

SAPIENZA
UNIVERSITÀ DI ROMA

Approximation Algorithms

Exercises 1 — Approximation algorithms

Worked exercises on the chapters of Volume 1, approximation algorithms: each exercise is followed by its complete solution.

Fabio Furini

fabiofurini.github.io

Department of Computer, Control and Management Engineering — Sapienza University of Rome

Online version and downloadable material:
fabiofurini.github.io/approximation-algorithms/
github.com/fabiofurini/approximation-algorithms

Academic year 2026–2027

Contents

1 Principles of approximation algorithms **3**

Knapsack problems **9**

2 Approximation algorithms for the knapsack problem **10**

2.1 Greedy algorithm 10
2.2 Extended greedy algorithm 15
2.3 The H^2 heuristic 20
2.4 The H^ℓ heuristic 25
2.5 The FPTAS 30

Packing problems **35**

3 Approximation algorithms for bin packing **36**

3.1 Next Fit 36
3.2 First Fit 42
3.3 Best Fit 46
3.4 Worst Fit 51

Coloring problems **57**

4 Vertex coloring: limits of approximation **58**

Clique problems **63**

5 Maximum clique: limits of approximation **64**

Routing problems **68**

6 The TSP: the general case and the metric case **69**

Scheduling problems **74**

7 Approximation for $P\|C_{\max}$ **75**

Chapter 1

Principles of approximation algorithms

Definitions used in these exercises

The measures of the chapter Principles of approximation algorithms, for an algorithm $\mathscr{A}$ with value $\mathscr{A}(I)$ on an instance I of optimal value $z(I)$:

$$\text{performance ratio:} \quad \frac{\mathscr{A}(I)}{z(I)} \text{ (minimization)}, \qquad \frac{z(I)}{\mathscr{A}(I)} \text{ (maximization)}; \tag{A1}$$

$$\text{additive error:} \quad \mathscr{A}(I) - z(I) \text{ (minimization)}, \qquad z(I) - \mathscr{A}(I) \text{ (maximization)}; \tag{A2}$$

$$\text{relative error:} \quad \frac{\mathscr{A}(I) - z(I)}{z(I)} \text{ (minimization)}, \qquad \frac{z(I) - \mathscr{A}(I)}{z(I)} \text{ (maximization)}. \tag{A3}$$

The worst-case performance ratio is the supremum of the performance ratio (A1) over all the instances:

$$r(\mathscr{A}) = \sup_I \frac{\mathscr{A}(I)}{z(I)} \text{ (minimization)}, \qquad r(\mathscr{A}) = \sup_I \frac{z(I)}{\mathscr{A}(I)} \text{ (maximization)}. \tag{A4}$$

Exercise 1.1 — Performance ratios and errors (1)

A heuristic algorithm $\mathscr{A}$ for a **minimization** problem and a heuristic algorithm $\mathscr{B}$ for a **maximization** problem give the following values on some instances:

$\mathscr{A}$ (min)	I_1	I_2	I_3	I_4	I_5
$\mathscr{A}(I)$	12	9	15	20	7
$z(I)$	10	9	10	16	6

$\mathscr{B}$ (max)	I_1	I_2	I_3
$\mathscr{B}(I)$	30	25	16
$z(I)$	36	25	24

Questions

1. Compute the performance ratio, the additive error and the relative error of $\mathscr{A}$ on its five instances.
2. Do the same for $\mathscr{B}$ on its three instances.
3. Using only these data, which of the constants $\alpha = \frac{7}{5}$ and $\alpha = \frac{3}{2}$ can be a relative performance guarantee of $\mathscr{A}$? What would a proof that $\mathscr{A}$ is a $\frac{3}{2}$-approximation algorithm need?

4. Suppose that $\mathscr{A}$ is proved to be a $\frac{3}{2}$-approximation algorithm. Is the guarantee tight? Give a lower bound on $r(\mathscr{A})$ and on $r(\mathscr{B})$.

Solution

We use the definitions (A1)–(A4) of the box at the beginning of the chapter.

1. For a minimization problem the ratio is $\frac{\mathscr{A}(I)}{z(I)}$:

instance	$\mathscr{A}(I)$	$z(I)$	ratio	additive error	relative error
I_1	12	10	$\frac{6}{5}$	2	$\frac{1}{5}$
I_2	9	9	1	0	0
I_3	15	10	$\frac{3}{2}$	5	$\frac{1}{2}$
I_4	20	16	$\frac{5}{4}$	4	$\frac{1}{4}$
I_5	7	6	$\frac{7}{6}$	1	$\frac{1}{6}$

2. For a maximization problem the ratio is $\frac{z(I)}{\mathscr{B}(I)}$:

instance	$\mathscr{B}(I)$	$z(I)$	ratio	additive error	relative error
I_1	30	36	$\frac{6}{5}$	6	$\frac{1}{6}$
I_2	25	25	1	0	0
I_3	16	24	$\frac{3}{2}$	8	$\frac{1}{3}$

   All the ratios are at least 1, as they must be for feasible solutions.

3. On I_3 the ratio is $\frac{3}{2} > \frac{7}{5}$: $\alpha = \frac{7}{5}$ is not a relative performance guarantee. The data do not exclude $\alpha = \frac{3}{2}$, since every ratio is at most $\frac{3}{2}$. A proof must show $\mathscr{A}(I) \leq \frac{3}{2}\, z(I)$ for *every* instance I: finitely many examples can only disprove a guarantee.

4. Yes: the instance I_3 attains the ratio $\frac{3}{2}$, so no smaller constant is a guarantee, and $r(\mathscr{A}) = \frac{3}{2}$. Without a proof, the data give only lower bounds: $r(\mathscr{A}) \geq \frac{3}{2}$ and $r(\mathscr{B}) \geq \frac{3}{2}$ (instance I_3 of $\mathscr{B}$).

Exercise 1.2 — A tight guarantee for a simple algorithm (1)

For $P\|C_{\max}$ consider the algorithm ONE, which assigns all the jobs to machine 1, so that $\mathrm{ONE}(I) = \sum_{j \in \mathscr{J}} t_j$. Every instance satisfies $z(I) \geq \frac{\sum_j t_j}{m}$, since the m loads sum to $\sum_j t_j$. Consider the instances

I_1: $m = 2$ and $t = (7, 5, 4, 3, 3)$;

I_2: $m = 3$ and $t = (9, 2, 2, 2, 1)$;

I_3: $m = 3$ and $t = (5, 5, 4, 4, 3, 3, 3)$;

Questions

1. Compute ONE(I), $z(I)$ (by enumeration) and the performance ratio on the three instances.
2. Prove that ONE is an m-approximation algorithm for every fixed number m of machines.
3. Build, for every $m \geq 2$, an instance on which the ratio is exactly m, and check it with $m = 4$. What can you conclude?
4. Does ONE have a finite additive performance guarantee? Use the instance I_1 with all the processing times multiplied by 10.

Solution

We use the definitions (A1)–(A4) of the box at the beginning of the chapter.

1.

instance	m	ONE(I)	$z(I)$	ratio	guarantee m
I_1	2	22	11	2	2
I_2	3	16	9	$\frac{16}{9}$	3
I_3	3	27	9	3	3

The ratio never exceeds m, and it is equal to m on I_1 and I_3.

2. For every instance, $z(I) \geq \frac{\sum_j t_j}{m}$ gives

$$\frac{\text{ONE}(I)}{z(I)} = \frac{\sum_j t_j}{z(I)} \leq \frac{\sum_j t_j}{\frac{\sum_j t_j}{m}} = m.$$

3. Take $n = 2m$ jobs of time 1. Two jobs per machine give $z(I) = 2$, while ONE(I) $= 2m$: the ratio is m. With $m = 4$: ONE(I) $= 8$, $z(I) = 2$ (enumeration) and the ratio is 4. The guarantee m is tight and $r(\text{ONE}) = m$.
4. Multiplying the times of I_1 by 10 gives $t = (70, 50, 40, 30, 30)$, with ONE(I) $= 220$ and $z(I) = 110$: the additive error goes from 11 to 110. Multiplying by any K multiplies the error by K, so no constant β bounds it: $e(\text{ONE}) = \infty$, while the ratio stays 2.

Exercise 1.3 — An algorithm without a guarantee (1)

For the BPP consider the algorithm SEP, which packs every item in a separate bin, so that SEP(I) $= n$. Consider the instance I_0 with $c = 10$ and weights $(6, 5, 4, 3, 2)$, and, for $k \in \mathbb{Z}_{\geq 2}$, the instances I_k with $c = k$ and $2k$ items of weight 1.

Questions

1. Compute SEP(I_0), $z(I_0)$, the performance ratio and the additive error on I_0.
2. Compute the performance ratio on I_k for $k = 3, 5, 10$ (use enumeration for $k = 3$), and its expression for every k.
3. Is SEP an α-approximation algorithm for some constant α? Give $r(\text{SEP})$ and $e(\text{SEP})$.

4. Using $z(I) \geq 2$, show that $\frac{\text{SEP}(I)}{z(I)} \leq \frac{n}{2}$. Why is $\frac{n}{2}$ not a relative performance guarantee?

Solution

We use the definitions (A1)–(A4) of the box at the beginning of the chapter.

1. $\text{SEP}(I_0) = 5$ and $z(I_0) = 2$ (patterns $\{1, 3\}$ and $\{2, 4, 5\}$). The ratio is $\frac{5}{2}$ and the additive error is $5 - 2 = 3$.

2. Two bins of capacity k hold the $2k$ unit items, and one bin is not enough, so $z(I_k) = 2$:

k	c	$\text{SEP}(I_k) = 2k$	$z(I_k)$	ratio
3	3	6	2	3
5	5	10	2	5
10	10	20	2	10

   In general the ratio is $\frac{2k}{2} = k$.

3. No: for every constant α, the instance I_k with $k > \alpha$ has ratio $k > \alpha$. Hence $r(\text{SEP}) = \infty$. The additive error on I_k is $2k - 2$, also unbounded: $e(\text{SEP}) = \infty$.

4. If $z(I) \geq 2$, then $\frac{\text{SEP}(I)}{z(I)} = \frac{n}{z(I)} \leq \frac{n}{2}$. The bound depends on the instance through n: a relative performance guarantee must be one constant α valid for all the instances.

Exercise 1.4 — Unbounded ratio of a greedy algorithm (1)

For the KP consider the algorithm GR: sort the items by non-increasing efficiency $\frac{p_j}{w_j}$ (ties: smallest index) and insert each item, in this order, if it fits in the residual capacity. Consider the instance I_0 with $c = 9$, $p = (10, 7, 6, 4)$, $w = (5, 4, 3, 3)$, and, for $M \in \mathbb{Z}_{\geq 3}$, the instances I_M with $c = M$ and two items: $p_1 = 2$, $w_1 = 1$ and $p_2 = M$, $w_2 = M$.

Questions

1. Apply GR to I_0, compute $z(I_0)$ by enumeration, the performance ratio and the relative error.

2. Apply GR to I_M and compute $z(I_M)$, the performance ratio and the additive error for $M = 3, 10, 100$.

3. Is GR an α-approximation algorithm for some constant α? Is it a β-additive approximation algorithm?

4. Which property of the instances I_M makes GR fail?

Solution

We use the definitions (A1)–(A4) of the box at the beginning of the chapter.

1. The efficiencies are 2, $\frac{7}{4}$, 2, $\frac{4}{3}$: the order is $(1, 3, 2, 4)$. GR inserts items 1, 3 and gets

$\mathrm{GR}(I_0) = 16$. Enumeration gives $z(I_0) = 17$ (items $\{1, 2\}$). The ratio is $\frac{17}{16}$ and the relative error is $\frac{17-16}{17} = \frac{1}{17}$.

2. Item 1 has efficiency 2 and item 2 has efficiency 1: GR inserts item 1, and then item 2 does not fit in the residual capacity $M - 1$. Hence $\mathrm{GR}(I_M) = 2$, while item 2 alone gives $z(I_M) = M$:

M	$\mathrm{GR}(I_M)$	$z(I_M)$	ratio	additive error
3	2	3	$\frac{3}{2}$	1
10	2	10	5	8
100	2	100	50	98

3. The ratio $\frac{M}{2}$ grows without limit, so no constant α works and $r(\mathrm{GR}) = \infty$. The additive error $M - 2$ is also unbounded: $e(\mathrm{GR}) = \infty$.

4. The efficient item is small and blocks the large item, whose profit is almost all the optimal value: the order by efficiency ignores how much profit an item brings in absolute terms.

Exercise 1.5 — Approximation schemes (1)

An approximation scheme $\{\mathscr{A}_\varepsilon\}_{\varepsilon \in (0,1)}$ is run on some instances. The table gives the type of problem, the accuracy parameter ε, the optimal value and the value returned:

case	problem	ε	$z(I)$	$\mathscr{A}_\varepsilon(I)$
(1)	minimization	$\frac{1}{10}$	100	108
(2)	minimization	$\frac{1}{10}$	60	67
(3)	maximization	$\frac{1}{5}$	50	41
(4)	maximization	$\frac{1}{5}$	50	38

Questions

1. For each case, compute the relative error and decide whether the value is compatible with the definition of an approximation scheme.

2. Translate $\varepsilon = \frac{1}{4}$ into a relative performance guarantee, for a minimization and for a maximization problem. What is the accuracy?

3. A scheme for a maximization problem must be a $\frac{5}{4}$-approximation algorithm. Which value of ε is needed?

4. Classify the following running times of an approximation scheme on instances of size n: $O(n^{1/\varepsilon})$, $O(n^3/\varepsilon)$, $O(n\,2^{1/\varepsilon})$, $O(n^2/\varepsilon^2)$, $O(2^n/\varepsilon)$.

Solution

1. (1) $\frac{108-100}{100} = \frac{2}{25} \leq \frac{1}{10}$: compatible.
   (2) $\frac{67-60}{60} = \frac{7}{60} > \frac{1}{10}$: not compatible, the scheme is wrong on this instance.
   (3) $\frac{50-41}{50} = \frac{9}{50} \leq \frac{1}{5}$: compatible.
   (4) $\frac{50-38}{50} = \frac{6}{25} > \frac{1}{5}$: not compatible, the scheme is wrong on this instance.

2. For a minimization problem $\mathscr{A}_\varepsilon(I) \leq (1+\varepsilon)\, z(I)$, so the guarantee is $1 + \frac{1}{4} = \frac{5}{4}$. For a maximization problem $\mathscr{A}_\varepsilon(I) \geq (1-\varepsilon)\, z(I)$, so the guarantee is $\frac{1}{1-\varepsilon} = \frac{4}{3}$. The accuracy is $\frac{1}{\varepsilon} = 4$.

3. We need $\frac{1}{1-\varepsilon} \leq \frac{5}{4}$, that is $1 - \varepsilon \geq \frac{4}{5}$ and $\varepsilon \leq \frac{1}{5}$: any $\varepsilon \leq \frac{1}{5}$ works, and $\varepsilon = \frac{1}{5}$ is the largest one.

4. - $O(n^{1/\varepsilon})$: PTAS, not FPTAS.
   - $O(n^3/\varepsilon)$: FPTAS.
   - $O(n\, 2^{1/\varepsilon})$: PTAS, not FPTAS.
   - $O(n^2/\varepsilon^2)$: FPTAS.
   - $O(2^n/\varepsilon)$: not a PTAS.

   A PTAS needs a polynomial in n for every fixed ε; an FPTAS needs a polynomial in both n and $\frac{1}{\varepsilon}$. The terms $n^{1/\varepsilon}$ and $2^{1/\varepsilon}$ are not polynomial in $\frac{1}{\varepsilon}$, and 2^n is not polynomial in n.

Knapsack problems

Chapter 2

Approximation algorithms for the knapsack problem

2.1 Greedy algorithm

Exercise 2.1 — Greedy 1

We consider a knapsack of capacity $c = 15$ and a set $\mathscr{I} = \{1, 2, 3, 4, 5\}$ of 5 items ($n = 5$), with the following profits and weights:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
p_i	12	6	6	8	6
w_i	3	2	3	9	7

Questions

1. Verify that the items are sorted by non-increasing efficiency, and determine the split item s and the residual capacity rc.
2. Apply the greedy algorithm (GR) to this instance. Show the execution step by step, indicating at each iteration whether the item is selected and the resulting residual capacity. State the solution S returned by GR and $GR(I)$.
3. Compute the upper bound $U = \sum_{i=1}^{s-1} p_i + p_s \frac{rc}{w_s}$ on $z(I)$ given by the LP relaxation, and deduce an upper bound on the performance ratio $\frac{z(I)}{GR(I)}$.
4. Determine whether GR finds an optimal solution on this instance, either by showing that $GR(I)$ attains the upper bound $\lfloor U \rfloor$, or by exhibiting a feasible solution of larger profit.

Solution

1. The efficiencies of the items are

$$\frac{p_1}{w_1} = \frac{12}{3} = 4, \quad \frac{p_2}{w_2} = \frac{6}{2} = 3, \quad \frac{p_3}{w_3} = \frac{6}{3} = 2, \quad \frac{p_4}{w_4} = \frac{8}{9} \approx 0.89, \quad \frac{p_5}{w_5} = \frac{6}{7} \approx 0.86$$

   hence the items are sorted by non-increasing efficiency. Since

$$\sum_{i=1}^{3} w_i = 3 + 2 + 3 = 8 \leq 15 \qquad \text{and} \qquad \sum_{i=1}^{4} w_i = 3 + 2 + 3 + 9 = 17 > 15,$$

   the split item is $s = 4$ and the residual capacity is $rc = 15 - 8 = 7$.

2. Initially $rc := c = 15$. The execution of GR proceeds as follows:

   (a) Item 1 (weight $w_1 = 3$): $w_1 = 3 \leq rc = 15$, so the item is selected and $rc \leftarrow 15 - 3 = 12$.

   (b) Item 2 (weight $w_2 = 2$): $w_2 = 2 \leq rc = 12$, so the item is selected and $rc \leftarrow 12 - 2 = 10$.

   (c) Item 3 (weight $w_3 = 3$): $w_3 = 3 \leq rc = 10$, so the item is selected and $rc \leftarrow 10 - 3 = 7$.

   (d) Item 4 (weight $w_4 = 9$): $w_4 = 9 > rc = 7$, so the item is not selected.

   (e) Item 5 (weight $w_5 = 7$): $w_5 = 7 \leq rc = 7$, so the item is selected and $rc \leftarrow 7 - 7 = 0$.

   Hence GR returns the solution $S = \{1, 2, 3, 5\}$ with

$$GR(I) = 12 + 6 + 6 + 6 = 30.$$

   GR selects all the items preceding the split item, does not select the split item 4, and also selects item 5, which follows it.

3. The LP relaxation gives

$$z(I) \leq U = 12 + 6 + 6 + 8 \cdot \frac{7}{9} = \frac{272}{9} \approx 30.22,$$

   and since $z(I)$ is integer, $z(I) \leq \lfloor U \rfloor = 30$. Hence

$$\frac{z(I)}{GR(I)} \leq \frac{30}{30} = 1.$$

4. Since $GR(I) = 30 = \lfloor U \rfloor$, no feasible solution has a larger profit: GR finds an **optimal** solution on this instance, and $z(I) = 30$.

Exercise 2.2 — Greedy 2

We consider a knapsack of capacity $c = 13$ and a set $\mathscr{I} = \{1, 2, 3, 4, 5, 6\}$ of 6 items ($n = 6$), with the following profits and weights:

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$
p_i	10	6	3	7	11	2
w_i	1	1	1	4	8	3

Questions

1. Verify that the items are sorted by non-increasing efficiency, and determine the split item s and the residual capacity rc.
2. Apply the greedy algorithm (GR) to this instance. Show the execution step by step, indicating at each iteration whether the item is selected and the resulting residual capacity. State the solution S returned by GR and $GR(I)$.
3. Compute the upper bound $U = \sum_{i=1}^{s-1} p_i + p_s \frac{rc}{w_s}$ on $z(I)$ given by the LP relaxation, and deduce an upper bound on the performance ratio $\frac{z(I)}{GR(I)}$.
4. Determine whether GR finds an optimal solution on this instance, either by showing that $GR(I)$ attains the upper bound $\lfloor U \rfloor$, or by exhibiting a feasible solution of larger profit.

Solution

1. The efficiencies of the items are

$$\frac{p_1}{w_1} = \frac{10}{1} = 10, \quad \frac{p_2}{w_2} = \frac{6}{1} = 6, \quad \frac{p_3}{w_3} = \frac{3}{1} = 3, \quad \frac{p_4}{w_4} = \frac{7}{4} = 1.75, \quad \frac{p_5}{w_5} = \frac{11}{8} \approx 1.38, \quad \frac{p_6}{w_6} = \frac{2}{3} \approx 0.67,$$

   hence the items are sorted by non-increasing efficiency. Since

$$\sum_{i=1}^{4} w_i = 1 + 1 + 1 + 4 = 7 \leq 13 \qquad \text{and} \qquad \sum_{i=1}^{5} w_i = 1 + 1 + 1 + 4 + 8 = 15 > 13,$$

   the split item is $s = 5$ and the residual capacity is $rc = 13 - 7 = 6$.

