Lecture 1: Definition and Complexity of Packing Problems

Helmut Alt, Freie Universität Berlin
at 10th Winter School on Computational Geometry
Amirkabir University of Technology, Tehran

28 February 2018

Packing Problems

Packing Problems

Decision Problem

Packing Problems

Decision Problem

given:
a set $\mathcal{O}$ of geometric objects and a container C

Packing Problems

Decision Problem

given:
a set $\mathcal{O}$ of geometric objects and a container C

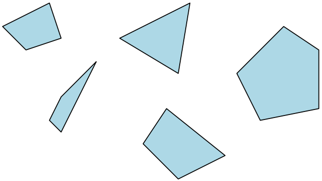
decide:
can the objects in $\mathcal{O}$ be packed without overlap into C?

Packing Problems

Decision Problem

given:
a set $\mathcal{O}$ of geometric objects and a container C

decide:
can the objects in $\mathcal{O}$ be packed without overlap into C?

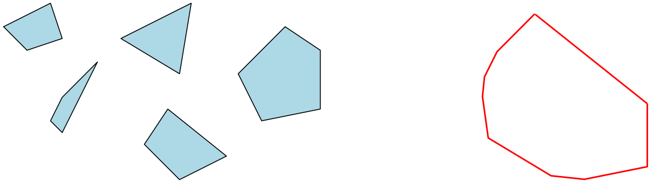


Packing Problems

Decision Problem

given:
a set $\mathcal{O}$ of geometric objects and a container C

decide:
can the objects in $\mathcal{O}$ be packed without overlap into C?

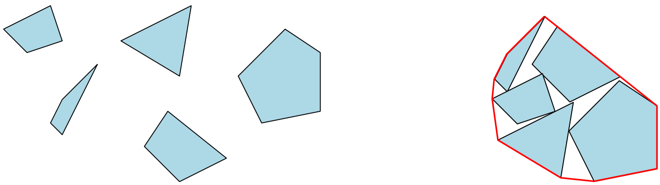


Packing Problems

Decision Problem

given:
a set $\mathcal{O}$ of geometric objects and a container C

decide:
can the objects in $\mathcal{O}$ be packed without overlap into C?



Packing Problems

Packing Problems

Optimization Problem

Packing Problems

Optimization Problem

given:
a set $\mathcal{O}$ of geometric objects

Packing Problems

Optimization Problem

given:
a set $\mathcal{O}$ of geometric objects

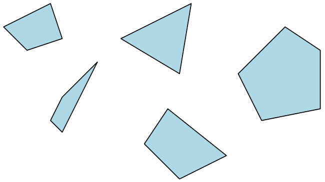
find:
the smallest container C into which the objects in $\mathcal{O}$ be packed

Packing Problems

Optimization Problem

given:
a set $\mathcal{O}$ of geometric objects

find:
the smallest container C into which the objects in $\mathcal{O}$ be packed

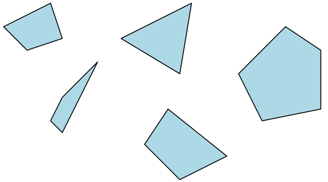


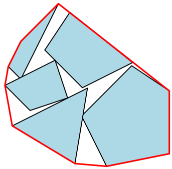
Packing Problems

Optimization Problem

given:
a set $\mathcal{O}$ of geometric objects

find:
the smallest container C into which the objects in $\mathcal{O}$ be packed





Very Many Variants

Very Many Variants

- **objects:**
  convex polygons (simple polygons, line segments, circles, polyhedra, spheres...)

Very Many Variants

- **objects:**
  convex polygons (simple polygons, line segments, circles, polyhedra, spheres...)
- **feasible motion:**
  translation or rigid motion (= translation and rotation)

Very Many Variants

- **objects:**
  convex polygons (simple polygons, line segments, circles, polyhedra, spheres...)
- **feasible motion:**
  translation or rigid motion (= translation and rotation)
- **containers:**
  convex polygons/polyhedra, rectangles/cuboids, strips of a certain width/cross section

Very Many Variants

- **objects:**
  convex polygons (simple polygons, line segments, circles, polyhedra, spheres...)
- **feasible motion:**
  translation or rigid motion (= translation and rotation)
- **containers:**
  convex polygons/polyhedra, rectangles/cuboids, strips of a certain width/cross section
- **objective function:**
  area/volume, circumference/surface area, diameter

Applications

Applications

- apparel industry

Applications

- apparel industry
- sheet metal processing

Applications

- apparel industry
- sheet metal processing
- storage

NP-completeness 1

NP-completeness 1

complexity class **P**

NP-completeness 1

complexity class **P**
(Decision-)problems, which can be answered efficiently, i.e., in running time polynomial in the size of the input.

