Optimization

Fall semester 2022/23

Kristóf Bérczi

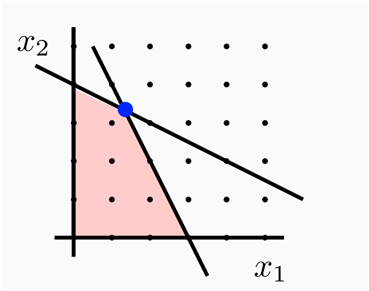
Eötvös Loránd University
Institute of Mathematics
Department of Operations Research

Lecture 2: Integer programming

Example revisited

Example:

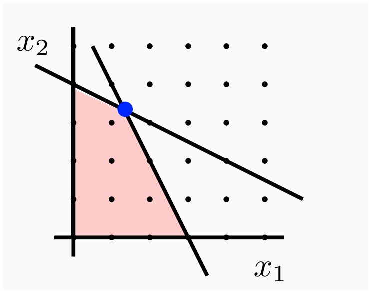
$$x_1 + 2 \cdot x_2 \leqslant 8$$
$$2 \cdot x_1 + x_2 \leqslant 6$$
$$x_1, x_2 \geqslant 0$$



Example revisited

Example:

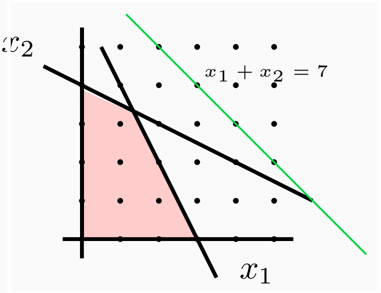
$$x_1 + 2 \cdot x_2 \leqslant 8$$
$$2 \cdot x_1 + x_2 \leqslant 6$$
$$x_1, x_2 \geqslant 0$$
$$x_1, x_2 \in \mathbb{Z}$$



Example revisited

Example:

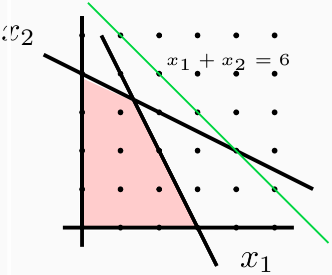
$$x_1 + 2 \cdot x_2 \leqslant 8$$
$$2 \cdot x_1 + x_2 \leqslant 6$$
$$x_1, x_2 \geqslant 0$$
$$x_1, x_2 \in \mathbb{Z}$$



Example revisited

Example:

$$x_1 + 2 \cdot x_2 \leqslant 8$$
$$2 \cdot x_1 + x_2 \leqslant 6$$
$$x_1, x_2 \geqslant 0$$
$$x_1, x_2 \in \mathbb{Z}$$



Another example

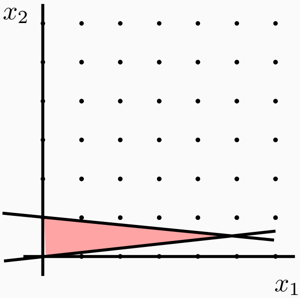
Example:

$$x_1 + 10 \cdot x_2 \leqslant 10$$

$$x_1 - 10 \cdot x_2 \leqslant 0$$

$$x_1, x_2 \geqslant 0$$

$$\max\{x_1\}$$



Another example

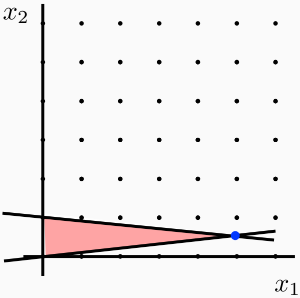
Example:

$$x_1 + 10 \cdot x_2 \leqslant 10$$

$$x_1 - 10 \cdot x_2 \leqslant 0$$

$$x_1, x_2 \geqslant 0$$

$$\max\{x_1\}$$



Another example

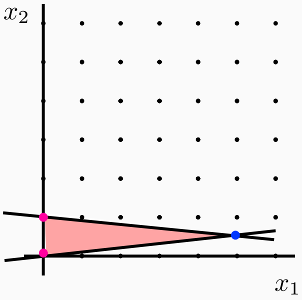
Example:

$$x_1 + 10 \cdot x_2 \leqslant 10$$

$$x_1 - 10 \cdot x_2 \leqslant 0$$

$$x_1, x_2 \geqslant 0$$

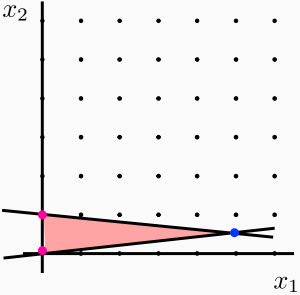
$$\max\{x_1\}$$



Another example

Example:

$$x_1 + 10 \cdot x_2 \leqslant 10$$
$$x_1 - 10 \cdot x_2 \leqslant 0$$
$$x_1, x_2 \geqslant 0$$
$$\max\{x_1\}$$



The fractional optimum can be far from the integer one.

Approaches

Bad news: integer programming is NP-complete

Approaches

Bad news: integer programming is NP-complete

Good news: there exist efficient algorithms

- totally unimodular matrices
  - every square submatrix has determinant 0, +1 or -1
- cutting plane methods
  - adding further inequalities that separate the actual optimum from the convex hull of the true feasible set
- branch and bound methods
  - systematically enumerating the candidate solutions, forming a rooted tree
- rounding methods (threshold rounding, iterative rounding)
  - rounding the coordinates of an optimal fractional solution
- heuristic methods (tabu search, hill climbing, simulated annealing, ant colony optimization, etc)
  - some would call these 'voodoo'...

Branch and bound I

$$\begin{aligned} &\min \ c(x) \\ &\text{s.t. } x \in F \end{aligned}$$

Here F is the set of integer feasible solutions to the problem.

Ideas:

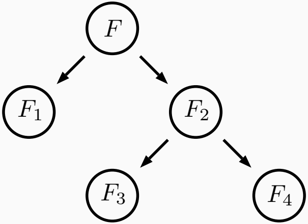
- Partition F into subsets $F_1, \ldots, F_k$, and solve the subproblems $\min c(x)$ s.t. $x \in F_i$. [May be as difficult as the original one, hence split into further subproblems - **branching part**.]
- Compute lower bounds $b(F_i)$ for the subproblems. [A lower bound might be easy to obtain, e.g. LP relaxation - **bounding part**.]
- Mainatin an upper bound U on the optimal cost. [E.g. the cost of the best feasible solution thus far.]

Key observation: If $b(F_i) \geqslant U$, then the subproblem need not be considered further.

Branch and bound II

Algorithm (general step):

1. Select an active subproblem F_i.
2. If the subproblem is infeasibe, delete it; otherwise compute $b(F_i)$.
    - If $b(F_i) \geqslant U$, delete the subproblem.
    - If $b(F_i) < U$, either determine an optimal solution for F_i, or break it into further (active) subproblems.



Remarks:

- Choosing the subproblem, e.g. BFS or DFS.
- Computing the lower bounds, e.g. LP relaxation.
- Breaking into subproblems.

Rounding methods

Given a minimization problem, an α-approximation algorithm provides a solution of value at most $\alpha \cdot OPT$.

Integer program
$\min c^T \cdot x$
$A \cdot x \leqslant b$
$x \in \mathbb{Z}^n$

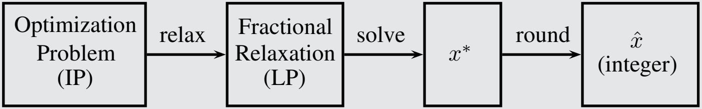
Naiv approach:

1. remove the integrality constraint,
2. solve the corresponding LP, and
3. round the entries of the solution to get an integer solution.

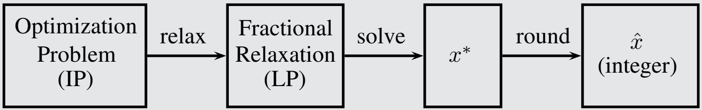
Problems:

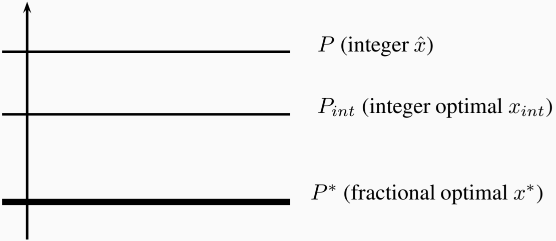
- the solution may not be feasible — Maintain feasibility.
- the solution may not be optimal — Approximation?