2. Initially $rc := c = 13$. The execution of GR proceeds as follows:
   (a) Item 1 (weight $w_1 = 1$): $w_1 = 1 \leq rc = 13$, so the item is selected and $rc \leftarrow 13 - 1 = 12$.
   (b) Item 2 (weight $w_2 = 1$): $w_2 = 1 \leq rc = 12$, so the item is selected and $rc \leftarrow 12 - 1 = 11$.

(c) Item 3 (weight $w_3 = 1$): $w_3 = 1 \leq rc = 11$, so the item is selected and $rc \leftarrow 11 - 1 = 10$.

(d) Item 4 (weight $w_4 = 4$): $w_4 = 4 \leq rc = 10$, so the item is selected and $rc \leftarrow 10 - 4 = 6$.

(e) Item 5 (weight $w_5 = 8$): $w_5 = 8 > rc = 6$, so the item is not selected.

(f) Item 6 (weight $w_6 = 3$): $w_6 = 3 \leq rc = 6$, so the item is selected and $rc \leftarrow 6 - 3 = 3$.

Hence GR returns the solution $S = \{1, 2, 3, 4, 6\}$ with

$$GR(I) = 10 + 6 + 3 + 7 + 2 = 28.$$

GR selects all the items preceding the split item, does not select the split item 5, and also selects item 6, which follows it.

3. The LP relaxation gives

$$z(I) \leq U = 10 + 6 + 3 + 7 + 11 \cdot \frac{6}{8} = \frac{137}{4} \approx 34.25,$$

and since $z(I)$ is integer, $z(I) \leq \lfloor U \rfloor = 34$. Hence

$$\frac{z(I)}{GR(I)} \leq \frac{34}{28} = \frac{17}{14} \approx 1.21.$$

4. The solution $\tilde{S} = \{1, 2, 3, 5\}$, of weight $1 + 1 + 1 + 8 = 11 \leq 13$ and profit $z(I) = 10 + 6 + 3 + 11 = 30$ is feasible and has a larger profit than $GR(I) = 28$. Hence GR does **not** find an optimal solution on this instance; the solution $\tilde{S}$ is optimal (this can be checked by enumerating the feasible solutions), and the performance ratio is

$$\frac{z(I)}{GR(I)} = \frac{30}{28} = \frac{15}{14} \approx 1.07.$$

The item of maximum profit, item 5 with $p_5 = 11$, is not selected by GR: its low efficiency puts it late in the order, when the residual capacity is no longer sufficient.

Exercise 2.3 — Greedy 3

We consider a knapsack of capacity $c = 12$ and a set $\mathscr{I} = \{1, 2, 3, 4, 5\}$ of 5 items ($n = 5$), with the following profits and weights:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
p_i	8	7	18	14	10
w_i	3	3	9	8	7

Questions

1. Verify that the items are sorted by non-increasing efficiency, and determine the split item s and the residual capacity rc.
2. Apply the greedy algorithm (GR) to this instance. Show the execution step by step, indicating at each iteration whether the item is selected and the resulting residual capacity. State the solution S returned by GR and $GR(I)$.
3. Compute the upper bound $U = \sum_{i=1}^{s-1} p_i + p_s \frac{rc}{w_s}$ on $z(I)$ given by the LP relaxation, and deduce an upper bound on the performance ratio $\frac{z(I)}{GR(I)}$.
4. Determine whether GR finds an optimal solution on this instance, either by showing that $GR(I)$ attains the upper bound $\lfloor U \rfloor$, or by exhibiting a feasible solution of larger profit.

Solution

1. The efficiencies of the items are

$$\frac{p_1}{w_1} = \frac{8}{3} \approx 2.67, \quad \frac{p_2}{w_2} = \frac{7}{3} \approx 2.33, \quad \frac{p_3}{w_3} = \frac{18}{9} = 2, \quad \frac{p_4}{w_4} = \frac{14}{8} = 1.75, \quad \frac{p_5}{w_5} = \frac{10}{7} \approx 1.43,$$

   hence the items are sorted by non-increasing efficiency. Since

$$\sum_{i=1}^{2} w_i = 3 + 3 = 6 \leq 12 \qquad \text{and} \qquad \sum_{i=1}^{3} w_i = 3 + 3 + 9 = 15 > 12,$$

   the split item is $s = 3$ and the residual capacity is $rc = 12 - 6 = 6$.

2. Initially $rc := c = 12$. The execution of GR proceeds as follows:

   (a) Item 1 (weight $w_1 = 3$): $w_1 = 3 \leq rc = 12$, so the item is selected and $rc \leftarrow 12 - 3 = 9$.

   (b) Item 2 (weight $w_2 = 3$): $w_2 = 3 \leq rc = 9$, so the item is selected and $rc \leftarrow 9 - 3 = 6$.

   (c) Item 3 (weight $w_3 = 9$): $w_3 = 9 > rc = 6$, so the item is not selected.

   (d) Item 4 (weight $w_4 = 8$): $w_4 = 8 > rc = 6$, so the item is not selected.

   (e) Item 5 (weight $w_5 = 7$): $w_5 = 7 > rc = 6$, so the item is not selected.

   Hence GR returns the solution $S = \{1, 2\}$ with

$$GR(I) = 8 + 7 = 15.$$

   Here GR selects exactly the items preceding the split item.

3. The LP relaxation gives

$$z(I) \leq U = 8 + 7 + 18 \cdot \frac{6}{9} = 27,$$

and since $z(I)$ is integer, $z(I) \leq \lfloor U \rfloor = 27$. Hence

$$\frac{z(I)}{GR(I)} \leq \frac{27}{15} = \frac{9}{5} = 1.8.$$

4. The solution $\tilde{S} = \{1, 3\}$, of weight $3 + 9 = 12 \leq 12$ and profit $z(I) = 8 + 18 = 26$ is feasible and has a larger profit than $GR(I) = 15$. Hence GR does **not** find an optimal solution on this instance; the solution $\tilde{S}$ is optimal (this can be checked by enumerating the feasible solutions), and the performance ratio is

$$\frac{z(I)}{GR(I)} = \frac{26}{15} \approx 1.73.$$

The item of maximum profit, item 3 with $p_3 = 18$, is not selected by GR: its low efficiency puts it late in the order, when the residual capacity is no longer sufficient.

2.2 Extended greedy algorithm

Exercise 2.4 — Ext-Greedy 1

We consider a knapsack of capacity $c = 22$ and a set $\mathscr{I} = \{1, 2, 3, 4, 5\}$ of 5 items ($n = 5$), with the following profits and weights:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
p_i	11	5	3	4	3
w_i	8	4	3	9	8

Questions

1. Apply the greedy algorithm (GR) to this instance step by step, and state $GR(I)$.
2. Determine the maximum profit $p_{\max}$ and the best single-item feasible solution $\bar{S}$. State the solution returned by the Ext-Greedy algorithm (GR^e) and $GR^e(I)$.
3. Determine the split item s and verify the two inequalities $\sum_{i=1}^{s-1} p_i \leq GR^e(I)$ and $p_s \leq GR^e(I)$ used in the proof of the performance guarantee of GR^e. Deduce an upper bound on $z(I)$.
4. Determine whether GR^e finds an optimal solution on this instance, and compute the performance ratio $\frac{z(I)}{GR^e(I)}$, comparing it with the relative performance guarantee 2 of GR^e.

Solution

1. The efficiencies of the items are

$$\frac{p_1}{w_1} = \frac{11}{8} \approx 1.38, \quad \frac{p_2}{w_2} = \frac{5}{4} = 1.25, \quad \frac{p_3}{w_3} = \frac{3}{3} = 1, \quad \frac{p_4}{w_4} = \frac{4}{9} \approx 0.44, \quad \frac{p_5}{w_5} = \frac{3}{8} \approx 0.38$$

   so the items are sorted by non-increasing efficiency. Initially $rc := c = 22$, and GR proceeds as follows:

   (a) Item 1 (weight $w_1 = 8$): $w_1 = 8 \leq rc = 22$, so the item is selected and $rc \leftarrow 22 - 8 = 14$.

   (b) Item 2 (weight $w_2 = 4$): $w_2 = 4 \leq rc = 14$, so the item is selected and $rc \leftarrow 14 - 4 = 10$.

   (c) Item 3 (weight $w_3 = 3$): $w_3 = 3 \leq rc = 10$, so the item is selected and $rc \leftarrow 10 - 3 = 7$.

   (d) Item 4 (weight $w_4 = 9$): $w_4 = 9 > rc = 7$, so the item is not selected.

   (e) Item 5 (weight $w_5 = 8$): $w_5 = 8 > rc = 7$, so the item is not selected.

   Hence GR returns $S = \{1, 2, 3\}$ and $GR(I) = 11 + 5 + 3 = 19$.

2. The maximum profit is $p_{\max} = p_1 = 11$, and item 1 fits into the knapsack ($w_1 = 8 \leq 22$), so $\bar{S} = \{1\}$. Since $GR(I) = 19 \geq p_{\max} = 11$, GR^e keeps the greedy solution:

$$GR^e(I) = GR(I) = 19.$$

3. Since $\sum_{i=1}^{3} w_i = 15 \leq 22$ and $\sum_{i=1}^{4} w_i = 24 > 22$, the split item is $s = 4$. Then

$$\sum_{i=1}^{s-1} p_i = 11 + 5 + 3 = 19 \leq GR^e(I) = 19 \qquad \text{and} \qquad p_s = p_4 = 4 \leq GR^e(I) = 19,$$

and, as in the proof of the performance guarantee,

$$z(I) \;\leq\; \sum_{i=1}^{s-1} p_i + p_s \;=\; 23 \;\leq\; 2\,GR^e(I) = 38.$$

4. The solution $\tilde{S} = \{1, 2, 4\}$, of weight $8 + 4 + 9 = 21 \leq 22$ and profit $z(I) = 11 + 5 + 4 = 20$ is feasible and better than the one returned by GR^e, so GR^e does **not** find an optimal solution; $\tilde{S}$ is optimal (this can be checked by enumerating the feasible solutions). The performance ratio is

$$\frac{z(I)}{GR^e(I)} = \frac{20}{19} \approx 1.05 \;\leq\; 2,$$

as guaranteed.

Exercise 2.5 — Ext-Greedy 2

We consider a knapsack of capacity $c = 8$ and a set $\mathscr{I} = \{1, 2, 3, 4, 5\}$ of 5 items ($n = 5$), with the following profits and weights:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
p_i	16	19	18	14	8
w_i	5	8	8	7	6

Questions

1. Apply the greedy algorithm (GR) to this instance step by step, and state $GR(I)$.
2. Determine the maximum profit $p_{\max}$ and the best single-item feasible solution $\bar{S}$. State the solution returned by the Ext-Greedy algorithm (GR^e) and $GR^e(I)$.
3. Determine the split item s and verify the two inequalities $\sum_{i=1}^{s-1} p_i \leq GR^e(I)$ and $p_s \leq GR^e(I)$ used in the proof of the performance guarantee of GR^e. Deduce an upper bound on $z(I)$.

4. Determine whether GR^e finds an optimal solution on this instance, and compute the performance ratio $\frac{z(I)}{GR^e(I)}$, comparing it with the relative performance guarantee 2 of GR^e.

Solution

1. The efficiencies of the items are

$$\frac{p_1}{w_1} = \frac{16}{5} = 3.2, \quad \frac{p_2}{w_2} = \frac{19}{8} \approx 2.38, \quad \frac{p_3}{w_3} = \frac{18}{8} = 2.25, \quad \frac{p_4}{w_4} = \frac{14}{7} = 2, \quad \frac{p_5}{w_5} = \frac{8}{6} \approx 1.33$$

   so the items are sorted by non-increasing efficiency. Initially $rc := c = 8$, and GR proceeds as follows:

   (a) Item 1 (weight $w_1 = 5$): $w_1 = 5 \leq rc = 8$, so the item is selected and $rc \leftarrow 8 - 5 = 3$.

   (b) Item 2 (weight $w_2 = 8$): $w_2 = 8 > rc = 3$, so the item is not selected.

   (c) Item 3 (weight $w_3 = 8$): $w_3 = 8 > rc = 3$, so the item is not selected.

   (d) Item 4 (weight $w_4 = 7$): $w_4 = 7 > rc = 3$, so the item is not selected.

   (e) Item 5 (weight $w_5 = 6$): $w_5 = 6 > rc = 3$, so the item is not selected.

   Hence GR returns $S = \{1\}$ and $GR(I) = 16$.

2. The maximum profit is $p_{\max} = p_2 = 19$, and item 2 fits into the knapsack ($w_2 = 8 \leq 8$), so $\bar{S} = \{2\}$. Since $GR(I) = 16 < p_{\max} = 19$, GR^e replaces the greedy solution with $\bar{S}$:

$$GR^e(I) = p_{\max} = 19.$$

3. Since $\sum_{i=1}^{1} w_i = 5 \leq 8$ and $\sum_{i=1}^{2} w_i = 13 > 8$, the split item is $s = 2$. Then

$$\sum_{i=1}^{s-1} p_i = 16 \leq GR^e(I) = 19 \qquad \text{and} \qquad p_s = p_2 = 19 \leq GR^e(I) = 19,$$

   and, as in the proof of the performance guarantee,

$$z(I) \leq \sum_{i=1}^{s-1} p_i + p_s = 35 \leq 2\,GR^e(I) = 38.$$

4. The solution returned by GR^e is **optimal**: $\tilde{S} = \{2\}$, of weight $8 \leq 8$ and profit $z(I) = 19$. The performance ratio is $\frac{z(I)}{GR^e(I)} = 1$.

Exercise 2.6 — Ext-Greedy 3

We consider a knapsack of capacity $c = 16$ and a set $\mathscr{I} = \{1, 2, 3\}$ of 3 items ($n = 3$), with the following profits and weights:

	$i = 1$	$i = 2$	$i = 3$
p_i	2	8	8
w_i	1	8	8

Questions

1. Apply the greedy algorithm (GR) to this instance step by step, and state $GR(I)$.
2. Determine the maximum profit $p_{\max}$ and the best single-item feasible solution $\bar{S}$. State the solution returned by the Ext-Greedy algorithm (GR^e) and $GR^e(I)$.
3. Determine the split item s and verify the two inequalities $\sum_{i=1}^{s-1} p_i \leq GR^e(I)$ and $p_s \leq GR^e(I)$ used in the proof of the performance guarantee of GR^e. Deduce an upper bound on $z(I)$.
4. Determine whether GR^e finds an optimal solution on this instance, and compute the performance ratio $\frac{z(I)}{GR^e(I)}$, comparing it with the relative performance guarantee 2 of GR^e.

Solution

1. The efficiencies of the items are

$$\frac{p_1}{w_1} = \frac{2}{1} = 2, \quad \frac{p_2}{w_2} = \frac{8}{8} = 1, \quad \frac{p_3}{w_3} = \frac{8}{8} = 1,$$

so the items are sorted by non-increasing efficiency. Initially $rc := c = 16$, and GR proceeds as follows:

   (a) Item 1 (weight $w_1 = 1$): $w_1 = 1 \leq rc = 16$, so the item is selected and $rc \leftarrow 16 - 1 = 15$.

   (b) Item 2 (weight $w_2 = 8$): $w_2 = 8 \leq rc = 15$, so the item is selected and $rc \leftarrow 15 - 8 = 7$.

   (c) Item 3 (weight $w_3 = 8$): $w_3 = 8 > rc = 7$, so the item is not selected.

   Hence GR returns $S = \{1, 2\}$ and $GR(I) = 2 + 8 = 10$.

2. The maximum profit is $p_{\max} = p_2 = 8$, and item 2 fits into the knapsack ($w_2 = 8 \leq 16$), so $\bar{S} = \{2\}$. Since $GR(I) = 10 \geq p_{\max} = 8$, GR^e keeps the greedy solution:

$$GR^e(I) = GR(I) = 10.$$

3. Since $\sum_{i=1}^{2} w_i = 9 \leq 16$ and $\sum_{i=1}^{3} w_i = 17 > 16$, the split item is $s = 3$. Then

$$\sum_{i=1}^{s-1} p_i = 2 + 8 = 10 \leq GR^e(I) = 10 \qquad \text{and} \qquad p_s = p_3 = 8 \leq GR^e(I) = 10,$$

and, as in the proof of the performance guarantee,

$$z(I) \;\leq\; \sum_{i=1}^{s-1} p_i + p_s \;=\; 18 \;\leq\; 2\,GR^e(I) = 20.$$

4. The solution $\tilde{S} = \{2, 3\}$, of weight $8 + 8 = 16 \leq 16$ and profit $z(I) = 8 + 8 = 16$ is feasible and better than the one returned by GR^e, so GR^e does **not** find an optimal solution; $\tilde{S}$ is optimal (this can be checked by enumerating the feasible solutions). The performance ratio is

$$\frac{z(I)}{GR^e(I)} = \frac{16}{10} = \frac{8}{5} = 1.6 \;\leq\; 2,$$

as guaranteed.

Remark. This instance belongs to the family of instances used to show that the relative performance guarantee 2 of GR^e is tight, with $\vartheta = 8$: there the ratio is $\frac{2\vartheta}{\vartheta+2}$, and here $\frac{16}{10} = 1.6$; it tends to 2 as ϑ grows.

2.3 The H^2 heuristic

Exercise 2.7 — H^2 1

We consider a knapsack of capacity $c = 11$ and a set $\mathscr{I} = \{1, 2, 3, 4, 5\}$ of 5 items ($n = 5$), with the following profits and weights:

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
p_i	8	9	2	3	2
w_i	5	8	3	7	9

Questions

1. Verify that the items are sorted by non-increasing efficiency, and determine the set $\mathscr{P}$ of the feasible pairs of items.
2. Apply the H^2 algorithm to this instance: for each feasible pair (i, k) compute the residual capacity rc, the residual set $\mathscr{I}(i, k)$, the value of GR^e on the residual instance and the value $lb(i, k)$.

3. State the solution returned by H^2 and $H^2(I)$.
4. Compare $H^2(I)$ with $GR^e(I)$ and with $z(I)$, and compute the performance ratio $\frac{z(I)}{H^2(I)}$, comparing it with the relative performance guarantee $\frac{4}{3}$ of H^2.

Solution

1. The efficiencies of the items are

$$\frac{p_1}{w_1} = \frac{8}{5} = 1.6, \quad \frac{p_2}{w_2} = \frac{9}{8} \approx 1.12, \quad \frac{p_3}{w_3} = \frac{2}{3} \approx 0.67, \quad \frac{p_4}{w_4} = \frac{3}{7} \approx 0.43, \quad \frac{p_5}{w_5} = \frac{2}{9} \approx 0.22$$

   so the items are sorted by non-increasing efficiency. A pair (i, k) with $i < k$ is feasible if $w_i + w_k \leq c = 11$, hence

$$\mathscr{P} = \{(1,3), (2,3), (3,4)\}.$$

2. The algorithm starts from the best single item, $lb = p_{\max} = p_2 = 9$. For each feasible pair, the residual set keeps the items $j \notin \{i, k\}$ with $p_j \leq \min\{p_i, p_k\}$ and $w_j \leq rc$:

pair	weight	rc	residual set	GR^e on the residual	value
$(1,2)$	$13 > 11$			not feasible	
$(1,3)$	8	3	$\varnothing$	0	10
$(1,4)$	$12 > 11$			not feasible	
$(1,5)$	$14 > 11$			not feasible	
$(2,3)$	11	0	$\varnothing$	0	11
$(2,4)$	$15 > 11$			not feasible	
$(2,5)$	$17 > 11$			not feasible	
$(3,4)$	10	1	$\varnothing$	0	5
$(3,5)$	$12 > 11$			not feasible	
$(4,5)$	$16 > 11$			not feasible	

3. The largest value is 11, obtained with the solution $\{2, 3\}$. Hence

$$H^2(I) = 11.$$

4. On this instance $GR^e(I) = 10$, so H^2 improves on GR^e. The solution returned by H^2 is **optimal**: $\tilde{S} = \{2, 3\}$, of weight $8 + 3 = 11 \leq 11$ and profit $z(I) = 9 + 2 = 11$, and the performance ratio is 1.