NP-completeness 1

complexity class **P**
(Decision-)problems, which can be answered efficiently, i.e., in running time polynomial in the size of the input.

complexity class **NP**

NP-completeness 1

complexity class **P**
(Decision-)problems, which can be answered efficiently, i.e., in running time polynomial in the size of the input.

complexity class **NP**
problems, for which a given positive solution can be verified efficiently.

NP-completeness 1

complexity class **P**
(Decision-)problems, which can be answered efficiently, i.e., in running time polynomial in the size of the input.

complexity class **NP**
problems, for which a given positive solution can be verified efficiently.

Example: PARTITION

NP-completeness 1

complexity class **P**
(Decision-)problems, which can be answered efficiently, i.e., in running time polynomial in the size of the input.

complexity class **NP**
problems, for which a given positive solution can be verified efficiently.

Example: PARTITION

given: a sequence $S = (a_1, ..., a_n)$ of integers

NP-completeness 1

complexity class **P**

(Decision-)problems, which can be answered efficiently, i.e., in running time polynomial in the size of the input.

complexity class **NP**

problems, for which a given positive solution can be verified efficiently.

Example: PARTITION

given: a sequence $S = (a_1, ..., a_n)$ of integers

question: Is it possible to decompose S into two subsequences whose sums are equal?

NP-completeness 1

complexity class **P**
(Decision-)problems, which can be answered efficiently, i.e., in running time polynomial in the size of the input.

complexity class **NP**
problems, for which a given positive solution can be verified efficiently.

Example: PARTITION

given: a sequence $S = (a_1, ..., a_n)$ of integers

question: Is it possible to decompose S into two subsequences whose sums are equal?

example:

NP-completeness 1

complexity class **P**
(Decision-)problems, which can be answered efficiently, i.e., in running time polynomial in the size of the input.

complexity class **NP**
problems, for which a given positive solution can be verified efficiently.

Example: PARTITION

given: a sequence $S = (a_1, ..., a_n)$ of integers

question: Is it possible to decompose S into two subsequences whose sums are equal?

example:
15 217 123 82 28 17 46 47 53 3 63 21 93

NP-completeness 1

complexity class **P**
(Decision-)problems, which can be answered efficiently, i.e., in running time polynomial in the size of the input.

complexity class **NP**
problems, for which a given positive solution can be verified efficiently.

Example: PARTITION

given: a sequence $S = (a_1, ..., a_n)$ of integers

question: Is it possible to decompose S into two subsequences whose sums are equal?

example:
15 217 123 82 28 17 46 47 53 3 63 21 93
15 217 123 82 28 17 46 47 53 3 63 21 93

NP-completeness 2

NP-completeness 2

grand open problem: Is $\mathbf{P = NP?}$

NP-completeness 2

grand open problem: Is **$\mathbf{P = NP}$?**

A problem A is called **NP-hard**, if any problem in NP can be efficiently **reduced** to A, **NP-complete**, if it is in NP itself.

NP-completeness 2

grand open problem: Is $\mathbf{P = NP?}$

A problem A is called **NP-hard**, if any problem in NP can be efficiently **reduced** to A, **NP-complete**, if it is in NP itself.

Example: PARTITION

NP-completeness 2

grand open problem: Is **P = NP?**

A problem A is called **NP-hard**, if any problem in NP can be efficiently **reduced** to A, **NP-complete**, if it is in NP itself.

Example: PARTITION

Consequently:

NP-completeness 2

grand open problem: Is **P = NP?**

A problem A is called **NP-hard**, if any problem in NP can be efficiently **reduced** to A, **NP-complete**, if it is in NP itself.

Example: PARTITION

Consequently:

- If there is an efficient algorithm for an NP-hard problem, then P=NP.

NP-completeness 2

grand open problem: Is **P = NP?**

A problem A is called **NP-hard**, if any problem in NP can be efficiently **reduced** to A, **NP-complete**, if it is in NP itself.

Example: PARTITION

Consequently:

- If there is an efficient algorithm for an NP-hard problem, then P=NP.
- If an NP-hard problem A can be reduced efficiently to a problem B, then B is NP-hard, as well.

NP-completeness 2

grand open problem: Is **$\mathbf{P = NP}$?**

A problem A is called **NP-hard**, if any problem in NP can be efficiently **reduced** to A, **NP-complete**, if it is in NP itself.

