Approximation Algorithms for Rectangle Packing

Arindam Khan
IISc Bangalore
Talk Overview

• Approximation Algorithms.
• Bin Packing and Knapsack.
• Rectangle Packing Problems:
  1. 2D Geometric Bin Packing (2BP),
  2. 2D Strip Packing (2SP),
  3. Dynamic Storage Allocation (DSA),
  4. 2D Geometric Knapsack (2GK),
  5. Unsplittable Flow on a Path (UFP),
  6. Storage Allocation Problem (SAP),
  7. Maximum Weight Independent Set of Rectangles (MISR).
Approximation Algorithms

• Approximation algorithms are efficient algorithms that find near-optimal solution.

• For a minimization problem, an algorithm $A$ is $\alpha$-(absolute) approximation ($\alpha>1$) if $A(I) \leq \alpha \cdot OPT(I)$ for all input instances $I$.

• For a minimization problem, an algorithm $A$ is $\alpha$-asymptotic approximation ($\alpha>1$) if $A(I) \leq \alpha \cdot OPT(I) + O(1)$ for all input instances $I$. 
PTAS
PTAS

• Polynomial Time Approximation Schemes (PTAS):
  If for every $\varepsilon > 0$, there exists a poly-time \( O(n^{f(\varepsilon)}) \)-time) algorithm \( A_\varepsilon \) such that
  \[ A_\varepsilon(I) \leq (1 + \varepsilon) \text{OPT}(I). \]

• Efficient PTAS (EPTAS): if running time is \( O(f(\varepsilon).n^c) \).

• Fully PTAS (FPTAS): if running time is \( O((n/\varepsilon)^c) \).

• Asymptotic PTAS (APTAS): \( A_\varepsilon(I) \leq (1 + \varepsilon) \text{OPT}(I) + O(1). \)

• QuasiPTAS (QPTAS): \((1 + \varepsilon)\)-approximation in \( n^{(\log n)^{O(1)}} \)-time.

• PseudoPTAS (PPTAS): \((1 + \varepsilon)\)-approximation in \( n^{O(1)} \)-time, where \( n \) is the number of items and the numeric data is polynomially bounded in \( n \).
I get ideas about what's essential when packing my suitcase.

— Diane von Furstenberg
Hoffman’s Packing Puzzle

• **Given:** Twenty-seven identical blocks with dimensions \(A \times B \times C\) where \(\frac{A + B + C}{4} < A < B < C\).
• **Goal:** Pack all blocks into a box with sides \(A + B + C\).
  
  *(e.g., \(A = 4, B = 5, C = 6\).)*
Packing Problems

• **Goal:** Pack some items under some **constraints**.
• **Example:** Bin Packing, Knapsack – two classical **NP-hard** problems.
Packing Problems

• Studied in computer science and optimization from 1960’s. [Gilmore-Gomory, Operations Research ‘61]
• Among Karp's 21 NP-complete problems.
• The cornerstone of approximation algorithms.
• The term approximation algorithms was first coined for near-optimal bin packing algorithms [Johnson, STOC ’73].
• Knapsack and Bin Packing has most needed implementations among all NP-hard problems [Market Research by Skiena, ‘99].
Bin Packing Problem

- **Given**: $n$ items with sizes $s_1, s_2, \ldots, s_n$, s.t. $s_i \in (0,1]$.
- **Goal**: Pack all items into min # of unit bins.

  - Example: items $\{0.8, 0.6, 0.3, 0.2, 0.1\}$ can be packed in 2 unit bins: $\{0.8, 0.2\}$ and $\{0.6, 0.3, 0.1\}$.
  - 3/2 hardness of approximation (from *Partition*).
    - This does not rule out OPT+1 guarantee.
  - delaVega-Lueker, Combinatorica ’81: APTAS,
  - Karp-Karmarkar, FOCS ’82: $\text{OPT} + O(\log^2(\text{OPT}))$,
  - Hoberg-Rothvoss, SODA ’17: $\text{OPT} + O(\log(\text{OPT}))$. 
Knapsack Problem

- **Given:** \( I \), a set of \( n \) items where item \( i \) has profit \( p_i \in \mathbb{Z} \) and size \( s_i \in \mathbb{Z} \); Knapsack of size \( W \in \mathbb{Z} \).