Exercise 2.8 — H^2 2

We consider a knapsack of capacity $c = 15$ and a set $\mathscr{I} = \{1, 2, 3, 4, 5\}$ of 5 items ($n = 5$),

with the following profits and weights:

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
p_i	14	9	5	10	15
w_i	1	1	2	5	8

Questions

1. Verify that the items are sorted by non-increasing efficiency, and determine the set $\mathscr{P}$ of the feasible pairs of items.
2. Apply the H^2 algorithm to this instance: for each feasible pair (i,k) compute the residual capacity rc, the residual set $\mathscr{I}(i,k)$, the value of GR^e on the residual instance and the value $lb(i,k)$.
3. State the solution returned by H^2 and $H^2(I)$.
4. Compare $H^2(I)$ with $GR^e(I)$ and with $z(I)$, and compute the performance ratio $\frac{z(I)}{H^2(I)}$, comparing it with the relative performance guarantee $\frac{4}{3}$ of H^2.

Solution

1. The efficiencies of the items are

$$\frac{p_1}{w_1}=\frac{14}{1}=14,\quad \frac{p_2}{w_2}=\frac{9}{1}=9,\quad \frac{p_3}{w_3}=\frac{5}{2}=2.5,\quad \frac{p_4}{w_4}=\frac{10}{5}=2,\quad \frac{p_5}{w_5}=\frac{15}{8}\approx 1.88$$

   so the items are sorted by non-increasing efficiency. A pair (i,k) with $i<k$ is feasible if $w_i+w_k\le c=15$, hence

$$\mathscr{P}=\big\{(1,2),(1,3),(1,4),(1,5),(2,3),(2,4),(2,5),(3,4),(3,5),(4,5)\big\}.$$

2. The algorithm starts from the best single item, $lb=p_{\max}=p_5=15$. For each feasible pair, the residual set keeps the items $j\notin\{i,k\}$ with $p_j\le\min\{p_i,p_k\}$ and $w_j\le rc$:

pair	weight	rc	residual set	GR^e on the residual	value
$(1,2)$	2	13	$\{3\}$	5	28
$(1,3)$	3	12	$\varnothing$	0	19
$(1,4)$	6	9	$\{2,3\}$	14	38
$(1,5)$	9	6	$\{2,3,4\}$	14	43
$(2,3)$	3	12	$\varnothing$	0	14
$(2,4)$	6	9	$\{3\}$	5	24
$(2,5)$	9	6	$\{3\}$	5	29
$(3,4)$	7	8	$\varnothing$	0	15
$(3,5)$	10	5	$\varnothing$	0	20
$(4,5)$	13	2	$\{2,3\}$	9	34

3. The largest value is 43, obtained with the solution $\{1, 2, 3, 5\}$. Hence

$$H^2(I) = 43.$$

4. On this instance $GR^e(I) = 38$, so H^2 improves on GR^e. The solution $\tilde{S} = \{1, 2, 4, 5\}$, of weight $1+1+5+8 = 15 \leq 15$ and profit $z(I) = 14+9+10+15 = 48$ is better, so H^2 does **not** find an optimal solution; $\tilde{S}$ is optimal (this can be checked by enumerating the feasible solutions). The performance ratio is

$$\frac{z(I)}{H^2(I)} = \frac{48}{43} \approx 1.12 \leq \frac{4}{3} \approx 1.33,$$

as guaranteed.

Exercise 2.9 — H^2 3

We consider a knapsack of capacity $c = 10$ and a set $\mathscr{I} = \{1, 2, 3, 4, 5\}$ of 5 items ($n = 5$), with the following profits and weights:

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
p_i	13	5	8	5	7
w_i	1	1	3	2	5

Questions

1. Verify that the items are sorted by non-increasing efficiency, and determine the set $\mathscr{P}$ of the feasible pairs of items.
2. Apply the H^2 algorithm to this instance: for each feasible pair (i, k) compute the residual capacity rc, the residual set $\mathscr{I}(i, k)$, the value of GR^e on the residual instance and the value $lb(i, k)$.
3. State the solution returned by H^2 and $H^2(I)$.
4. Compare $H^2(I)$ with $GR^e(I)$ and with $z(I)$, and compute the performance ratio $\frac{z(I)}{H^2(I)}$, comparing it with the relative performance guarantee $\frac{4}{3}$ of H^2.

Solution

1. The efficiencies of the items are

$$\frac{p_1}{w_1} = \frac{13}{1} = 13, \quad \frac{p_2}{w_2} = \frac{5}{1} = 5, \quad \frac{p_3}{w_3} = \frac{8}{3} \approx 2.67, \quad \frac{p_4}{w_4} = \frac{5}{2} = 2.5, \quad \frac{p_5}{w_5} = \frac{7}{5} = 1.4,$$

so the items are sorted by non-increasing efficiency. A pair (i,k) with $i<k$ is feasible if $w_i + w_k \le c = 10$, hence

$$\mathscr{P} = \{(1,2),(1,3),(1,4),(1,5),(2,3),(2,4),(2,5),(3,4),(3,5),(4,5)\}.$$

2. The algorithm starts from the best single item, $lb = p_{\max} = p_1 = 13$. For each feasible pair, the residual set keeps the items $j \notin \{i,k\}$ with $p_j \le \min\{p_i, p_k\}$ and $w_j \le rc$:

pair	weight	rc	residual set	GR^e on the residual	value
$(1,2)$	2	8	$\{4\}$	5	23
$(1,3)$	4	6	$\{2,4,5\}$	10	31
$(1,4)$	3	7	$\{2\}$	5	23
$(1,5)$	6	4	$\{2,4\}$	10	30
$(2,3)$	4	6	$\{4\}$	5	18
$(2,4)$	3	7	$\varnothing$	0	10
$(2,5)$	6	4	$\{4\}$	5	17
$(3,4)$	5	5	$\{2\}$	5	18
$(3,5)$	8	2	$\{2,4\}$	5	20
$(4,5)$	7	3	$\{2\}$	5	17

3. The largest value is 31, obtained with the solution $\{1,2,3,4\}$. Hence

$$H^2(I) = 31.$$

4. On this instance $GR^e(I) = 31$, the same value as H^2. The solution $\tilde{S} = \{1,2,3,5\}$, of weight $1+1+3+5 = 10 \le 10$ and profit $z(I) = 13+5+8+7 = 33$ is better, so H^2 does **not** find an optimal solution; $\tilde{S}$ is optimal (this can be checked by enumerating the feasible solutions). The performance ratio is

$$\frac{z(I)}{H^2(I)} = \frac{33}{31} \approx 1.06 \;\le\; \frac{4}{3} \approx 1.33,$$

as guaranteed.

2.4 The H^ℓ heuristic

Exercise 2.10 — H^ℓ 1

We consider a knapsack of capacity $c = 16$ and a set $\mathscr{I} = \{1, 2, 3, 4, 5\}$ of 5 items ($n = 5$), with the following profits and weights:

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
p_i	14	13	13	10	11
w_i	1	2	3	3	9

Questions

1. We apply the H^ℓ algorithm with $\ell = 3$. Compute the best feasible solution made of at most $\ell - 1 = 2$ items.
2. For each feasible set L of exactly $\ell = 3$ items, compute the residual capacity rc, the residual set $\mathscr{I}(L)$, the value of GR^e on the residual instance and the value $lb(L)$.
3. State the solution returned by H^3 and $H^3(I)$.
4. Determine whether H^3 finds an optimal solution on this instance, and compute the performance ratio $\frac{z(I)}{H^3(I)}$, comparing it with the relative performance guarantee $\frac{\ell+2}{\ell+1} = \frac{5}{4}$.

Solution

1. The best feasible solution with at most 2 items is $\{1, 2\}$, of weight $1+2 = 3 \leq 16$ and profit $14 + 13 = 27$.
2. For each set L of 3 items, the residual set keeps the items $j \notin L$ with $p_j \leq \min\{p_i : i \in L\}$ and $w_j \leq rc$:

set L	weight	rc	residual set	GR^e on the residual	value
$\{1,2,3\}$	6	10	$\{4,5\}$	11	51
$\{1,2,4\}$	6	10	$\varnothing$	0	37
$\{1,2,5\}$	12	4	$\{4\}$	10	48
$\{1,3,4\}$	7	9	$\varnothing$	0	37
$\{1,3,5\}$	13	3	$\{4\}$	10	48
$\{1,4,5\}$	13	3	$\varnothing$	0	35
$\{2,3,4\}$	8	8	$\varnothing$	0	36
$\{2,3,5\}$	14	2	$\varnothing$	0	37
$\{2,4,5\}$	14	2	$\varnothing$	0	34
$\{3,4,5\}$	15	1	$\varnothing$	0	34

3. The largest value is 51, obtained with the solution $\{1, 2, 3, 5\}$. Hence

$$H^3(I) = 51.$$

4. The solution returned by H^3 is **optimal**: $\tilde{S} = \{1, 2, 3, 5\}$, of weight $1 + 2 + 3 + 9 = 15 \leq 16$ and profit $z(I) = 14 + 13 + 13 + 11 = 51$, and the performance ratio is 1. For comparison, on this instance $H^2(I) = 50$: guessing three items instead of two is enough to find the optimal solution.

Exercise 2.11 — H^ℓ 2

We consider a knapsack of capacity $c = 20$ and a set $\mathscr{I} = \{1, 2, 3, 4, 5, 6\}$ of 6 items ($n = 6$), with the following profits and weights:

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$
p_i	2	4	4	4	4	4
w_i	1	4	4	4	4	4

Questions

1. We apply the H^ℓ algorithm with $\ell = 3$. Compute the best feasible solution made of at most $\ell - 1 = 2$ items.
2. For each feasible set L of exactly $\ell = 3$ items, compute the residual capacity rc, the residual set $\mathscr{I}(L)$, the value of GR^e on the residual instance and the value $lb(L)$.
3. State the solution returned by H^3 and $H^3(I)$.
4. Determine whether H^3 finds an optimal solution on this instance, and compute the performance ratio $\frac{z(I)}{H^3(I)}$, comparing it with the relative performance guarantee $\frac{\ell+2}{\ell+1} = \frac{5}{4}$.

Solution

1. The best feasible solution with at most 2 items is $\{2, 3\}$, of weight $4 + 4 = 8 \leq 20$ and profit $4 + 4 = 8$.
2. For each set L of 3 items, the residual set keeps the items $j \notin L$ with $p_j \leq \min\{p_i : i \in L\}$ and $w_j \leq rc$:

set L	weight	rc	residual set	GR^e on the residual	value
$\{1,2,3\}$	9	11	$\varnothing$	0	10
$\{1,2,4\}$	9	11	$\varnothing$	0	10
$\{1,2,5\}$	9	11	$\varnothing$	0	10
$\{1,2,6\}$	9	11	$\varnothing$	0	10
$\{1,3,4\}$	9	11	$\varnothing$	0	10
$\{1,3,5\}$	9	11	$\varnothing$	0	10
$\{1,3,6\}$	9	11	$\varnothing$	0	10
$\{1,4,5\}$	9	11	$\varnothing$	0	10
$\{1,4,6\}$	9	11	$\varnothing$	0	10
$\{1,5,6\}$	9	11	$\varnothing$	0	10
$\{2,3,4\}$	12	8	$\{1,5,6\}$	6	18
$\{2,3,5\}$	12	8	$\{1,4,6\}$	6	18
$\{2,3,6\}$	12	8	$\{1,4,5\}$	6	18
$\{2,4,5\}$	12	8	$\{1,3,6\}$	6	18
$\{2,4,6\}$	12	8	$\{1,3,5\}$	6	18
$\{2,5,6\}$	12	8	$\{1,3,4\}$	6	18
$\{3,4,5\}$	12	8	$\{1,2,6\}$	6	18
$\{3,4,6\}$	12	8	$\{1,2,5\}$	6	18
$\{3,5,6\}$	12	8	$\{1,2,4\}$	6	18
$\{4,5,6\}$	12	8	$\{1,2,3\}$	6	18

3. The largest value is 18, obtained with the solution $\{1,2,3,4,5\}$. Hence

$$H^3(I) = 18.$$

4. The solution $\tilde{S} = \{2,3,4,5,6\}$, of weight $4+4+4+4+4 = 20 \leq 20$ and profit $z(I) = 4+4+4+4+4 = 20$ is better, so H^3 does **not** find an optimal solution; $\tilde{S}$ is optimal (this can be checked by enumerating the feasible solutions). The performance ratio is

$$\frac{z(I)}{H^3(I)} = \frac{20}{18} = \frac{10}{9} \approx 1.11 \;\leq\; \frac{5}{4} = 1.25,$$

as guaranteed. For comparison, on this instance $H^2(I) = 18$.

Remark. This instance belongs to the family of instances used to show that the relative performance guarantee $\frac{\ell+2}{\ell+1}$ of H^ℓ is tight, with $\ell = 3$ and $M = 4$: there $H^\ell(I_M) = (\ell+1)\,M + 2$ and $z(I_M) = (\ell+2)\,M$, here 18 and 20.

Exercise 2.12 — H^ℓ 3

We consider a knapsack of capacity $c = 18$ and a set $\mathscr{I} = \{1,2,3,4,5,6\}$ of 6 items ($n = 6$), with the following profits and weights:

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$	$i=6$
p_i	15	15	14	8	12	13
w_i	1	2	3	2	6	9

Questions

1. We apply the H^ℓ algorithm with $\ell = 3$. Compute the best feasible solution made of at most $\ell - 1 = 2$ items.
2. For each feasible set L of exactly $\ell = 3$ items, compute the residual capacity rc, the residual set $\mathscr{I}(L)$, the value of GR^e on the residual instance and the value $lb(L)$.
3. State the solution returned by H^3 and $H^3(I)$.
4. Determine whether H^3 finds an optimal solution on this instance, and compute the performance ratio $\frac{z(I)}{H^3(I)}$, comparing it with the relative performance guarantee $\frac{\ell+2}{\ell+1} = \frac{5}{4}$.

Solution

1. The best feasible solution with at most 2 items is $\{1, 2\}$, of weight $1+2 = 3 \leq 18$ and profit $15 + 15 = 30$.
2. For each set L of 3 items, the residual set keeps the items $j \notin L$ with $p_j \leq \min\{p_i : i \in L\}$ and $w_j \leq rc$:

set L	weight	rc	residual set	GR^e on the residual	value
$\{1,2,3\}$	6	12	$\{4,5,6\}$	20	64
$\{1,2,4\}$	5	13	$\varnothing$	0	38
$\{1,2,5\}$	9	9	$\{4\}$	8	50
$\{1,2,6\}$	12	6	$\{4,5\}$	12	55
$\{1,3,4\}$	6	12	$\varnothing$	0	37
$\{1,3,5\}$	10	8	$\{4\}$	8	49
$\{1,3,6\}$	13	5	$\{4\}$	8	50
$\{1,4,5\}$	9	9	$\varnothing$	0	35
$\{1,4,6\}$	12	6	$\varnothing$	0	36
$\{1,5,6\}$	16	2	$\{4\}$	8	48
$\{2,3,4\}$	7	11	$\varnothing$	0	37
$\{2,3,5\}$	11	7	$\{4\}$	8	49
$\{2,3,6\}$	14	4	$\{4\}$	8	50
$\{2,4,5\}$	10	8	$\varnothing$	0	35
$\{2,4,6\}$	13	5	$\varnothing$	0	36
$\{2,5,6\}$	17	1	$\varnothing$	0	40
$\{3,4,5\}$	11	7	$\varnothing$	0	34
$\{3,4,6\}$	14	4	$\varnothing$	0	35
$\{3,5,6\}$	18	0	$\varnothing$	0	39
$\{4,5,6\}$	17	1	$\varnothing$	0	33

3. The largest value is 64, obtained with the solution $\{1,2,3,4,5\}$. Hence
$$H^3(I) = 64.$$

4. The solution $\tilde{S} = \{1,2,3,4,6\}$, of weight $1+2+3+2+9 = 17 \leq 18$ and profit $z(I) = 15+15+14+8+13 = 65$ is better, so H^3 does **not** find an optimal solution; $\tilde{S}$ is optimal (this can be checked by enumerating the feasible solutions). The performance ratio is
$$\frac{z(I)}{H^3(I)} = \frac{65}{64} \approx 1.02 \leq \frac{5}{4} = 1.25,$$
as guaranteed. For comparison, on this instance $H^2(I) = 64$.

2.5 The FPTAS

Exercise 2.13 — Basic-FPTAS 1

We consider a knapsack of capacity $c = 13$ and a set $\mathscr{I} = \{1, 2, 3, 4\}$ of 4 items ($n = 4$), with the following profits and weights:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$
p_i	24	11	23	5
w_i	2	3	9	3

We apply the Basic-FPTAS algorithm with accuracy $\varepsilon = \frac{1}{2}$.

Questions

1. Compute the lower bound $GR^e(I)$ used by the Basic-FPTAS algorithm.
2. Compute the scaling factor and the scaled profits
$$K = \frac{\varepsilon\, GR^e(I)}{n} \qquad \text{and} \qquad p'_j = \left\lfloor \frac{p_j}{K} \right\rfloor \quad \text{for all } j \in \mathscr{I}.$$
3. Solve the scaled instance by dynamic programming by profits, i.e., compute the minimum weights $u(j, q)$ and find the largest profit q such that $u(4, q) \le c$.
4. State the solution returned by FP^ε, its profit $FP^\varepsilon(I)$ in the original instance, and compare it with $GR^e(I)$ and $z(I)$. Compute the performance ratio and compare it with the guarantee $\frac{1}{1-\varepsilon}$.

Solution

1. The items are sorted by non-increasing efficiency. GR selects the items $\{1, 2, 4\}$, with $GR(I) = 24 + 11 + 5 = 40$; the maximum profit is $p_{\max} = p_1 = 24$. Hence
$$GR^e(I) = 40.$$
2. The scaling factor is
$$K = \frac{\varepsilon\, GR^e(I)}{n} = \frac{\frac{1}{2} \cdot 40}{4} = 5,$$
and the scaled profits are
$$p'_1 = \left\lfloor \frac{24}{5} \right\rfloor = 4, \quad p'_2 = \left\lfloor \frac{11}{5} \right\rfloor = 2, \quad p'_3 = \left\lfloor \frac{23}{5} \right\rfloor = 4, \quad p'_4 = \left\lfloor \frac{5}{5} \right\rfloor = 1.$$
The total scaled profit is $P' = 4 + 2 + 4 + 1 = 11$.
3. The minimum weights $u(j, q)$ of a subset of $\{1, \ldots, j\}$ with scaled profit exactly q are (only the values of q that can be attained are reported; ∞ means that q cannot be attained with the first j items):

q	0	1	2	3	4	5	6	7	8	9	10
$u(0,q)$	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$u(1,q)$	0	∞	∞	∞	2	∞	∞	∞	∞	∞	∞
$u(2,q)$	0	∞	3	∞	2	∞	5	∞	∞	∞	∞
$u(3,q)$	0	∞	3	∞	2	∞	5	∞	11	∞	14
$u(4,q)$	0	3	3	6	2	5	5	8	11	14	14

q	11
$u(0,q)$	∞
$u(1,q)$	∞
$u(2,q)$	∞
$u(3,q)$	∞
$u(4,q)$	17

The largest q with $u(4,q) \leq c = 13$ is $q = 8$, of weight $u(4,8) = 11$, attained by the items $\{1,3\}$.

4. FP^{ε} returns $S = \{1,3\}$, of profit

$$FP^{\varepsilon}(I) = 24 + 23 = 47$$

in the original instance, better than $GR^{e}(I) = 40$. The solution is **optimal**: $\tilde{S} = \{1,3\}$, of weight $2+9 = 11 \leq 13$ and profit $z(I) = 24+23 = 47$, and the performance ratio is 1.

Exercise 2.14 — Basic-FPTAS 2

We consider a knapsack of capacity $c = 16$ and a set $\mathscr{I} = \{1,2,3,4\}$ of 4 items ($n = 4$), with the following profits and weights:

	$i=1$	$i=2$	$i=3$	$i=4$
p_i	14	12	14	18
w_i	3	4	5	7

We apply the Basic-FPTAS algorithm with accuracy $\varepsilon = \frac{1}{2}$.

Questions

1. Compute the lower bound $GR^{e}(I)$ used by the Basic-FPTAS algorithm.
2. Compute the scaling factor and the scaled profits

$$K = \frac{\varepsilon\, GR^{e}(I)}{n} \qquad \text{and} \qquad p'_j = \left\lfloor \frac{p_j}{K} \right\rfloor \quad \text{for all } j \in \mathscr{I}.$$

3. Solve the scaled instance by dynamic programming by profits, i.e., compute the minimum weights $u(j, q)$ and find the largest profit q such that $u(4, q) \leq c$.

4. State the solution returned by FP^{ε}, its profit $FP^{\varepsilon}(I)$ in the original instance, and compare it with $GR^{e}(I)$ and $z(I)$. Compute the performance ratio and compare it with the guarantee $\frac{1}{1-\varepsilon}$.

