

6 Mehrdimensionales Packen

*Algorithmen und Datenstrukturen 2
Sommer 2023*

Prof. Dr. Sándor Fekete

Autos, Quader, Rechtecke

Reisezeit!

Reisezeit!



Wie viel schluckt
der Kofferraum? Beim
Beladen des Reise-
gepäckes erleben Autofahrer
oft Überraschungen

TEST 20 MODELLE IM KOFFERRAUM-VERGLEICH



Die Test-Kandidaten 1 BMW 3er-Touring • 2 Hyundai i30 • 3 Audi A6 Avant • 4 Opel Antara • 5 Peugeot 207 • 6 Fiat Grande Punto
 • 7 Mercedes B-Klasse • 8 Opel Astra • 9 VW Touran • 10 Opel Tigra TwinTop • 11 VW Golf Plus • 12 Mini • 13 Mercedes C-Klasse • 14 Ford Mondeo
 • 15 VW Golf Variant • 16 Ford C-Max • 17 VW Passat Variant • 18 Toyota Yaris • 19 Smart Fortwo • 20 Audi Q7

Wie viel schluckt der Kofferraum? Beim Beladen des Reisegepäckes erleben Autofahrer oft Überraschungen

Der große Kofferraum-Vergleich

DAS PASST WIRKLICH REIN!

Bei kaum einer anderen Angabe tricksen Autobauer mehr als bei der Kofferraumgröße. Wie die Hersteller uns verladen, entlarvt der große AUTO BILD-Test mit dem TÜV Süd

dem kleine Möbellaster. Zumindest auf dem Papier. Doch stimmen die Herstellerwerte? Um das herauszufinden, hat AUTO BILD zusammen mit dem TÜV Süd in München 20 wichtige Modelle aus mehreren Segmenen einen großen Kofferraumtest unterzogen. Ergebnis: Viele Marken tricksten mit falschen Inhaltsangaben und übertreiben zum Teil maßlos.

Um das oftmals kaufentscheidende Ladevolumen zu beziffern, verwendet die Autoindustrie die Literangabe. Ermittelt wird diese mithilfe von Quadern, deren Abmessungen der Verband der Automobilindustrie (VDA) auf 200 x 100 x 50 Millimeter festgelegt hat. Wie mit denen ein Kofferraum zu beladen ist, wird in den DIN-Vorschriften 70020-1

und ISO 3832 detailliert beschrieben. Trotzdem haben die Hersteller offenbar ihre eigenen Gewohnheiten beim Auslasten der Kofferräume. Denn das TÜV-Resultat ist erstaunlich: Keiner – wirklich keiner – der 20 Testkandidaten trifft seine Prospektangabe. Dabei bietet kurioserweise der Ford C-Max sogar mehr Kofferraum, als vom Werk angegeben. Bei den 19 anderen

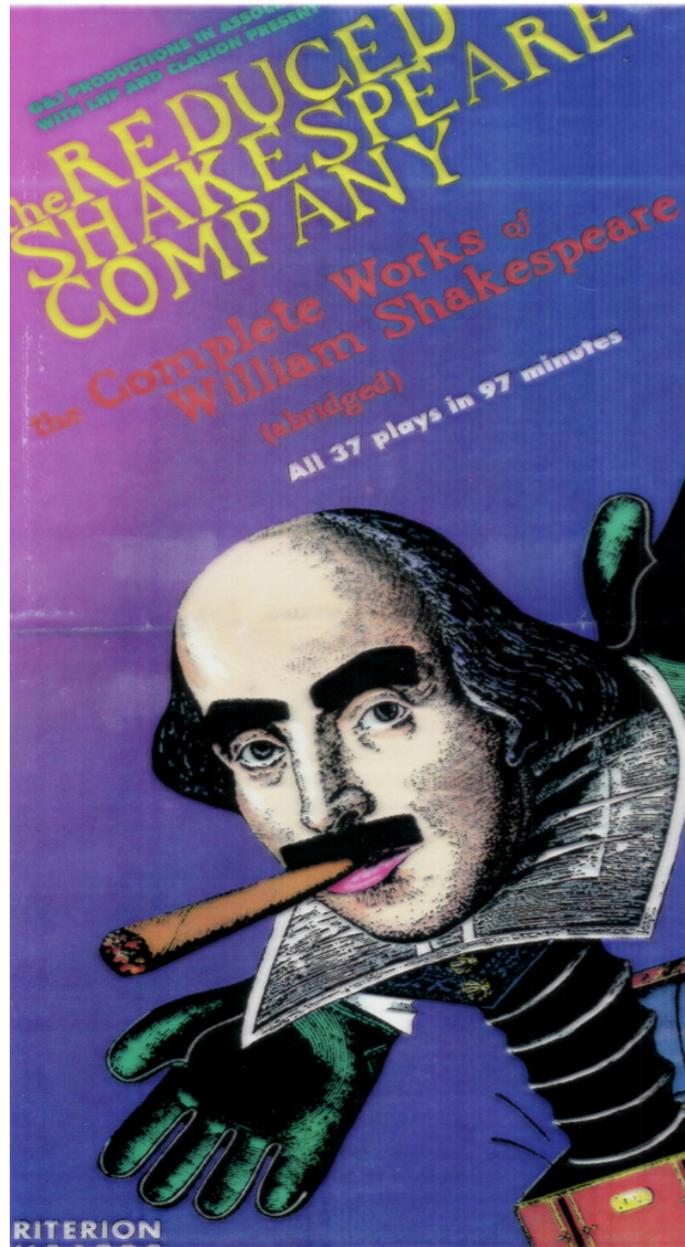
Modellen liegt das tatsächliche Volumen einer Praxisprüfung unterzogen. Wie gut passen Koffer und Kinderwagen in die einzelnen Modelle? Auch bei diesem Familientest gab es überraschende Erkenntnisse. Zum Beispiel diese: Probieren geht über Studieren. Wer genau wissen will, wie sich sein Gepäck in einzelne Modelle einladen lässt, sollte das beim Händler mit seinem Ladegut prüfen. Nur das verrät die wahre Größe. JM

milien beliebten Vans, Kombis und SUV außerdem einer Praxisprüfung unterzogen. Wie gut passen Koffer und Kinderwagen in die einzelnen Modelle? Auch bei diesem Familientest gab es überraschende Erkenntnisse. Zum Beispiel diese: Probieren geht über Studieren. Wer genau wissen will, wie sich sein Gepäck in einzelne Modelle einladen lässt, sollte das beim Händler mit seinem Ladegut prüfen. Nur das verrät die wahre Größe. JM

Ein Test mit TÜV und Siegel



Wieviel passt in eine Vorlesung?



Wieviel passt in einen Kofferraum?

Wieviel passt in einen Kofferraum?

TEST KOFFERRAUM-VERGLEICH

VAN GEGEN SUV

Touran fasst mehr als Q7



134
Liter zu wenig

Werksangabe: 695 Liter
Gemessen: 561 Liter

VW Touran



so sehen Testleser aus: quadratisch, praktisch, gut. Als Lade-
meister erfüllt der Touran vorzügliche Dienste. Ob großes
Reisegepäck oder Sportgeräte, er schluckt jede Menge Ladegut

Wenn das keine Überraschung ist:
Audi's Riesen-SUV ist nur außen groß.
Der sehr viel kürzere und kompaktere
VW Touran packt mehr ein. Mit
561 Liter gemessenem Volumen
übertrifft er sogar den Q7. Allerdings
liegt auch der Touran erheblich unter
seiner Werksangabe von 695 Liter.
Wird seine Rückbank vorge-
schoben, schluckt er sogar 716 Liter
- wieder mehr als der Q7 (650 Liter
mit vorgeklippten Sitzen). Ein echter
Raumrieser, der Touran. Ganz anders
der Audi Q7: Bei der Innenraum-
ausstattung enttäuscht der XXL-SUV.
Das Werk nennt ein Volumen von 775
Litern hinter den Rücksitzen. Klüfft
gewaltig. Tatsächlich sind es aber
nur 541 Liter. Macht minus 234 Liter.
Das entspricht fast dem Stauvolumen
des Peugeot 207, mit dem hier
übertrieben wird. Durch seine
verchiebbare Rückbank eröffnet der
Q7 mehrere Beladungsmöglichkeiten.
Wenn das Heckrollen entfernt ist,
näher sich der Q7 seiner Prospekt-
angabe und erreicht immerhin 732
Liter. Bei dachhoher Beladung sind
es bis zu 1070 Liter.



VW Touran

Audi Q7



234
Liter zu wenig

Werksangabe: 775 Liter
Gemessen: 541 Liter

Audi Q7



Außen riesig, innen
vergleichsweise
mickrig - der Audi-
SUV verspricht viel
mehr, als er halten
kann. Das normale
Kofferraumvolumen
liegt gerade mal
auf dem Niveau
eines Ford Mondeo

ZWEIMAL VARIANT

Golf fast so groß wie Passat!



Golf Variant

VW Passat Variant



5
Liter zu wenig

Werksangabe: 505 Liter
Gemessen: 500 Liter

VW Golf Variant



95
Liter zu wenig

Werksangabe: 603 Liter
Gemessen: 508 Liter

VW Passat Variant

Das üppige Gepäckabteil
des Golf ist gut geris-
tet für Transportaufgaben
aller Art. Die TÜV-
Messung bestätigt den Werks-
wert bis auf fünf Liter

Der Passat-Laderraum
präsentiert sich edel,
glattflächig und deshalb
gut nutzbar. Doch die
VW-Werksangabe
ist stark übertrieben

Das VW-interne Duell hat viel
Brisanz. Greift doch der 5625 Euro
günstigere Golf Kombi (als 105-PS-
Diesel) seinen größeren Variant-
Bruder ungeniert an. Aber was heißt
hier größer? Wo gibt es ein
Ladefachvolumen von 603 Litern
für den Passat an und wartet damit
ein Sicherheitsabstand von fast
100 Litern zum Golf, doch die TÜV-
Messung enthüllt: Tatsächlich pa-
cken beide Variant nahezu gleich
ein. Der Golf erreicht bis auf fünf
Liter seine Werksangabe, der Passat
liegt 95 Liter darunter. Der Kanibi-
sierungseffekt wird noch deutlicher
bei Beladung ohne Gepäckraum-
rolle. Dann ist der Golf sogar um vier
Liter besser. 83 Liter Volumen, die
in der TÜV-Messung enthalten sind,
entfallen beim Passat allein auf die
Reserveraumkulde.

CABRIO GEGEN SCHRÄGHECK

TwinTop verlädt Corsa

Die Preise des Tigra TwinTop
prahlt mit 440 Liter Volumen bei
geschlossenem Dach. Stark über-
trieben: Tatsächlich passen nur
372 Liter rein. Opel hat sich nach
eigener Aussage auf Computer-
daten verlassen und das AUTO
BILD-Ergebnis bestätigt. Bei ech-
ter Beladung von Hand schaffen
auch die Opel-Experten nur 381 Li-
ter, wobei davon sieben Liter auf
die Reserveraumkulde und 50 Liter

auf die Ablage hinter den Sitzen
entfallen. Immerhin schlägt der
Tigra damit Markenbruder Corsa
deutlich. Satte 134 Liter bleibt das
Schrägheckmodell hinter dem
Cabrio zurück und verfehlt seine
Werksangabe um 47 Liter. Wird
die Hutablage entfernt und der
Corsa bis zur Oberkante der Rück-
sitze beladen, passen noch 18
Liter mehr rein und das Volumen
steigt auf 256 Liter.