Analysing the solution

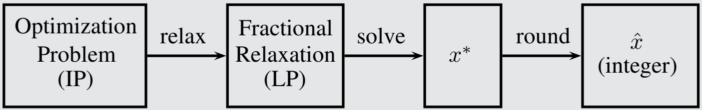


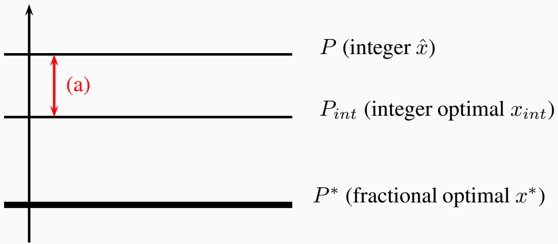
Analysing the solution





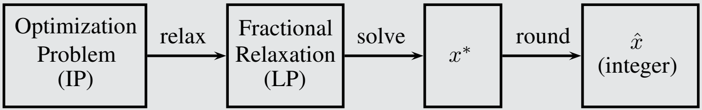
Analysing the solution

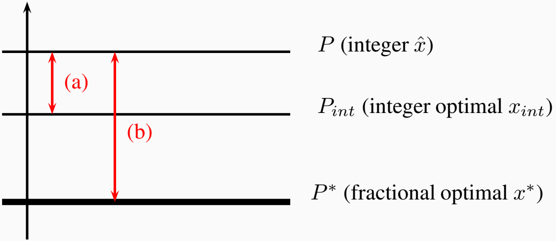




(a) = Approximation ratio between $\hat{x}$ and x_{int}.

Analysing the solution





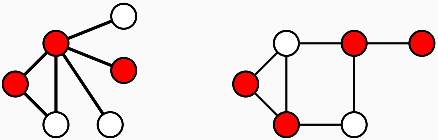
(a) = Approximation ratio between $\hat{x}$ and x_{int}.

(b) = Approximation ration between $\hat{x}$ and x^*.

Vertex cover I

Problem

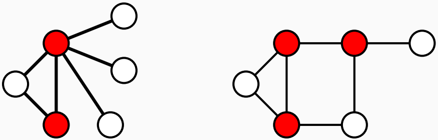
Find a minimum number of vertices covering every edge of a graph.



Vertex cover I

Problem

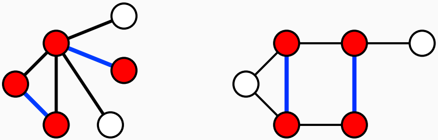
Find a minimum number of vertices covering every edge of a graph.



Vertex cover I

Problem

Find a minimum number of vertices covering every edge of a graph.



Simple algorithm:

Step 1. Take an inclusionwise maximal matching M.

Step 2. Consider the end vertices of the matching edges.

Observation

This gives a 2-approximation.

- One of Karp's 21 NP-complete problems.
- Moreover, it is APX-complete.
  - No better than 1.3606-approx. unless **P** = **NP**.
  - No better than 2-approx. assuming **UGC**.

Vertex cover II

IP formulation

$$\min \sum_{v \in V} x_v$$

$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$

$$x_v \in \{0, 1\} \quad \text{for } v \in V$$

Vertex cover II

IP formulation

$$\min \sum_{v \in V} x_v$$

$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$

$$x_v \in \{0, 1\} \quad \text{for } v \in V$$

LP relaxation

$$\min \sum_{v \in V} x_v$$

$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$

$$x_v \geqslant 0 \quad \text{for } v \in V$$

Vertex cover II

IP formulation

$$\min \sum_{v \in V} x_v$$
$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$
$$x_v \in \{0,1\} \quad \text{for } v \in V$$

LP relaxation

$$\min \sum_{v \in V} x_v$$
$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$
$$x_v \geqslant 0 \quad \text{for } v \in V$$

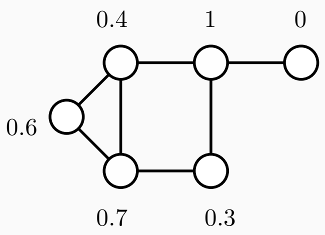
Step 1.