Solution

1. The items are sorted by non-increasing efficiency. GR selects the items $\{1, 2, 3\}$, with $GR(I) = 14 + 12 + 14 = 40$; the maximum profit is $p_{\max} = p_4 = 18$. Hence
$$GR^{e}(I) = 40.$$

2. The scaling factor is
$$K = \frac{\varepsilon\, GR^{e}(I)}{n} = \frac{\frac{1}{2} \cdot 40}{4} = 5,$$
and the scaled profits are
$$p'_1 = \left\lfloor \frac{14}{5} \right\rfloor = 2, \quad p'_2 = \left\lfloor \frac{12}{5} \right\rfloor = 2, \quad p'_3 = \left\lfloor \frac{14}{5} \right\rfloor = 2, \quad p'_4 = \left\lfloor \frac{18}{5} \right\rfloor = 3.$$
The total scaled profit is $P' = 2 + 2 + 2 + 3 = 9$.

3. The minimum weights $u(j, q)$ of a subset of $\{1, \ldots, j\}$ with scaled profit exactly q are (only the values of q that can be attained are reported; ∞ means that q cannot be attained with the first j items):

q	0	2	3	4	5	6	7	9
$u(0, q)$	0	∞	∞	∞	∞	∞	∞	∞
$u(1, q)$	0	3	∞	∞	∞	∞	∞	∞
$u(2, q)$	0	3	∞	7	∞	∞	∞	∞
$u(3, q)$	0	3	∞	7	∞	12	∞	∞
$u(4, q)$	0	3	7	7	10	12	14	19

The largest q with $u(4, q) \leq c = 16$ is $q = 7$, of weight $u(4, 7) = 14$, attained by the items $\{1, 2, 4\}$.

4. FP^{ε} returns $S = \{1, 2, 4\}$, of profit
$$FP^{\varepsilon}(I) = 14 + 12 + 18 = 44$$
in the original instance, better than $GR^{e}(I) = 40$. The solution $\tilde{S} = \{1, 3, 4\}$, of weight $3 + 5 + 7 = 15 \leq 16$ and profit $z(I) = 14 + 14 + 18 = 46$ is better, so FP^{ε} does **not** find an optimal solution; $\tilde{S}$ is optimal (this can be checked by enumerating the feasible solutions). The performance ratio is
$$\frac{z(I)}{FP^{\varepsilon}(I)} = \frac{46}{44} = \frac{23}{22} \approx 1.05 \leq \frac{1}{1-\varepsilon} = 2,$$
as guaranteed.

Exercise 2.15 — Basic-FPTAS 3

We consider a knapsack of capacity $c = 16$ and a set $\mathscr{I} = \{1, 2, 3, 4\}$ of 4 items ($n = 4$), with the following profits and weights:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$
p_i	14	12	14	18
w_i	3	4	5	7

We apply the Basic-FPTAS algorithm with accuracy $\varepsilon = \frac{1}{4}$.

Questions

1. Compute the lower bound $GR^e(I)$ used by the Basic-FPTAS algorithm.
2. Compute the scaling factor and the scaled profits
$$K = \frac{\varepsilon\, GR^e(I)}{n} \qquad \text{and} \qquad p'_j = \left\lfloor \frac{p_j}{K} \right\rfloor \quad \text{for all } j \in \mathscr{I}.$$
3. Solve the scaled instance by dynamic programming by profits, i.e., compute the minimum weights $u(j, q)$ and find the largest profit q such that $u(4, q) \leq c$.
4. State the solution returned by FP^ε, its profit $FP^\varepsilon(I)$ in the original instance, and compare it with $GR^e(I)$ and $z(I)$. Compute the performance ratio and compare it with the guarantee $\frac{1}{1-\varepsilon}$.

Solution

1. The items are sorted by non-increasing efficiency. GR selects the items $\{1, 2, 3\}$, with $GR(I) = 14 + 12 + 14 = 40$; the maximum profit is $p_{\max} = p_4 = 18$. Hence
$$GR^e(I) = 40.$$
2. The scaling factor is
$$K = \frac{\varepsilon\, GR^e(I)}{n} = \frac{\frac{1}{4} \cdot 40}{4} = \frac{5}{2} = 2.5,$$
and the scaled profits are
$$p'_1 = \left\lfloor \frac{14}{\frac{5}{2}} \right\rfloor = 5, \quad p'_2 = \left\lfloor \frac{12}{\frac{5}{2}} \right\rfloor = 4, \quad p'_3 = \left\lfloor \frac{14}{\frac{5}{2}} \right\rfloor = 5, \quad p'_4 = \left\lfloor \frac{18}{\frac{5}{2}} \right\rfloor = 7.$$
The total scaled profit is $P' = 5 + 4 + 5 + 7 = 21$.
3. The minimum weights $u(j, q)$ of a subset of $\{1, \ldots, j\}$ with scaled profit exactly q are (only the values of q that can be attained are reported; ∞ means that q cannot be attained with the first j items):

q	0	4	5	7	9	10	11	12	14	16	17
$u(0,q)$	0	∞	∞	∞	∞	∞	∞	∞	∞	∞	∞
$u(1,q)$	0	∞	3	∞	∞	∞	∞	∞	∞	∞	∞
$u(2,q)$	0	4	3	∞	7	∞	∞	∞	∞	∞	∞
$u(3,q)$	0	4	3	∞	7	8	∞	∞	12	∞	∞
$u(4,q)$	0	4	3	7	7	8	11	10	12	14	15

q	21
$u(0,q)$	∞
$u(1,q)$	∞
$u(2,q)$	∞
$u(3,q)$	∞
$u(4,q)$	19

The largest q with $u(4,q) \leq c = 16$ is $q = 17$, of weight $u(4,17) = 15$, attained by the items $\{1,3,4\}$.

4. FP^{ε} returns $S = \{1,3,4\}$, of profit

$$FP^{\varepsilon}(I) = 14 + 14 + 18 = 46$$

in the original instance, better than $GR^{e}(I) = 40$. The solution is **optimal**: $\tilde{S} = \{1,3,4\}$, of weight $3 + 5 + 7 = 15 \leq 16$ and profit $z(I) = 14 + 14 + 18 = 46$, and the performance ratio is 1.

Remark. This is the instance of the exercise Basic-FPTAS 2, solved there with $\varepsilon = \frac{1}{2}$. Halving ε halves the scaling factor K: the scaled profits are finer, the dynamic programming table is longer (P' grows), and the optimal solution is found.

Packing problems

Chapter 3

Approximation algorithms for bin packing

Property used in these exercises

The **half-full property** of the chapter Approximation algorithms for bin packing: a feasible solution with packing patterns $P_1, \dots, P_k$ satisfies it when at most one packing pattern has total weight at most $c/2$, that is

$$\left|\left\{ b \in \{1, \dots, k\} \ : \ \sum_{i \in P_b} w_i \leq \frac{c}{2} \right\}\right| \ \leq \ 1. \tag{H}$$

3.1 Next Fit

Exercise 3.1 — Next Fit 1

We consider a bin of capacity $c = 10$ and a set $\mathscr{I} = \{1, 2, 3, 4\}$ of 4 items ($n = 4$), with the following item weights:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$
w_i	6	6	4	4

Questions

1. Apply the Next Fit (NF) algorithm to this instance. Show the execution step by step, indicating at each iteration whether the item is packed into the currently open bin or a new bin is opened, and the resulting residual capacity. State the final bin assignment and $NF(I)$.
2. Construct the packing patterns $P_1, \dots, P_{NF(I)}$ from the bin vector obtained in Question 1.
3. Compute a lower bound on $z(I)$ and determine whether NF finds an optimal solution on this instance.

4. Verify whether the packing patterns found by NF satisfy the half-full property (H), and determine whether the solution found is optimal for this BPP instance, either by comparing $NF(I)$ with a lower bound on $z(I)$, or by exhibiting a feasible solution of strictly smaller value.

Solution

1. The NF algorithm produces the following feasible solution:

   (a) Item 1 (weight 6) is packed into bin 1. The residual capacity of bin 1 becomes $10 - 6 = 4$.

   (b) Item 2 (weight 6) does not fit into bin 1 (residual capacity 4). Thus bin 1 is closed and item 2 is packed into a new bin 2. The residual capacity of bin 2 becomes $10 - 6 = 4$.

   (c) Item 3 (weight 4) is packed into bin 2. The residual capacity of bin 2 becomes $4 - 4 = 0$.

   (d) Item 4 (weight 4) does not fit into bin 2 (residual capacity 0). Thus bin 2 is closed and item 4 is packed into a new bin 3. The residual capacity of bin 3 becomes $10 - 4 = 6$.

   Hence NF returns:
   $$bin = \begin{pmatrix} 1 & 2 & 2 & 3 \end{pmatrix}, \qquad nb = 3.$$
   and therefore
   $$NF(I) = nb = 3.$$

2. From the bin vector $bin = (1, 2, 2, 3)$, the packing patterns are
   $$\begin{aligned} P_1 &= \{i \in \{1,2,3,4\} : bin_i = 1\} = \{1\}, \\ P_2 &= \{i \in \{1,2,3,4\} : bin_i = 2\} = \{2,3\}, \\ P_3 &= \{i \in \{1,2,3,4\} : bin_i = 3\} = \{4\}. \end{aligned}$$

3. Accordingly, 3 is an upper bound on the optimal number of bins for this instance:
   $$z(I) \leq 3.$$
   Moreover,
   $$z(I) \geq \left\lceil \frac{\sum_{i \in \mathscr{I}} w_i}{c} \right\rceil = \left\lceil \frac{6+6+4+4}{10} \right\rceil = \left\lceil \frac{20}{10} \right\rceil = 2.$$
   In addition, the following is a feasible BPP solution:
   $$\tilde{P}_1 = \{1,3\}, \qquad \tilde{P}_2 = \{2,4\}$$
   which uses 2 bins; hence $z(I) \leq 2$. Therefore $z(I) = 2$, and NF does **not** compute an optimal solution on this instance.

4. From Question 2, the packing patterns found by NF have total weights

$$\sum_{i \in P_1} w_i = 6, \qquad \sum_{i \in P_2} w_i = 6 + 4 = 10, \qquad \sum_{i \in P_3} w_i = 4.$$

Since $c/2 = 5$, only P_3 has total weight at most $c/2$ (the other two exceed it); hence the solution **satisfies** the half-full property.

As shown in Question 3, $NF(I) = 3$ while the feasible solution $\tilde{P}_1 = \{1, 3\}$, $\tilde{P}_2 = \{2, 4\}$ uses only 2 bins; since this is strictly smaller than $NF(I)$, the solution found by NF is **not optimal** (indeed $z(I) = 2 < NF(I) = 3$).

Exercise 3.2 — Next Fit 2

We consider a bin of capacity $c = 10$ and a set $\mathscr{I} = \{1, 2, 3, 4\}$ of 4 items ($n = 4$), with the following item weights:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$
w_i	7	7	3	3

Questions

1. Apply the Next Fit (NF) algorithm to this instance. Show the execution step by step, indicating at each iteration whether the item is packed into the currently open bin or a new bin is opened, and the resulting residual capacity. State the final bin assignment and $NF(I)$.
2. Construct the packing patterns $P_1, \ldots, P_{NF(I)}$ from the bin vector obtained in Question 1.
3. Compute a lower bound on $z(I)$ and determine whether NF finds an optimal solution on this instance.
4. Verify whether the packing patterns found by NF satisfy the half-full property (H), and determine whether the solution found is optimal for this BPP instance, either by comparing $NF(I)$ with a lower bound on $z(I)$, or by exhibiting a feasible solution of strictly smaller value.

Solution

1. The NF algorithm produces the following feasible solution:
   (a) Item 1 (weight 7) is packed into bin 1. The residual capacity of bin 1 becomes $10 - 7 = 3$.
   (b) Item 2 (weight 7) does not fit into bin 1 (residual capacity 3). Thus bin 1 is closed and item 2 is packed into a new bin 2. The residual capacity of bin 2

becomes $10 - 7 = 3$.

(c) Item 3 (weight 3) is packed into bin 2 (residual capacity 3). The residual capacity of bin 2 becomes $3 - 3 = 0$.

(d) Item 4 (weight 3) does not fit into bin 2 (residual capacity 0). Thus bin 2 is closed and item 4 is packed into a new bin 3. The residual capacity of bin 3 becomes $10 - 3 = 7$.

Hence NF returns:

$$bin = \begin{pmatrix} 1 & 2 & 2 & 3 \end{pmatrix}, \qquad nb = 3.$$

and therefore

$$NF(I) = nb = 3.$$

2. From the bin vector $bin = (1, 2, 2, 3)$, the packing patterns are

$$P_1 = \{i \in \{1,2,3,4\} : bin_i = 1\} = \{1\},$$
$$P_2 = \{i \in \{1,2,3,4\} : bin_i = 2\} = \{2,3\},$$
$$P_3 = \{i \in \{1,2,3,4\} : bin_i = 3\} = \{4\}.$$

3. Accordingly, 3 is an upper bound on the optimal number of bins for this instance:

$$z(I) \leq 3.$$

Moreover,

$$z(I) \geq \left\lceil \frac{\sum_{i \in \mathscr{I}} w_i}{c} \right\rceil = \left\lceil \frac{7+7+3+3}{10} \right\rceil = \left\lceil \frac{20}{10} \right\rceil = 2.$$

In addition, the following is a feasible BPP solution:

$$\tilde{P}_1 = \{1,3\}, \qquad \tilde{P}_2 = \{2,4\}$$

which uses 2 bins (since $w_1 + w_3 = 7 + 3 = 10$ and $w_2 + w_4 = 7 + 3 = 10$); hence $z(I) \leq 2$. Therefore $z(I) = 2$, and NF does **not** compute an optimal solution on this instance.

4. From Question 2, the packing patterns found by NF have total weights

$$\sum_{i \in P_1} w_i = 7, \qquad \sum_{i \in P_2} w_i = 7 + 3 = 10, \qquad \sum_{i \in P_3} w_i = 3.$$

Since $c/2 = 5$, only P_3 has total weight at most $c/2$ (the other two exceed it); hence the solution **satisfies** the half-full property.

As shown in Question 3, $NF(I) = 3$ while the feasible solution $\tilde{P}_1 = \{1,3\}$, $\tilde{P}_2 = \{2,4\}$ uses only 2 bins; since this is strictly smaller than $NF(I)$, the solution found by NF is **not optimal** (indeed $z(I) = 2 < NF(I) = 3$).

Exercise 3.3 — Next Fit 3

We consider a bin of capacity $c = 10$ and a set $\mathscr{I} = \{1, 2, 3, 4, 5\}$ of 5 items ($n = 5$), with the following item weights:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
w_i	4	7	6	3	5

Questions

1. Apply the Next Fit (NF) algorithm to this instance. Show the execution step by step, indicating at each iteration whether the item is packed into the currently open bin or a new bin is opened, and the resulting residual capacity. State the final bin assignment and $NF(I)$.
2. Construct the packing patterns $P_1, \ldots, P_{NF(I)}$ from the bin vector obtained in Question 1.
3. Compute a lower bound on $z(I)$ and determine whether NF finds an optimal solution on this instance.
4. Verify whether the packing patterns found by NF satisfy the half-full property (H), and determine whether the solution found is optimal for this BPP instance, either by comparing $NF(I)$ with a lower bound on $z(I)$, or by exhibiting a feasible solution of strictly smaller value.

Solution

1. The NF algorithm produces the following feasible solution:
   (a) Item 1 (weight 4) is packed into bin 1. The residual capacity of bin 1 becomes $10 - 4 = 6$.
   (b) Item 2 (weight 7) does not fit into bin 1 (residual capacity 6). Thus bin 1 is closed and item 2 is packed into a new bin 2. The residual capacity of bin 2 becomes $10 - 7 = 3$.
   (c) Item 3 (weight 6) does not fit into bin 2 (residual capacity 3). Thus bin 2 is closed and item 3 is packed into a new bin 3. The residual capacity of bin 3 becomes $10 - 6 = 4$.
   (d) Item 4 (weight 3) is packed into bin 3 (residual capacity 4). The residual capacity of bin 3 becomes $4 - 3 = 1$.
   (e) Item 5 (weight 5) does not fit into bin 3 (residual capacity 1). Thus bin 3 is closed and item 5 is packed into a new bin 4. The residual capacity of bin 4 becomes $10 - 5 = 5$.

Hence NF returns:

$$bin = \begin{pmatrix} 1 & 2 & 3 & 3 & 4 \end{pmatrix}, \qquad nb = 4.$$

and therefore

$$NF(I) = nb = 4.$$

2. From the bin vector $bin = (1, 2, 3, 3, 4)$, the packing patterns are

$$\begin{aligned} P_1 &= \{i : bin_i = 1\} = \{1\}, \\ P_2 &= \{i : bin_i = 2\} = \{2\}, \\ P_3 &= \{i : bin_i = 3\} = \{3, 4\}, \\ P_4 &= \{i : bin_i = 4\} = \{5\}. \end{aligned}$$

3. Accordingly, 4 is an upper bound on the optimal number of bins for this instance:

$$z(I) \leq 4.$$

Moreover,

$$z(I) \geq \left\lceil \frac{\sum_{i \in \mathscr{I}} w_i}{c} \right\rceil = \left\lceil \frac{4 + 7 + 6 + 3 + 5}{10} \right\rceil = \left\lceil \frac{25}{10} \right\rceil = 3.$$

In addition, the following is a feasible BPP solution:

$$\tilde{P}_1 = \{2, 4\}, \qquad \tilde{P}_2 = \{1, 3\}, \qquad \tilde{P}_3 = \{5\}$$

which uses 3 bins (since $w_2 + w_4 = 7 + 3 = 10$ and $w_1 + w_3 = 4 + 6 = 10$); hence $z(I) \leq 3$. Therefore $z(I) = 3$, and NF does **not** compute an optimal solution on this instance.

4. From Question 2, the packing patterns found by NF have total weights

$$\sum_{i \in P_1} w_i = 4, \qquad \sum_{i \in P_2} w_i = 7, \qquad \sum_{i \in P_3} w_i = 6 + 3 = 9, \qquad \sum_{i \in P_4} w_i = 5.$$

Since $c/2 = 5$, *both* P_1 (weight 4) and P_4 (weight 5) have total weight at most $c/2$; hence the solution found by NF **does not satisfy** the half-full property (this is possible precisely because NF is not an Any Fit algorithm).

As shown in Question 3, $NF(I) = 4$ while the feasible solution $\tilde{P}_1 = \{2, 4\}$, $\tilde{P}_2 = \{1, 3\}$, $\tilde{P}_3 = \{5\}$ uses only 3 bins; since this is strictly smaller than $NF(I)$, the solution found by NF is **not optimal** (indeed $z(I) = 3 < NF(I) = 4$).

3.2 First Fit

Exercise 3.4 — First Fit 1

We consider a bin of capacity $c = 10$ and a set $\mathscr{I} = \{1, 2, 3, 4\}$ of 4 items ($n = 4$), with the following item weights:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$
w_i	6	6	4	4

Questions

1. Apply the First Fit (FF) algorithm to this instance. Show the execution step by step, indicating at each iteration which bin the item is packed into (scanning bins in the order they were opened) and the resulting residual capacity. State the final bin assignment and $FF(I)$.
2. Construct the packing patterns $P_1, \ldots, P_{FF(I)}$ from the bin vector obtained in Question 1.
3. Compute a lower bound on $z(I)$ and determine whether FF finds an optimal solution on this instance.
4. Verify whether the packing patterns found by FF satisfy the half-full property (H), and determine whether the solution found is optimal for this BPP instance, either by comparing $FF(I)$ with a lower bound on $z(I)$, or by exhibiting a feasible solution of strictly smaller value.

Solution

1. Recall the FF pseudocode: for each item i, $s := -1$ and, for $b := 1$ to nb, if $w[i] \le rc[b]$ then $s \leftarrow b$ and stop the scan; if $s = -1$ after the scan, a new bin is opened.

   (a) Item 1 (weight 6): bin 1 is open at the start ($nb = 1$, $rc[1] = c = 10$) and $w[1] \le rc[1]$, so $s = 1$ and FF packs item 1 into bin 1, and $rc[1] \leftarrow 10 - 6 = 4$.

   (b) Item 2 (weight 6): $b = 1$: $w[2] = 6 \le rc[1] = 4$? No. No bin fits, so FF opens bin 2 ($rc[2] \leftarrow 10$). Item 2 is packed into bin 2, and $rc[2] \leftarrow 10 - 6 = 4$.