Opel Tigra TwinTop



Opel Corsa



Werksangabe: 440 Liter
Gemessen: 372 Liter

68
Liter zu wenig

Erstaunlich, was in
das kleine Cabrio passt.
Bei geöffnetem Dach
schrumpft der Kofferraum
natürlich extrem

Die starken Einzüge am
Corsa-Heck schränken
die Kofferraum-Nutz-
barkeit ein. Seine Werks-
angabe verfehlt er klar



Opel Corsa

Werksangabe: 285 Liter
Gemessen: 238 Liter

47
Liter zu wenig

DIE KOFFERRAUM-KÖNIGE

1 **VW Touran**
561 Liter
Siegler ist der
Touran. Keiner
steckt mehr weg.
Bei dachhoher
Beladung sogar
928 Liter

2 **Audi Q7**
541 Liter
Obwohl der
mächtige Audi-
SUV seine Werks-
angabe extrem
verfehlt, gehört
er zu den Größ-
ten im Testfeld

3 **Ford Mondeo**
523 Liter
Das klassische Stufenheck-Konzept bietet sehr

Klötzchen für Klötzchen: So hat der TÜV getestet



Maßarbeit: Beladen bis zur Fensterkante. Anschließend muss die Klappe problemlos schlie-

■ 20 Autos, drei TÜV-Experten und 3000 VDA-
Quader - der Kofferraum-Vergleich dauerte
eine ganze Woche. Allein das Beladen des
Audi Q7 brauchte rund sechs Stunden. Akribisch
stapelten die Spezialisten die Styroporklötz-
chen in jedes Auto. Basierend auf DIN 70020-1
und ISO 3832, wurde nach TÜV-Spezifikation
getestet. Dazu befanden sich die Sitze der Rück-
sitzebank - wo möglich - in hinterster Posi-
tion, die Lehne war für eine entspannte Reiseposi-
tion justiert, und die Höhe des ge-
messenen Kofferraumvolumens
war begrenzt durch Hutablage
oder die Gepäckraumabdeckung.



Konzentration: Die 1
Quader (200 mal 100 mal
100 Millimeter) müssen ge-
stapelt und ausgezählt we-

LIMOUSINE GEGEN KOMBI

C-Klasse schluckt mehr als 3er-Touring

■ Dass der Lifestylefaktor von
BMW kein Raumwunder ist, war
zu erwarten. Doch dass er nur 370
Liter einladen kann, statt der ver-
sprochenen 460, ist eine Riesen-
Enttäuschung. Zumal davon auch
noch 19 Liter auf die Staufächer
im Heck fallen. Ohne die sind es

sogar nur 351 Liter. Das kann ein
klassische Limousine besser. Bei
spiel: Mercedes C-Klasse. Zwei
versprechen auch die Stuttgart-
er zu viel, aber mit 464 Litern liegt
der Mercedes deutlich über den
BMW. 79 davon entfallen aller-
dings auf die Reserveraumkulde.

464 Liter
Werksangabe: 475 Liter
Mercedes C-Klasse

370 Liter
Werksangabe: 448 Liter
BMW 3er-Touring

LIMO GEGEN KOMBI II

Mondeo größer als A6

■ Auch beim Vergleich Mondeo
gegen A6 Avant hat das Stufen-
heck mehr Platz. Mit sattem 523
Liter Stauvolumen gehört der
große Ford zu den Platzkönigen
im Vergleich. Er kann sich sogar
ein Notrad erlauben, das ihn ei-

nige Liter kostet. Denn zur Wert-
angabe fehlen 27 Liter. Der Au-
tombi füllt nur 470 Liter ein -
wenn er ohne Jalousie (517 Li-
ter) oder dachhoch (698 Liter) be-
laden wird, ist er konkurrenzlos

523 Liter
Werksangabe: 550 Liter
Ford Mondeo

470 Liter
Werksangabe: 545 Liter
Audi A6 Avant

Kofferpacken für die Reise



Kofferpacken für die Reise

Gegeben:

Eine Menge von n Objekten, jedes mit einer Größe l_i

Gesamtgröße ist $\sum_{i=1}^n l_i = 2K$

Kofferpacken für die Reise

Gegeben:

Eine Menge von n Objekten, jedes mit einer Größe l_i

Gesamtgröße ist $\sum_{i=1}^n l_i = 2K$

29

13

17

31

35

24

31

7

53

Kofferpacken für die Reise

Gegeben:

Eine Menge von n Objekten, jedes mit einer Größe l_i

Gesamtgröße ist $\sum_{i=1}^n l_i = 2K$

29

13

17

31

35

24

31

7

53

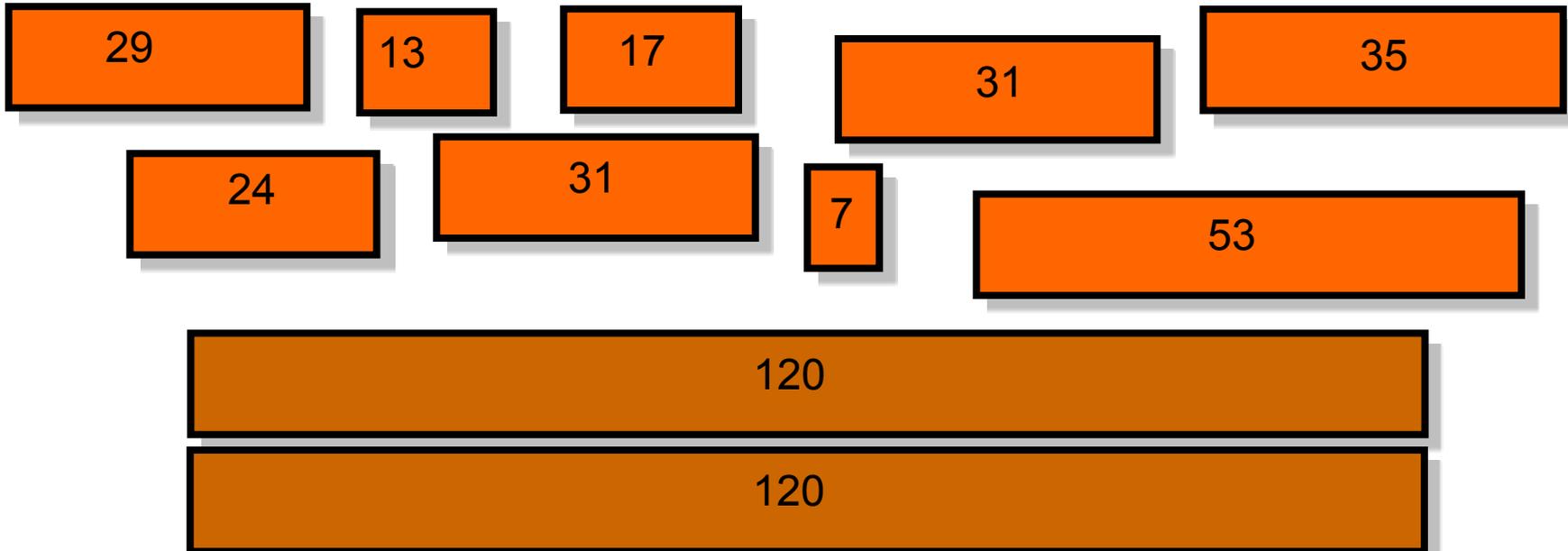
Gesucht:

Kofferpacken für die Reise

Gegeben:

Eine Menge von n Objekten, jedes mit einer Größe l_i

Gesamtgröße ist $\sum_{i=1}^n l_i = 2K$



Gesucht:

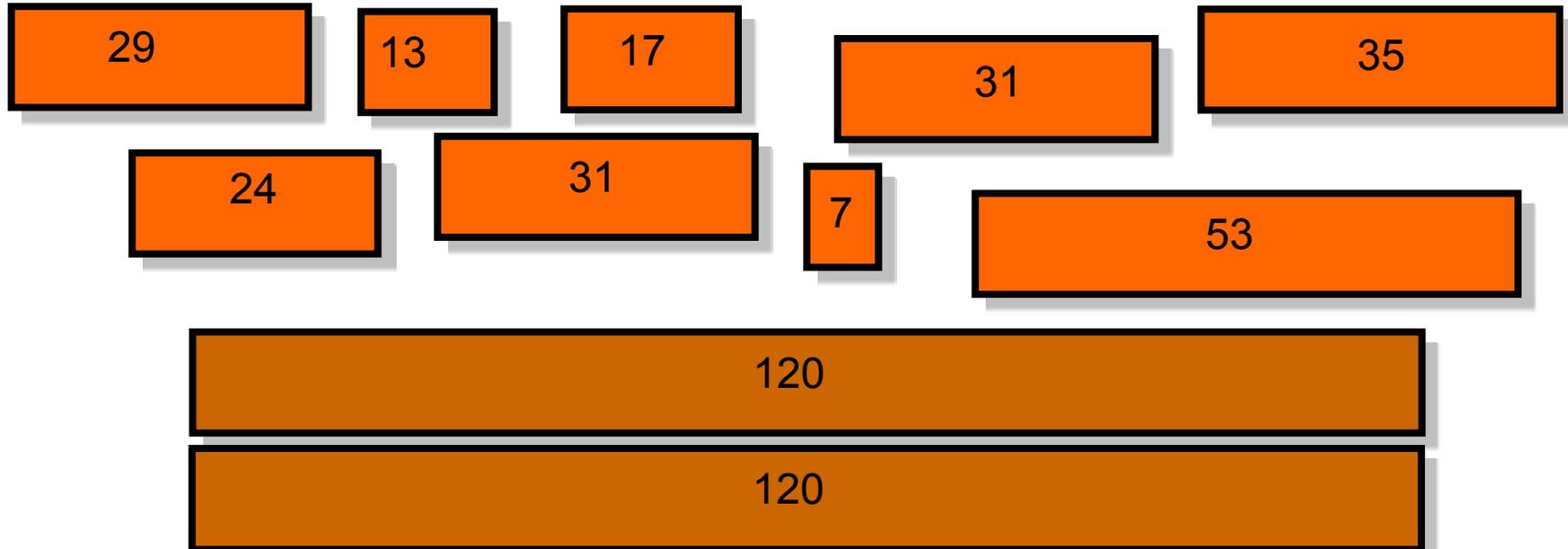
Eine Packung in zwei Container der Größe K

Kofferpacken für die Reise

Gegeben:

Eine Menge von n Objekten, jedes mit einer Größe l_i

Gesamtgröße ist $\sum_{i=1}^n l_i = 2K$



Gesucht:

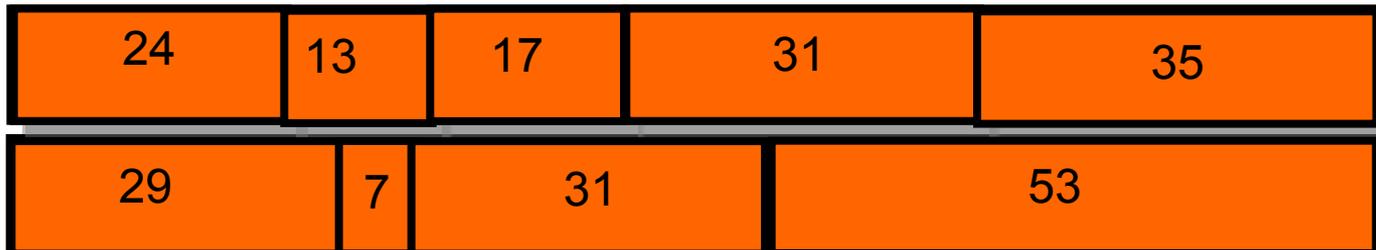
Eine Packung in zwei Container der Größe K

Kofferpacken für die Reise

Gegeben:

Eine Menge von n Objekten, jedes mit einer Größe l_i

Gesamtgröße ist $\sum_{i=1}^n l_i = 2K$



Gesucht:

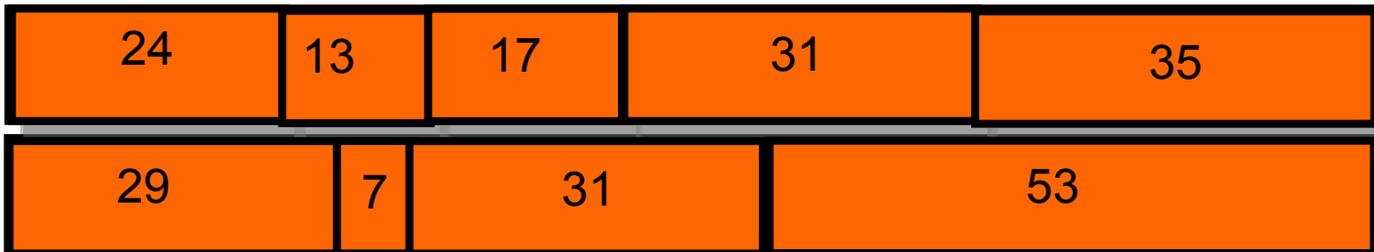
Eine Packung in zwei Container der Größe K

Kofferpacken für die Reise

Gegeben:

Eine Menge von n Objekten, jedes mit einer Größe l_i

Gesamtgröße ist $\sum_{i=1}^n l_i = 2K$



Gesucht:

Eine Packung in zwei Container der Größe K

Karp (1972): Das Finden einer optimalen Partition ist NP-vollständig.