Take a fractional solution x^*.

Step 2.

Define

$$\hat{x}_v = \begin{cases} 1 & \text{if } x^*_v \geqslant 1/2, \\ 0 & \text{otherwise.} \end{cases}$$



Vertex cover II

IP formulation

$$\min \sum_{v \in V} x_v$$

$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$

$$x_v \in \{0,1\} \quad \text{for } v \in V$$

LP relaxation

$$\min \sum_{v \in V} x_v$$

$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$

$$x_v \geqslant 0 \quad \text{for } v \in V$$

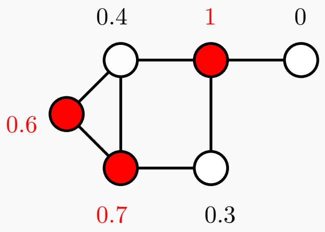
Step 1.

Take a fractional solution x^*.

Step 2.

Define

$$\hat{x}_v = \begin{cases} 1 & \text{if } x_v^* \geqslant 1/2, \\ 0 & \text{otherwise.} \end{cases}$$



Vertex cover II

IP formulation

$$\min \sum_{v \in V} x_v$$
$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$
$$x_v \in \{0,1\} \quad \text{for } v \in V$$

LP relaxation

$$\min \sum_{v \in V} x_v$$
$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$
$$x_v \geqslant 0 \quad \text{for } v \in V$$

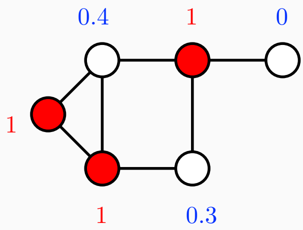
Step 1.

Take a fractional solution x^*.

Step 2.

Define

$$\hat{x}_v = \begin{cases} 1 & \text{if } x^*_v \geqslant 1/2, \\ 0 & \text{otherwise.} \end{cases}$$



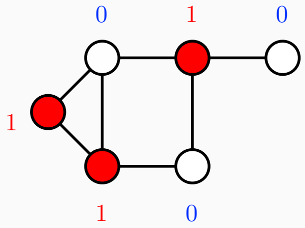
Vertex cover II

IP formulation

$$\min \sum_{v \in V} x_v$$
$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$
$$x_v \in \{0,1\} \quad \text{for } v \in V$$

LP relaxation

$$\min \sum_{v \in V} x_v$$
$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$
$$x_v \geqslant 0 \quad \text{for } v \in V$$



Step 1.

Take a fractional solution x^*.

Step 2.

Define

$$\hat{x}_v = \begin{cases} 1 & \text{if } x_v^* \geqslant 1/2, \\ 0 & \text{otherwise.} \end{cases}$$

Vertex cover II

IP formulation

$$\min \sum_{v \in V} x_v$$
$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$
$$x_v \in \{0,1\} \quad \text{for } v \in V$$

LP relaxation

$$\min \sum_{v \in V} x_v$$
$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$
$$x_v \geqslant 0 \quad \text{for } v \in V$$

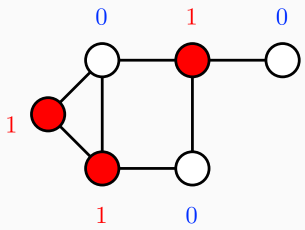
Step 1.

Take a fractional solution x^*.

Step 2.

Define

$$\hat{x}_v = \begin{cases} 1 & \text{if } x_v^* \geqslant 1/2, \\ 0 & \text{otherwise.} \end{cases}$$



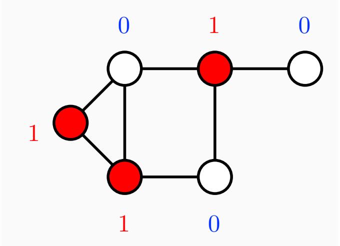
Vertex cover II

IP formulation

$$\min \sum_{v \in V} x_v$$
$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$
$$x_v \in \{0,1\} \quad \text{for } v \in V$$

LP relaxation

$$\min \sum_{v \in V} x_v$$
$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$
$$x_v \geqslant 0 \quad \text{for } v \in V$$



Step 1.

