs is fixed.

What to do with the remaining jobs?

Four types:

$a_j \leq \varepsilon L$ $b_j \leq \varepsilon L$	$a_j > \varepsilon L$ on B $b_j \leq \varepsilon L$
$a_j \leq \varepsilon L$ on A $b_j > \varepsilon L$	$a_j > \varepsilon L$ do not exist $b_j > \varepsilon L$

$\frac{1}{2}p_{\text{sum}} = L \leq \text{OPT} \leq 2L = p_{\text{sum}} = \Sigma_{j=1}^{n} \min\{a_j, b_j\}$

In a single group $\mathcal{F}^{(l)}$ allocation of big jobs is fixed.

What to do with the remaining jobs?

Four types:

$a_j \leq \varepsilon L$ $b_j \leq \varepsilon L$	$a_j > \varepsilon L$ on B $b_j \leq \varepsilon L$
$a_j \leq \varepsilon L$ on A $b_j > \varepsilon L$	$a_j > \varepsilon L$ do not exist $b_j > \varepsilon L$

$\longrightarrow$ only need to schedule jobs that are small for both machines.

Example 2: makespan on two different machines

Define problem | Define groups | Define single-group approx. algo.

For a certain group is given: schedule of all jobs big for A and/or B.

Define $\alpha, \beta :=$ workload of A, B in that schedule.

To do: schedule for jobs $J_j(j = \{1, ..., k\})$ small for A *and* B.

ILP with variables $x_j \in \{0, 1\}$ (1 iff J_j on A; 0 iff on B).

$\frac{1}{2}p_{\textsf{sum}} = L \leq \textsf{OPT} \leq 2L = p_{\textsf{sum}} = \Sigma_{j=1}^{n} \min\{a_j, b_j\}$

Example 2: makespan on two different machines

Define problem | Define groups | Define single-group approx. algo.

For a certain group is given: schedule of all jobs big for A and/or B.

Define $\alpha, \beta :=$ workload of A, B in that schedule.

To do: schedule for jobs $J_j (j = \{1, ..., k\})$ small for A *and* B.

ILP with variables $x_j \in \{0, 1\}$ (1 iff J_j on A; 0 iff on B).

Minimize $makespan$, such that:

$\frac{1}{2} p_{\text{sum}} = L \leq \text{OPT} \leq 2L = p_{\text{sum}} = \Sigma_{j=1}^{n} \min\{a_j, b_j\}$

Example 2: makespan on two different machines

Define problem | Define groups | Define single-group approx. algo.

For a certain group is given: schedule of all jobs big for A and/or B.

Define $\alpha, \beta :=$ workload of A, B in that schedule.

To do: schedule for jobs $J_j(j = \{1, ..., k\})$ small for A *and* B.

ILP with variables $x_j \in \{0, 1\}$ (1 iff J_j on A; 0 iff on B).

Minimize $makespan$, such that:

(1) $\alpha + \Sigma_{j=1}^{k} a_j x_j \leq makespan$ (A can make it in time)

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

Example 2: makespan on two different machines

Define problem | Define groups | Define single-group approx. algo.

For a certain group is given: schedule of all jobs big for A and/or B.

Define $\alpha, \beta :=$ workload of A, B in that schedule.

To do: schedule for jobs $J_j (j = \{1, ..., k\})$ small for A *and* B.

ILP with variables $x_j \in \{0, 1\}$ (1 iff J_j on A; 0 iff on B).

Minimize $makespan$, such that:

(1) $\alpha + \Sigma_{j=1}^{k} a_j x_j \leq makespan$ (A can make it in time)

(2) $\beta + \Sigma_{j=1}^{k} a_j (1 - x_j) \leq makespan$ (B can make it in time)

$\frac{1}{2} p_{\textsf{sum}} = L \leq \textsf{OPT} \leq 2L = p_{\textsf{sum}} = \Sigma_{j=1}^{n} \min\{a_j, b_j\}$

Example 2: makespan on two different machines

Define problem | Define groups | Define single-group approx. algo.

For a certain group is given: schedule of all jobs big for A and/or B.

Define $\alpha, \beta :=$ workload of A, B in that schedule.

To do: schedule for jobs $J_j (j = \{1, ..., k\})$ small for A *and* B.

ILP with variables $x_j \in \{0, 1\}$ (1 iff J_j on A; 0 iff on B).

Minimize $makespan$, such that:

(1) $\alpha + \Sigma_{j=1}^{k} a_j x_j \leq makespan$ (A can make it in time)

(2) $\beta + \Sigma_{j=1}^{k} a_j (1 - x_j) \leq makespan$ (B can make it in time)

(3) $x_j \in \{0, 1\}$ (each job on A *or* B)

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

For a certain group is given: schedule of all jobs big for A and/or B.

Define $\alpha, \beta :=$ workload of A, B in that schedule.

To do: schedule for jobs $J_j (j = \{1, ..., k\})$ small for A *and* B.

~~I~~LP with variables $x_j \in \cancel{\{0, 1\}}$ (1 iff J_j on A; 0 iff on B).

Minimize $makespan$, such that:

(1) $\alpha + \Sigma_{j=1}^{k} a_j x_j \leq makespan$ (A can make it in time)

(2) $\beta + \Sigma_{j=1}^{k} a_j (1 - x_j) \leq makespan$ (B can make it in time)

~~(3) $x_j \in \{0, 1\}$ (each job on A or B)~~

(3) $0 \leq x_j \leq 1$ (each job on A or B, or "in between")

$\frac{1}{2} p_{\text{sum}} = L \leq \text{OPT} \leq 2L = p_{\text{sum}} = \Sigma_{j=1}^{n} \min\{a_j, b_j\}$

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_j that isn't 0 or 1

Minimize $makespan$, such that:

(1) $\alpha + \Sigma_{j=1}^{k} a_j x_j \leq makespan$ (A can make it in time)

(2) $\beta + \Sigma_{j=1}^{k} a_j (1 - x_j) \leq makespan$ (B can make it in time)

~~(3)~~ $\cancel{x_j \in \{0,1\}}$ ~~(each job on A or B)~~

(3) $0 \leq x_j \leq 1$ (each job on A or B, or "in between")

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

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_j 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:

$$\text{APP}^{(l)} \leq \textit{makespan} + \varepsilon L \leq \text{OPT}^{(l)} + \varepsilon L \leq (1+\varepsilon)\text{OPT}^{(l)}$$

$$\frac{1}{2}p_{\text{sum}} = L \leq \text{OPT} \leq 2L = p_{\text{sum}} = \Sigma_{j=1}^{n} \min\{a_j, b_j\}$$