Packen und Presse(n)

BRAUNSCHWEIGER ZEITUNG

WOLFENBÜTTELER ZEITUNG UND ANZEIGER

Samstag, 28. Januar 2006

unabhängig · nicht parteigebunden

Nr. 24 · 61. Jahrgang · 1,20 €

Wenn Mathematiker Autos beladen

TU-Professor Sándor Fekete tüftelt an ganz handfesten Alltagsproblemen, kombiniert und optimiert

Von Cornelia Steiner

Sieben große Probleme müssen Mathematiker weltweit lösen, jeweils eine Million Dollar Preisgeld sind dafür ausgeschrieben. Bei manchen von ihnen versteht man selbst die Fragestellung erst nach einigen Fachsemestern. Das gesteht sogar Sándor Fekete, Professor für mathematische Optimierung an der TU Braunschweig. Er hat auch nicht den Anspruch, eines dieser abstrakten Probleme zu lösen. Stattdessen widmet er sich mit Kollegen und Studenten den handfesten Fragen des alltäglichen Lebens. Denn Mathematik hat eben doch etwas mit dem Hier und Jetzt zu tun: mit Routenplanung zum Beispiel, mit dem Packen eines Kofferraumes oder sogar mit Rasenmähen. All das sind Probleme, bei denen es darum geht, aus einer bestimmten Menge möglicher Kombinationen die beste Lösung zu finden.

In jedem Fall gilt: „Man braucht immer einen Algorithmus, der unabhängig vom Einzelfall eine systematische Lösung anbietet“, sagt Fekete. Für die Routenplanung bedeutet das: Ein einziger Algorithmus liefert sowohl den kürzesten Weg von

„Spannend wird das Lösen von Optimierungsproblemen dadurch, dass man oft nicht alle Informationen zur Verfügung hat.“

Sándor Fekete

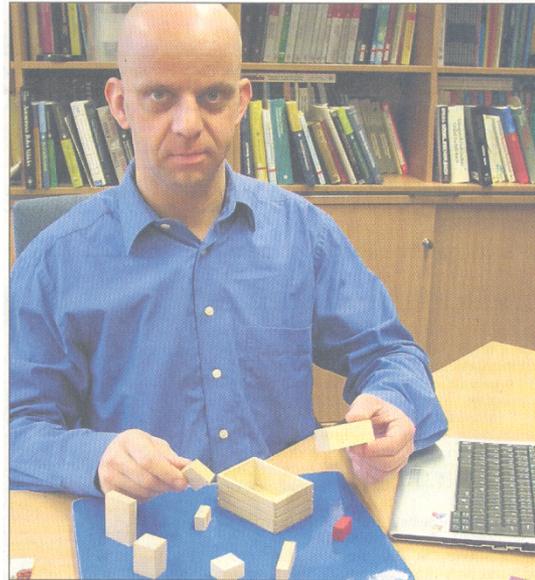
dar. In unserem Fall wären die Knoten Kreuzungen und die Verbindungen Wege. Bevor gerechnet wird, muss der Umfang des Graphen allerdings eingeschränkt werden: Für die Fahrt nach Peine ist schließlich das Umfeld von Helmstedt uninteressant. „Wenn so eine Struktur vorhanden ist, kann man sehr gut ermitteln, wie man am schnellsten zum Wunschort kommt“, sagt er.

Doch alles hat auch seine Tücken, denn es gibt einfache und schwere Probleme: Einfach ist es, den kürzesten Weg zwischen zwei Punkten zu berechnen. Einfach ist es auch, die schnellste Verbindung in einem

Fahrplan zu finden, oder die billigste Flugverbindung zu ermitteln.

Schwierig wird das Ganze aber, sobald mehrere Wegeprobleme ineinander greifen: Das ist der Fall, wenn der kürzeste Weg über mehrere Punkte in unbestimmter Reihenfolge berechnet wird. Wer also eine schnelle Rundreise

durch alle Braunschweiger Kneipen plant, sollte wissen, wie das funktioniert. Kompliziert ist es auch, einen guten Bahnfahrplan zu erstellen, oder Flugpläne. Denn dann müssen zusätzliche Aspekte beachtet werden – Art der Flugzeuge, Größe der Besatzung, Strecken, Anzahl der



Professor Sándor Fekete zeigt auf einfache Weise die Probleme, die sich beim Packen und Beladen eines Kofferraumes ergeben können. Fotos (2): Steiner

nicht alle wichtigen Informationen zur Verfügung hat“, sagt er. So ist zum Beispiel ungewiss, ob ein Stau auftritt oder wie viele Passagiere tat-

sich mit modernen Computersystemen und großen Kommunikationsnetzen beschäftigen, aber auch mit selbsttätigen Paketen und da-

Kofferraumdesign

Kofferraumdesign



Kofferraumdesign



Kofferraumdesign



Wie groß ist der Kofferraum?

Kofferraumdesign



Wie groß ist der Kofferraum?

DIN-Messmethode: Zahl der 5cm*10cm*20cm-Quader (Tetra Paks), die sich packen lassen

Kofferraumdesign

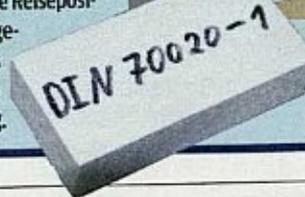


Klötzchen für Klötzchen: So hat der TÜV getestet



Maßarbeit: Beladen bis zur Fensterkante. Anschließend muss die Klappe problemlos schließen.

■ 20 Autos, drei TÜV-Experten und 3000 VDA-Quader - der Kofferraum-Vergleich dauerte eine ganze Woche. Allein das Beladen des Audi Q7 brauchte rund sechs Stunden. Akribisch stapelten die Spezialisten die Styroporklötzchen in jedes Auto. Basierend auf DIN 70020-1 und ISO 3832, wurde nach TÜV-Spezifikation getestet. Dazu befanden sich die Sitze der Rückbank - wo möglich - in hinterster Position, die Lehne war für eine entspannte Reiseposition justiert, und die Höhe des gemessenen Kofferraumvolumens war begrenzt durch Hutablage oder die Gepäckraumabdeckung.



Konzentration: Die Quader (200 mal 100 mm) müssen gestapelt und ausgezählt werden.

DIN-Mess
Quader

20cm-
Quader

Kofferraumdesign



Wie groß ist der Kofferraum?

DIN-Messmethode: Zahl der 5cm*10cm*20cm-Quader (Tetra Paks), die sich packen lassen

Kofferraumdesign



Wie groß ist der Kofferraum?

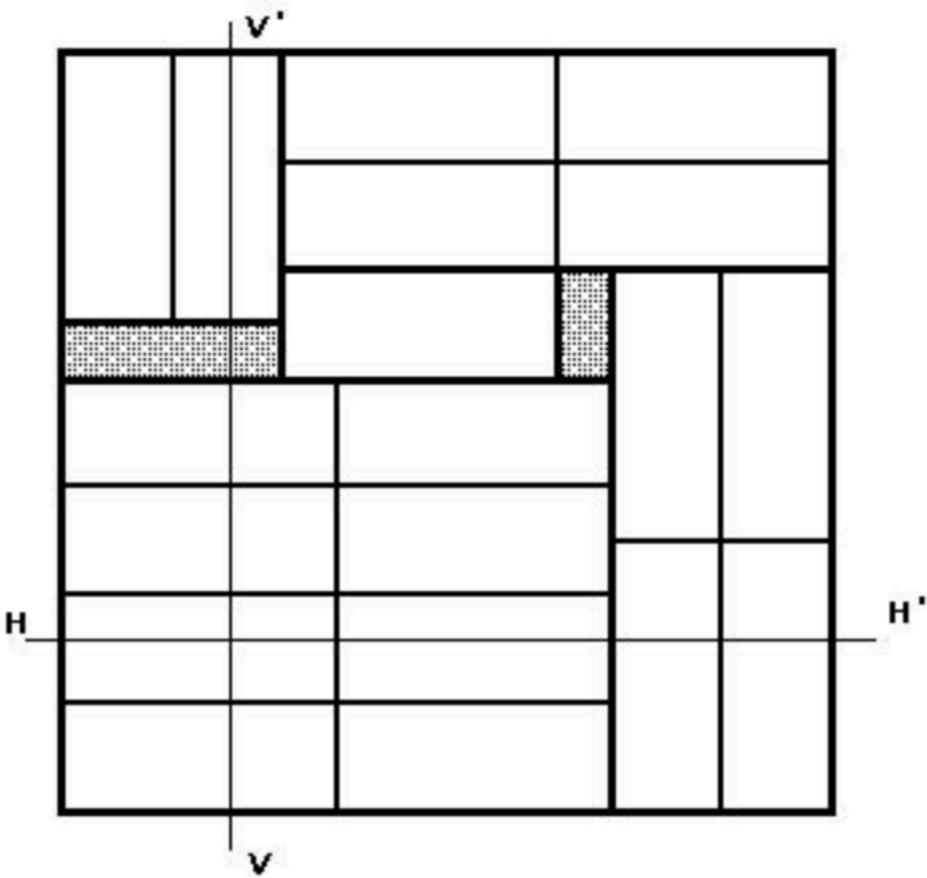
DIN-Messmethode: Zahl der 5cm*10cm*20cm-Quader (Tetra Paks), die sich packen lassen

Schwierigkeit: Mehrdimensionales Packen ist praktisch noch schwerer als eindimensionales!





Kofferraumdesign



[Next](#) [Up](#) [Previous](#)

Next: [Numerical List of All](#)

The Open Problems Project

edited by [Erik D. Demaine](#)

[Joseph S. B. Mitchell](#)

[Joseph O'Rourke](#)

Introduction

This is the beginning of a project¹ to record open problems of interest to researchers in computational geometry and related fields. It commenced with the publication of thirty problems in Computational Geometry Column 42 [[MO01](#)] (see [Problems 1-30](#)), but has grown much beyond that. We encourage correspondence to improve the entries; please send email to topp@csail.mit.edu. If you would like to submit a new problem, please fill out [this template](#).

Each problem is assigned a unique number for citation purposes. Problem numbers also indicate the order in which the problems were entered. Each problem is classified as belonging to one or more categories.

The problems are also available as a single [Postscript](#) or [PDF](#) file.

To begin navigating through the open problems, you may select from a category of interest below, or view [a list of all problems sorted numerically](#).

Kofferraumdesign

[Next](#) [Up](#) [Previous](#)

Next: [Problem 56: Packing Unit](#) **Up:** [The Open Problems Project](#) **Previous:** [Problem 54: Traveling Salesman](#)

Problem 55: Pallet Loading

Statement

What is the complexity of the pallet loading problem? Given two pairs of numbers, (A, B) and (a, b) , and a number n , decide whether n small rectangles of size $a \times b$, in either axis-parallel orientation, can be packed into a large rectangle of size $A \times B$.