   (c) Item 3 (weight 4): $b = 1$: $w[3] = 4 \le rc[1] = 4$? Yes, so $s \leftarrow 1$ and the scan stops. Item 3 is packed into bin 1, and $rc[1] \leftarrow 4 - 4 = 0$.

   (d) Item 4 (weight 4): $b = 1$: $w[4] = 4 \le rc[1] = 0$? No. $b = 2$: $w[4] = 4 \le rc[2] = 4$? Yes, so $s \leftarrow 2$. Item 4 is packed into bin 2, and $rc[2] \leftarrow 4 - 4 = 0$.

   Hence FF returns:

$$bin = \begin{pmatrix} 1 & 2 & 1 & 2 \end{pmatrix}, \qquad nb = 2.$$

and therefore

$$FF(I) = nb = 2.$$

2. From the bin vector $bin = (1, 2, 1, 2)$, the packing patterns are

$$P_1 = \{i \in \{1, 2, 3, 4\} : bin_i = 1\} = \{1, 3\},$$
$$P_2 = \{i \in \{1, 2, 3, 4\} : bin_i = 2\} = \{2, 4\}.$$

3. Accordingly,

$$z(I) \leq 2.$$

Moreover,

$$z(I) \geq \left\lceil \frac{\sum_{i \in \mathscr{I}} w_i}{c} \right\rceil = \left\lceil \frac{6 + 6 + 4 + 4}{10} \right\rceil = \left\lceil \frac{20}{10} \right\rceil = 2.$$

Hence $z(I) = 2$ and FF computes an **optimal** solution on this instance.

4. From Question 2, the packing patterns found by FF have total weights

$$\sum_{i \in P_1} w_i = 6 + 4 = 10, \qquad \sum_{i \in P_2} w_i = 6 + 4 = 10.$$

Since $c/2 = 5$, no packing pattern has total weight at most $c/2$; in particular *at most one* does, so the solution **satisfies** the half-full property.

Since $FF(I) = 2$ matches the lower bound $z(I) \geq 2$ found in Question 3, the solution found by FF is **optimal**.

Exercise 3.5 — First Fit 2

We consider a bin of capacity $c = 10$ and a set $\mathscr{I} = \{1, 2, 3, 4\}$ of 4 items ($n = 4$), with the following item weights:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$
w_i	7	7	3	3

Questions

1. Apply the First Fit (FF) algorithm to this instance. Show the execution step by step, indicating at each iteration which bin the item is packed into (scanning bins in the order they were opened) and the resulting residual capacity. State the final bin assignment and $FF(I)$.

2. Construct the packing patterns $P_1, \ldots, P_{FF(I)}$ from the bin vector obtained in Question 1.

3. Compute a lower bound on $z(I)$ and determine whether FF finds an optimal solution on this instance.

4. Verify whether the packing patterns found by FF satisfy the half-full property (H), and determine whether the solution found is optimal for this BPP instance, either by comparing $FF(I)$ with a lower bound on $z(I)$, or by exhibiting a feasible solution of strictly smaller value.

Solution

1. Recall the FF pseudocode: for each item i, $s := -1$ and, for $b := 1$ to nb, if $w[i] \leq rc[b]$ then $s \leftarrow b$ and stop the scan; if $s = -1$ after the scan, a new bin is opened.

   (a) Item 1 (weight 7): bin 1 is open at the start ($nb = 1$, $rc[1] = c = 10$) and $w[1] \leq rc[1]$, so $s = 1$ and FF packs item 1 into bin 1, and $rc[1] \leftarrow 10 - 7 = 3$.

   (b) Item 2 (weight 7): $b = 1$: $w[2] = 7 \leq rc[1] = 3$? No. No bin fits, so FF opens bin 2 ($rc[2] \leftarrow 10$). Item 2 is packed into bin 2, and $rc[2] \leftarrow 10 - 7 = 3$.

   (c) Item 3 (weight 3): $b = 1$: $w[3] = 3 \leq rc[1] = 3$? Yes, so $s \leftarrow 1$ and the scan stops. Item 3 is packed into bin 1, and $rc[1] \leftarrow 3 - 3 = 0$.

   (d) Item 4 (weight 3): $b = 1$: $w[4] = 3 \leq rc[1] = 0$? No. $b = 2$: $w[4] = 3 \leq rc[2] = 3$? Yes, so $s \leftarrow 2$. Item 4 is packed into bin 2, and $rc[2] \leftarrow 3 - 3 = 0$.

   Hence FF returns:

   $$bin = \begin{pmatrix} 1 & 2 & 1 & 2 \end{pmatrix}, \qquad nb = 2.$$

   and therefore

   $$FF(I) = nb = 2.$$

2. From the bin vector $bin = (1, 2, 1, 2)$, the packing patterns are

   $$P_1 = \{i \in \{1, 2, 3, 4\} : bin_i = 1\} = \{1, 3\},$$
   $$P_2 = \{i \in \{1, 2, 3, 4\} : bin_i = 2\} = \{2, 4\}.$$

3. Accordingly,

   $$z(I) \leq 2.$$

   Moreover,

   $$z(I) \geq \left\lceil \frac{\sum_{i \in \mathscr{I}} w_i}{c} \right\rceil = \left\lceil \frac{7 + 7 + 3 + 3}{10} \right\rceil = \left\lceil \frac{20}{10} \right\rceil = 2.$$

   Hence $z(I) = 2$ and FF computes an **optimal** solution on this instance.

4. From Question 2, the packing patterns found by FF have total weights

   $$\sum_{i \in P_1} w_i = 7 + 3 = 10, \qquad \sum_{i \in P_2} w_i = 7 + 3 = 10.$$

   Since $c/2 = 5$, no packing pattern has total weight at most $c/2$; in particular *at most one* does, so the solution **satisfies** the half-full property.

   Since $FF(I) = 2$ matches the lower bound $z(I) \geq 2$ found in Question 3, the solution found by FF is **optimal**.

Exercise 3.6 — First Fit 3

We consider a bin of capacity $c = 10$ and a set $\mathscr{I} = \{1, 2, 3, 4, 5\}$ of 5 items ($n = 5$), with the following item weights:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$	$i = 5$
w_i	4	7	6	3	5

Questions

1. Apply the First Fit (FF) algorithm to this instance. Show the execution step by step, indicating at each iteration which bin the item is packed into (scanning bins in the order they were opened) and the resulting residual capacity. State the final bin assignment and $FF(I)$.
2. Construct the packing patterns $P_1, \ldots, P_{FF(I)}$ from the bin vector obtained in Question 1.
3. Compute a lower bound on $z(I)$ and determine whether FF finds an optimal solution on this instance.
4. Verify whether the packing patterns found by FF satisfy the half-full property (H), and determine whether the solution found is optimal for this BPP instance, either by comparing $FF(I)$ with a lower bound on $z(I)$, or by exhibiting a feasible solution of strictly smaller value.

Solution

1. Recall the FF pseudocode: for each item i, $s := -1$ and, for $b := 1$ to nb, if $w[i] \le rc[b]$ then $s \leftarrow b$ and stop the scan; if $s = -1$ after the scan, a new bin is opened.

   (a) Item 1 (weight 4): bin 1 is open at the start ($nb = 1$, $rc[1] = c = 10$) and $w[1] \le rc[1]$, so $s = 1$ and FF packs item 1 into bin 1, and $rc[1] \leftarrow 10 - 4 = 6$.

   (b) Item 2 (weight 7): $b = 1$: $w[2] = 7 \le rc[1] = 6$? No. No bin fits, so FF opens bin 2 ($rc[2] \leftarrow 10$). Item 2 is packed into bin 2, and $rc[2] \leftarrow 10 - 7 = 3$.

   (c) Item 3 (weight 6): $b = 1$: $w[3] = 6 \le rc[1] = 6$? Yes, so $s \leftarrow 1$ and the scan stops. Item 3 is packed into bin 1, and $rc[1] \leftarrow 6 - 6 = 0$.

   (d) Item 4 (weight 3): $b = 1$: $w[4] = 3 \le rc[1] = 0$? No. $b = 2$: $w[4] = 3 \le rc[2] = 3$? Yes, so $s \leftarrow 2$. Item 4 is packed into bin 2, and $rc[2] \leftarrow 3 - 3 = 0$.

   (e) Item 5 (weight 5): $b = 1$: $w[5] = 5 \le rc[1] = 0$? No. $b = 2$: $w[5] = 5 \le rc[2] = 0$? No. No bin fits, so FF opens bin 3 ($rc[3] \leftarrow 10$). Item 5 is packed into bin 3, and $rc[3] \leftarrow 10 - 5 = 5$.

   Hence FF returns:

$$bin = \begin{pmatrix} 1 & 2 & 1 & 2 & 3 \end{pmatrix}, \qquad nb = 3.$$

and therefore

$$FF(I) = nb = 3.$$

2. From the bin vector $bin = (1, 2, 1, 2, 3)$, the packing patterns are

$$P_1 = \{i : bin_i = 1\} = \{1, 3\},$$
$$P_2 = \{i : bin_i = 2\} = \{2, 4\},$$
$$P_3 = \{i : bin_i = 3\} = \{5\}.$$

3. Accordingly,

$$z(I) \leq 3.$$

Moreover,

$$z(I) \geq \left\lceil \frac{\sum_{i \in \mathscr{I}} w_i}{c} \right\rceil = \left\lceil \frac{4+7+6+3+5}{10} \right\rceil = \left\lceil \frac{25}{10} \right\rceil = 3.$$

Hence $z(I) = 3$ and FF computes an **optimal** solution on this instance.

4. From Question 2, the packing patterns found by FF have total weights

$$\sum_{i \in P_1} w_i = 4 + 6 = 10, \qquad \sum_{i \in P_2} w_i = 7 + 3 = 10, \qquad \sum_{i \in P_3} w_i = 5.$$

Since $c/2 = 5$, only P_3 has total weight at most $c/2$ (the other two exceed it); hence the solution **satisfies** the half-full property.

Since $FF(I) = 3$ matches the lower bound $z(I) \geq 3$ found in Question 3, the solution found by FF is **optimal**.

3.3 Best Fit

Exercise 3.7 — Best Fit 1

We consider a bin of capacity $c = 10$ and a set $\mathscr{I} = \{1, 2, 3, 4\}$ of 4 items ($n = 4$), with the following item weights:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$
w_i	6	6	4	4

Questions

1. Apply the Best Fit (BF) algorithm to this instance. Show the execution step by step: for each item, trace the pseudocode variables $minrc$ and s over the scan of the open bins, and state into which bin the item is packed and the resulting residual capacity.

2. Construct the packing patterns $P_1, \ldots, P_{BF(I)}$ from the bin vector obtained in Question 1.

3. Compute a lower bound on $z(I)$ and determine whether BF finds an optimal solution on this instance.

4. Verify whether the packing patterns found by BF satisfy the half-full property (H), and determine whether the solution found is optimal for this BPP instance, either by comparing $BF(I)$ with a lower bound on $z(I)$, or by exhibiting a feasible solution of strictly smaller value.

Solution

1. Recall the BF pseudocode: for each item, $minrc := c + 1$, $s := -1$, and for $b := 1$ to nb, if $w[i] \leq rc[b]$ and $rc[b] < minrc$ then $s \leftarrow b$, $minrc \leftarrow rc[b]$.

   (a) Item 1 (weight 6): bin 1 is open at the start ($nb = 1$, $rc[1] = c$) and item 1 fits into it, so $s = 1$ and BF packs item 1 into bin 1. Residual capacity of bin 1: $10 - 6 = 4$.

   (b) Item 2 (weight 6): $minrc := 11$, $s := -1$. For $b = 1$: $w[2] = 6 \leq rc[1] = 4$? No. No bin fits, so BF opens bin 2 and packs item 2 into it. Residual capacity of bin 2: $10 - 6 = 4$.

   (c) Item 3 (weight 4): $minrc := 11$, $s := -1$. For $b = 1$: $w[3] = 4 \leq rc[1] = 4$ and $rc[1] = 4 < 11$, so $s \leftarrow 1$, $minrc \leftarrow 4$. For $b = 2$: $w[3] = 4 \leq rc[2] = 4$, but $rc[2] = 4$ is *not* strictly smaller than $minrc = 4$, so s is not updated. Hence $s = 1$: item 3 is packed into bin 1 (the first bin opened, since the tie between bin 1 and bin 2 is broken by the strict inequality in the pseudocode). Residual capacity of bin 1: $4 - 4 = 0$.

   (d) Item 4 (weight 4): $minrc := 11$, $s := -1$. For $b = 1$: $w[4] = 4 \leq rc[1] = 0$? No. For $b = 2$: $w[4] = 4 \leq rc[2] = 4$ and $rc[2] = 4 < 11$, so $s \leftarrow 2$, $minrc \leftarrow 4$. Hence $s = 2$: item 4 is packed into bin 2. Residual capacity of bin 2: $4 - 4 = 0$.

   Hence BF returns:

   $$bin = \begin{pmatrix} 1 & 2 & 1 & 2 \end{pmatrix}, \qquad nb = 2.$$

   and therefore

   $$BF(I) = nb = 2.$$

2. From the bin vector $bin = (1, 2, 1, 2)$, the packing patterns are

   $$P_1 = \{i \in \{1, 2, 3, 4\} : bin_i = 1\} = \{1, 3\},$$
   $$P_2 = \{i \in \{1, 2, 3, 4\} : bin_i = 2\} = \{2, 4\}.$$

3. Accordingly,

   $$z(I) \leq 2.$$

   Moreover,

   $$z(I) \geq \left\lceil \frac{\sum_{i \in \mathscr{I}} w_i}{c} \right\rceil = \left\lceil \frac{6 + 6 + 4 + 4}{10} \right\rceil = \left\lceil \frac{20}{10} \right\rceil = 2.$$

Hence $z(I) = 2$ and BF computes an **optimal** solution on this instance.

4. From Question 2, the packing patterns found by BF have total weights

$$\sum_{i \in P_1} w_i = 6 + 4 = 10, \qquad \sum_{i \in P_2} w_i = 6 + 4 = 10.$$

Since $c/2 = 5$, no packing pattern has total weight at most $c/2$; in particular *at most one* does, so the solution **satisfies** the half-full property.

Since $BF(I) = 2$ matches the lower bound $z(I) \geq 2$ found in Question 3, the solution found by BF is **optimal**.

Exercise 3.8 — Best Fit 2

We consider a bin of capacity $c = 10$ and a set $\mathscr{I} = \{1, 2, 3, 4\}$ of 4 items ($n = 4$), with the following item weights:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$
w_i	7	7	3	3

Questions

1. Apply the Best Fit (BF) algorithm to this instance. Show the execution step by step: for each item, trace the pseudocode variables $minrc$ and s over the scan of the open bins, and state into which bin the item is packed and the resulting residual capacity.
2. Construct the packing patterns $P_1, \ldots, P_{BF(I)}$ from the bin vector obtained in Question 1.
3. Compute a lower bound on $z(I)$ and determine whether BF finds an optimal solution on this instance.
4. Verify whether the packing patterns found by BF satisfy the half-full property (H), and determine whether the solution found is optimal for this BPP instance, either by comparing $BF(I)$ with a lower bound on $z(I)$, or by exhibiting a feasible solution of strictly smaller value.

Solution

1. Recall the BF pseudocode: for each item i, $minrc := c + 1$, $s := -1$, and for $b := 1$ to nb, if $w[i] \leq rc[b]$ and $rc[b] < minrc$ then $s \leftarrow b$, $minrc \leftarrow rc[b]$.

   (a) Item 1 (weight 7): bin 1 is open at the start ($nb = 1$, $rc[1] = c$) and item 1 fits into it, so $s = 1$ and BF packs item 1 into bin 1. Residual capacity of bin 1: $10 - 7 = 3$.

(b) Item 2 (weight 7): $minrc := 11$, $s := -1$. For $b = 1$: $w[2] = 7 \leq rc[1] = 3$? No. No bin fits, so BF opens bin 2 and packs item 2 into it. Residual capacity of bin 2: $10 - 7 = 3$.

(c) Item 3 (weight 3): $minrc := 11$, $s := -1$. For $b = 1$: $w[3] = 3 \leq rc[1] = 3$ and $rc[1] = 3 < 11$, so $s \leftarrow 1$, $minrc \leftarrow 3$. For $b = 2$: $w[3] = 3 \leq rc[2] = 3$, but $rc[2] = 3$ is *not* strictly smaller than $minrc = 3$, so s is not updated. Hence $s = 1$: item 3 is packed into bin 1 (the first bin opened, since the tie between bin 1 and bin 2 is broken by the strict inequality in the pseudocode). Residual capacity of bin 1: $3 - 3 = 0$.

(d) Item 4 (weight 3): $minrc := 11$, $s := -1$. For $b = 1$: $w[4] = 3 \leq rc[1] = 0$? No. For $b = 2$: $w[4] = 3 \leq rc[2] = 3$ and $rc[2] = 3 < 11$, so $s \leftarrow 2$, $minrc \leftarrow 3$. Hence $s = 2$: item 4 is packed into bin 2. Residual capacity of bin 2: $3 - 3 = 0$.

Hence BF returns:

$$bin = \begin{pmatrix} 1 & 2 & 1 & 2 \end{pmatrix}, \qquad nb = 2.$$

and therefore

$$BF(I) = nb = 2.$$

2. From the bin vector $bin = (1, 2, 1, 2)$, the packing patterns are

$$P_1 = \{i \in \{1, 2, 3, 4\} : bin_i = 1\} = \{1, 3\},$$
$$P_2 = \{i \in \{1, 2, 3, 4\} : bin_i = 2\} = \{2, 4\}.$$

3. Accordingly,

$$z(I) \leq 2.$$

Moreover,

$$z(I) \geq \left\lceil \frac{\sum_{i \in \mathscr{I}} w_i}{c} \right\rceil = \left\lceil \frac{7 + 7 + 3 + 3}{10} \right\rceil = \left\lceil \frac{20}{10} \right\rceil = 2.$$

Hence $z(I) = 2$ and BF computes an **optimal** solution on this instance.

4. From Question 2, the packing patterns found by BF have total weights

$$\sum_{i \in P_1} w_i = 7 + 3 = 10, \qquad \sum_{i \in P_2} w_i = 7 + 3 = 10.$$

Since $c/2 = 5$, no packing pattern has total weight at most $c/2$; in particular *at most one* does, so the solution **satisfies** the half-full property.

Since $BF(I) = 2$ matches the lower bound $z(I) \geq 2$ found in Question 3, the solution found by BF is **optimal**.

Exercise 3.9 — Best Fit 3

We consider a bin of capacity $c = 10$ and a set $\mathscr{I} = \{1, 2, 3, 4, 5\}$ of 5 items ($n = 5$), with the following item weights:

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
w_i	4	7	6	3	5

Questions

1. Apply the Best Fit (BF) algorithm to this instance. Show the execution step by step: for each item, trace the pseudocode variables $minrc$ and s over the scan of the open bins, and state into which bin the item is packed and the resulting residual capacity.
2. Construct the packing patterns $P_1, \ldots, P_{BF(I)}$ from the bin vector obtained in Question 1.
3. Compute a lower bound on $z(I)$ and determine whether BF finds an optimal solution on this instance.
4. Verify whether the packing patterns found by BF satisfy the half-full property (H), and determine whether the solution found is optimal for this BPP instance, either by comparing $BF(I)$ with a lower bound on $z(I)$, or by exhibiting a feasible solution of strictly smaller value.

Solution

1. Recall the BF pseudocode: for each item i, $minrc := c+1$, $s := -1$, and for $b := 1$ to nb, if $w[i] \le rc[b]$ and $rc[b] < minrc$ then $s \leftarrow b$, $minrc \leftarrow rc[b]$.

   (a) Item 1 (weight 4): bin 1 is open at the start ($nb = 1$, $rc[1] = c$) and item 1 fits into it, so $s = 1$ and BF packs item 1 into bin 1. Residual capacity of bin 1: $10 - 4 = 6$.

   (b) Item 2 (weight 7): $minrc := 11$, $s := -1$. For $b = 1$: $w[2] = 7 \le rc[1] = 6$? No. No bin fits, so BF opens bin 2 and packs item 2 into it. Residual capacity of bin 2: $10 - 7 = 3$.

   (c) Item 3 (weight 6): $minrc := 11$, $s := -1$. For $b = 1$: $w[3] = 6 \le rc[1] = 6$ and $rc[1] = 6 < 11$, so $s \leftarrow 1$, $minrc \leftarrow 6$. For $b = 2$: $w[3] = 6 \le rc[2] = 3$? No. Hence $s = 1$: item 3 is packed into bin 1. Residual capacity of bin 1: $6 - 6 = 0$.

