An Efficient Data Structure for Dynamic Two-Dimensional Reconfiguration

Sándor Fekete, Jan-Marc Reinhardt, and **Christian Scheffer**

Technische Universität Braunschweig

7.4.2016

An Efficient Data Structure for Dynamic Two-Dimensional Reconfiguration (Online packing of squares)

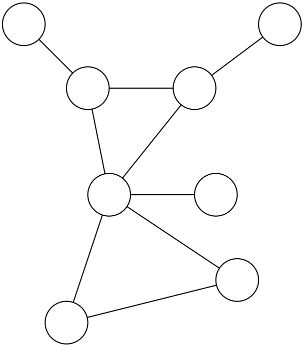
Sándor Fekete, Jan-Marc Reinhardt, and **Christian Scheffer**

Technische Universität Braunschweig

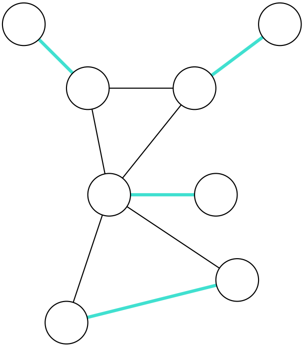
7.4.2016

Introduction

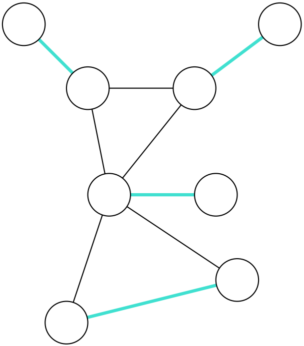
Packing Problems

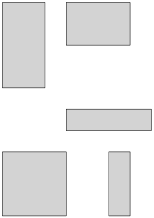


Packing Problems

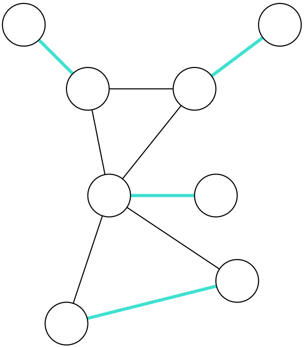


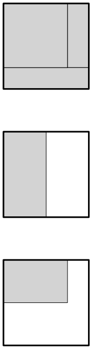
Packing Problems





Packing Problems



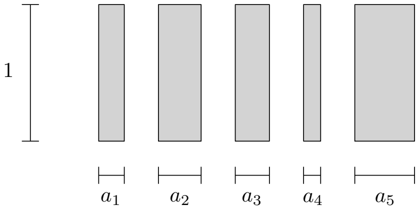


Complexity of geometric packing problems

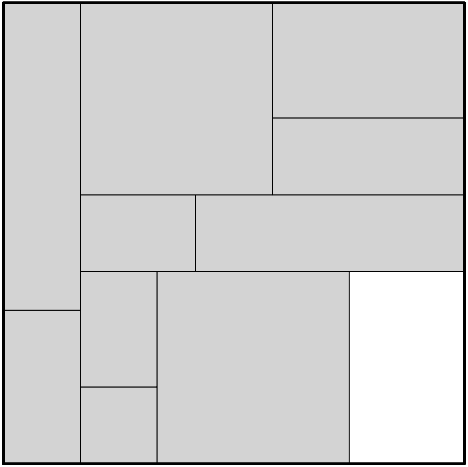
- Reduction of Bin Packing
- Sequence a_1, a_2, a_3, a_4, a_5, $\ldots$

Complexity of geometric packing problems

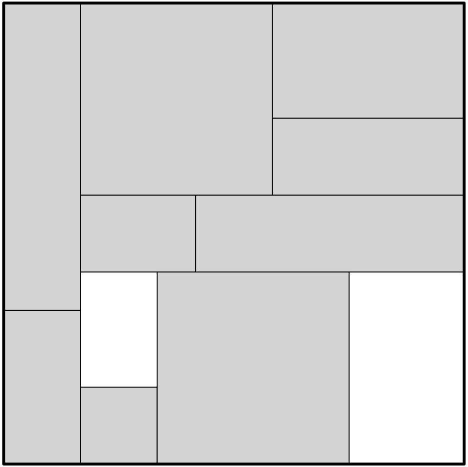
- Reduction of Bin Packing
- Sequence a_1, a_2, a_3, a_4, a_5, $\ldots$