This problem is not even known to be in NP, because of the compact input description, and the possibly complicated structure of a packing, if there is one.

Origin

Uncertain, pending investigation.

Status/Conjectures

Open.

Motivation

Natural packing problem; first-rate example of the relevance of coding input and output.

Partial and Related Results

Tarnowsky [[Tar92](#)] showed that the problem can be solved in time polynomial in the size of the input if we are restricted to "guillotine" patterns, i.e., arrangements of items that can be obtained by a recursive sequence of edge-to-edge cuts. This result uses some nontrivial algebraic methods.

Related Open Problems

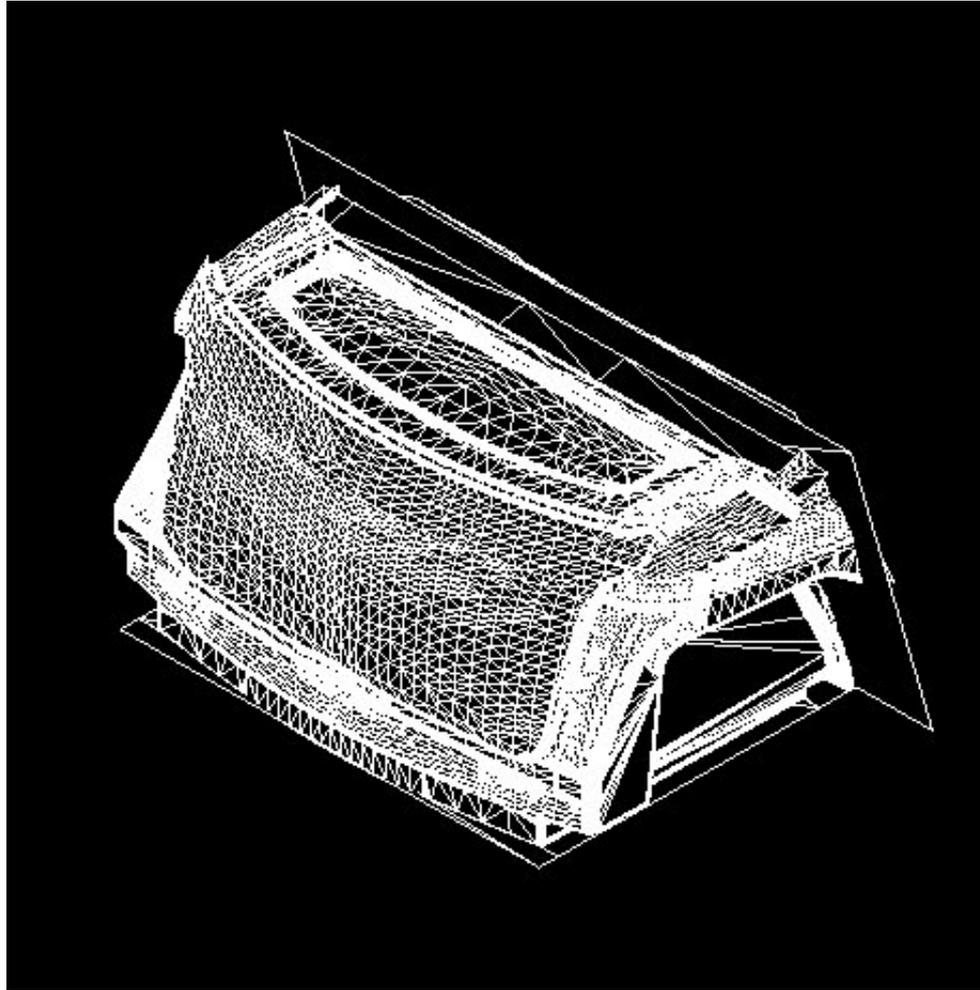
What is the complexity of packing a maximal number of unit squares in a simple polygon? (Problem [54](#))

Appearances

[Dow87](#) claims the problem to be NP-hard; [Exe88](#) claims the problem to be in NP; but both claims are erroneous. The precise nature of the difficulty is stated in

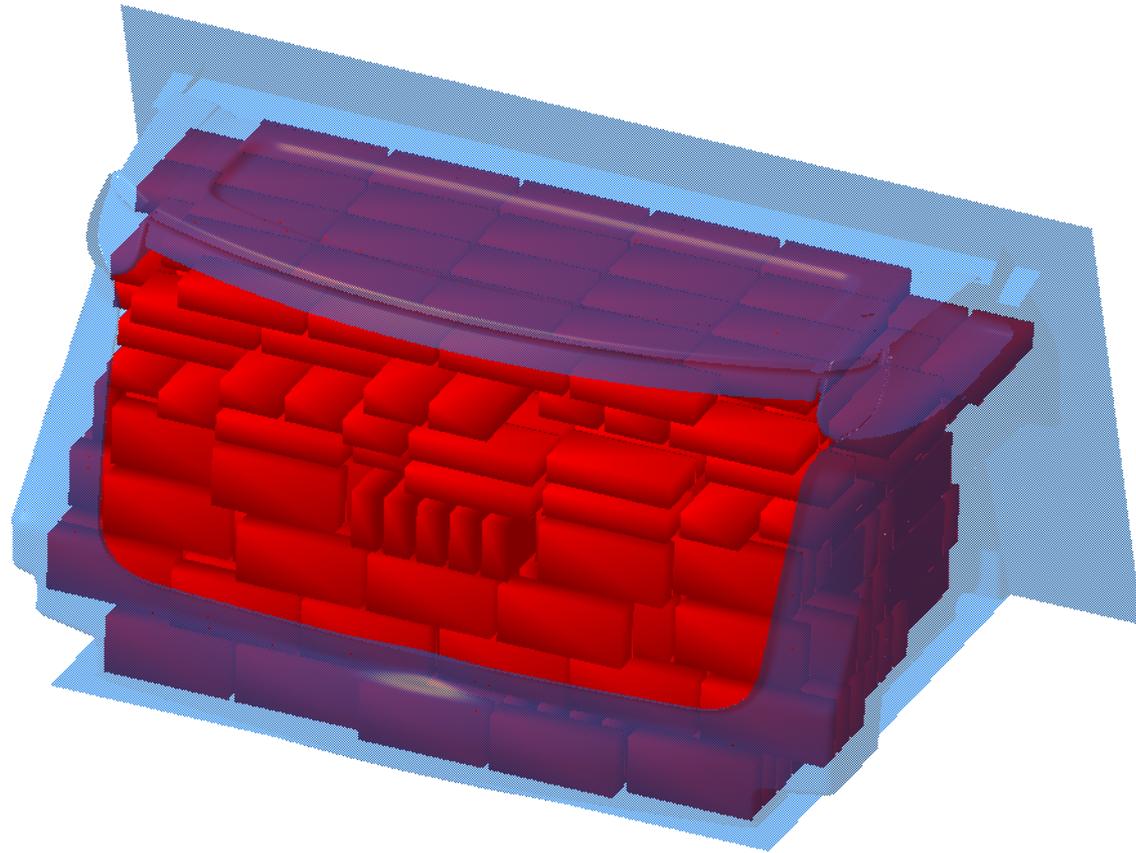
Ergebnisse





Ergebnisse

Ergebnisse



Ergebnisse

Mathematische Methoden

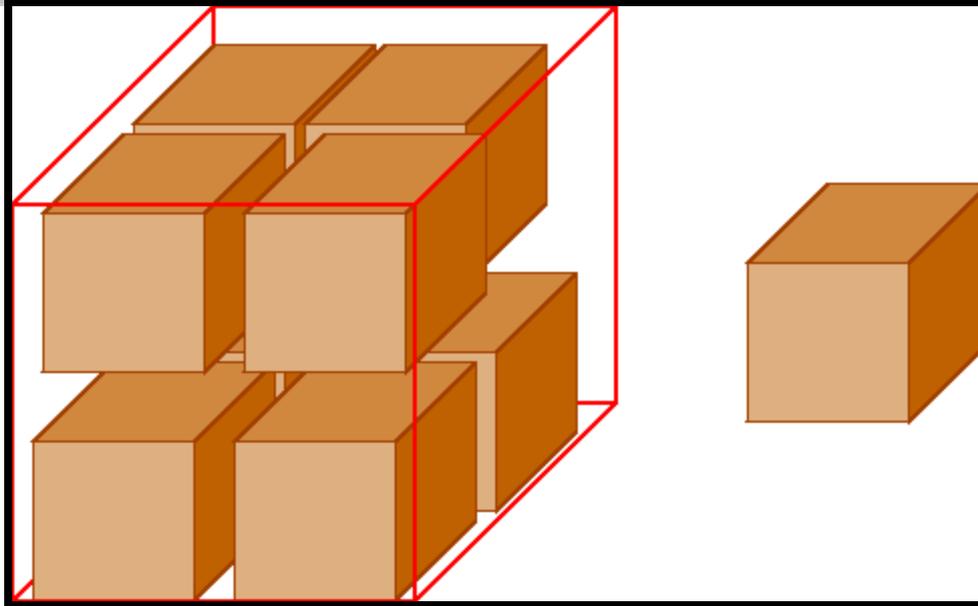
Ein einfaches Beispiel

Ein einfaches Beispiel

Gegeben: 9 Würfel der Kantenlänge 0.4

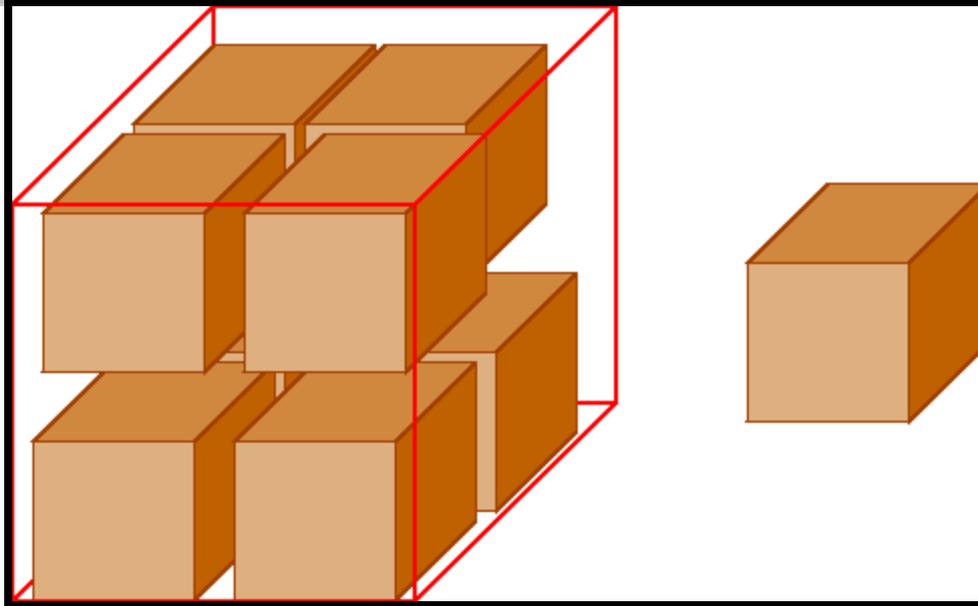
Ein einfaches Beispiel

Gegeben: 9 Würfel der Kantenlänge 0.4



Ein einfaches Beispiel

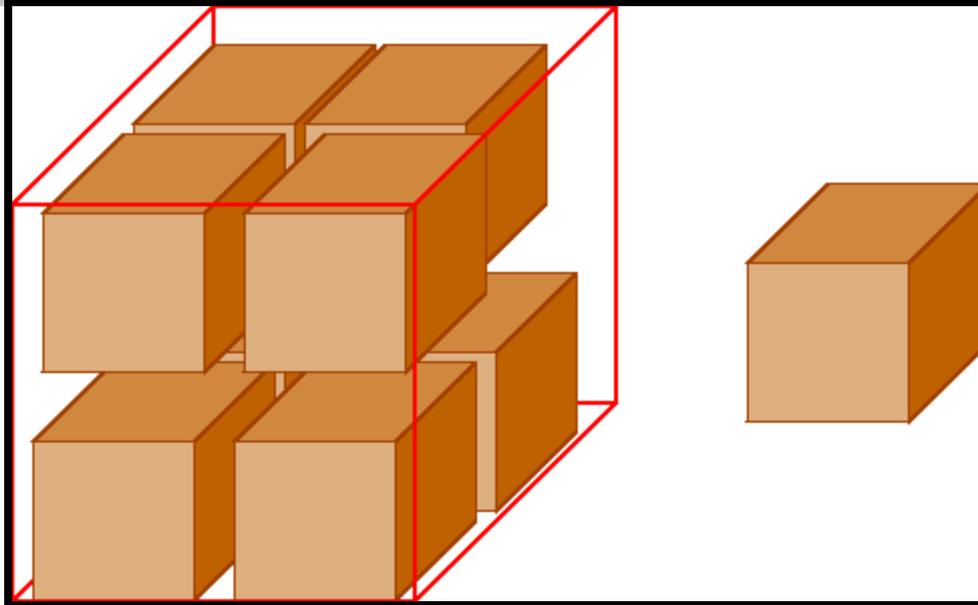
Gegeben: 9 Würfel der Kantenlänge 0.4



Frage: Passen die Würfel in einen würfelförmigen Container der Kantenlänge 1?