   (d) Item 4 (weight 3): $minrc := 11$, $s := -1$. For $b = 1$: $w[4] = 3 \le rc[1] = 0$? No. For $b = 2$: $w[4] = 3 \le rc[2] = 3$ and $rc[2] = 3 < 11$, so $s \leftarrow 2$, $minrc \leftarrow 3$. Hence $s = 2$: item 4 is packed into bin 2. Residual capacity of bin 2: $3 - 3 = 0$.

   (e) Item 5 (weight 5): $minrc := 11$, $s := -1$. For $b = 1$: $w[5] = 5 \le rc[1] = 0$? No. For $b = 2$: $w[5] = 5 \le rc[2] = 0$? No. No bin fits, so BF opens bin 3 and packs item 5 into it. Residual capacity of bin 3: $10 - 5 = 5$.

   Hence BF returns:

$$bin = \begin{pmatrix} 1 & 2 & 1 & 2 & 3 \end{pmatrix}, \qquad nb = 3.$$

   and therefore

$$BF(I) = nb = 3.$$

2. From the bin vector $bin = (1, 2, 1, 2, 3)$, the packing patterns are

$$P_1 = \{i : bin_i = 1\} = \{1, 3\},$$
$$P_2 = \{i : bin_i = 2\} = \{2, 4\},$$
$$P_3 = \{i : bin_i = 3\} = \{5\}.$$

3. Accordingly,

$$z(I) \leq 3.$$

Moreover,

$$z(I) \geq \left\lceil \frac{\sum_{i \in \mathscr{I}} w_i}{c} \right\rceil = \left\lceil \frac{4 + 7 + 6 + 3 + 5}{10} \right\rceil = \left\lceil \frac{25}{10} \right\rceil = 3.$$

Hence $z(I) = 3$ and BF computes an **optimal** solution on this instance.

4. From Question 2, the packing patterns found by BF have total weights

$$\sum_{i \in P_1} w_i = 4 + 6 = 10, \qquad \sum_{i \in P_2} w_i = 7 + 3 = 10, \qquad \sum_{i \in P_3} w_i = 5.$$

Since $c/2 = 5$, only P_3 has total weight at most $c/2$ (the other two exceed it); hence the solution **satisfies** the half-full property.

Since $BF(I) = 3$ matches the lower bound $z(I) \geq 3$ found in Question 3, the solution found by BF is **optimal**.

3.4 Worst Fit

Exercise 3.10 — Worst Fit 1

We consider a bin of capacity $c = 10$ and a set $\mathscr{I} = \{1, 2, 3, 4\}$ of 4 items ($n = 4$), with the following item weights:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$
w_i	6	6	4	4

Questions

1. Apply the Worst Fit (WF) algorithm to this instance. Show the execution step by step: for each item, trace the pseudocode variables *maxrc* and s over the scan of the open bins, and state into which bin the item is packed and the resulting residual capacity.

2. Construct the packing patterns $P_1, \ldots, P_{WF(I)}$ from the bin vector obtained in Question 1.

3. Compute a lower bound on $z(I)$ and determine whether WF finds an optimal solution on this instance.

4. Verify whether the packing patterns found by WF satisfy the half-full property (H), and determine whether the solution found is optimal for this BPP instance, either by comparing $WF(I)$ with a lower bound on $z(I)$, or by exhibiting a feasible solution of strictly smaller value.

Solution

1. Recall the WF pseudocode: for each item i, $maxrc := -1$, $s := -1$, and for $b := 1$ to nb, if $w[i] \leq rc[b]$ and $rc[b] > maxrc$ then $s \leftarrow b$, $maxrc \leftarrow rc[b]$.

   (a) Item 1 (weight 6): bin 1 is open at the start ($nb = 1$, $rc[1] = c$) and item 1 fits into it, so $s = 1$ and WF packs item 1 into bin 1. Residual capacity of bin 1: $10 - 6 = 4$.

   (b) Item 2 (weight 6): $maxrc := -1$, $s := -1$. For $b = 1$: $w[2] = 6 \leq rc[1] = 4$? No. No bin fits, so WF opens bin 2 and packs item 2 into it. Residual capacity of bin 2: $10 - 6 = 4$.

   (c) Item 3 (weight 4): $maxrc := -1$, $s := -1$. For $b = 1$: $w[3] = 4 \leq rc[1] = 4$ and $rc[1] = 4 > -1$, so $s \leftarrow 1$, $maxrc \leftarrow 4$. For $b = 2$: $w[3] = 4 \leq rc[2] = 4$, but $rc[2] = 4$ is *not* strictly greater than $maxrc = 4$, so s is not updated. Hence $s = 1$: item 3 is packed into bin 1 (the first bin opened, since the tie between bin 1 and bin 2 is broken by the strict inequality in the pseudocode). Residual capacity of bin 1: $4 - 4 = 0$.

   (d) Item 4 (weight 4): $maxrc := -1$, $s := -1$. For $b = 1$: $w[4] = 4 \leq rc[1] = 0$? No. For $b = 2$: $w[4] = 4 \leq rc[2] = 4$ and $rc[2] = 4 > -1$, so $s \leftarrow 2$, $maxrc \leftarrow 4$. Hence $s = 2$: item 4 is packed into bin 2. Residual capacity of bin 2: $4 - 4 = 0$.

   Hence WF returns:

   $$bin = \begin{pmatrix} 1 & 2 & 1 & 2 \end{pmatrix}, \qquad nb = 2.$$

   and therefore

   $$WF(I) = nb = 2.$$

2. From the bin vector $bin = (1, 2, 1, 2)$, the packing patterns are

   $$P_1 = \{i \in \{1, 2, 3, 4\} : bin_i = 1\} = \{1, 3\},$$
   $$P_2 = \{i \in \{1, 2, 3, 4\} : bin_i = 2\} = \{2, 4\}.$$

3. Accordingly,

   $$z(I) \leq 2.$$

   Moreover,

   $$z(I) \geq \left\lceil \frac{\sum_{i \in \mathscr{I}} w_i}{c} \right\rceil = \left\lceil \frac{6 + 6 + 4 + 4}{10} \right\rceil = \left\lceil \frac{20}{10} \right\rceil = 2.$$

   Hence $z(I) = 2$ and WF computes an **optimal** solution on this instance.

4. From Question 2, the packing patterns found by WF have total weights

$$\sum_{i \in P_1} w_i = 6 + 4 = 10, \qquad \sum_{i \in P_2} w_i = 6 + 4 = 10.$$

Since $c/2 = 5$, no packing pattern has total weight at most $c/2$; in particular *at most one* does, so the solution **satisfies** the half-full property.

Since $WF(I) = 2$ matches the lower bound $z(I) \geq 2$ found in Question 3, the solution found by WF is **optimal**.

Exercise 3.11 — Worst Fit 2

We consider a bin of capacity $c = 10$ and a set $\mathscr{I} = \{1, 2, 3, 4\}$ of 4 items ($n = 4$), with the following item weights:

	$i = 1$	$i = 2$	$i = 3$	$i = 4$
w_i	7	7	3	3

Questions

1. Apply the Worst Fit (WF) algorithm to this instance. Show the execution step by step: for each item, trace the pseudocode variables *maxrc* and s over the scan of the open bins, and state into which bin the item is packed and the resulting residual capacity.
2. Construct the packing patterns $P_1, \ldots, P_{WF(I)}$ from the bin vector obtained in Question 1.
3. Compute a lower bound on $z(I)$ and determine whether WF finds an optimal solution on this instance.
4. Verify whether the packing patterns found by WF satisfy the half-full property (H), and determine whether the solution found is optimal for this BPP instance, either by comparing $WF(I)$ with a lower bound on $z(I)$, or by exhibiting a feasible solution of strictly smaller value.

Solution

1. Recall the WF pseudocode: for each item i, $maxrc := -1$, $s := -1$, and for $b := 1$ to nb, if $w[i] \leq rc[b]$ and $rc[b] > maxrc$ then $s \leftarrow b$, $maxrc \leftarrow rc[b]$.

   (a) Item 1 (weight 7): bin 1 is open at the start ($nb = 1$, $rc[1] = c$) and item 1 fits into it, so $s = 1$ and WF packs item 1 into bin 1. Residual capacity of bin 1: $10 - 7 = 3$.

   (b) Item 2 (weight 7): $maxrc := -1$, $s := -1$. For $b = 1$: $w[2] = 7 \leq rc[1] = 3$? No.

No bin fits, so WF opens bin 2 and packs item 2 into it. Residual capacity of bin 2: $10 - 7 = 3$.

(c) Item 3 (weight 3): $maxrc := -1$, $s := -1$. For $b = 1$: $w[3] = 3 \leq rc[1] = 3$ and $rc[1] = 3 > -1$, so $s \leftarrow 1$, $maxrc \leftarrow 3$. For $b = 2$: $w[3] = 3 \leq rc[2] = 3$, but $rc[2] = 3$ is *not* strictly greater than $maxrc = 3$, so s is not updated. Hence $s = 1$: item 3 is packed into bin 1 (the first bin opened, since the tie between bin 1 and bin 2 is broken by the strict inequality in the pseudocode). Residual capacity of bin 1: $3 - 3 = 0$.

(d) Item 4 (weight 3): $maxrc := -1$, $s := -1$. For $b = 1$: $w[4] = 3 \leq rc[1] = 0$? No. For $b = 2$: $w[4] = 3 \leq rc[2] = 3$ and $rc[2] = 3 > -1$, so $s \leftarrow 2$, $maxrc \leftarrow 3$. Hence $s = 2$: item 4 is packed into bin 2. Residual capacity of bin 2: $3 - 3 = 0$.

Hence WF returns:

$$bin = \begin{pmatrix} 1 & 2 & 1 & 2 \end{pmatrix}, \qquad nb = 2.$$

and therefore

$$WF(I) = nb = 2.$$

2. From the bin vector $bin = (1, 2, 1, 2)$, the packing patterns are

$$P_1 = \{i \in \{1, 2, 3, 4\} : bin_i = 1\} = \{1, 3\},$$
$$P_2 = \{i \in \{1, 2, 3, 4\} : bin_i = 2\} = \{2, 4\}.$$

3. Accordingly,

$$z(I) \leq 2.$$

Moreover,

$$z(I) \geq \left\lceil \frac{\sum_{i \in \mathscr{I}} w_i}{c} \right\rceil = \left\lceil \frac{7 + 7 + 3 + 3}{10} \right\rceil = \left\lceil \frac{20}{10} \right\rceil = 2.$$

Hence $z(I) = 2$ and WF computes an **optimal** solution on this instance.

4. From Question 2, the packing patterns found by WF have total weights

$$\sum_{i \in P_1} w_i = 7 + 3 = 10, \qquad \sum_{i \in P_2} w_i = 7 + 3 = 10.$$

Since $c/2 = 5$, no packing pattern has total weight at most $c/2$; in particular *at most one* does, so the solution **satisfies** the half-full property.

Since $WF(I) = 2$ matches the lower bound $z(I) \geq 2$ found in Question 3, the solution found by WF is **optimal**.

Exercise 3.12 — Worst Fit 3

We consider a bin of capacity $c = 10$ and a set $\mathscr{I} = \{1, 2, 3, 4, 5\}$ of 5 items ($n = 5$), with the following item weights:

	$i=1$	$i=2$	$i=3$	$i=4$	$i=5$
w_i	4	7	6	3	5

Questions

1. Apply the Worst Fit (WF) algorithm to this instance. Show the execution step by step: for each item, trace the pseudocode variables $maxrc$ and s over the scan of the open bins, and state into which bin the item is packed and the resulting residual capacity.
2. Construct the packing patterns $P_1, \ldots, P_{WF(I)}$ from the bin vector obtained in Question 1.
3. Compute a lower bound on $z(I)$ and determine whether WF finds an optimal solution on this instance.
4. Verify whether the packing patterns found by WF satisfy the half-full property (H), and determine whether the solution found is optimal for this BPP instance, either by comparing $WF(I)$ with a lower bound on $z(I)$, or by exhibiting a feasible solution of strictly smaller value.

Solution

1. Recall the WF pseudocode: for each item i, $maxrc := -1$, $s := -1$, and for $b := 1$ to nb, if $w[i] \le rc[b]$ and $rc[b] > maxrc$ then $s \leftarrow b$, $maxrc \leftarrow rc[b]$.

   (a) Item 1 (weight 4): bin 1 is open at the start ($nb = 1$, $rc[1] = c$) and item 1 fits into it, so $s = 1$ and WF packs item 1 into bin 1. Residual capacity of bin 1: $10 - 4 = 6$.

   (b) Item 2 (weight 7): $maxrc := -1$, $s := -1$. For $b = 1$: $w[2] = 7 \le rc[1] = 6$? No. No bin fits, so WF opens bin 2 and packs item 2 into it. Residual capacity of bin 2: $10 - 7 = 3$.

   (c) Item 3 (weight 6): $maxrc := -1$, $s := -1$. For $b = 1$: $w[3] = 6 \le rc[1] = 6$ and $rc[1] = 6 > -1$, so $s \leftarrow 1$, $maxrc \leftarrow 6$. For $b = 2$: $w[3] = 6 \le rc[2] = 3$? No. Hence $s = 1$: item 3 is packed into bin 1. Residual capacity of bin 1: $6 - 6 = 0$.

   (d) Item 4 (weight 3): $maxrc := -1$, $s := -1$. For $b = 1$: $w[4] = 3 \le rc[1] = 0$? No. For $b = 2$: $w[4] = 3 \le rc[2] = 3$ and $rc[2] = 3 > -1$, so $s \leftarrow 2$, $maxrc \leftarrow 3$. Hence $s = 2$: item 4 is packed into bin 2. Residual capacity of bin 2: $3 - 3 = 0$.

   (e) Item 5 (weight 5): $maxrc := -1$, $s := -1$. For $b = 1$: $w[5] = 5 \le rc[1] = 0$? No. For $b = 2$: $w[5] = 5 \le rc[2] = 0$? No. No bin fits, so WF opens bin 3 and packs item 5 into it. Residual capacity of bin 3: $10 - 5 = 5$.

   Hence WF returns:

$$bin = \begin{pmatrix} 1 & 2 & 1 & 2 & 3 \end{pmatrix}, \qquad nb = 3.$$

   and therefore

$$WF(I) = nb = 3.$$

2. From the bin vector $bin = (1, 2, 1, 2, 3)$, the packing patterns are

$$P_1 = \{i : bin_i = 1\} = \{1, 3\},$$
$$P_2 = \{i : bin_i = 2\} = \{2, 4\},$$
$$P_3 = \{i : bin_i = 3\} = \{5\}.$$

3. Accordingly,

$$z(I) \leq 3.$$

Moreover,

$$z(I) \geq \left\lceil \frac{\sum_{i \in \mathscr{I}} w_i}{c} \right\rceil = \left\lceil \frac{4 + 7 + 6 + 3 + 5}{10} \right\rceil = \left\lceil \frac{25}{10} \right\rceil = 3.$$

Hence $z(I) = 3$ and WF computes an **optimal** solution on this instance.

4. From Question 2, the packing patterns found by WF have total weights

$$\sum_{i \in P_1} w_i = 4 + 6 = 10, \qquad \sum_{i \in P_2} w_i = 7 + 3 = 10, \qquad \sum_{i \in P_3} w_i = 5.$$

Since $c/2 = 5$, only P_3 has total weight at most $c/2$ (the other two exceed it); hence the solution **satisfies** the half-full property.

Since $WF(I) = 3$ matches the lower bound $z(I) \geq 3$ found in Question 3, the solution found by WF is **optimal**.

Coloring problems

Chapter 4

Vertex coloring: limits of approximation

Results used in these exercises

From the chapter Vertex coloring: limits of approximation, for a graph $G = (V, E)$ with n vertices, m edges and maximum degree $\Delta(G)$:

no ratio below $\frac{4}{3}$:	$\frac{\mathscr{A}(I)}{\chi(G)} < \frac{4}{3}$ on every graph would decide $\chi(G) \leq 3$, unless $P = NP$;	(C1)
bipartite graphs:	$\chi(G) \leq 2 \iff$ no odd cycle; breadth-first search in $O(n+m)$;	(C2)
greedy coloring:	$GC(I) \leq \Delta(G) + 1$ for every order of the vertices;	(C3)
planar graphs:	$m \leq 3n - 6$ $(n \geq 3)$, $PL(I) - \chi(G) \leq 2$ (≤ 1 with 4 colors);	(C4)
interval graphs:	GC by non-decreasing left endpoint uses $\omega(G) = \chi(G)$ colors.	(C5)

Exercise 4.1 — The ratio four thirds (1)

Consider the two graphs

G_1: the wheel with 6 vertices: the cycle $1, 2, 3, 4, 5, 1$ and the vertex 6 adjacent to all the others, $E_1 = \{\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{1,5\},\{1,6\},\{2,6\},\{3,6\},\{4,6\},\{5,6\}\}$;

G_2: the cycle with 7 vertices, $E_2 = \{\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{5,6\},\{6,7\},\{1,7\}\}$.

Suppose that a polynomial algorithm $\mathscr{A}$ for the VCP had the guarantee $\frac{\mathscr{A}(I)}{\chi(G)} < \frac{4}{3}$ on every graph.

Questions

1. Compute $\chi(G_1)$ and $\chi(G_2)$, with a coloring and a proof that fewer colors are not enough.
2. Which numbers of colors can $\mathscr{A}$ return on G_1 and on G_2?
3. On a third graph G_3, $\mathscr{A}$ returns 5 colors. What can you say about $\chi(G_3)$? Would $\mathscr{A}$ decide whether $\chi(G_3) \leq 3$?
4. The same argument with $\frac{\mathscr{A}(I)}{\chi(G)} < \frac{3}{2}$ would decide whether $\chi(G) \leq 2$. Why does this prove nothing?

Solution

We use the results (C1)–(C5) of the box at the beginning of the chapter.

1. G_2 is a cycle of odd length, which needs 3 colors, and $1,2,1,2,1,2,3$ is a coloring: $\chi(G_2) = 3$. In G_1 the outer cycle $1,2,3,4,5$ is odd and needs 3 colors, and vertex 6 is adjacent to all of them and needs a fourth one; the coloring $1,2,1,2,3$ of the cycle and 4 for vertex 6 gives $\chi(G_1) = 4$. Enumeration confirms both values.

2. On G_2: $\mathscr{A}(I) \geq \chi(G_2) = 3$ and $\mathscr{A}(I) < \frac{4}{3} \cdot 3 = 4$, so $\mathscr{A}(I) = 3$. On G_1: $4 \leq \mathscr{A}(I) < \frac{4}{3} \cdot 4 = \frac{16}{3}$, so $\mathscr{A}(I) \in \{4,5\}$. As in (C1), $\mathscr{A}$ returns at most 3 colors exactly on the graphs with $\chi(G) \leq 3$.

3. From $5 < \frac{4}{3}\chi(G_3)$ we get $\chi(G_3) > \frac{15}{4}$, so $\chi(G_3) \geq 4$; feasibility gives $\chi(G_3) \leq 5$. Yes: $\mathscr{A}$ returned more than 3 colors, so $\chi(G_3) \geq 4$ and G_3 is not 3-colorable.

4. Deciding whether $\chi(G) \leq 2$ takes polynomial time by breadth-first search (C2): an algorithm that decides it gives no contradiction with $P \neq NP$. The threshold $\frac{4}{3}$ comes from 3-colorability, which is NP-complete.

Exercise 4.2 — Bipartite graphs and breadth-first search (1)

Consider the graphs

G_a: $V = \{1,\ldots,8\}$, $E_a = \{\{1,2\},\{1,4\},\{1,5\},\{2,3\},\{2,6\},\{3,4\},\{3,7\},\{4,8\},\{5,6\},\{5,8\},\{6,7\},\{7,8\}\}$;

G_b: $V = \{1,\ldots,7\}$, $E_b = \{\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{5,6\},\{1,6\},\{1,4\},\{2,7\},\{5,7\}\}$.

Run the breadth-first search from vertex 1, visiting the neighbours of a vertex in increasing order.

Questions

1. Compute the levels of the vertices of G_a and decide whether G_a is bipartite. If it is, give a coloring with 2 colors.

2. Compute the levels of the vertices of G_b and find an edge between two levels of the same parity (the first one in lexicographic order).

3. From that edge, build a cycle of odd length of G_b and conclude on $\chi(G_b)$.

4. Give a coloring of G_b with 3 colors. What is $\chi(G_b)$?

Solution

We use the results (C1)–(C5) of the box at the beginning of the chapter.

1. The levels are: level 0: $\{1\}$; level 1: $\{2,4,5\}$; level 2: $\{3,6,8\}$; level 3: $\{7\}$. No edge joins two levels of the same parity, so G_a is bipartite (C2): color 1 for the even levels, $\{1,3,6,8\}$, and color 2 for the odd levels, $\{2,4,5,7\}$. Hence $\chi(G_a) = 2$.