Take a fractional solution x^*.

Step 2.

Define

$$\hat{x}_v = \begin{cases} 1 & \text{if } x_v^* \geqslant 1/2, \\ 0 & \text{otherwise.} \end{cases}$$

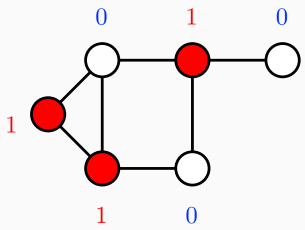
Vertex cover II

IP formulation

$$\min \sum_{v \in V} x_v$$
$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$
$$x_v \in \{0,1\} \quad \text{for } v \in V$$

LP relaxation

$$\min \sum_{v \in V} x_v$$
$$x_u + x_v \geqslant 1 \quad \text{for } uv \in E$$
$$x_v \geqslant 0 \quad \text{for } v \in V$$



Step 1.

Take a fractional solution x^*.

Step 2.

Define

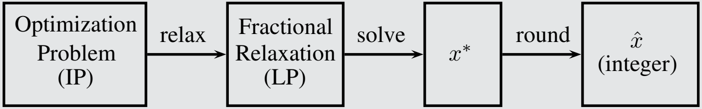
$$\hat{x}_v = \begin{cases} 1 & \text{if } x_v^* \geqslant 1/2, \\ 0 & \text{otherwise.} \end{cases}$$

Proof.
Note that $\hat{x}$ is integral, feasible, and $\hat{x}_v \leqslant 2 \cdot x_v^*$. Hence

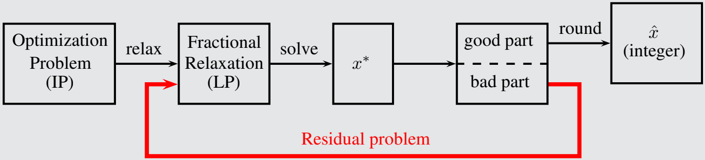
$$\sum_{v \in V} \hat{x}_v \leqslant 2 \cdot \sum_{v \in V} x_v^* \leqslant 2 \cdot OPT.$$

Threshold vs. iterative rounding

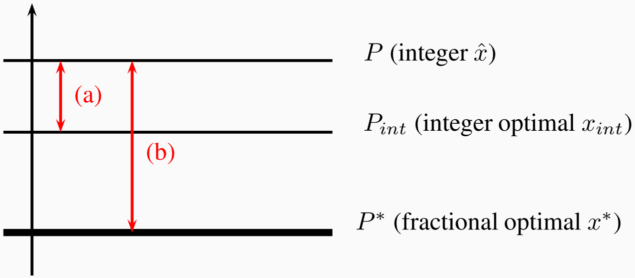
Threshold rounding



Iterative rounding



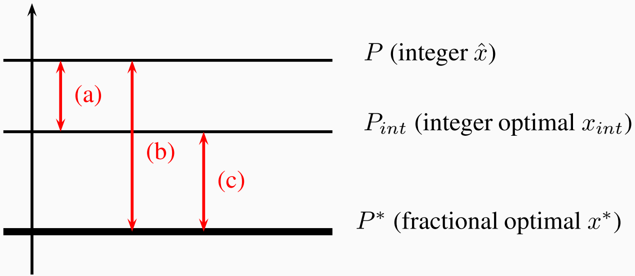
Integrality gap



(a) = Approximation ratio between $\hat{x}$ and x_{int}.

(b) = Approximation ration between $\hat{x}$ and x^*.

Integrality gap



(a) = Approximation ratio between $\hat{x}$ and x_{int}.

(b) = Approximation ration between $\hat{x}$ and x^*.

(c) = Integrality gap.

Heuristics - Local search

$$\min \; c(x)$$
$$\text{s.t.} \; x \in F$$

Algorithm:

1. Start at some $x \in F$.
2. Evaluate $c(x)$, and evaluate $c(y)$ for "neighbors" $y \in F$ of x.
   - If $c(y) < c(x)$, the move to y and repeat.
   - Otherwise stop: *local optimum* has been found.