- **Goal:** Find the maximum profit subset \( S \subseteq I \) that can be packed in the knapsack, i.e., the total size of items in \( S \) is at most \( W \).

- **Weakly NP-hard:** Admits pseudo-polynomial time exact algorithm.

- **FPTAS:** in \( O \left( n \log \frac{1}{\epsilon} + \frac{1}{\epsilon^4} \right) \)-time. [Lawler FOCS ’77].
1. 2-D Geometric Bin Packing
2-D Geometric Bin Packing

- **Given:** Collection of rectangles (by width, height),
- **Goal:** Pack them into minimum number of unit square bins.

- **Orthogonal Packing:** rectangles packed parallel to bin edges.
- **With 90 degree rotations and without rotations.**
Applications:

• Cloth cutting, steel cutting, wood cutting
• Placing ads in newspapers
• Memory allocation in paging systems

• Truck Loading
• Palletization by robots
### 2D BP: Tale of approximability

<table>
<thead>
<tr>
<th>Algorithm (Asymptotic)</th>
<th>Hardness</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.125 [Chung Garey Johnson, JACM ‘82]</td>
<td>No APTAS (from 3D Matching)</td>
</tr>
<tr>
<td>2+(\varepsilon) [Kenyon-Remilla, FOCS’96]</td>
<td>[Bansal-Sviridenko SODA’04],</td>
</tr>
<tr>
<td>1.69 [Caprara, FOCS’02]</td>
<td>3793/3792 (with rotation),</td>
</tr>
<tr>
<td>1.52 [Bansal-Caprara-Sviridenko, FOCS’06]</td>
<td>2197/2196 (w/o rotation)</td>
</tr>
<tr>
<td>1.5 [Jansen-Praedel, SODA’13]</td>
<td>[Chlebik-Chlebikova ‘09]</td>
</tr>
<tr>
<td><strong>1.405 [Bansal-K., SODA’14]</strong> (with and w/o rotations)</td>
<td></td>
</tr>
</tbody>
</table>

- \(d\)-dimensional \((d>2)\) geometric bin packing: \(1.69^{d-1}\) [Caprara, FOCS’02].
- APTAS for \(d\)-dimensional squares: [Bansal-Sviridenko, SODA’04].
Configuration LP

- \( \mathbb{C} \): set of configurations (possible way of feasibly packing a bin).

**Primal:**
\[
\min \left\{ \sum_{C} x_C : \sum_{C \ni i} x_C \geq 1 \ (i \in I), x_C \geq 0 \ (C \in \mathbb{C}) \right\}
\]

**Dual:**
\[
\max \left\{ \sum_{i \in I} v_i : \sum_{i \in C} v_i \leq 1 \ (C \in \mathbb{C}), v_i \geq 0 \ (i \in I) \right\}
\]

- **Problem:** Exponential number of configurations!
- **Solution:** Can be solved within \((1 + \epsilon)\) accuracy using separation problem for the dual.
Configuration LP

• $\mathbb{C}$: set of configurations (possible way of feasibly packing a bin).

Primal:
\[
\min \left\{ \sum_{C} x_C: \sum_{C \in i} x_C \geq 1 \text{ for all } i \in I, x_C \geq 0 \text{ (} C \in \mathbb{C} \text{)} \right\}
\]

Dual Separation problem $\Rightarrow$
2-D Geometric Knapsack:
Given one bin, pack as much area as possible.
- PTAS [BCJPS ISAAC 2009]

Dual:
\[
\max \left\{ \sum_{i \in I} v_i: \sum_{i \in C} v_i \leq 1 \text{ for all } C \in \mathbb{C}, v_i \geq 0 \text{ (} i \in I \) \right\}
\]

• Problem: Exponential number of configurations!
• Solution: Can be solved within $(1 + \epsilon)$ accuracy using separation problem for the dual.
Round and Approx (R&A) Framework [Bansal-K. ‘14]

• Given a packing problem $\Pi$

1. If the configuration LP is solved within $(1 + \epsilon)$ factor

\[
\min \left\{ \sum_{C} x_{C} \mid \sum_{C \ni i} x_{C} \geq 1 \ (i \in I), x_{C} \geq 0 \ (C \in \mathcal{C}) \right\}
\]

2. There is a $\rho$ approximation rounding-based algorithm.

• Then there is $(1 + \ln \rho)$ approximation for $\Pi$. 