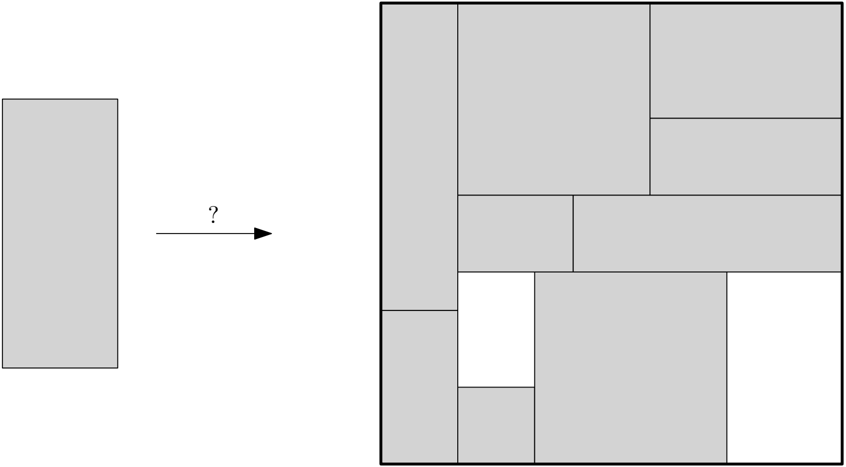
Reallocation Problems



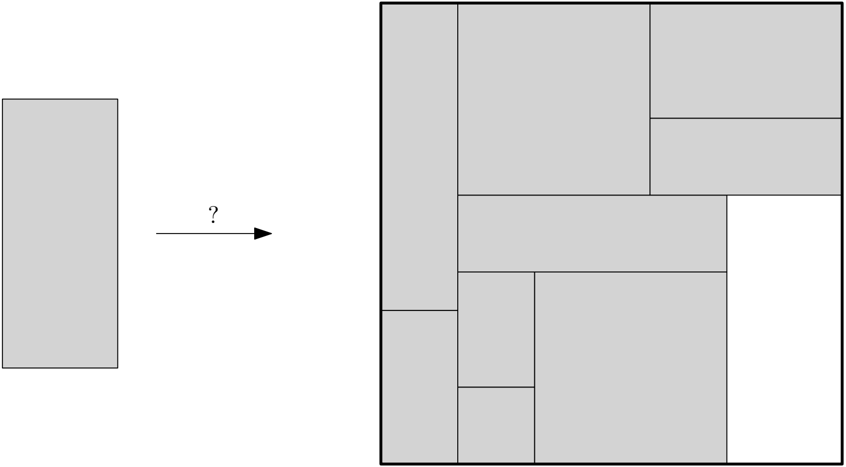
Reallocation Problems



Reallocation Problems

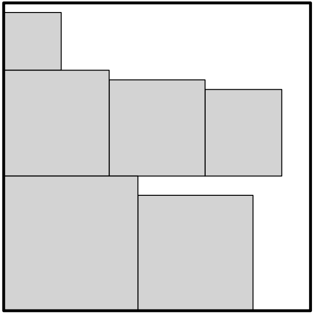


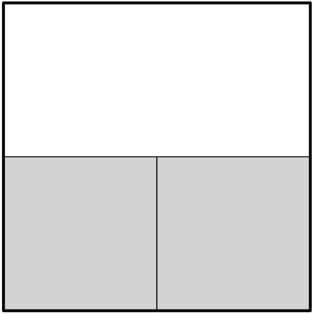
Reallocation Problems



Preliminaries

- Reallocations in 1D
- Packing of squares in the unit square: Density 1/2





Overview

Problem Setting

Problem Setting

- Online packing of squares in the unit square
- Sequence of operations $\mathrm{INSERT}(x)$, $\mathrm{DELETE}(x)$ with

$$\text{side lengths of } x: \ \frac{1}{2^k}, \ \text{for } k \in \mathbb{N}_0$$

Objective

- All operations have to be doable
- Minimize cost of reallocations

Problem Setting

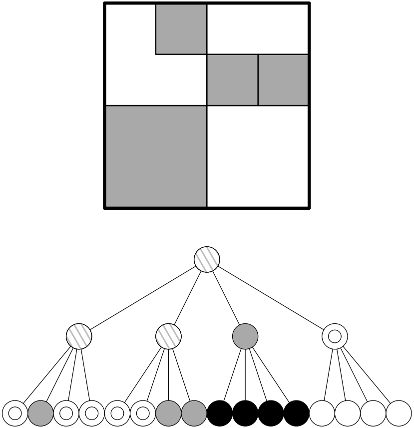
- Online packing of squares in the unit square
- Sequence of operations $\mathrm{INSERT}(x)$, $\mathrm{DELETE}(x)$ with

$$\text{side lengths of } x: \ \frac{1}{2^k}, \text{ for } k \in \mathbb{N}_0$$

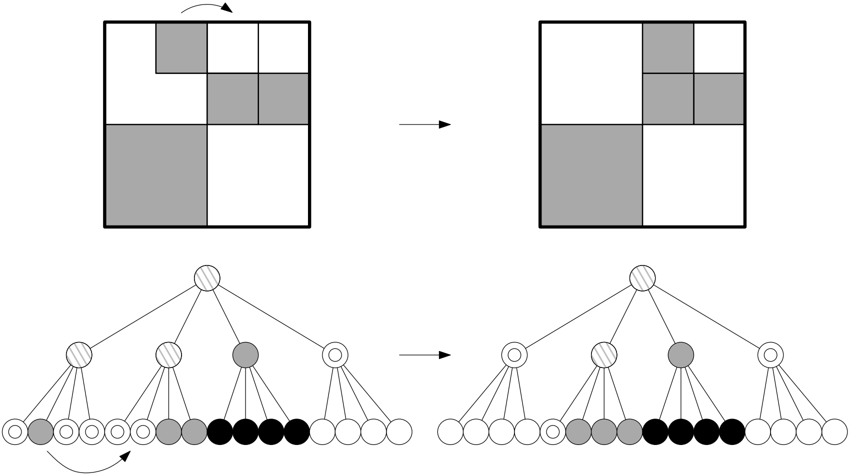
Objective

- All operations have to be doable
- Minimize cost of reallocations

Quadtree Configurations



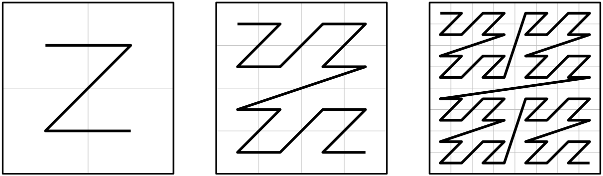
Reallocations for Quadtree Configurations