Ein einfaches Beispiel

Gegeben: 9 Würfel der Kantenlänge 0.4

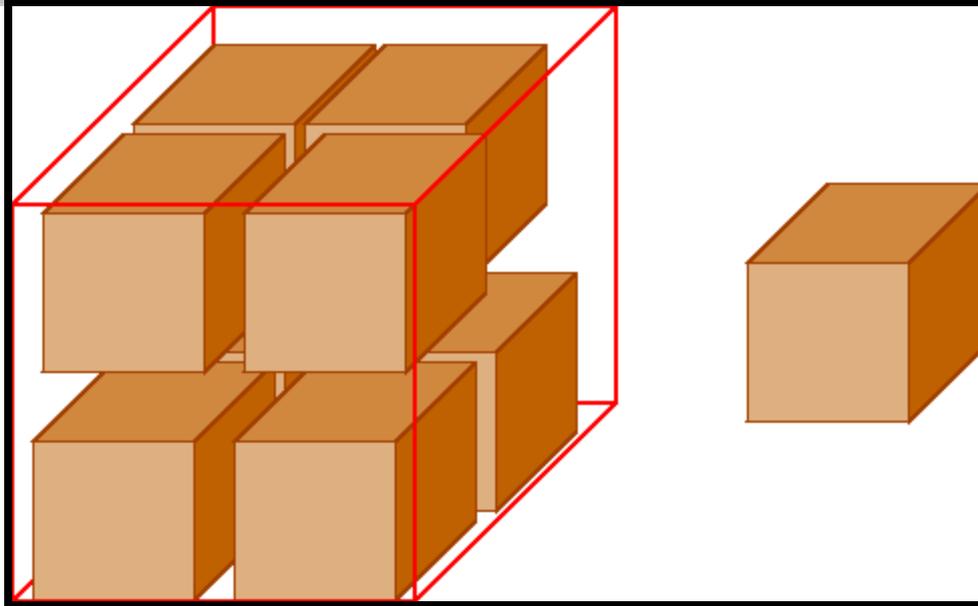


Frage: Passen die Würfel in einen würfelförmigen Container der Kantenlänge 1?

$$\text{Volumen: } 9 * 0.4^3 = 0.576$$

Ein einfaches Beispiel

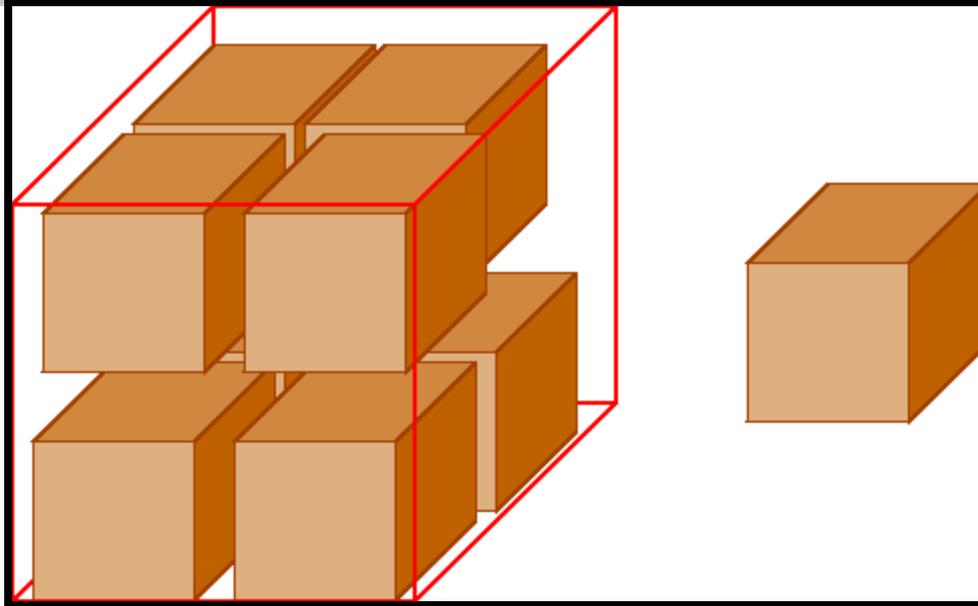
Gegeben: 9 Würfel der Kantenlänge 0.4



Frage: Passen die Würfel in einen würfelförmigen Container der Kantenlänge 1?

Ein einfaches Beispiel

Gegeben: 9 Würfel der Kantenlänge 0.4



Frage: Passen die Würfel in einen würfelförmigen Container der Kantenlänge 1?

„Geht's nicht noch etwas schneller?“

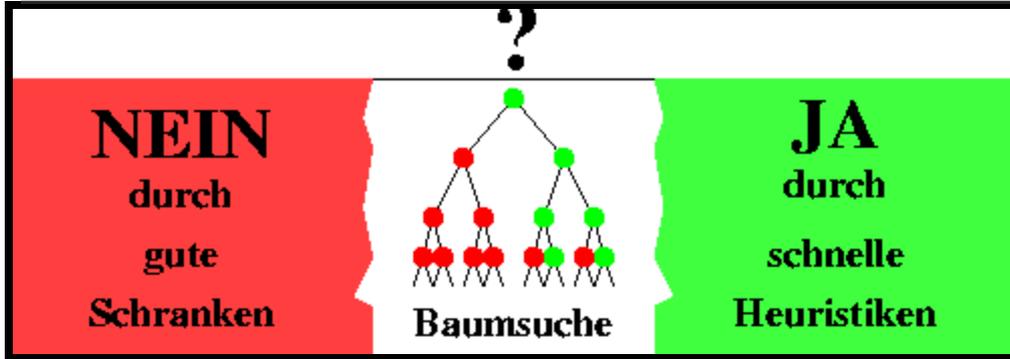
Lösen schwerer Packungsprobleme

Lösen schwerer Packungsprobleme

Ist der Container groß genug?

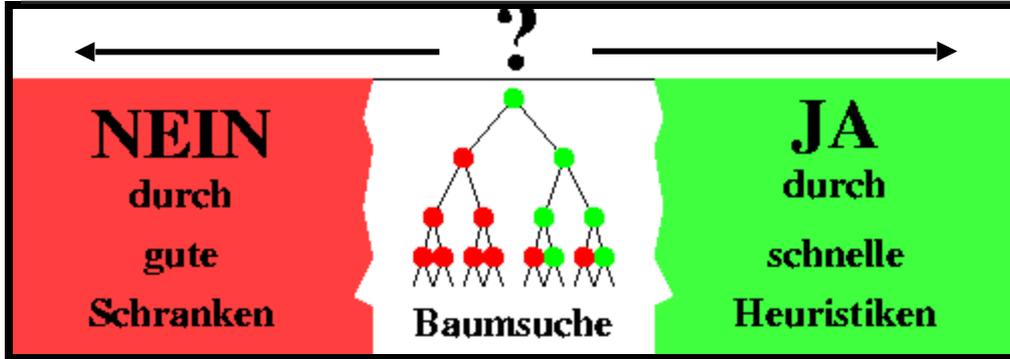
Lösen schwerer Packungsprobleme

Ist der Container groß genug?



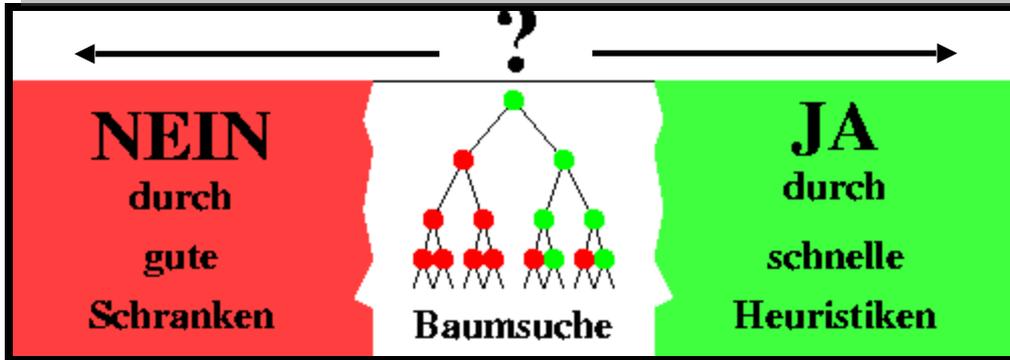
Lösen schwerer Packungsprobleme

Ist der Container groß genug?



Lösen schwerer Packungsprobleme

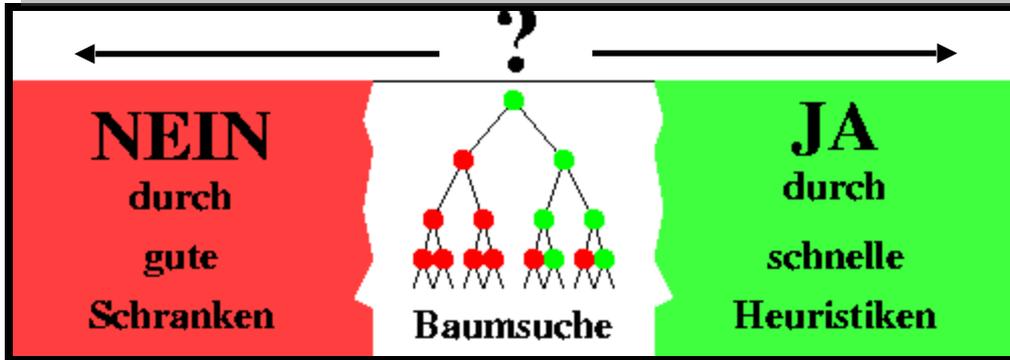
Ist der Container groß genug?



Schnelle Heuristiken:

Lösen schwerer Packungsprobleme

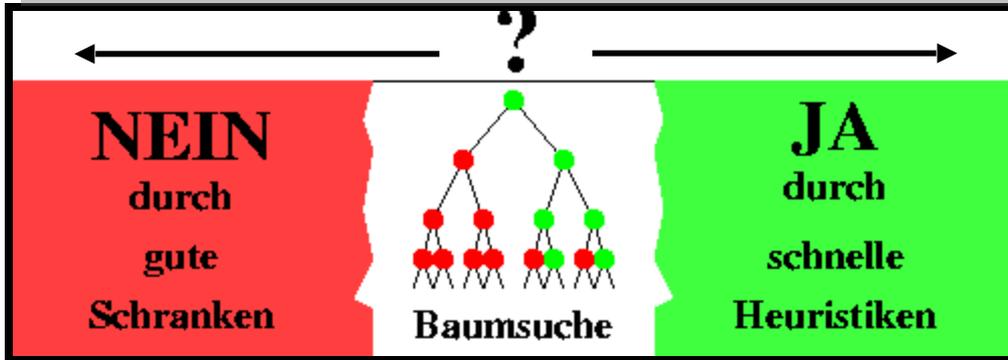
Ist der Container groß genug?