2. The levels are: level 0: $\{1\}$; level 1: $\{2,4,6\}$; level 2: $\{3,5,7\}$. The first edge with

both endpoints at levels of the same parity is $\{5, 7\}$, with levels 2 and 2.

3. The tree paths from the common ancestor to 5 and to 7, with the edge $\{5, 7\}$, give the cycle $5, 4, 1, 2, 7, 5$ with 5 edges, an odd number. An odd cycle needs 3 colors: $\chi(G_b) \geq 3$.

4. The greedy coloring in the order $1, \ldots, 7$ gives the colors $1, 2, 1, 2, 1, 2, 3$, a feasible coloring with 3 colors. Hence $\chi(G_b) = 3$.

Exercise 4.3 — The greedy coloring and the degree bound (1)

The crown graph G_k, $k \geq 3$, has the vertices $a_i = i$ and $b_i = k + i$, $i = 1, \ldots, k$, and the edges $\{a_i, b_j\}$ for all $i \neq j$. For $k = 4$: $E = \{\{1, 6\}, \{1, 7\}, \{1, 8\}, \{2, 5\}, \{2, 7\}, \{2, 8\}, \{3, 5\}, \{3, 6\}, \{3, 8\}, \{4, 5\}, \{4, 6\}, \{4, 7\}\}$. The greedy coloring GC gives each vertex, in a given order, the smallest color not used by its neighbours already colored.

Questions

1. Compute $\Delta(G_4)$ and $\chi(G_4)$.

2. Apply GC to G_4 with the order $\lambda_1 = (1, 2, 3, 4, 5, 6, 7, 8)$ and with the order $\lambda_2 = (1, 5, 2, 6, 3, 7, 4, 8)$.

3. Compare the two results with the bound (C3) and with $\chi(G_4)$. Is the bound $\Delta(G)+1$ attained?

4. Generalize the order λ_2 to G_k and compute the ratio $\frac{GC(I)}{\chi(G_k)}$ for $k = 3, \ldots, 6$. Is GC, with an arbitrary order, an α-approximation algorithm for some constant α?

Solution

We use the results (C1)–(C5) of the box at the beginning of the chapter.

1. Every vertex is adjacent to $k - 1 = 3$ vertices of the other side: $\Delta(G_4) = 3$. The sets $\{1, 2, 3, 4\}$ and $\{5, 6, 7, 8\}$ are stable, and there are edges: $\chi(G_4) = 2$.

2. With λ_1 the vertices $1, 2, 3, 4$ have no colored neighbour and take color 1; then $5, \ldots, 8$ take color 2: $GC(I) = 2$. With λ_2, vertex a_i is adjacent to $b_1, \ldots, b_{i-1}$, which have the colors $1, \ldots, i - 1$, and takes color i; vertex b_i does the same. The colors are $1, 1, 2, 2, 3, 3, 4, 4$ and $GC(I) = 4$.

3. Both results satisfy $GC(I) \leq \Delta(G_4) + 1 = 4$, and λ_2 attains the bound. With λ_2 the greedy coloring uses twice the optimal number of colors, while with λ_1 it is optimal: the order matters.

4. With the order $a_1, b_1, a_2, b_2, \ldots, a_k, b_k$ the vertices a_i and b_i take color i, so $GC(I) = k$:

k	n	$\Delta(G_k)$	$GC(I)$	$\frac{GC(I)}{\chi(G_k)}$
3	6	2	3	$\frac{3}{2}$
4	8	3	4	2
5	10	4	5	$\frac{5}{2}$
6	12	5	6	3

The ratio $\frac{k}{2}$ grows without limit: on all graphs, GC has no constant guarantee. On graphs of maximum degree at most Δ the ratio is at most $\frac{\Delta+1}{2}$, attained here.

Exercise 4.4 — Interval graphs (1)

Seven requests use a room in the time intervals

request v	1	2	3	4	5	6	7
$[a_v, b_v]$	$[1,3]$	$[2,6]$	$[4,7]$	$[3,5]$	$[6,10]$	$[8,9]$	$[9,12]$

Two requests conflict if their closed intervals intersect, and a color is a room.

Questions

1. Build the interval graph G: list its edges.
2. Apply the greedy coloring with the requests in order of non-decreasing left endpoint.
3. Find a point covered by the largest number of intervals, and deduce $\omega(G)$. Is the coloring of question 2 optimal?
4. Apply the greedy coloring with the order $(1, 2, 4, 5, 3, 6, 7)$. What does this show?

Solution

We use the results (C1)–(C5) of the box at the beginning of the chapter.

1. $E = \{\{1,2\},\{1,4\},\{2,3\},\{2,4\},\{2,5\},\{3,4\},\{3,5\},\{5,6\},\{5,7\},\{6,7\}\}$: 10 edges.
2. The order is $(1, 2, 4, 3, 5, 6, 7)$, and the colors are $1 \to 1$, $2 \to 2$, $4 \to 3$, $3 \to 1$, $5 \to 3$, $6 \to 1$, $7 \to 2$: $GC(I) = 3$.
3. The point 3 lies in the intervals $\{1, 2, 4\}$, which pairwise intersect and form a clique: $\omega(G) \geq 3$; no point lies in more than 3 intervals, so $\omega(G) = 3$. By (C5) and the clique bound, $GC(I) = \omega(G) = \chi(G) = 3$: the coloring is optimal, with 3 rooms.
4. The colors are $1 \to 1$, $2 \to 2$, $4 \to 3$, $5 \to 1$, $3 \to 4$, $6 \to 2$, $7 \to 3$: vertex 3 sees the colors $\{1, 2, 3\}$ on its neighbours and takes color 4. $GC(I) = 4 > \chi(G)$: the order of the left endpoints is what makes the greedy coloring exact.

Exercise 4.5 — Planar graphs and an additive guarantee (1)

Consider the graphs

W: the wheel, $E = \{\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{1,5\},\{1,6\},\{2,6\},\{3,6\},\{4,6\},\{5,6\}\}$;

O: the octahedron, with $V = \{1,\ldots,6\}$ and all the edges except $\{1,6\},\{2,5\},\{3,4\}$;

K_5: the complete graph with 5 vertices.

The algorithm PL for planar graphs uses 1 color if there is no edge, 2 colors if the graph is bipartite, and otherwise the 5 colors of the theorem of Heawood.

Questions

1. Compute n, m and $3n-6$ for the three graphs. Which of them can be planar?
2. W and O are planar (draw them). Find in each a vertex of degree at most 5, as the lemma of the chapter guarantees.
3. Is W bipartite? Is O? What does PL guarantee on them, and what are $\chi(W)$ and $\chi(O)$ (by enumeration)?
4. Bound the additive error of PL on W and on O, and the additive error with the four color algorithm in place of the five color one.

Solution

We use the results (C1)–(C5) of the box at the beginning of the chapter.

1.

graph	n	m	$3n-6$
W	6	10	12
O	6	12	12
K_5	5	10	9

By (C4) a planar graph has $m \le 3n-6$: K_5, with $10 > 9$, is not planar. The bound does not exclude W and O.

2. In W the vertices of the cycle have degree 3, and in O every vertex has degree 4: both values are at most 5.
3. Both graphs contain triangles ($\{1,2,6\}$ in W, $\{1,2,3\}$ in O), odd cycles: neither is bipartite, so PL uses at most 5 colors. Enumeration gives $\chi(W) = 4$ (the odd cycle $1,\ldots,5$ needs 3 colors and vertex 6 a fourth) and $\chi(O) = 3$ (the classes $\{1,6\},\{2,5\},\{3,4\}$).
4. On W: $PL(I) - \chi(W) \le 5 - 4 = 1$; on O: $PL(I) - \chi(O) \le 5 - 3 = 2$, the bound of (C4). With 4 colors the errors are at most 0 on W, where the four color algorithm is exact, and 1 on O.

Clique problems

Chapter 5

Maximum clique: limits of approximation

Results used in these exercises

From the chapter Maximum clique: limits of approximation, for graphs $G = (V, E)$ and $H = (W, F)$:

strong product: $(u, x) \sim (v, y) \iff (u = v \text{ or } \{u, v\} \in E) \text{ and } (x = y \text{ or } \{x, y\} \in F)$; (Q1)

clique number: $\omega(G \boxtimes H) = \omega(G)\,\omega(H), \quad |K| \le |p(K)| \cdot \max_u |K_u|$; (Q2)

amplification: ratio α on $G \boxtimes G \Rightarrow$ ratio $\sqrt{\alpha}$ on G; t products: $\alpha^{1/2^t}$; (Q3)

bounded degree: $\omega(G) \le \Delta(G) + 1$, enumeration of $\sum_v 2^{\deg(v)}$ subsets; (Q4)

stable sets: $\alpha(G) = \omega(\overline{G}), \quad$ a maximal stable set S has $\frac{\alpha(G)}{|S|} \le \Delta(G) + 1$. (Q5)

Exercise 5.1 — The strong product (1)

Let G be the path with the vertices $1, 2, 3$ and the edges $\{1, 2\}, \{2, 3\}$, and consider the strong product $G \boxtimes G$, whose vertices are the pairs (i, j) with $i, j \in \{1, 2, 3\}$.

Questions

1. List the neighbours of $(2, 2)$ and of $(1, 1)$ in $G \boxtimes G$, and count its edges.
2. Is $\{(1, 1), (2, 2), (3, 3)\}$ a clique of $G \boxtimes G$?
3. Compute $\omega(G)$ and $\omega(G \boxtimes G)$, and give all the maximum cliques of $G \boxtimes G$.
4. For the clique $K = \{(1, 1), (1, 2), (2, 1), (2, 2)\}$, compute the projection $p(K)$ and the fibres K_u, and check the inequality of (Q2).

Solution

We use the results (Q1)–(Q5) of the box at the beginning of the chapter.

1. By (Q1), $(2, 2)$ is adjacent to every other pair, since 2 is adjacent to 1 and 3: its 8 neighbours are all the other vertices. The vertex $(1, 1)$ has the neighbours $(1, 2), (2, 1), (2, 2)$. Counting the neighbours of all the vertices and halving, $G \boxtimes G$ has 20 edges.

2. No: $(1,1)$ and $(3,3)$ are not adjacent, because $1 \neq 3$ and $\{1,3\}$ is not an edge of G.
3. $\omega(G) = 2$ (the edges), and by (Q2) $\omega(G \boxtimes G) = 2 \cdot 2 = 4$. Enumeration confirms it, with the 4 maximum cliques $\{i, i+1\} \times \{j, j+1\}$, $i, j \in \{1,2\}$: $\{(1,1),(1,2),(2,1),(2,2)\}$, $\{(1,2),(1,3),(2,2),(2,3)\}$, $\{(2,1),(2,2),(3,1),(3,2)\}$, $\{(2,2),(2,3),(3,2),(3,3)\}$.
4. $p(K) = \{1,2\}$, with the fibres $K_1 = K_2 = \{1,2\}$: all are cliques of G, and $|K| = 4 \leq |p(K)| \cdot \max_u |K_u| = 2 \cdot 2 = 4$.

Exercise 5.2 — Amplifying a ratio (1)

Suppose that a polynomial algorithm $\mathscr{A}$ for the MCP had the guarantee $\frac{\omega(G)}{\mathscr{A}(I)} \leq \alpha$ with $\alpha = 4$ on every graph, and apply the construction of the chapter to a graph G with $n = 50$ vertices: t strong products, then $\mathscr{A}$, then t extractions of a clique of at least the square root of the size.

Questions

1. Compute the guaranteed ratio and the number of vertices of the graph given to $\mathscr{A}$, for $t = 0, 1, 2, 3, 4$.
2. Which is the smallest t that gives the ratio $\beta = \frac{3}{2}$? And $\beta = \frac{5}{4}$, $\beta = \frac{11}{10}$?
3. On $G \boxtimes G$ the algorithm returns a clique K with 10 vertices. How large is the clique of G extracted from it, at least? What does the guarantee of $\mathscr{A}$ say on $\omega(G)$?
4. Why can t not grow with n, for example $t = \log_2 n$, to obtain the ratio 1?

Solution

We use the results (Q1)–(Q5) of the box at the beginning of the chapter.

1. By (Q3) the ratio after t products is $\alpha^{1/2^t}$, and the graph has n^{2^t} vertices:

t	ratio $4^{1/2^t}$	vertices 50^{2^t}
0	4.000	50
1	2.000	2 500
2	1.414	6 250 000
3	1.189	$3.9 \cdot 10^{13}$
4	1.091	$1.5 \cdot 10^{27}$

2. We need $4^{1/2^t} \leq \beta$, i.e., $2^t \geq \frac{\ln 4}{\ln \beta}$:
   - $\beta = \frac{3}{2}$: $\frac{\ln 4}{\ln \beta} \approx 3.42$, so $t = 2$ (ratio 1.414).
   - $\beta = \frac{5}{4}$: $\frac{\ln 4}{\ln \beta} \approx 6.21$, so $t = 3$ (ratio 1.189).
   - $\beta = \frac{11}{10}$: $\frac{\ln 4}{\ln \beta} \approx 14.55$, so $t = 4$ (ratio 1.091).
3. By (Q2), $|p(K)| \cdot \max_u |K_u| \geq 10$; both are integers, and $3 \cdot 3 = 9 < 10$, so the larger one has at least 4 vertices, a clique of G. The guarantee gives $10 \geq \frac{\omega(G)^2}{4}$, so

$\omega(G)^2 \leq 40$ and $\omega(G) \leq 6$: the extracted clique is within the ratio $\sqrt{4} = 2$, since $2 \cdot 4 \geq 6$.

4. The graph given to $\mathscr{A}$ has n^{2^t} vertices: for $t = \log_2 n$ it has n^n vertices, and the algorithm is no longer polynomial. The construction gives every *constant* ratio $\beta > 1$, with a constant t.

Exercise 5.3 — Special cases of the MCP and the MSSP (1)

Consider the prism G, with $V = \{1, \ldots, 6\}$ and

$$E = \{\{1,2\},\{1,3\},\{2,3\},\{4,5\},\{4,6\},\{5,6\},\{1,4\},\{2,5\},\{3,6\}\},$$

and the star S_4, with the centre 1 adjacent to the vertices $2, 3, 4, 5$ and no other edge.

Questions

1. Compute $\Delta(G)$ and the number of subsets examined by the enumeration of (Q4). Which maximum cliques does it find?
2. Check that G satisfies $m \leq 3n-6$ (G is planar). Which bound on $\omega(G)$ does planarity give, and which the degree?
3. Build a maximal stable set of G by adding the vertices in the order $1, \ldots, 6$ while possible, and compare it with $\alpha(G)$ and with the bound (Q5).
4. Do the same on the star S_4. How close is the ratio to the bound $\Delta + 1$?

Solution

We use the results (Q1)–(Q5) of the box at the beginning of the chapter.

1. Every vertex has degree 3: $\Delta(G) = 3$, and the enumeration examines $\sum_v 2^{\deg(v)} = 6 \cdot 2^3 = 48$ subsets. It finds the two maximum cliques $\{1,2,3\}$ and $\{4,5,6\}$ (each from any of its vertices): $\omega(G) = 3$.
2. $m = 9 \leq 3 \cdot 6 - 6 = 12$. Planarity gives $\omega(G) \leq 4$, the degree $\omega(G) \leq \Delta(G) + 1 = 4$; the true value is 3.
3. Vertex 1 is taken; $2, 3, 4$ are adjacent to 1; 5 is taken; 6 is adjacent to 5: $S = \{1, 5\}$. A stable set has at most one vertex per triangle, so $\alpha(G) = 2$ and S is optimal. The bound (Q5) guarantees $|S| \geq \frac{n}{\Delta+1} = \frac{3}{2}$, so $|S| \geq 2$.
4. The centre 1 is taken first, and every other vertex is adjacent to it: $S = \{1\}$. The leaves form a stable set, $\alpha(S_4) = 4$, and the ratio is $\frac{4}{1} = 4$, against the bound $\Delta+1 = 5$: a maximal stable set can be far from a maximum one.

Exercise 5.4 — Complement graph and stable sets (1)

Let G be the Petersen graph, the reference instance of the VCP, with $V = \{1, \dots, 10\}$ and $E = \{\{1,2\},\{1,5\},\{1,6\},\{2,3\},\{2,7\},\{3,4\},\{3,8\},\{4,5\},\{4,9\},\{5,10\},\{6,8\},\{6,9\},\{7,9\},\{7,10\},\{8,10\}\}$.

Questions

1. Compute the number of edges and the degrees of the complement graph $\overline{G}$. Is $\overline{G}$ planar?
2. Compute $\alpha(G)$ and deduce $\omega(\overline{G})$. How many subsets does the enumeration of (Q4) examine on $\overline{G}$?
3. Build a maximal stable set of G by adding the vertices in the order $1, \dots, 10$ while possible. What does (Q5) guarantee?
4. An algorithm for the MCP has ratio at most ρ on every graph with 10 vertices. What does it give for the MSSP on G?

Solution

We use the results (Q1)–(Q5) of the box at the beginning of the chapter.

1. $\overline{G}$ has $\binom{10}{2} - 15 = 30$ edges, and every vertex has degree $9 - 3 = 6$. Since $30 > 3 \cdot 10 - 6 = 24$, $\overline{G}$ is not planar: the special case of planar graphs does not apply to the MSSP on G.
2. The two 5-cycles $1, \dots, 5$ and $6, \dots, 10$ contain at most 2 vertices of a stable set each, and $\{1,4,7,8\}$ is stable: $\alpha(G) = 4$, so $\omega(\overline{G}) = 4$, as enumeration confirms. On $\overline{G}$ the enumeration examines $10 \cdot 2^6 = 640$ subsets.
3. The order gives $S = \{1,3,7\}$: each other vertex is adjacent to one of them. By (Q5), $|S| \geq \frac{n}{\Delta(G)+1} = \frac{5}{2}$, so $|S| \geq 3$, and the ratio is $\frac{\alpha(G)}{|S|} = \frac{4}{3} \leq \Delta(G) + 1 = 4$.
4. Applied to $\overline{G}$, which has the same 10 vertices, it returns a clique of $\overline{G}$, i.e., a stable set of G, with at least $\frac{\omega(\overline{G})}{\rho} = \frac{\alpha(G)}{\rho}$ vertices: the same ratio ρ for the MSSP.

Routing problems

Chapter 6

The TSP: the general case and the metric case

Exercise 6.1 — Metric TSP approximation 1

We consider a set $\mathcal{V} = \{1, \ldots, 6\}$ of 6 cities with coordinates 1: $(3, 4)$, 2: $(6, 1)$, 3: $(3, 8)$, 4: $(8, 8)$, 5: $(0, 4)$, 6: $(8, 0)$ and distances equal to the Euclidean distances rounded to the nearest integer (the instance is metric):

d_{ij}	1	2	3	4	5	6
1	—	4	4	6	3	6
2	4	—	8	7	7	2
3	4	8	—	5	5	9
4	6	7	5	—	9	8
5	3	7	5	9	—	9
6	6	2	9	8	9	—

Questions

1. Compute a minimum spanning tree T with Kruskal's algorithm (edges by non-decreasing distance, ties by the pair of indices) and its weight $d(T)$.
2. Apply the double tree algorithm DT: give the Eulerian circuit of the doubled tree from city 1 (Fleury's rule), the tour obtained by shortcutting, and $DT(I)$.
3. Apply the Christofides algorithm CH: give the set O of odd-degree cities, the minimum weight perfect matching M (compare all the perfect matchings of O), the Eulerian circuit of $T \cup M$ from city 1, the tour and $CH(I)$.
4. The optimal tour has length $z(I) = 27$. Compute the two ratios and check the inequalities $DT(I) \leq 2\, d(T) \leq 2\, z(I)$ and $CH(I) \leq d(T) + d(M) \leq \frac{3}{2}\, z(I)$.

Solution

1. Kruskal's algorithm examines $\{2,6\}$ (2, taken), $\{1,5\}$ (3, taken), $\{1,2\}$ (4, taken), $\{1,3\}$ (4, taken), $\{3,4\}$ (5, taken). After five edges the tree spans the six cities:

$$T = \big\{\{2,6\},\{1,5\},\{1,2\},\{1,3\},\{3,4\}\big\}, \qquad d(T) = 2+3+4+4+5 = 18.$$

   It is the unique MST of the instance.