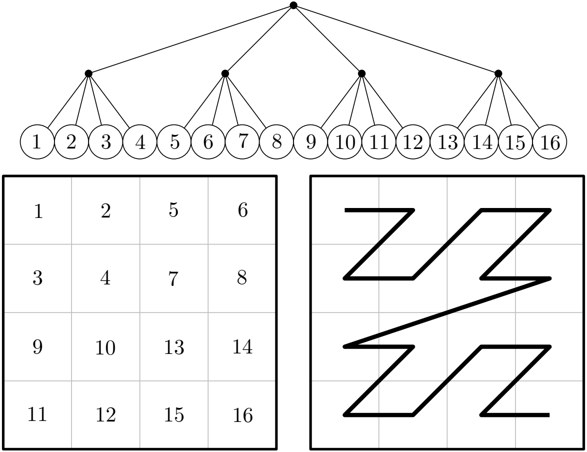
2D First-fit

First-fit

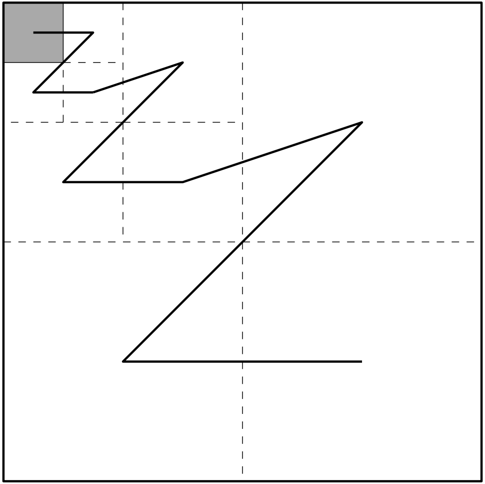
First-fit



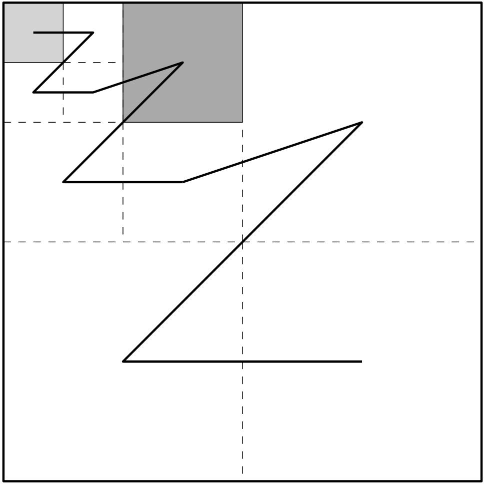
Z-Order



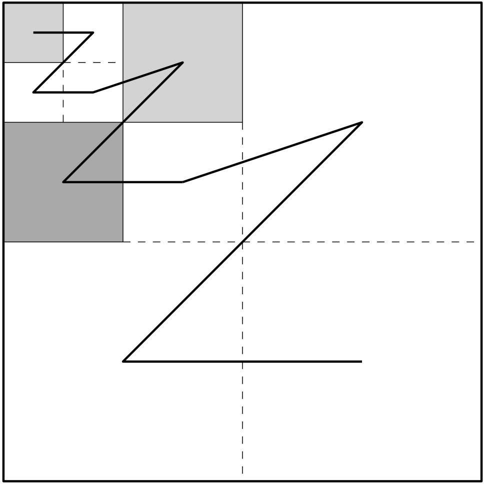
Strategy



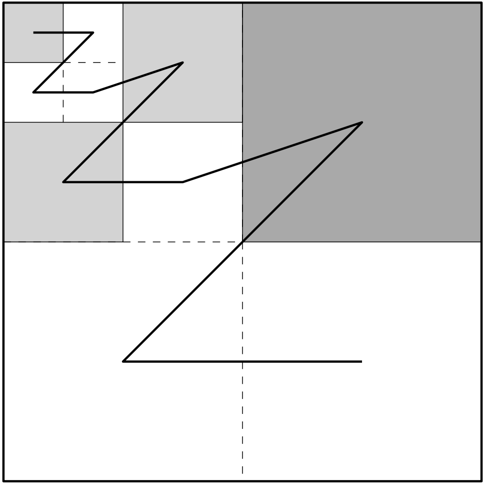
Strategy



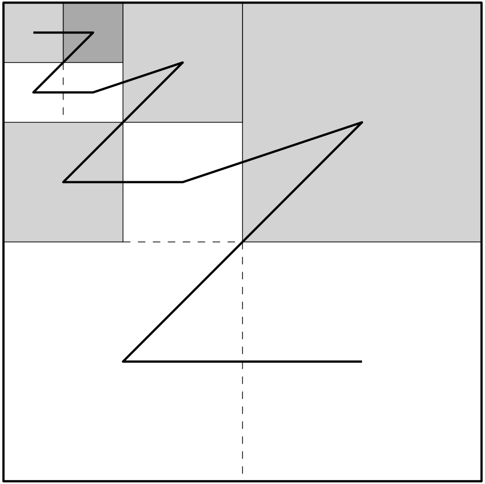
Strategy



Strategy

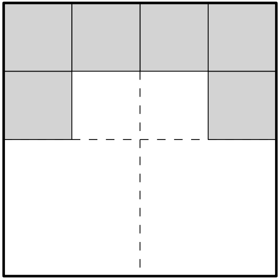


Strategy

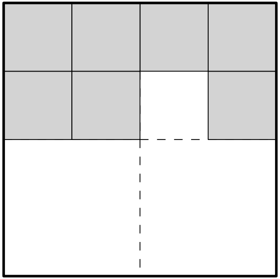


Compactness

- On each level at most one "partially occupied" square



not compact



compact

- Lemma: First-fit produces compact allocations

First-fit: Analysis

- Lemma: A compact configuration contains on each level at most three empty pixel (maximum sized squares that are empty)
- Theorem: $\mathrm{INSERT}(\cdot)$ with First-fit always works (if there is enough space) and needs no Reallocations

Proof

Assumption: There is no empty pixel with volume 4^{-i}. Then

$$\mathrm{cap}(T) \leq \sum_{k=i+1}^{s} 3 \cdot 4^{-k} = 4^{-i} - 4^{-s} < 4^{-i}.$$