Remarks:

- Specifics are determined once "neighbors" are defined.
- Simplex method can be viewed as a special case.
- In practice: run repeatedly starting from different initial solutions.
- Tradeoff: **better solution** is likely to obtained when considering **larger neighborhood**, but this results in **slower running time**.

Heuristics - Simulated annealing I

Main drawback of local search: Only finds local minimum.

Idea: Allow occasional moves to feasible solutions with higher costs.

Algorithm: For every state $x \in F$, a set $N(x) \subseteq F$ of neighbors is given ($y \in N(x) \Leftrightarrow x \in N(y)$).

1. Start from state $x \in F$.
2. Select a random neighbor y of x with probability q_{xy}.
   [Here $q_{xy} \geqslant 0$ and $\sum_{y \in N(x)} q_{xy} = 1$.]
3. Compute the difference $c(y) - c(x)$.
   - If $c(y) \leqslant c(x)$, then move to state y.
   - If $c(y) > c(x)$, then move to state y with probability $e^{-(c(y)-c(x))/T}$.

Remarks:

- When the temperature T is small - cost increases are unlikely.
- When T is large - the value of $c(y) - c(x)$ has insignificant effect.

Heuristics - Simulated annealing II

The procedure evolves as a Markov chain. Let $A = \sum_{z \in F} e^{-c(z)/T}$.

Steady-state distribution:

$$\pi(x) = \frac{e^{-c(x)/T}}{A},$$

$\Rightarrow \pi(x)$ falls exponentially with $c(x)$. Hence if T is small, then almost all of the steady-state probability is concentrated on states minimizing $c(x)$ **globally**. Should we set T to some very small constant then?

Drawback: the lower the value of T, the harder it is to escape from a local minimum and the longer it takes to reach steady-state.

Instead: Let the temperature vary with time:

$$T(t) = \frac{C}{\log t}.$$

Thm.

If C is sufficiently large, then $\lim_{t \to \infty} P(x(t) \text{ is optimal}) = 1$.

Reading assignment

D. Bertsimas, J.N. Tsitsiklis. Introduction to linear optimization.

- Chapter 11, Sections 11.2, 11.6, and 11.7

Exercises

Submission deadline: The starting time of the next lecture.

1. Consider the following integer programming problem.

$$\begin{aligned} \text{maximize} \quad & x_1 + 2x_2 \\ \text{subject to} \quad & -3x_1 + 4x_2 \leqslant 4 \\ & 3x_1 + 2x_2 \leqslant 11 \\ & 2x_1 - x_2 \leqslant 5 \\ & x_1, x_2 \geqslant 0 \\ & x_1, x_2 \quad \text{integer} \end{aligned}$$

   Use a figure to answer the following questions.

   a. What is the optimal cost of the linear programming relaxation? What is the optimal cost of the integer programming problem? (1pt)
   b. What is the convex hull of the set of all solutions to the integer programming problem? (1pt)

Exercises

2. A company is manufacturing k different products using m resources. The amounts of available resources are given, together with the requirement of each of them for the different products. The selling price of the products are also known.
   a. Write up an IP model that aims at maximizing the total profit. (1pt)
   b. Adjust the model if starting the production of product i requires a cost of s_i. (1pt)
3. Consider the integer programming problem

$$\begin{aligned} \text{minimize} \quad & x_{n+1} \\ \text{subject to} \quad & 2x_1 + 2x_2 + \cdots + 2x_n + x_{n+1} = n \\ & x_i \in \{0, 1\} \end{aligned}$$

   Show that any branch and bound algorithm that uses LP relaxations to compute lower bounds, and branches by setting a fractional variable to either zero or one, will require the enumeration of an exponential number of subproblems when n is odd. (2pts)

Exercises

4. The pagination problem faced by a document processing program like LaTeX can be abstracted as follows. The text consists of a sequence $1, \ldots, n$ of n items (words, formulas, etc.). A page that starts with item i and ends with item j is assigned an attractiveness factor c_{ij}. Assuming that the factors c_{ij} are available, we wish to maximize the total attractiveness of the paginated text. Develop an algorithm for this problem. (Hint: try to use recursive approach.) (2pts)