Example: PARTITION

Consequently:

- If there is an efficient algorithm for an NP-hard problem, then P=NP.
- If an NP-hard problem A can be reduced efficiently to a problem B, then B is NP-hard, as well.

It is well known: NP $\subseteq$ PSPACE, the class of all problems solvable in polynomial *space*.

Exact Solution

Exact Solution

Milenkovic, Daniels STOC 1996, PhD thesis

Exact Solution

Milenkovic, Daniels STOC 1996, PhD thesis

packing by translation, techniques from linear programming

Exact Solution

Milenkovic, Daniels STOC 1996, PhD thesis

packing by translation, techniques from linear programming

all bounds are of the form $(m + n)^{O(k)}$
for k simple m-gons, container a simple n-gon, even if not convex

Exact Solution

Milenkovic, Daniels STOC 1996, PhD thesis

packing by translation, techniques from linear programming

all bounds are of the form $(m+n)^{O(k)}$
for k simple m-gons, container a simple n-gon, even if not convex

cooperation with apparel industry

Packing Axes-Parallel Recangles PAR

Packing Axes-Parallel Recangles PAR

given: a set of rectangles $\mathcal{R}$, and a rectangular container R
question: can the rectangles of $\mathcal{R}$ be packed into R under translation?

Packing Axes-Parallel Recangles PAR

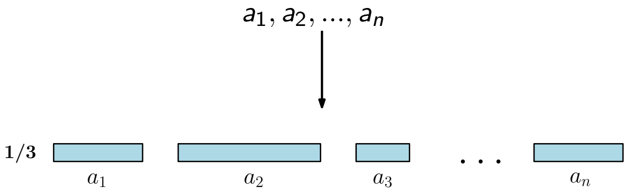
given: a set of rectangles $\mathcal{R}$, and a rectangular container R
question: can the rectangles of $\mathcal{R}$ be packed into R under translation?

- Claim: PARTITION is efficiently reducible to PAR.

Packing Axes-Parallel Recangles PAR

given: a set of rectangles $\mathcal{R}$, and a rectangular container R
question: can the rectangles of $\mathcal{R}$ be packed into R under translation?

- Claim: PARTITION is efficiently reducible to PAR.

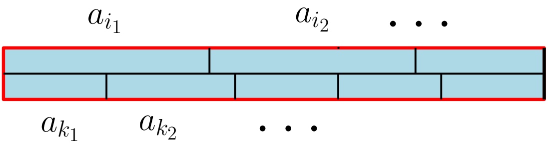


Packing Axes-Parallel Recangles PAR

given: a set of rectangles $\mathcal{R}$, and a rectangular container R
question: can the rectangles of $\mathcal{R}$ be packed into R under translation?

- Claim: PARTITION is efficiently reducible to PAR.

partition $a_{i_1} + \ldots + a_{i_l} = a_{k_1} + \ldots + a_{k_m}$ is possible, iff



perfect packing is possible

Packing Axes-Parallel Recangles PAR

given: a set of rectangles $\mathcal{R}$, and a rectangular container R
question: can the rectangles of $\mathcal{R}$ be packed into R under translation?

- Claim: PARTITION is efficiently reducible to PAR.
- Consequently: PAR is NP-complete.

Packing Axes-Parallel Recangles PAR

given: a set of rectangles $\mathcal{R}$, and a rectangular container R
question: can the rectangles of $\mathcal{R}$ be packed into R under translation?

- Claim: PARTITION is efficiently reducible to PAR.

- Consequently: PAR is NP-complete.

- also, it follows or can easily be shown:
  - for rectangular/convex polygonal containers the corresponding optimization problems are NP-hard.

Packing Axes-Parallel Recangles PAR

given: a set of rectangles $\mathcal{R}$, and a rectangular container R
question: can the rectangles of $\mathcal{R}$ be packed into R under translation?

- Claim: PARTITION is efficiently reducible to PAR.

- Consequently: PAR is NP-complete.

- also, it follows or can easily be shown:
  - packing of rectangles/arbitrary polygons into rectangular/polygonal containers under translation/rigid motion is NP-hard.
  - for rectangular/convex polygonal containers the corresponding optimization problems are NP-hard.

3-Approximation

3-Approximation

Approximation algorithms for NP-hard optimization problems:

3-Approximation

Approximation algorithms for NP-hard optimization problems: Find efficiently an α-**approximation**, i.e., a solution at most a factor α worse than an optimal one.