2. The neighbours in T are $1\colon\{2,3,5\}$, $2\colon\{1,6\}$, $3\colon\{1,4\}$, $4\colon\{3\}$, $5\colon\{1\}$, $6\colon\{2\}$. The Eulerian circuit of the doubled tree from city 1 is the depth-first visit with the children in increasing order:

$$W = (1,2,6,2,1,3,4,3,1,5,1), \qquad 4+2+2+4+4+5+5+4+3+3 = 36 = 2\,d(T).$$

   Keeping each city at its first occurrence gives $\sigma = (1,2,6,3,4,5)$, of length

$$DT(I) = d_{12} + d_{26} + d_{63} + d_{34} + d_{45} + d_{51} = 4+2+9+5+9+3 = 32.$$

3. The degrees in T are $\deg(1) = 3$, $\deg(2) = 2$, $\deg(3) = 2$, $\deg(4) = 1$, $\deg(5) = 1$, $\deg(6) = 1$, so $O = \{1,4,5,6\}$. Its three perfect matchings weigh

$$\{\{1,4\},\{5,6\}\}\colon 6+9 = 15, \quad \{\{1,5\},\{4,6\}\}\colon 3+8 = 11, \quad \{\{1,6\},\{4,5\}\}\colon 6+9 = 15,$$

   so $M = \{\{1,5\},\{4,6\}\}$, with $d(M) = 11$. The multigraph $T \cup M$ has two copies of $\{1,5\}$. Fleury's rule from city 1 (neighbours $2,3,5,5$) goes to 2, then 6, 4 (along M), 3, back to 1, and finally 5 and 1 along the two copies of $\{1,5\}$:

$$W = (1,2,6,4,3,1,5,1), \qquad 4+2+8+5+4+3+3 = 29 = d(T) + d(M).$$

   City 1 is repeated in the middle; shortcutting it gives $\sigma = (1,2,6,4,3,5)$, of length

$$CH(I) = d_{12} + d_{26} + d_{64} + d_{43} + d_{35} + d_{51} = 4+2+8+5+5+3 = 27.$$

   The shortcut replaces $3 \to 1 \to 5$ $(4+3)$ by $3 \to 5$ (5) and saves 2.

4. The ratios are

$$\frac{DT(I)}{z(I)} = \frac{32}{27}, \qquad \frac{CH(I)}{z(I)} = \frac{27}{27} = 1,$$

   so CH finds the optimal tour, which is unique. The inequalities of the proofs are

$$DT(I) = 32 \le 2\,d(T) = 36 \le 2\,z(I) = 54,$$
$$CH(I) = 27 \le d(T) + d(M) = 29 \le \frac{3}{2}\,z(I) = \frac{81}{2}.$$

Exercise 6.2 — Metric TSP approximation 2

We consider a set $\mathcal{V} = \{1,\dots,6\}$ of 6 cities with coordinates $1\colon(2,4)$, $2\colon(6,0)$, $3\colon(7,3)$, $4\colon(0,7)$, $5\colon(0,1)$, $6\colon(2,1)$ and distances equal to the Euclidean distances rounded to the

nearest integer (the instance is metric):

d_{ij}	1	2	3	4	5	6
1	—	6	5	4	4	3
2	6	—	3	9	6	4
3	5	3	—	8	7	5
4	4	9	8	—	6	6
5	4	6	7	6	—	2
6	3	4	5	6	2	—

Questions

1. Compute a minimum spanning tree T with Kruskal's algorithm and its weight $d(T)$.
2. Apply CH: give O, the weights of all the perfect matchings of O, the matching M, the Eulerian circuit of $T \cup M$ from city 1 (Fleury's rule), the tour and $CH(I)$.
3. Apply DT and give $DT(I)$.
4. The unique optimal tour is $(1, 3, 2, 6, 5, 4)$. Compute $z(I)$ and the two ratios. Is CH optimal on this instance? Check the bound $d(M) \leq \frac{z(I)}{2}$ by shortcutting the optimal tour to the cities of O, as in the proof of the notes.

Solution

1. Kruskal's algorithm examines $\{5,6\}$ (2, taken), $\{1,6\}$ (3, taken), $\{2,3\}$ (3, taken), $\{1,4\}$ (4, taken), $\{1,5\}$ (4, it closes the cycle $1,5,6$: rejected), $\{2,6\}$ (4, taken). Hence

$$T = \big\{\{5,6\},\{1,6\},\{2,3\},\{1,4\},\{2,6\}\big\}, \qquad d(T) = 2+3+3+4+4 = 16,$$

the unique MST of the instance.

2. The degrees in T are $\deg(1) = 2$, $\deg(2) = 2$, $\deg(3) = 1$, $\deg(4) = 1$, $\deg(5) = 1$, $\deg(6) = 3$, so $O = \{3,4,5,6\}$. Its perfect matchings weigh

$$\{\{3,4\},\{5,6\}\}\text{: } 8+2=10, \quad \{\{3,5\},\{4,6\}\}\text{: } 7+6=13, \quad \{\{3,6\},\{4,5\}\}\text{: } 5+6=11,$$

so $M = \{\{3,4\},\{5,6\}\}$, with $d(M) = 10$; $T \cup M$ has two copies of $\{5,6\}$. Fleury's rule from city 1 (neighbours 4, 6) goes to 4, then to 3 along M, to 2 and to 6. At city 6 the unused edges are $\{6,1\}$ and the two copies of $\{6,5\}$: city 1 has the smallest index, but $\{6,1\}$ is a bridge of the unused edges (removing it leaves city 1 isolated from the two copies of $\{5,6\}$), so the walk goes to 5, back to 6 and finally to 1:

$$W = (1,4,3,2,6,5,6,1), \qquad 4+8+3+4+2+2+3 = 26 = d(T) + d(M).$$

Shortcutting the second visit of city 6 gives $\sigma = (1,4,3,2,6,5)$, of length

$$CH(I) = d_{14} + d_{43} + d_{32} + d_{26} + d_{65} + d_{51} = 4+8+3+4+2+4 = 25.$$

3. The neighbours in T are $1{:}\{4,6\}$, $2{:}\{3,6\}$, $3{:}\{2\}$, $4{:}\{1\}$, $5{:}\{6\}$, $6{:}\{1,2,5\}$. The depth-first visit from city 1 is

$$W = (1,4,1,6,2,3,2,6,5,6,1), \qquad \text{of length } 32 = 2\,d(T),$$

and shortcutting gives $\sigma = (1,4,6,2,3,5)$, of length

$$DT(I) = 4+6+4+3+7+4 = 28.$$

4. $z(I) = d_{13} + d_{32} + d_{26} + d_{65} + d_{54} + d_{41} = 5+3+4+2+6+4 = 24$, and

$$\frac{CH(I)}{z(I)} = \frac{25}{24}, \qquad \frac{DT(I)}{z(I)} = \frac{28}{24} = \frac{7}{6}.$$

CH is not optimal here, but it is only one unit above the optimum. Shortcutting the optimal tour to $O = \{3,4,5,6\}$ gives the cycle $(3,6,5,4)$, of length $d_{36} + d_{65} + d_{54} + d_{43} = 5+2+6+8 = 21 \le 24$. Its alternate edges form the matchings $\{\{3,6\},\{5,4\}\}$, of weight 11, and $\{\{6,5\},\{4,3\}\}$, of weight 10, and indeed

$$d(M) = 10 \le \min\{11,10\} \le \frac{21}{2} \le \frac{z(I)}{2} = 12.$$

Exercise 6.3 — Metric TSP approximation 3

We consider two graphs with 5 vertices, $V = \{1,2,3,4,5\}$:

$$G_1: \; E_1 = \big\{\{1,2\},\{2,3\},\{3,4\},\{4,5\},\{5,1\},\{1,3\}\big\},$$

$$G_2: \; E_2 = \big\{\{1,3\},\{1,4\},\{1,5\},\{2,3\},\{2,4\},\{2,5\}\big\}.$$

Suppose that a polynomial algorithm $\mathscr{A}$ for the TSP had the guarantee $\frac{\mathscr{A}(I)}{z(I)} \le \alpha$ with $\alpha = 2$ on every TSP instance.

Questions

1. Build the TSP instances I_{G_1} and I_{G_2} of the proof of the notes: $d_{ij} = 1$ if $\{i,j\}$ is an edge and $d_{ij} = \lceil \alpha n \rceil + 1$ otherwise.
2. Determine whether G_1 and G_2 are Hamiltonian, and compute $z(I_{G_1})$ and $z(I_{G_2})$.
3. What can $\mathscr{A}$ return on the two instances? Explain how $\mathscr{A}$ would decide whether a graph is Hamiltonian.
4. Is I_{G_2} metric? Could $\mathscr{A}$ be the Christofides algorithm?

Solution

1. With $n = 5$ and $\alpha = 2$, the non-edges get distance $\lceil 2 \cdot 5 \rceil + 1 = 11$:

I_{G_1}	1	2	3	4	5
1	—	1	1	11	1
2	1	—	1	11	11
3	1	1	—	1	11
4	11	11	1	—	1
5	1	11	11	1	—

I_{G_2}	1	2	3	4	5
1	—	11	1	1	1
2	11	—	1	1	1
3	1	1	—	11	11
4	1	1	11	—	11
5	1	1	11	11	—

2. G_1 contains the cycle $(1, 2, 3, 4, 5)$, which is Hamiltonian, so the tour $(1, 2, 3, 4, 5)$ has length 5 and $z(I_{G_1}) = 5$.

   G_2 is the complete bipartite graph with sides $\{1, 2\}$ and $\{3, 4, 5\}$. A cycle in a bipartite graph alternates between the two sides, so it has as many vertices on each side: no cycle can contain the three vertices $3, 4, 5$ and only two vertices of the other side, and G_2 is not Hamiltonian. Every tour of I_{G_2} uses at least one distance 11. The tour $(1, 3, 2, 4, 5)$ uses the edges $\{1, 3\}, \{3, 2\}, \{2, 4\}, \{1, 5\}$ and the non-edge $\{4, 5\}$, so its length is $4 \cdot 1 + 11 = 15$, and

   $$z(I_{G_2}) = 15.$$

   (Enumerating the 12 tours, their lengths are only 15 and 35.)

3. On I_{G_1}: $\mathscr{A}(I_{G_1}) \leq 2 \cdot 5 = 10$. A tour that uses a distance 11 has length at least $4 + 11 = 15 > 10$, so $\mathscr{A}$ must return a tour of length 5, i.e., a Hamiltonian cycle of G_1.

   On I_{G_2}: $\mathscr{A}(I_{G_2}) \geq z(I_{G_2}) = 15 > 10$.

   So the test "$\mathscr{A}(I_G) \leq \alpha n = 10$" answers "yes" on G_1 and "no" on G_2. In general, building I_G and running $\mathscr{A}$ takes polynomial time and gives the right answer on every graph, so the NP-complete Hamiltonian cycle problem would be solved in polynomial time: such an $\mathscr{A}$ cannot exist unless $P = NP$.

4. I_{G_2} is not metric: for example $d_{12} = 11 > d_{13} + d_{32} = 1 + 1 = 2$. The guarantee $\frac{3}{2}$ of the Christofides algorithm holds only for metric instances, so it says nothing on I_{G_2}, and there is no contradiction with the proposition of the notes: it is the triangle inequality that makes the TSP approximable.

Scheduling problems

Approximation for $P||C_{\max}$

Exercise 7.1 — $P\,||\,C_{\max}$ approximation 1

We consider $m = 3$ identical machines and a set $\mathscr{J} = \{1, \ldots, 7\}$ of 7 jobs ($n = 7$), with the following processing times:

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$
t_j	2	2	4	8	8	6	6

Questions

1. Apply $LIST$ with the list $\lambda = (1, 2, \ldots, 7)$. Show, for each job, the loads before the assignment and the chosen machine, and give $LIST(I)$.
2. Identify a job ℓ that ends last in the schedule of Question 1 and check the inequality $S_\ell \leq \frac{1}{m}\sum_{j\neq\ell} t_j$ used in the proof of the guarantee of $LIST$.
3. Apply LPT and give $LPT(I)$.
4. Compute the lower bound $\max\{\max_j t_j, \frac{1}{m}\sum_j t_j\}$ on $z(I)$ and determine $z(I)$. Compute the two ratios $\frac{LIST(I)}{z(I)}$ and $\frac{LPT(I)}{z(I)}$ and compare them with the guarantees $2 - \frac{1}{m}$ and $\frac{4}{3} - \frac{1}{3m}$.

Solution

1. Each job goes to a machine with the smallest load (smallest index in case of ties):

job j	t_j	loads (L_1, L_2, L_3) before	machine
1	2	$(0,0,0)$	1
2	2	$(2,0,0)$	2
3	4	$(2,2,0)$	3
4	8	$(2,2,4)$	1
5	8	$(10,2,4)$	2
6	6	$(10,10,4)$	3
7	6	$(10,10,10)$	1

Hence $mach = (1,2,3,1,2,3,1)$, $J_1 = \{1,4,7\}$, $J_2 = \{2,5\}$, $J_3 = \{3,6\}$, the loads are $(16,10,10)$ and

$$LIST(I) = 16.$$

2. Job $\ell = 7$ ends last: it starts at $S_7 = 10$ on machine 1 and ends at 16. The bound gives

$$S_7 = 10 \;\le\; \frac{1}{3}\sum_{j\neq 7} t_j = \frac{2+2+4+8+8+6}{3} = \frac{30}{3} = 10,$$

which holds with equality: when job 7 arrives the three loads are equal, so the smallest one coincides with the average.

3. The jobs sorted by non-increasing processing time (ties by index) are $\lambda = (4,5,6,7,3,1,2)$:

job j	t_j	loads (L_1, L_2, L_3) before	machine
4	8	$(0,0,0)$	1
5	8	$(8,0,0)$	2
6	6	$(8,8,0)$	3
7	6	$(8,8,6)$	3
3	4	$(8,8,12)$	1
1	2	$(12,8,12)$	2
2	2	$(12,10,12)$	2

Hence $J_1 = \{3,4\}$, $J_2 = \{1,2,5\}$, $J_3 = \{6,7\}$, the loads are $(12,12,12)$ and

$$LPT(I) = 12.$$

4. The lower bound is

$$z(I) \;\ge\; \max\left\{8,\; \frac{2+2+4+8+8+6+6}{3}\right\} = \max\left\{8,\; \frac{36}{3}\right\} = 12.$$

LPT attains it, so $z(I) = 12$ and the schedule of LPT is optimal. The ratios are

$$\frac{LIST(I)}{z(I)} = \frac{16}{12} = \frac{4}{3} \;\le\; 2 - \frac{1}{3} = \frac{5}{3}, \qquad \frac{LPT(I)}{z(I)} = \frac{12}{12} = 1 \;\le\; \frac{4}{3} - \frac{1}{9} = \frac{11}{9}.$$

Both guarantees hold. With the list in the input order, the two longest jobs arrive when the machines are already loaded, and $LIST$ is 4 units above the optimum; sorting the list puts them first and closes the gap.

Exercise 7.2 — $P\,||\,C_{\max}$ approximation 2

We consider $m = 3$ identical machines and a set $\mathscr{J} = \{1, \dots, 7\}$ of 7 jobs ($n = 7$), with the following processing times:

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$
t_j	1	1	1	1	1	1	3

Questions

1. Apply $LIST$ with the list $\lambda = (1, 2, \dots, 7)$ and give $LIST(I)$.
2. Determine $z(I)$, by exhibiting an optimal schedule and a matching lower bound.
3. Compute $\frac{LIST(I)}{z(I)}$ and compare it with the guarantee of $LIST$. Which instance of the notes is this?
4. Apply LPT and give $LPT(I)$. Explain why sorting the list removes the bad case.

Solution

1.

job j	t_j	loads (L_1, L_2, L_3) before	machine
1	1	$(0,0,0)$	1
2	1	$(1,0,0)$	2
3	1	$(1,1,0)$	3
4	1	$(1,1,1)$	1
5	1	$(2,1,1)$	2
6	1	$(2,2,1)$	3
7	3	$(2,2,2)$	1

The loads are $(5, 2, 2)$, with $J_1 = \{1, 4, 7\}$, $J_2 = \{2, 5\}$, $J_3 = \{3, 6\}$, and

$$LIST(I) = 5.$$

2. The schedule $J_1 = \{7\}$, $J_2 = \{1, 2, 3\}$, $J_3 = \{4, 5, 6\}$ has loads $(3, 3, 3)$ and makespan 3. The lower bound gives

$$z(I) \;\geq\; \max\left\{3,\; \frac{6 \cdot 1 + 3}{3}\right\} = \max\{3, 3\} = 3.$$

Hence $z(I) = 3$.

3.
$$\frac{LIST(I)}{z(I)} = \frac{5}{3} = 2 - \frac{1}{3},$$

exactly the guarantee $2 - \frac{1}{m}$ for $m = 3$: the bound of $LIST$ is attained. This is the tight instance I_m of the notes for $m = 3$: $m(m-1) = 6$ unit jobs followed by one job

of length $m = 3$. The unit jobs fill all the machines up to $m - 1 = 2$, and the long job starts on top of one of them.

4. The sorted list is $\lambda = (7, 1, 2, 3, 4, 5, 6)$. Job 7 goes to machine 1, with loads $(3, 0, 0)$; the unit jobs then go to machines $2, 3, 2, 3, 2, 3$ (machine 1 is never the least loaded again, since its load 3 is never smaller than the others'), giving the loads $(3, 3, 3)$. Hence
$$LPT(I) = 3 = z(I).$$
The long job is the one that hurts when it comes last; LPT schedules it first, and the short jobs level the loads around it.

Exercise 7.3 — $P\,||\,C_{\max}$ approximation 3

We consider $m = 4$ identical machines and a set $\mathscr{J} = \{1, \ldots, 9\}$ of 9 jobs ($n = 9$), with the following processing times:

	$j=1$	$j=2$	$j=3$	$j=4$	$j=5$	$j=6$	$j=7$	$j=8$	$j=9$
t_j	7	7	6	6	5	5	4	4	4

Questions

1. Apply LPT and give $LPT(I)$.
2. Determine $z(I)$.
3. Compute $\frac{LPT(I)}{z(I)}$ and compare it with the guarantee $\frac{4}{3} - \frac{1}{3m}$.
4. In the proof of the guarantee of LPT, which of the two cases applies to the job ℓ that ends last? Check that every inequality of that case holds with equality on this instance.

Solution

1. The jobs are already sorted, $\lambda = (1, 2, \ldots, 9)$:

job j	t_j	loads (L_1, L_2, L_3, L_4) before	machine
1	7	$(0,0,0,0)$	1
2	7	$(7,0,0,0)$	2
3	6	$(7,7,0,0)$	3
4	6	$(7,7,6,0)$	4
5	5	$(7,7,6,6)$	3
6	5	$(7,7,11,6)$	4
7	4	$(7,7,11,11)$	1
8	4	$(11,7,11,11)$	2
9	4	$(11,11,11,11)$	1

Hence $J_1 = \{1,7,9\}$, $J_2 = \{2,8\}$, $J_3 = \{3,5\}$, $J_4 = \{4,6\}$, the loads are $(15,11,11,11)$ and

$$LPT(I) = 15.$$

2. The total processing time is 48, so $z(I) \geq \max\{7, \frac{48}{4}\} = 12$. The schedule $J_1 = \{1,5\}$, $J_2 = \{2,6\}$, $J_3 = \{3,4\}$, $J_4 = \{7,8,9\}$ has loads $(12,12,12,12)$. Hence $z(I) = 12$.

3. $$\frac{LPT(I)}{z(I)} = \frac{15}{12} = \frac{5}{4} = \frac{4}{3} - \frac{1}{12} = \frac{4}{3} - \frac{1}{3 \cdot 4}.$$

   The guarantee of LPT is attained: this is the tight instance of the notes for $m = 4$, with two jobs of each length $2m-1 = 7$, 6, $m+1 = 5$ and three jobs of length $m = 4$; after $2m = 8$ jobs all the loads are $3m - 1 = 11$, and the last job gives $4m - 1 = 15$, while $z(I) = 3m = 12$.

4. The last job is $\ell = 9$, with $t_9 = 4 = \frac{z(I)}{3}$, so Case 1 ($t_\ell \leq \frac{z}{3}$) applies, at its border. Its inequalities are:

$$\begin{aligned} LPT(I) = S_9 + t_9 &= 11 + 4 = 15, \\ \frac{1}{m}\sum_j t_j + \Big(1 - \frac{1}{m}\Big) t_9 &= \frac{48}{4} + \frac{3}{4} \cdot 4 = 15, \\ z(I) + \Big(1 - \frac{1}{m}\Big)\frac{z(I)}{3} &= 12 + \frac{3}{4} \cdot 4 = 15, \\ \Big(\frac{4}{3} - \frac{1}{3m}\Big) z(I) &= \frac{5}{4} \cdot 12 = 15. \end{aligned}$$

   All the quantities are equal: the starting time $S_9 = 11$ equals the average $\frac{48-4}{4} = 11$ of the other loads (the four machines are perfectly balanced), $\frac{1}{m}\sum_j t_j = z(I)$, and $t_9 = \frac{z(I)}{3}$.