2/10/20
Rounding based Algorithms:

• Rounding based algorithms are ubiquitous in bin packing.
• Items are replaced by slightly larger items from $O(1)$ types.
• **Loss:**
  Due to larger items.
• **Gain:**
  **Fewer configurations.** $O(1)$ types of large items imply rounded instance can be solved optimally.
• Example: Linear grouping [delaVega-Luker, Kenyon-Remilla], Geometric Grouping [Karp-Karmarkar], Harmonic Rounding [Lee-Lee, Caprara, Bansal et al.], JP rounding [JansenPradel].
Rounding based Algorithms in 2D

• **Classification** of items into big, wide, long, medium and small by defining two parameters $f(\epsilon)$ and $g(\epsilon) (\ll f(\epsilon))$ such that total volume of medium rectangles is $\epsilon \cdot Area(I)$. 

![Diagram showing classification of items into big, wide, long, medium, and small categories based on two parameters $f(\epsilon)$ and $g(\epsilon)$.](image)
Rounding based Algorithms in 2D

• Classification of items into big, wide, long, medium and small by defining two parameters \( f(\varepsilon) \) and \( g(\varepsilon) \)(\( \ll f(\varepsilon) \)) such that total volume of medium rectangles is \( \varepsilon \cdot Area(I) \).

Negligible volume

Can be well packed
Rounding based Algorithms

• Skewed (wide/long) items are packed into containers. (i) it has large size in each dimensions and (ii) items are packed into containers with a negligible loss of volume.

• Containers and big items are rounded to $O(1)$ types so that we can find near-optimal packing of big items and containers in polynomial time.
Rounding based Algorithms

- Each item is packed in $O(1)$-type of containers.

- Existence of such packing implies that constructively we can find it.
Round and Approx Framework (R & A)

• 1. Solve configuration LP using APTAS. Let \( z^* = \sum_{\{C \in \mathbb{C}\}} x^*_C \).

Primal:

\[
\min \left\{ \sum_C x_C : \sum_{C \ni i} x_C \geq 1 \ (i \in I), x_C \geq 0 \ (C \in \mathbb{C}) \right\}
\]
Round and Approx Framework (R & A)

• 1. Solve configuration LP using APTAS. Let \( z^* = \sum_{\{C \in \mathbb{C}\}} x_C^* \).

• 2. Randomized Rounding: For \( q \) iterations:

   select a configuration \( C' \) at random with probability \( \frac{x_{C'}^*}{z^*} \).

Primal:
\[
\min \{ \sum_C x_C : \sum_{C \ni i} x_C \geq 1 \ (i \in I), x_C \geq 0 \ (C \in \mathbb{C}) \}
\]
Round and Approx Framework (R & A)

1. Solve configuration LP using APTAS. Let $z^* = \sum_{\mathcal{C} \in \mathcal{C}} x_{C}^*$. 
2. Randomized Rounding: For $q$ iterations:
   - Select a configuration $C'$ at random with probability $\frac{x_{C'}^*}{z^*}$.
3. Approx: Apply a $\rho$ approximation rounding based algorithm $A$ on the residual instance $S$.
R & A Rounding Based Algorithms

• Probability item $i$ left uncovered after rand. rounding
  \[ = \left( 1 - \sum_{c \in i} \frac{x^*_c}{z^*} \right)^q \leq \frac{1}{\rho} \text{ by choosing } q = (\ln \rho)LP(I). \]

• Number of items of each type shrinks by a factor $\rho$
  e.g., $\mathbb{E}[|B_j \cap S|] = \frac{|B_j|}{\rho}.for$ some item type $B_j$.

• Concentration using Independent Bounded Difference Inequality.
Proof Sketch

• Rounding based Algo: $O(1)$ types of items 
  $= O(1)$ number of constraints in configuration LP.

• $ALGO(S) \approx OPT(\tilde{S}) \approx LP(\tilde{S})$.