Schnelle Heuristiken:
Bekannt

Lösen schwerer Packungsprobleme

Ist der Container groß genug?

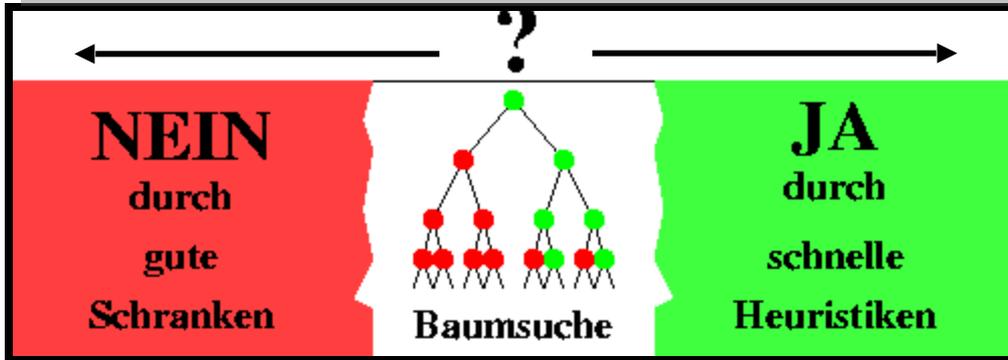


Schnelle Heuristiken:
Bekannt

Gute Schranken:

Lösen schwerer Packungsprobleme

Ist der Container groß genug?



[New classes of fast lower bounds for bin packing problems](#)

SP Fekete... - *Mathematical programming*, 2001 - Springer

... $\sum_{x_i \in T} x_i + \sum_{x_i \in S \setminus T} \lfloor (k+1)x_i \rfloor < (k+1) \sum_{x_i \in S} x_i$. Page 6. 16 Sándor P. Fekete, Jörg Schepers
By definition of $u(k)$, the terms $(k+1) \sum_{x \in T} u(k)(x)$ and $k \sum_{x \in S \setminus T} u(k)(x)$ are integer, so by virtue of $\sum_{x_i \in S} x_i \leq 1$, we have the inequality $(k+1) \sum_{x \in T} u(k)(x) + \dots$

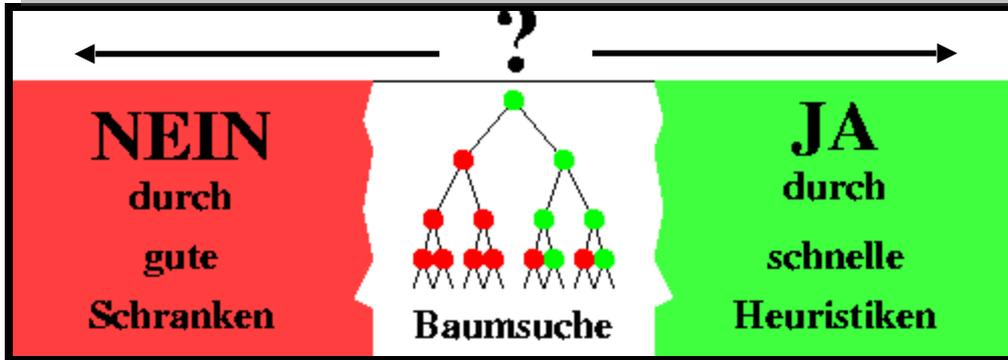
Zitiert durch: 95 - [Ähnliche Artikel](#) - [Alle 11 Versionen](#)

Schnelle Heuristiken:
Bekannt

Gute Schranken:
Fekete&Schepers 2001

Lösen schwerer Packungsprobleme

Ist der Container groß genug?



[New classes of fast lower bounds for bin packing problems](#)

SP Fekete... - [Mathematical programming, 2001 - Springer](#)

... $\sum_{x_i \in T} x_i + \sum_{x_i \in S \setminus T} \lfloor (k+1)x_i \rfloor < (k+1) \sum_{x_i \in S} x_i$. Page 6. 16 Sándor P. Fekete, Jörg Schepers
By definition of $u(k)$, the terms $(k+1) \sum_{x \in T} u(k)(x)$ and $k \sum_{x \in S \setminus T} u(k)(x)$ are integer, so
by virtue of $\sum_{x_i \in S} x_i \leq 1$, we have the inequality $(k+1) \sum_{x \in T} u(k)(x) + \dots$

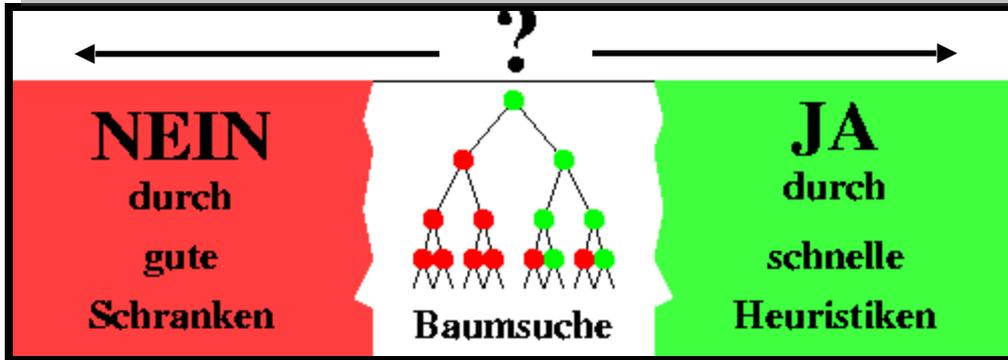
Zitiert durch: [95](#) - [Ähnliche Artikel](#) - [Alle 11 Versionen](#)

Schnelle Heuristiken:
Bekannt

Gute Schranken:
Fekete&Schepers 2001
(Bin Packing, also 1D)

Lösen schwerer Packungsprobleme

Ist der Container groß genug?



[New classes of fast lower bounds for bin packing problems](#)

SP Fekete... - Mathematical programming, 2001 - Springer

... $\sum_{x \in T} x_i + \sum_{x \in S \setminus T} \lfloor (k+1)x_i \rfloor < (k+1) \sum_{x \in S} x_i$. Page 6. 16 Sándor P. Fekete, Jörg Schepers
By definition of $u(k)$, the terms $(k+1) \sum_{x \in T} u(k)(x)$ and $k \sum_{x \in S \setminus T} u(k)(x)$ are integer, so
by virtue of $\sum_{x \in S} x_i \leq 1$, we have the inequality $(k+1) \sum_{x \in T} u(k)(x) + \dots$

Zitiert durch: 95 - [Ähnliche Artikel](#) - [Alle 11 Versionen](#)

[A general framework for bounds for higher-dimensional orthogonal packing problems](#)

SP Fekete... - Mathematical Methods of Operations Research, 2004 - Springer

... Sándor P. Fekete¹, Jörg Schepers²
¹ Department of Mathematical Optimization,
Braunschweig University of Technology, D-38106 Braunschweig, Germany (email:
s.fekete@tu-bs.de)
² IBM Germany, Gustav-Heinemann-Ufer, 120/122 D-50968, Köln, Germany (...

[Cited by 48](#) - [Related articles](#) - [BL Direct](#) - [All 5 versions](#) - [Import into BibTeX](#)

Schnelle Heuristiken:

Bekannt

Gute Schranken:

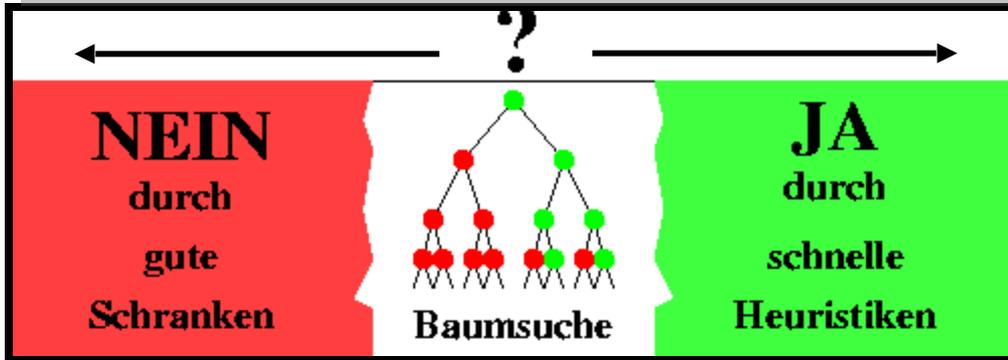
Fekete&Schepers 2001

(Bin Packing, also 1D)

Fekete&Schepers 2004

Lösen schwerer Packungsprobleme

Ist der Container groß genug?



[New classes of fast lower bounds for bin packing problems](#)

SP Fekete... - Mathematical programming, 2001 - Springer

... $\sum_{x \in T} x_i + \sum_{x \in S \setminus T} \lfloor (k+1)x_i \rfloor < (k+1) \sum_{x \in S} x_i$. Page 6. 16 Sándor P. Fekete, Jörg Schepers
By definition of $u(k)$, the terms $(k+1) \sum_{x \in T} u(k)(x)$ and $k \sum_{x \in S \setminus T} u(k)(x)$ are integer, so
by virtue of $\sum_{x \in S} x_i \leq 1$, we have the inequality $(k+1) \sum_{x \in T} u(k)(x) + \dots$

Zitiert durch: 95 - [Ähnliche Artikel](#) - [Alle 11 Versionen](#)

[A general framework for bounds for higher-dimensional orthogonal packing problems](#)

SP Fekete... - Mathematical Methods of Operations Research, 2004 - Springer

... Sándor P. Fekete¹, Jörg Schepers²
¹ Department of Mathematical Optimization,
Braunschweig University of Technology, D-38106 Braunschweig, Germany (email:
s.fekete@tu-bs.de)
² IBM Germany, Gustav-Heinemann-Ufer, 120/122 D-50968, Köln, Germany (...

[Cited by 48](#) - [Related articles](#) - [BL Direct](#) - [All 5 versions](#) - [Import into BibTeX](#)

Schnelle Heuristiken:

Bekannt

Gute Schranken:

Fekete&Schepers 2001

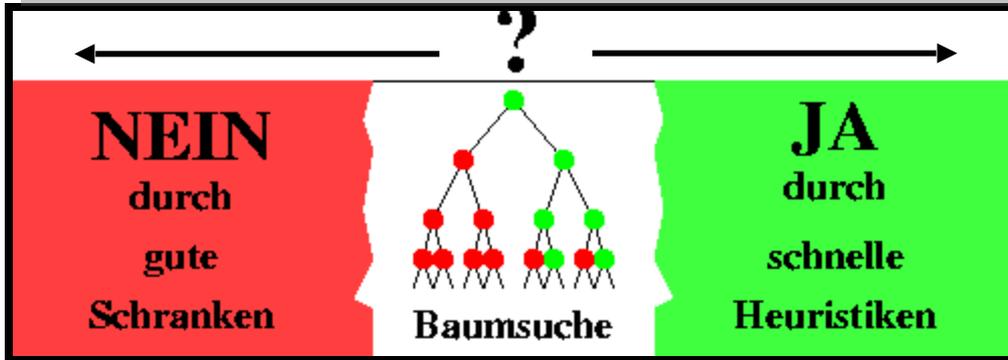
(Bin Packing, also 1D)

Fekete&Schepers 2004

(Höhere Dimensionen)

Lösen schwerer Packungsprobleme

Ist der Container groß genug?