First-fit: Analysis

- Lemma: A compact configuration contains on each level at most three empty pixel (maximum sized squares that are empty)
- Theorem: $\mathrm{INSERT}(\cdot)$ with First-fit always works (if there is enough space) and needs no Reallocations

Proof

Assumption: There is no empty pixel with volume 4^{-i}. Then

$$\mathrm{cap}(T) \leq \sum_{k=i+1}^{s} 3 \cdot 4^{-k} = 4^{-i} - 4^{-s} < 4^{-i}.$$

First-fit: Analysis

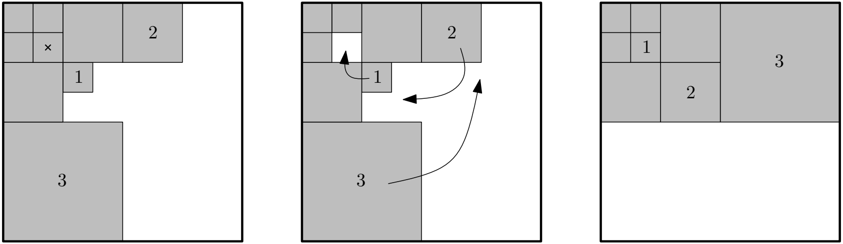
- Lemma: A compact configuration contains on each level at most three empty pixel (maximum sized squares that are empty)
- Theorem: $\mathrm{INSERT}(\cdot)$ with First-fit always works (if there is enough space) and needs no Reallocations

Proof

Assumption: There is no empty pixel with volume 4^{-i}. Then

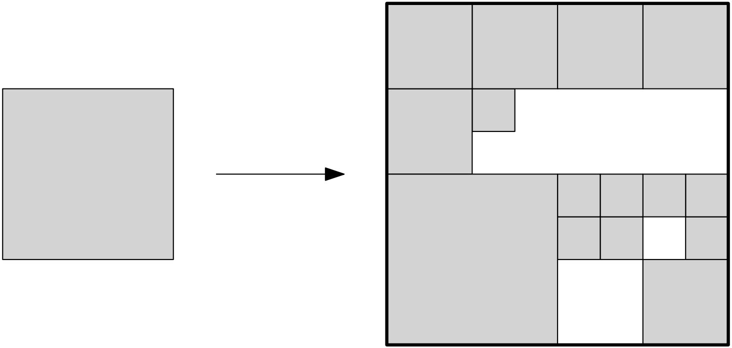
$$\mathrm{cap}(T) \leq \sum_{k=i+1}^{s} 3 \cdot 4^{-k} = 4^{-i} - 4^{-s} < 4^{-i}.$$