• As # items for each item type shrinks by $\rho$, $LP(\tilde{S}) \approx \frac{1+\epsilon}{\rho} LP(\tilde{I})$.

• $\rho$ – approximation: $ALGO(I) \approx LP(\tilde{I}) \leq \rho OPT(I) + O(1)$.

• $ALGO(S) \approx OPT(I)$. 
Proof Sketch

• **Thm:** R&A gives a \((1 + \ln \rho + \epsilon)\) approximation.

• **Proof:**

  • Randomized Rounding: \(q = \ln \rho \cdot LP(I)\)
  • Residual Instance: \(S = (1 + \epsilon)OPT(I) + O(1)\).

• **Round + Approx** \(\Rightarrow (\ln \rho + 1 + \epsilon)OPT(I) + O(1)\).
Guillotine Packing

Guillotine Cut: Edge to Edge cut across a bin

Objective: Minimize number of bins such that packing in each bin is a guillotine packing.
Guillotine Packing => General Bin packing

Guillotine cut: edge to edge cut across a bin

- APTAS for guillotine 2-D bin packing [Bansal Lodi Sviridenko, FOCS’05].
- Conjecture: Given any packing of $m$ bins, there is a guillotine packing in $4m/3$ bins. This will imply $(\frac{4}{3} + \varepsilon)$-approximation for 2-D BP.
2. Strip Packing
Strip Packing Problem: (2-D)

• Input:
  - Rectangles $R_1, R_2, \ldots, R_n$: Each $R_i$ has integral width and height $(w_i, h_i)$.
  - A strip of integral width $W$ and infinite height.

• Goal:
  - Pack all rectangles minimizing the height of the strip.
  - Axis-parallel non-overlapping packing.

Variant 1:
No rotations are allowed!
Strip Packing Problem: (2-D)

**Input:**
- Rectangles $R_1, R_2, \ldots, R_n$: Each $R_i$ has integral width and height $(w_i, h_i)$.
- A strip of integral width $W$ and infinite height.

**Goal:**
- Pack all rectangles minimizing the height of the strip.
- Axis-parallel non-overlapping packing.

**Variant 2:**
90° rotations are allowed!
Strip Packing:

• Strip Packing generalizes
  - bin packing (when all rectangles have same height),
  - makespan minimization (when all rectangles have same width).
Tale of approximability.

• Asymptotic Approximation:
  • Asymptotic PTAS [Kenyon-Remila, FOCS’96] (Without rotations),
  • Asymptotic PTAS [Jansen-vanStee, STOC‘05] (With rotations).

• Absolute Approximation:
  • 2.7-appx. [First-Fit-Decreasing-Height, Coffman-Garey-Johnson-Tarjan ‘80].
  • 5/3+ε [Harren-Jansen-Pradel-vanStee, Comp.Geom.‘14].

• Hardness of appx in poly-time: 3/2 (from Bin Packing).
• Hardness of appx in pseudo poly-time: 5/4 (from 3-partition).
• Pseudo-polytime: (5/4+ε)-appx [Jansen-Rau ESA’19].
3. Dynamic Storage Allocation (DSA)
DSA

• **Input:** Rectangles $R_1, R_2, \ldots, R_n$; Each $R_i$ has width $w_i$, height $h_i$, and fixed starting position on x-coordinate $x_i$;
- A strip of integral width 1 and infinite height.

• **Goal:** Pack (non-overlapping and axis-parallel) all rectangles into the strip of minimum height by sliding the rectangles vertically but not horizontally.
DSA

• Important applications in contiguous resource allocation (e.g., memory, bandwidth)
• Generalizes interval coloring (when items have same height),
• NP-hard [Stockmeyer ‘76], even for squares.
• Possibility of PTAS is open.
• $h_{max}$ := maximum height rectangle, $LOAD :=$ maximum sum of heights of rectangles that intersect any vertical line. Then, $OPT = LOAD \left(1 + O\left(\frac{h_{max}}{L}\right)^{1/7}\right)$.
• $(2 + \varepsilon)$-approx, even for squares [Buchsbaum et al., STOC’03],
• If we can drop $\varepsilon$-fraction of items, we can achieve a packing in height $(1 + \varepsilon)OPT$ [Momke et al., ‘20].