3-Approximation

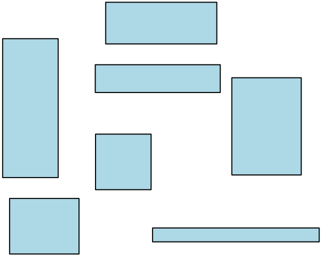
Approximation algorithms for NP-hard optimization problems: Find efficiently an **α-approximation**, i.e., a solution at most a factor α worse than an optimal one.

3-approximation for packing axes-parallel rectangles under translation

3-Approximation

Approximation algorithms for NP-hard optimization problems: Find efficiently an **α-approximation**, i.e., a solution at most a factor α worse than an optimal one.

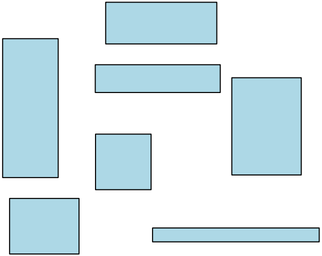
3-approximation for packing axes-parallel rectangles under translation

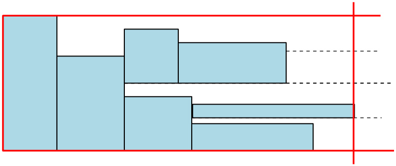


3-Approximation

Approximation algorithms for NP-hard optimization problems: Find efficiently an **α-approximation**, i.e., a solution at most a factor α worse than an optimal one.

3-approximation for packing axes-parallel rectangles under translation

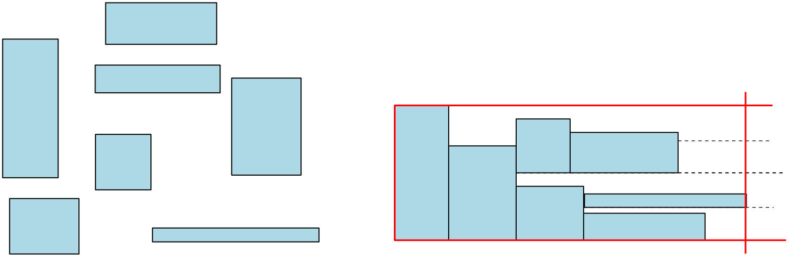




3-Approximation

Approximation algorithms for NP-hard optimization problems: Find efficiently an **α-approximation**, i.e., a solution at most a factor α worse than an optimal one.

3-approximation for packing axes-parallel rectangles under translation



also possible:
3-approximation for packing axes-parallel rectangles under rigid motion

12-Approximation for Convex Polygons

12-Approximation for Convex Polygons

under rigid motion, where the container is an axes-parallel rectangle.

12-Approximation for Convex Polygons

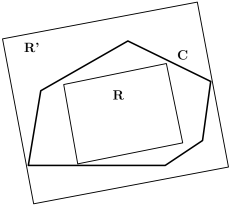
under rigid motion, where the container is an axes-parallel rectangle.

Fact:(Fuchs et al. 90)To any convex polygon C a rectangle $R \subseteq C$ can be computed efficiently where $C \subseteq R'$ and R' is a translate of $2R$. (cf. Theorem by Löwner/John for ellipses)

12-Approximation for Convex Polygons

under rigid motion, where the container is an axes-parallel rectangle.

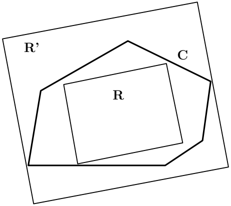
Fact:(Fuchs et al. 90)To any convex polygon C a rectangle $R \subseteq C$ can be computed efficiently where $C \subseteq R'$ and R' is a translate of $2R$. (cf. Theorem by Löwner/John for ellipses)



12-Approximation for Convex Polygons

under rigid motion, where the container is an axes-parallel rectangle.

Fact:(Fuchs et al. 90)To any convex polygon C a rectangle $R \subseteq C$ can be computed efficiently where $C \subseteq R'$ and R' is a translate of $2R$. (cf. Theorem by Löwner/John for ellipses)

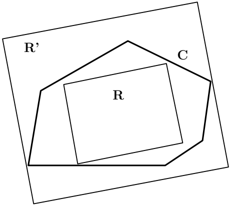


It follows:

12-Approximation for Convex Polygons

under rigid motion, where the container is an axes-parallel rectangle.

Fact:(Fuchs et al. 90)To any convex polygon C a rectangle $R \subseteq C$ can be computed efficiently where $C \subseteq R'$ and R' is a translate of $2R$. (cf. Theorem by Löwner/John for ellipses)



It follows: Any α-approximative packing algorithm for rectangular objects can be used to get a 4α-approximation algorithm for convex polygonal objects.