First-fit and Delete Operations

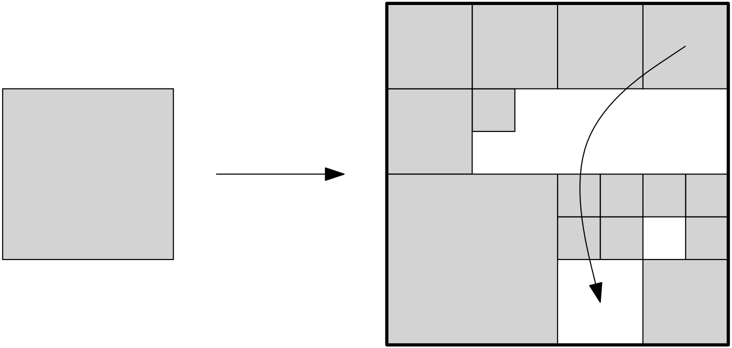


Insert in arbitrary Configurations

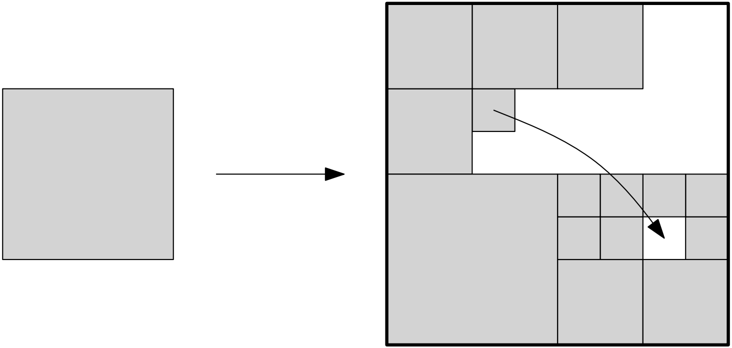
Defragmentation



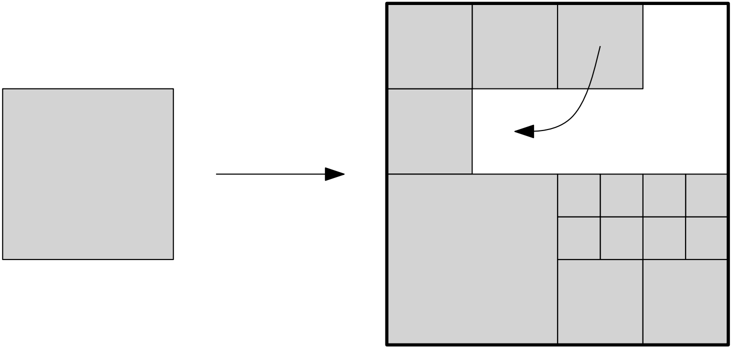
Defragmentation



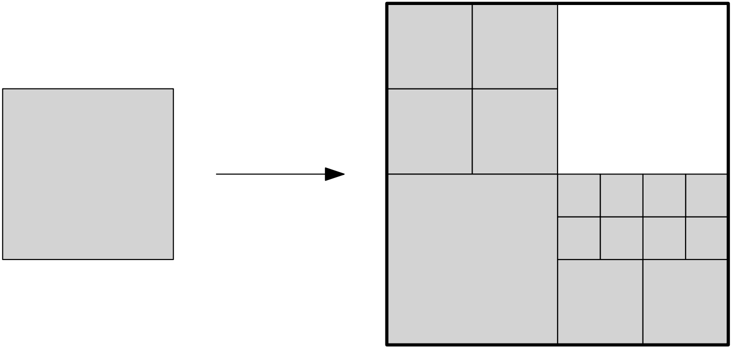
Defragmentation



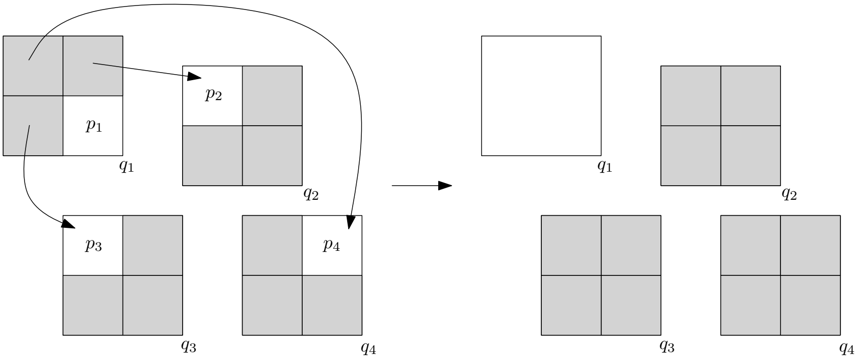
Defragmentation



Defragmentation

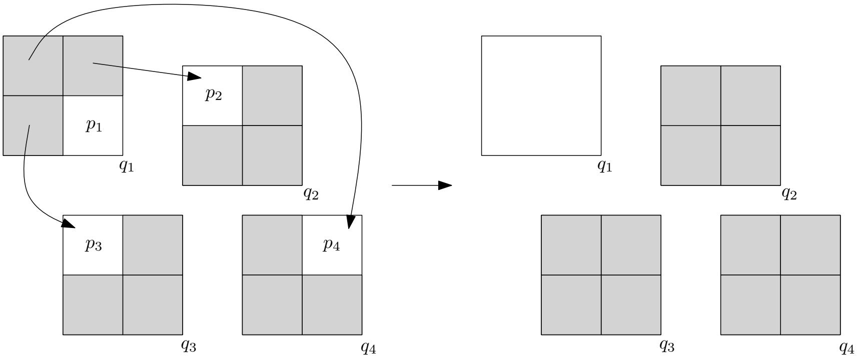


Defragmentation: Single step



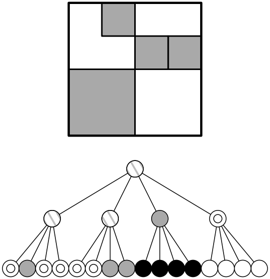
$$c_{\text{total,max}} \leq \frac{3}{4} \cdot 4^{-i} \cdot (s - i) \in O(|Q| \cdot h(T))$$

Defragmentation: Single step

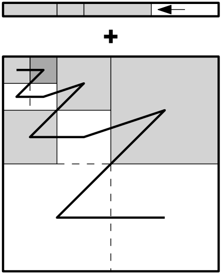


$$c_{\text{total,max}} \leq \frac{3}{4} \cdot 4^{-i} \cdot (s - i) \in O(|Q| \cdot h(T))$$