[New classes of fast lower bounds for bin packing problems](#)

SP Fekete... - Mathematical programming, 2001 - Springer

... $x_i \in T$ $x_i + \sum_{x_i \in S \setminus T} \lfloor (k+1)x_i \rfloor < (k+1) \sum_{x_i \in S} x_i$. Page 6. 16 Sándor P. Fekete, Jörg Schepers
By definition of $u(k)$, the terms $(k+1) \sum_{x \in T} u(k)(x)$ and $k \sum_{x \in S \setminus T} u(k)(x)$ are integer, so
by virtue of $\sum_{x_i \in S} x_i \leq 1$, we have the inequality $(k+1) \sum_{x \in T} u(k)(x) + \dots$

Zitiert durch: 95 - Ähnliche Artikel - Alle 11 Versionen

[A general framework for bounds for higher-dimensional orthogonal packing problems](#)

SP Fekete... - Mathematical Methods of Operations Research, 2004 - Springer

... Sándor P. Fekete¹, Jörg Schepers²
¹ Department of Mathematical Optimization,
Braunschweig University of Technology, D-38106 Braunschweig, Germany (email:
s.fekete@tu-bs.de)
² IBM Germany Gustav-Heinemann-Ufer, 120/122 D-50968, Köln, Germany (...

Cited by 48 - Related articles - BL Direct - All 5 versions - Import into BibTeX

Schnelle Heuristiken:

Bekannt

Gute Schranken:

Fekete&Schepers 2001

(Bin Packing, also 1D)

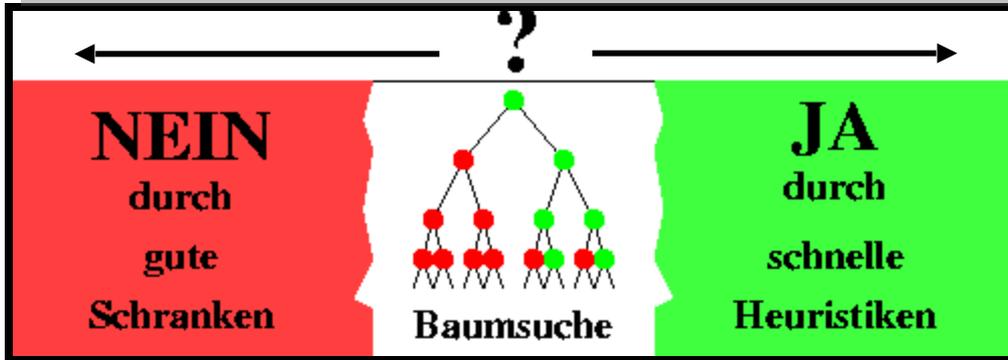
Fekete&Schepers 2004

(Höhere Dimensionen)

Schnelle Baumsuche:

Lösen schwerer Packungsprobleme

Ist der Container groß genug?



[New classes of fast lower bounds for bin packing problems](#)

SP [Fekete...](#) - *Mathematical programming*, 2001 - Springer

... $\sum_{x_i \in T} x_i + \sum_{x_i \in S \setminus T} \lfloor (k+1)x_i \rfloor < (k+1) \sum_{x_i \in S} x_i$. Page 6. 16 Sándor P. [Fekete](#), Jörg Schepers
By definition of $u(k)$, the terms $(k+1) \sum_{x \in T} u(k)(x)$ and $k \sum_{x \in S \setminus T} u(k)(x)$ are integer, so by virtue of $\sum_{x_i \in S} x_i \leq 1$, we have the inequality $(k+1) \sum_{x \in T} u(k)(x) + \dots$

Zitiert durch: 95 - [Ähnliche Artikel](#) - [Alle 11 Versionen](#)

[A general framework for bounds for higher-dimensional orthogonal packing problems](#)

SP [Fekete...](#) - *Mathematical Methods of Operations Research*, 2004 - Springer

... Sándor P. [Fekete](#), Jörg Schepers, 21 Department of Mathematical Optimization, Braunschweig University of Technology, D-38106 Braunschweig, Germany (email: s.fekete@tu-bs.de) 2 IBM Germany Gustav-Heinemann-Ufer, 120/122 D-50968, Köln, Germany (...

[Cited by 48](#) - [Related articles](#) - [BL Direct](#) - [All 5 versions](#) - [Import into BibTeX](#)

[A combinatorial characterization of higher-dimensional orthogonal packing](#)

SP [Fekete...](#) - *Mathematics of Operations Research*, 2004 - JSTOR

... Sándor P. [Fekete](#), Jörg Schepers Department of Mathematical Optimization, Braunschweig University of Technology, D-38106 Braunschweig, Germany, s.fekete@tu-bs.de IBM Germany, Gustav-Heinemann-Ufer 120/122, D-50968 Köln, Germany, schepers@de.ibm.com ...

[Cited by 58](#) - [Related articles](#) - [BL Direct](#) - [All 11 versions](#) - [Import into BibTeX](#)

Schnelle Heuristiken:

Bekannt

Gute Schranken:

Fekete&Schepers 2001

(Bin Packing, also 1D)

Fekete&Schepers 2004

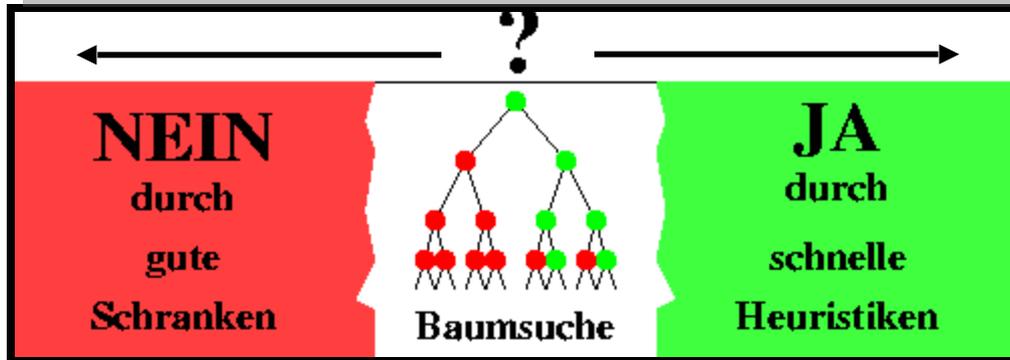
(Höhere Dimensionen)

Schnelle Baumsuche:

Fekete&Schepers 2004.

Lösen schwerer Packungsprobleme

Ist der Container groß genug?



[New classes of fast lower bounds for bin packing problems](#)

SP Fekete... - Mathematical programming, 2001 - Springer

... $\sum_{x_i \in T} x_i + \sum_{x_i \in S \setminus T} \lfloor (k+1)x_i \rfloor < (k+1) \sum_{x_i \in S} x_i$. Page 6. 16 Sándor P. Fekete, Jörg Schepers
By definition of $u(k)$, the terms $(k+1) \sum_{x \in T} u(k)(x)$ and $k \sum_{x \in S \setminus T} u(k)(x)$ are integer, so by virtue of $\sum_{x_i \in S} x_i \leq 1$, we have the inequality $(k+1) \sum_{x \in T} u(k)(x) + \dots$

Zitiert durch: 95 - [Ähnliche Artikel](#) - [Alle 11 Versionen](#)

[A general framework for bounds for higher-dimensional orthogonal packing problems](#)

SP Fekete... - Mathematical Methods of Operations Research, 2004 - Springer

... Sándor P. Fekete¹, Jörg Schepers²
¹ Department of Mathematical Optimization, Braunschweig University of Technology, D-38106 Braunschweig, Germany (email: s.fekete@tu-bs.de)
² IBM Germany, Gustav-Heinemann-Ufer, 120/122 D-50968, Köln, Germany (...

[Cited by 48](#) - [Related articles](#) - [BL Direct](#) - [All 5 versions](#) - [Import into BibTeX](#)

[A combinatorial characterization of higher-dimensional orthogonal packing](#)

SP Fekete... - Mathematics of Operations Research, 2004 - JSTOR

... Sándor P. Fekete¹, Jörg Schepers²
¹ Department of Mathematical Optimization, Braunschweig University of Technology, D-38106 Braunschweig, Germany, s.fekete@tu-bs.de
² IBM Germany, Gustav-Heinemann-Ufer 120/122, D-50968 Köln, Germany, schepers@de.ibm.com ...

[Cited by 58](#) - [Related articles](#) - [BL Direct](#) - [All 11 versions](#) - [Import into BibTeX](#)

Schnelle Heuristiken:
Bekannt

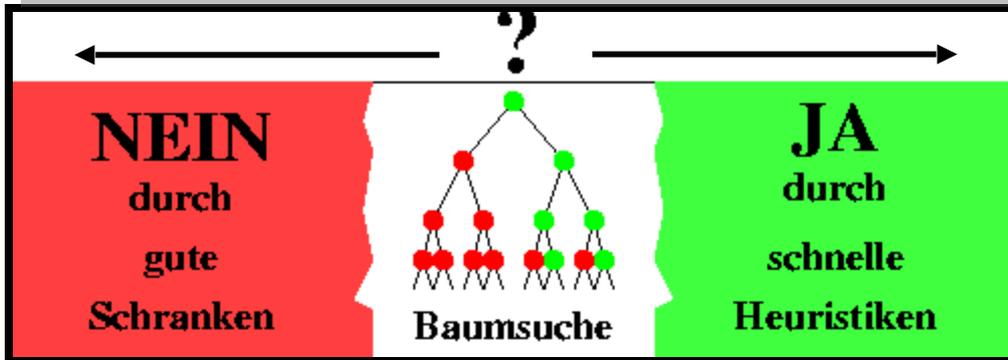
Gute Schranken:
Fekete&Schepers 2001
(Bin Packing, also 1D)
Fekete&Schepers 2004
(Höhere Dimensionen)

Schnelle Baumsuche:
Fekete&Schepers 2004.

Praktische Umsetzung:

Lösen schwerer Packungsprobleme

Ist der Container groß genug?



[New classes of fast lower bounds for bin packing problems](#)

SP Fekete... - *Mathematical programming*, 2001 - Springer

... $\sum_{x_i \in T} x_i + \sum_{x_i \in S \setminus T} \lfloor (k+1)x_i \rfloor < (k+1) \sum_{x_i \in S} x_i$. Page 6. 16 Sándor P. Fekete, Jörg Schepers
By definition of $u(k)$, the terms $(k+1) \sum_{x \in T} u(k)(x)$ and $k \sum_{x \in S \setminus T} u(k)(x)$ are integer, so by virtue of $\sum_{x_i \in S} x_i \leq 1$, we have the inequality $(k+1) \sum_{x \in T} u(k)(x) + \dots$

Zitiert durch: 95 - [Ähnliche Artikel](#) - [Alle 11 Versionen](#)

[A general framework for bounds for higher-dimensional orthogonal packing problems](#)

SP Fekete... - *Mathematical Methods of Operations Research*, 2004 - Springer

... Sándor P. Fekete¹, Jörg Schepers²
¹ Department of Mathematical Optimization, Braunschweig University of Technology, D-38106 Braunschweig, Germany (email: s.fekete@tu-bs.de)
² IBM Germany Gustav-Heinemann-Ufer, 120/122 D-50968, Köln, Germany (...

Cited by 48 - [Related articles](#) - [BL Direct](#) - [All 5 versions](#) - [Import into BibTeX](#)

[A combinatorial characterization of higher-dimensional orthogonal packing](#)

SP Fekete... - *Mathematics of Operations Research*, 2004 - JSTOR

... Sándor P. Fekete¹, Jörg Schepers²
¹ Department of Mathematical Optimization, Braunschweig University of Technology, D-38106 Braunschweig, Germany, s.fekete@tu-bs.de
² IBM Germany, Gustav-Heinemann-Ufer 120/122, D-50968 Köln, Germany, schepers@de.ibm.com ...