$h_{max} = 3,$
$LOAD = 4,$
$OPT = 5$
4. 2-D Geometric Knapsack
   (2-D GK)
Geometric Knapsack: (2-D)

• **Input:**
  - Rectangles \( I := \{R_1, R_2, \ldots, R_n\} \); Each \( R_i \) has integral width and height \((w_i, h_i)\) and profit \( p_i \).
  - A Square \( K \times K \) knapsack.

• **Goal:** Find an axis-parallel non-overlapping packing of a subset of input rectangles into the knapsack that maximizes the total profit.

**Variant 1: 2DK**
- No rotations are allowed!

\[ \text{OPT} = 155 \]
Geometric Knapsack: (2-D)

• Input:
  - Rectangles \( I = \{ R_1, R_2, ..., R_n \} \); Each \( R_i \) has integral width and height \( (w_i, h_i) \) and profit \( p_i \).
  - A Square \( K \times K \) knapsack.

• Goal: Find an axis-parallel non-overlapping packing of a subset of input rectangles into the knapsack that maximizes the total profit.

<table>
<thead>
<tr>
<th>Value</th>
<th>(x,y)</th>
</tr>
</thead>
<tbody>
<tr>
<td>100 $</td>
<td>(9,6)</td>
</tr>
<tr>
<td>95 $</td>
<td>(7,6)</td>
</tr>
<tr>
<td>90 $</td>
<td>(5,8)</td>
</tr>
<tr>
<td>60 $</td>
<td>(4,6)</td>
</tr>
<tr>
<td>5 $</td>
<td>(4,4)</td>
</tr>
<tr>
<td>1 $</td>
<td>(2,3)</td>
</tr>
</tbody>
</table>

Variant 2: (2DKR) 90 degree rotations are allowed!

\( \text{OPT}=165 \)
Geometric Knapsack: Complexity

• **Geometric Knapsack** is Strongly NP-hard
  (even when all items are squares with profit 1), [Leung et al., 1990]
  - Remains NP-hard even if the input is given in unary.
  - No exact algorithm even in pseudo-polynomial time (unless P=NP).
  
• **Not known** whether the problem is APX-hard. So, the existence of a
  PTAS/QPTAS/PPTAS is still open!

• (1+ε)-approximation known if
  - profit of an item is equal to its area. [Bansal et al., ISAAC ‘09].
  - items are relatively small [Fishkin et al., MFCS ‘05].
  - items are squares [Wiese-Heydrich, SODA ’17].
Geometric Knapsack:

• \((2+\varepsilon)\)-approximation [Jansen-Zhang, SODA’04]
  - for both with and without rotations.
  - even in the cardinality case (when all profits are 1).

• Broke the barrier of 2 [Galvez-Grandoni-Ingala-K.-Wiese, FOCS’17]
  - Without rotations: \((17/9+\varepsilon)<1.89\)-appx.
  - With rotations: \((1.5+\varepsilon)\)-appx.
  - Cardinality case: 1.72, \((4/3+\varepsilon)\)-appx., resp.
5. Storage Allocation Problem (SAP)
SAP

• **Input:** A path with edge capacities and a set of tasks (rectangles) that are specified by start and end vertices (fixed starting coordinate and width), demands (heights) and profits.

• **Goal:** Select a subset of tasks that can be drawn as non-overlapping rectangles underneath the capacity profile.
SAP: Tale of approximability

- Generalizes knapsack.
- Special case: Uniform-SAP (when all edges have same capacity)
  - 7-approx [Bar-Noy et al, STOC’00].
- General case: \((9 + \varepsilon)\)-approximation [Bar-Yehuda et al, SPAA’13].
- \((2 + \varepsilon)\)-approximation [Momke-Wiese, ICALP’15].
- Uniform-SAP: \(1.969\) [Momke-Wiese, ’20].
- General-SAP: QPTAS with resource augmentation. [Momke-Wiese, ’20].
6. Unsplittable Flow on a Path (UFP)
UFP (sliced version of SAP)

• **Input:** A path with edge capacities and a set of tasks (rectangles) that are specified by start and end vertices (fixed starting coordinate and width), demands (heights) and profits.