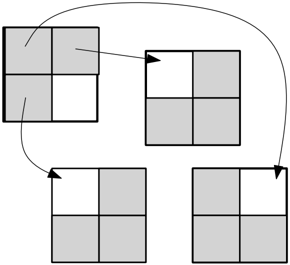
Conclusion



Quadtree



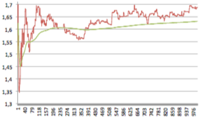
2D First-fit



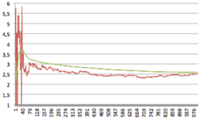
Arbitrary Configurations

Experiments

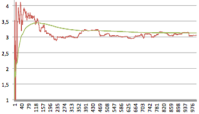
Experiments



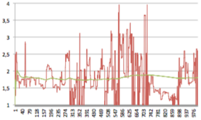
$k = 1,\ b = 0.125,\ c = 219$



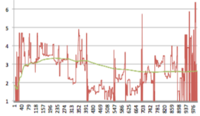
$k = 2,\ b = 0.125,\ c = 232$



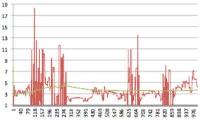
$k = 5,\ b = 0.125,\ c = 264$



$k = 1,\ b = 1,\ c = 419$

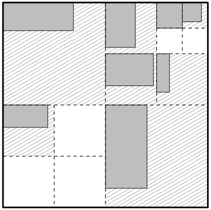


$k = 2,\ b = 1,\ c = 438$



$k = 5,\ b = 1,\ c = 421$

- Extension to arbitrary rectangles
- Randomly generated squares
- Worst case packing density $4k$
- k: bound for ratio of side lengths,
  b: bound for side lengths, and
  c: collisions



Insert in arbitrary Configurations: Details

Defragmentation: Strategy

- Sequence of all empty pixels ordered w.r.t. volume:

$$|p_1| \geq |p_2| \geq \dots$$

- Shortest prefix with total volume at least 4^{-i}:

$$p_1, p_2, \dots, p_k$$

- Representation of normed sum $1/(4^{-i}) \sum_{j=1}^{k-1} |p_j|$ with base 4:

$$0, ?????????\dots$$

- $|p_k| = |p_{k-1}| = |p_{k-2}| = |p_{k-3}|$
- Repeat until $k = 1$

Defragmentation: Strategy

- Sequence of all empty pixels ordered w.r.t. volume:

$$|p_1| \geq |p_2| \geq \dots$$

- Shortest prefix with total volume at least 4^{-i}:

$$p_1, p_2, \dots, p_k$$

- Representation of normed sum $1/(4^{-i}) \sum_{j=1}^{k-1} |p_j|$ with base 4:

$$0, ?????????\dots$$

- $|p_k| = |p_{k-1}| = |p_{k-2}| = |p_{k-3}|$
- Repeat until $k = 1$

Defragmentation: Strategy

- Sequence of all empty pixels ordered w.r.t. volume:

$$|p_1| \geq |p_2| \geq \dots$$

- Shortest prefix with total volume at least 4^{-i}:

$$p_1, p_2, \dots, p_k$$

- Representation of normed sum $1/(4^{-i}) \sum_{j=1}^{k-1} |p_j|$ with base 4:

$$0, ?????????\dots$$

- $|p_k| = |p_{k-1}| = |p_{k-2}| = |p_{k-3}|$
- Repeat until $k = 1$

Defragmentation: Strategy

- Sequence of all empty pixels ordered w.r.t. volume:

$$|p_1| \geq |p_2| \geq \dots$$

- Shortest prefix with total volume at least 4^{-i}:

$$p_1, p_2, \dots, p_k$$

- Representation of normed sum $1/(4^{-i}) \sum_{j=1}^{k-1} |p_j|$ with base 4:

$$0,3333333\dots300\dots$$

- $|p_k| = |p_{k-1}| = |p_{k-2}| = |p_{k-3}|$
- Repeat until $k = 1$

Defragmentation: Strategy

- Sequence of all empty pixels ordered w.r.t. volume:

$$|p_1| \geq |p_2| \geq \dots$$

- Shortest prefix with total volume at least 4^{-i}:

$$p_1, p_2, \dots, p_k$$

- Representation of normed sum $1/(4^{-i}) \sum_{j=1}^{k-1} |p_j|$ with base 4:

$$0,3333333\dots300\dots$$

- $|p_k| = |p_{k-1}| = |p_{k-2}| = |p_{k-3}|$
- Repeat until $k = 1$

Defragmentation: Strategy

- Sequence of all empty pixels ordered w.r.t. volume:

$$|p_1| \geq |p_2| \geq \dots$$

- Shortest prefix with total volume at least 4^{-i}:

$$p_1, p_2, \dots, p_k$$

- Representation of normed sum $1/(4^{-i}) \sum_{j=1}^{k-1} |p_j|$ with base 4:

$$0,3333333\dots300\dots$$

- $|p_k| = |p_{k-1}| = |p_{k-2}| = |p_{k-3}|$
- Repeat until $k = 1$

Costs of Reallocations: Details

Objective functions

- Costs of a single operation σ that causes that the squares $s_1, s_2, \ldots, s_k$ have to be reallocated:

$$c_{\text{move}}(\sigma) = k$$

$$c_{\text{total}}(\sigma) = \sum_{i=1}^{k} |s_i|$$

$$c_{\text{vol}}(\sigma) = \frac{c_{\text{total}}(\sigma)}{|\text{vol}(\sigma)|}$$

Objective functions

- Costs of a single operation σ that causes that the squares $s_1, s_2, \ldots, s_k$ have to be reallocated:

$$c_{\mathsf{move}}(\sigma) = k$$

$$c_{\mathsf{total}}(\sigma) = \sum_{i=1}^{k} |s_i|$$

$$c_{\mathsf{vol}}(\sigma) = \frac{c_{\mathsf{total}}(\sigma)}{|\mathrm{vol}(\sigma)|}$$

Objective functions

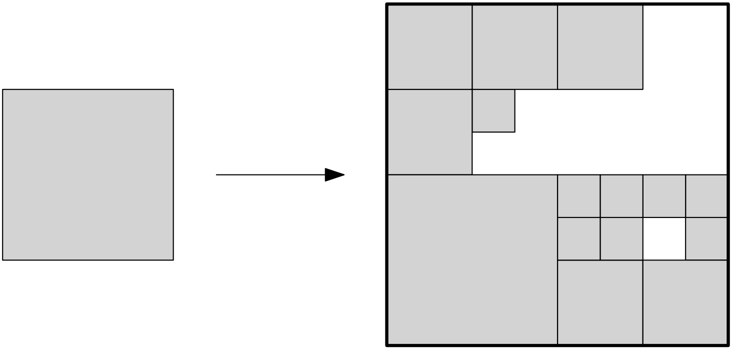
- Costs of a single operation σ that causes that the squares $s_1, s_2, \ldots, s_k$ have to be reallocated:

$$c_{\mathsf{move}}(\sigma) = k$$

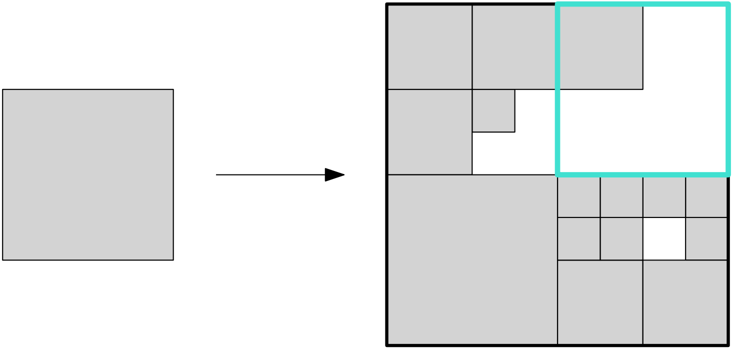
$$c_{\mathsf{total}}(\sigma) = \sum_{i=1}^{k} |s_i|$$

$$c_{\mathsf{vol}}(\sigma) = \frac{c_{\mathsf{total}}(\sigma)}{|\mathrm{vol}(\sigma)|}$$

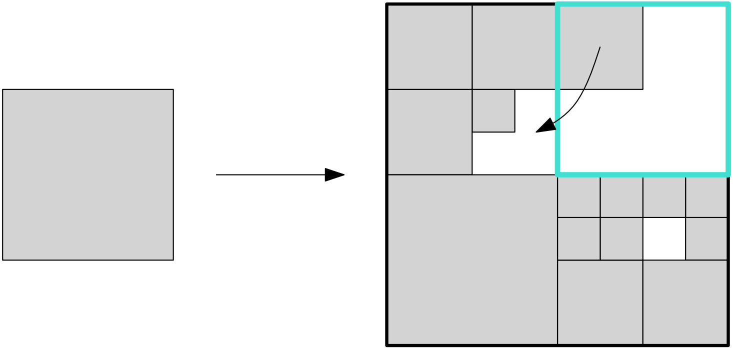
Cascading Reallocations



Cascading Reallocations



Cascading Reallocations



Absolute volume costs in the worst case

- Observation: Reallocating one square causes the same costs as reallocating four squares on the next higher level
- s: Level of the smallest square; insert square Q on level i

$$v_i \leq \sum_{j=i+1}^{s} \frac{3}{4^j} = 4^{-i} - 4^{-s}$$

$$x_i = v_i + \sum_{j=i+1}^{s} 3x_j$$

$$c_{\text{total,max}} \leq \frac{3}{4} \cdot 4^{-i} \cdot (s-i) \in O(|Q| \cdot h(T))$$

Absolute volume costs in the worst case

- Observation: Reallocating one square causes the same costs as reallocating four squares on the next higher level
- s: Level of the smallest square; insert square Q on level i

$$v_i \leq \sum_{j=i+1}^{s} \frac{3}{4^j} = 4^{-i} - 4^{-s}$$

$$x_i = v_i + \sum_{j=i+1}^{s} 3x_j$$

$$c_{\text{total,max}} \leq \frac{3}{4} \cdot 4^{-i} \cdot (s-i) \in O(|Q| \cdot h(T))$$