• **Goal:** Select a subset of tasks such that total demand of selected tasks at any edge is less than the edge capacity.
UFP: A tale of approximability

- Strongly NP-hard, even for uniform edge capacities and uniform profits.
- QPTAS [Bansal et al, STOC’06, Batra et al, SODA’15],
- $O(\log n)$-approx. [Bansal et al, SODA’09],
- $(7 + \varepsilon)$-approx [Bonsma et al, FOCS’11],
- $(2 + \varepsilon)$-approx [Anagnostopoulos et al, SODA’14],
- $(\frac{5}{3} + \varepsilon)$-approx [Grandoni et al. STOC’18],
- Possibility of PTAS is still open!
7. Maximum Weighted Independent Set of Rectangles (MWISR)
MWISR

- **Input**: n axis-parallel rectangles (each with associated profit) on a plane.
- **Goal**: Find maximum profit subset of disjoint rectangles.
- Special case: uniform profit (MISR).
- Applications: data-mining, map-labeling, etc.
MWISR: tale of approximability

- NP-hard.
- Folklore: $O(\log n)$
- MISR: $O(\log \log n)$-approximation [Chalermsook-Chuzhoy, SODA’09]
- MWISR:
  - $O(\log n/\log \log n)$ [Chan-HarPeled, SoCG’09]
  - PTAS for pseudodiscs (e.g. squares) [Chan-HarPeled, SoCG’09]
  - $(1 + \varepsilon)$-appx in $n^{poly(\log n)}$ [Adamaszek-Wiese, FOCS’13]
  - $(1 + \varepsilon)$-appx in $n^{poly(\log \log n)}$ [Chuzhoy-Ene, STOC’16]
- PTAS, even $O(1)$-appx is open!
Pach-Tardos Conjecture
Pach-Tardos Conjecture

• **Conjecture:** For any set of $n$ non-overlapping axis-parallel rectangles there is a guillotine cutting sequence with only axis-parallel cuts separating $\Omega(n)$ of them.

• Known upper bound: $n/2$ (also for squares).

• Known lower bound: $n/\log n$.

• The conjecture is true for squares! [Abed et al, APPROX’15]

• Theorem: [Abed et al, APPROX’15] If the conjecture is true, then there is a $O(1)$-approximation algorithm for MISR with running time $O(n^5)$. 
Other related problems.

• Round-UFP, Round-SAP, coloring of rectangles.
• Min-area rectangle packing: APTAS [Bansal-Sviridenko, SODA’04]
• Circle and other geometric objects.
• Vector: when items are multidimensional vectors.
  d-dim vector bin packing: $0.81 + O(\log d)$ [Bansal-K.-Elias, SODA’16]
• Graph: weighted bipartite edge coloring, weighted bipartite matching, generalized assignment problem. [K.-Singh, FSTTCS’15]
Summary of present status

- Though these variants are related, their approximability and techniques are quite diverse, e.g.,
  - Strip packing: admits APTAS,
  - Independent set of rectangles:
    - admits QPTAS and \((\log n / \log \log n)\)-polytime approximation.
  - Geometric knapsack:
    - may or may not admit PTAS/QPTAS/PPTAS, < 2- appx. Known.
  - SAP/DSA: APX-hardness not known, Barrier of 2-approximation.
  - \(d\)-dim vector packing: No APTAS. Best known approx.: \(O(\log d)\)
  - \(d\)-dim geometric bin packing: No APTAS. Best known approx.: \(1.69^{d-1}\).
Top 10 open problems

1. Algorithm with $\text{OPT} + O(1)$-guarantee for bin packing.
2. A $\text{poly}(d)$-approximation or hardness for $d$-dim geometric bin packing?
3. Resolve guillotine conjecture for 2-D bin packing.
4. $(\frac{3}{2} + \varepsilon)$-approximation for strip packing?
5. PTAS (or PPTAS or QPTAS) for 2-D geometric knapsack (even with rotations)?
6. PTAS for unsplittable flow on a path?
7. Break the barrier of 2 for dynamic storage allocation.
8. Break the barrier of 2 for storage allocation problem.
10. PTAS for maximum independent set of rectangles?