Absolute volume costs in the worst case

- Observation: Reallocating one square causes the same costs as reallocating four squares on the next higher level
- s: Level of the smallest square; insert square Q on level i

$$v_i \leq \sum_{j=i+1}^{s} \frac{3}{4^j} = 4^{-i} - 4^{-s}$$

$$x_i = v_i + \sum_{j=i+1}^{s} 3x_j$$

$$c_{\text{total,max}} \leq \frac{3}{4} \cdot 4^{-i} \cdot (s-i) \in O(|Q| \cdot h(T))$$

Absolute volume costs in the worst case

- Observation: Reallocating one square causes the same costs as reallocating four squares on the next higher level
- s: Level of the smallest square; insert square Q on level i

$$v_i \leq \sum_{j=i+1}^{s} \frac{3}{4^j} = 4^{-i} - 4^{-s}$$

$$x_i = v_i + \sum_{j=i+1}^{s} 3x_j$$

$$c_{\text{total,max}} \leq \frac{3}{4} \cdot 4^{-i} \cdot (s - i) \in O(|Q| \cdot h(T))$$

Absolute volume costs in the worst case

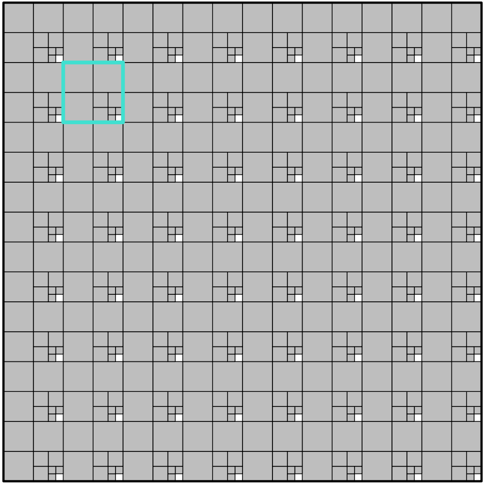
- Observation: Reallocating one square causes the same costs as reallocating four squares on the next higher level
- s: Level of the smallest square; insert square Q on level i

$$v_i \leq \sum_{j=i+1}^{s} \frac{3}{4^j} = 4^{-i} - 4^{-s}$$

$$x_i = v_i + \sum_{j=i+1}^{s} 3x_j$$

$$c_{\text{total,max}} \leq \frac{3}{4} \cdot 4^{-i} \cdot (s - i) \in O(|Q| \cdot h(T))$$

A lower bound construction



Analogue for other objective functions

- $c_{\mathsf{vol,max}} \leq \frac{3}{4} \cdot (s - i) \in O(h(T))$
- $c_{\mathsf{move,max}} \leq 4^{\min\{(s-i),i\}-1} \in O\big(4^{h(T)/2}\big)$

Applications to other scenarios

Quadratische Gitterpunktmengen

- Beschränkung der Höhe
- Untere Schranke von 1
- $i = \log_4 |Q|$
- $c_{\text{vol,max}} \in O(\log |Q|)$
- $c_{\text{move,max}} \in O(|Q|)$

Quadratische Gitterpunktmengen

- Beschränkung der Höhe
- Untere Schranke von 1
- $i = \log_4 |Q|$
- $c_{\text{vol,max}} \in O(\log |Q|)$
- $c_{\text{move,max}} \in O(|Q|)$

Quadratische Gitterpunktmengen

- Beschränkung der Höhe
- Untere Schranke von 1
- $i = \log_4 |Q|$
- $c_{\text{vol,max}} \in O(\log |Q|)$
- $c_{\text{move,max}} \in O(|Q|)$

Quadratische Gitterpunktmengen

- Beschränkung der Höhe
- Untere Schranke von 1
- $i = \log_4 |Q|$
- $c_{\text{vol,max}} \in O(\log |Q|)$
- $c_{\text{move,max}} \in O(|Q|)$

Höhere Dimensionen

- Analog!

$$c_{\text{vol,max}} \leq \frac{2^d - 1}{2^d} \cdot \min\{s-i, i\} \in \Theta(h(T))$$

$$c_{\text{move,max}} \leq 2^{d \cdot \min\{s-i,i\}} - 1 \in \Theta(2^{d \cdot h(T)/2})$$

- Sinnvoll?

Höhere Dimensionen

- Analog!

$$c_{\text{vol,max}} \leq \frac{2^d - 1}{2^d} \cdot \min\{s-i, i\} \in \Theta(h(T))$$

$$c_{\text{move,max}} \leq 2^{d \cdot \min\{s-i,i\}} - 1 \in \Theta(2^{d \cdot h(T)/2})$$

- Sinnvoll?

Höhere Dimensionen

- Analog!

$$c_{\text{vol,max}} \leq \frac{2^d - 1}{2^d} \cdot \min\{s-i, i\} \in \Theta(h(T))$$

$$c_{\text{move,max}} \leq 2^{d \cdot \min\{s-i,i\}} - 1 \in \Theta(2^{d \cdot h(T)/2})$$

- Sinnvoll?

Höhere Dimensionen

- Analog!

$$c_{\mathsf{vol,max}} \leq \frac{2^d - 1}{2^d} \cdot \min\{s-i, i\} \in \Theta(h(T))$$

$$c_{\mathsf{move,max}} \leq 2^{d \cdot \min\{s-i,i\}} - 1 \in \Theta(2^{d \cdot h(T)/2})$$

- Sinnvoll?

General squares