Cited by 58 - [Related articles](#) - [BL Direct](#) - [All 11 versions](#) - [Import into BibTeX](#)

[An exact algorithm for higher-dimensional orthogonal packing](#)

SP Fekete, J Schepers... - *Arxiv preprint cs/0604045*, 2006 - arxiv.org

Page 1. arXiv:cs/0604045v1 [cs.DS] 11 Apr 2006 An Exact Algorithm for Higher-Dimensional Orthogonal Packing • Sándor P. Fekete Department of Mathematical Optimization Braunschweig University of Technology D-38106 Braunschweig GERMANY s.fekete@tu-bs.de ...

Zitiert durch: 68 - [Ähnliche Artikel](#)

Schnelle Heuristiken:

Bekannt

Gute Schranken:

Fekete&Schepers 2001

(Bin Packing, also 1D)

Fekete&Schepers 2004

(Höhere Dimensionen)

Schnelle Baumsuche:

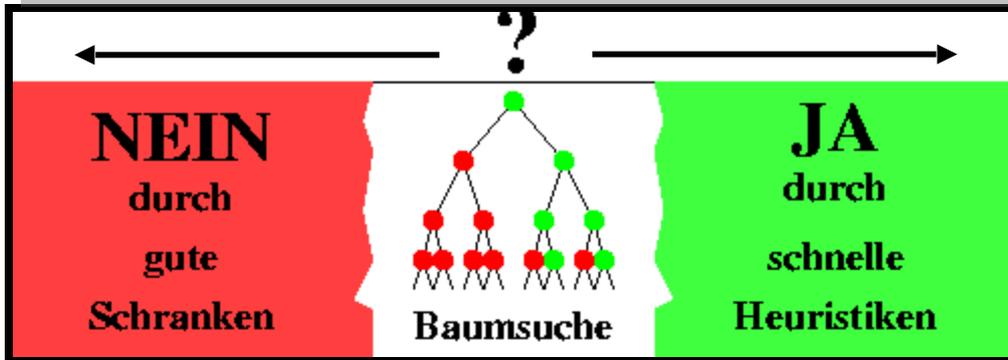
Fekete&Schepers 2004.

Praktische Umsetzung:

Fekete, Schepers, van der Veen 2006

Lösen schwerer Packungsprobleme

Ist der Container groß genug?



[New classes of fast lower bounds for bin packing problems](#)

SP Fekete... - Mathematical programming, 2001 - Springer

... $\sum_{i \in T} x_i + \sum_{i \in S \setminus T} \lfloor (k+1)x_i \rfloor < (k+1) \sum_{i \in S} x_i$. Page 6. 16 Sándor P. Fekete, Jörg Schepers
By definition of $u(k)$, the terms $(k+1) \sum_{x \in T} u(k)(x)$ and $k \sum_{x \in S \setminus T} u(k)(x)$ are integer, so by virtue of $\sum_{i \in S} x_i \leq 1$, we have the inequality $(k+1) \sum_{x \in T} u(k)(x) + \dots$

Zitiert durch: 95 - [Ähnliche Artikel](#) - [Alle 11 Versionen](#)

[A general framework for bounds for higher-dimensional orthogonal packing problems](#)

SP Fekete... - Mathematical Methods of Operations Research, 2004 - Springer

... Sándor P. Fekete¹, Jörg Schepers²
¹ Department of Mathematical Optimization, Braunschweig University of Technology, D-38106 Braunschweig, Germany (email: s.fekete@tu-bs.de)
² IBM Germany Gustav-Heinemann-Ufer, 120/122 D-50968, Köln, Germany (...

Cited by 48 - [Related articles](#) - [BL Direct](#) - [All 5 versions](#) - [Import into BibTeX](#)

[A combinatorial characterization of higher-dimensional orthogonal packing](#)

SP Fekete... - Mathematics of Operations Research, 2004 - JSTOR

... Sándor P. Fekete¹, Jörg Schepers²
¹ Department of Mathematical Optimization, Braunschweig University of Technology, D-38106 Braunschweig, Germany, s.fekete@tu-bs.de
² IBM Germany, Gustav-Heinemann-Ufer 120/122, D-50968 Köln, Germany, schepers@de.ibm.com ...

Cited by 58 - [Related articles](#) - [BL Direct](#) - [All 11 versions](#) - [Import into BibTeX](#)

[An exact algorithm for higher-dimensional orthogonal packing](#)

SP Fekete, J Schepers... - Arxiv preprint cs/0604045, 2006 - arxiv.org

Page 1. arXiv:cs/0604045v1 [cs.DS] 11 Apr 2006 An Exact Algorithm for Higher-Dimensional Orthogonal Packing • Sándor P. Fekete Department of Mathematical Optimization Braunschweig University of Technology D-38106 Braunschweig GERMANY s.fekete@tu-bs.de ...

Zitiert durch: 68 - [Ähnliche Artikel](#)

Schnelle Heuristiken:

Bekannt

Gute Schranken:

Fekete&Schepers 2001

(Bin Packing, also 1D)

Fekete&Schepers 2004

(Höhere Dimensionen)

Schnelle Baumsuche:

Fekete&Schepers 2004.

Praktische Umsetzung:

Fekete, Schepers, van der Veen 2006

Gute und schnelle Schranken

Idee: Betrachte *dualzulässige Funktionen*:

Idee: Betrachte *dualzulässige Funktionen*:

$$(1) f: [0,1] \rightarrow [0,1]$$

Idee: Betrachte *dualzulässige Funktionen*:

$$(1) f: [0,1] \rightarrow [0,1]$$

$$(2) \sum_{i \in S} w_i \leq 1 \Rightarrow \sum_{i \in S} f(w_i) \leq 1$$

Idee: Betrachte *dualzulässige Funktionen*:

$$(1) f: [0,1] \rightarrow [0,1]$$

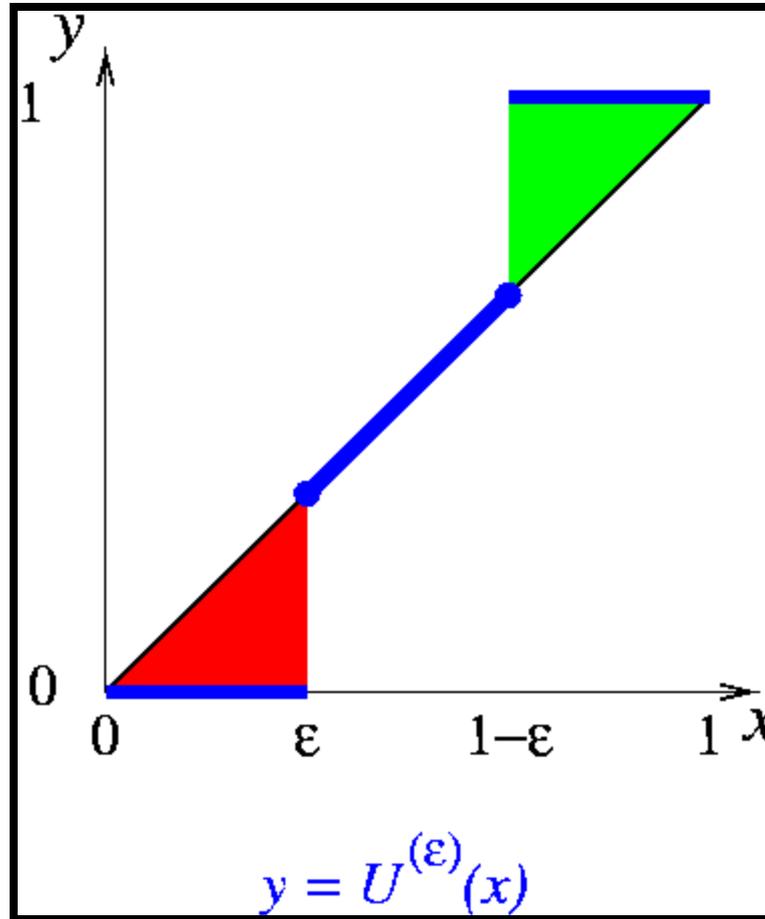
$$(2) \sum_{i \in S} w_i \leq 1 \Rightarrow \sum_{i \in S} f(w_i) \leq 1$$

Satz (Fekete & Schepers):

Ist f eine dualzulässige Funktion, dann ist $\sum_{i \in S} f(w_i)$ eine untere Schranke für das benötigte Volumen.

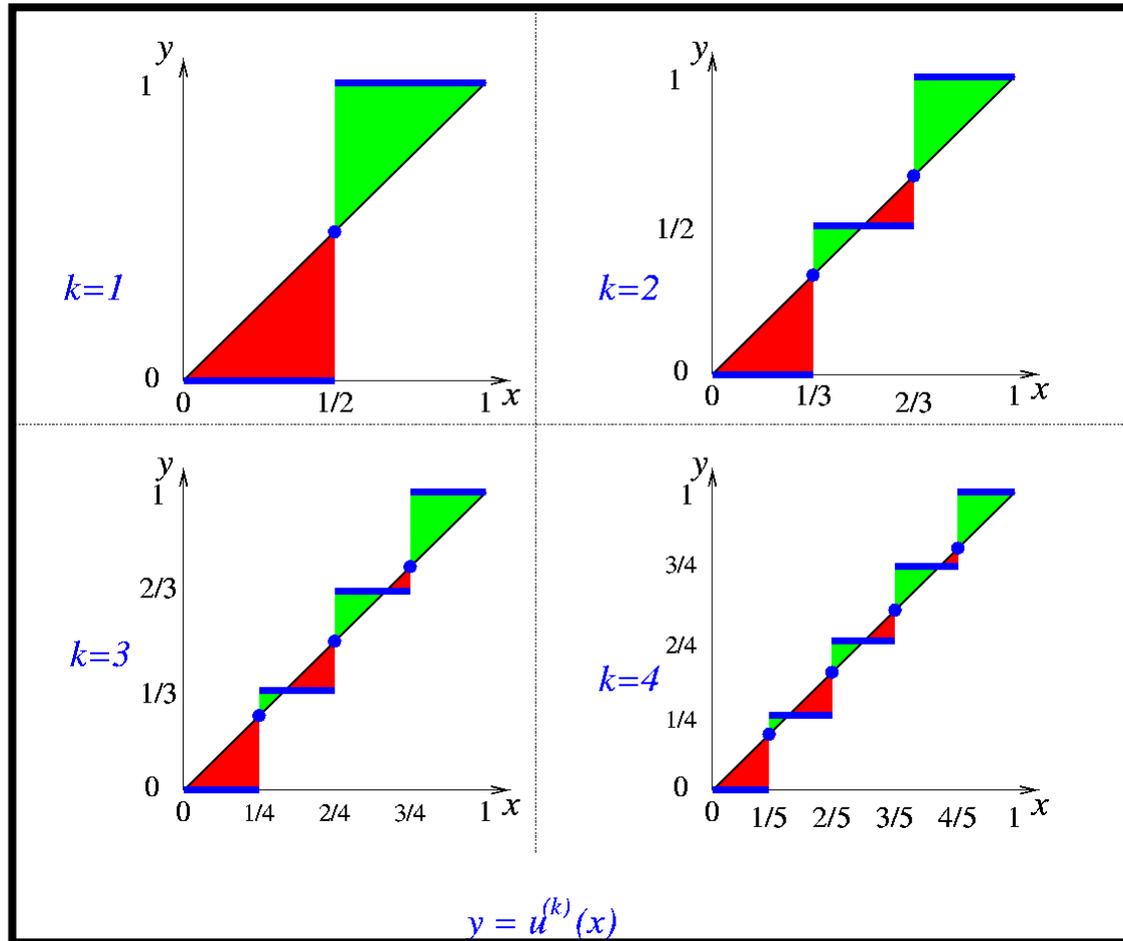
Gute und schnelle Schranken

Gute und schnelle Schranken

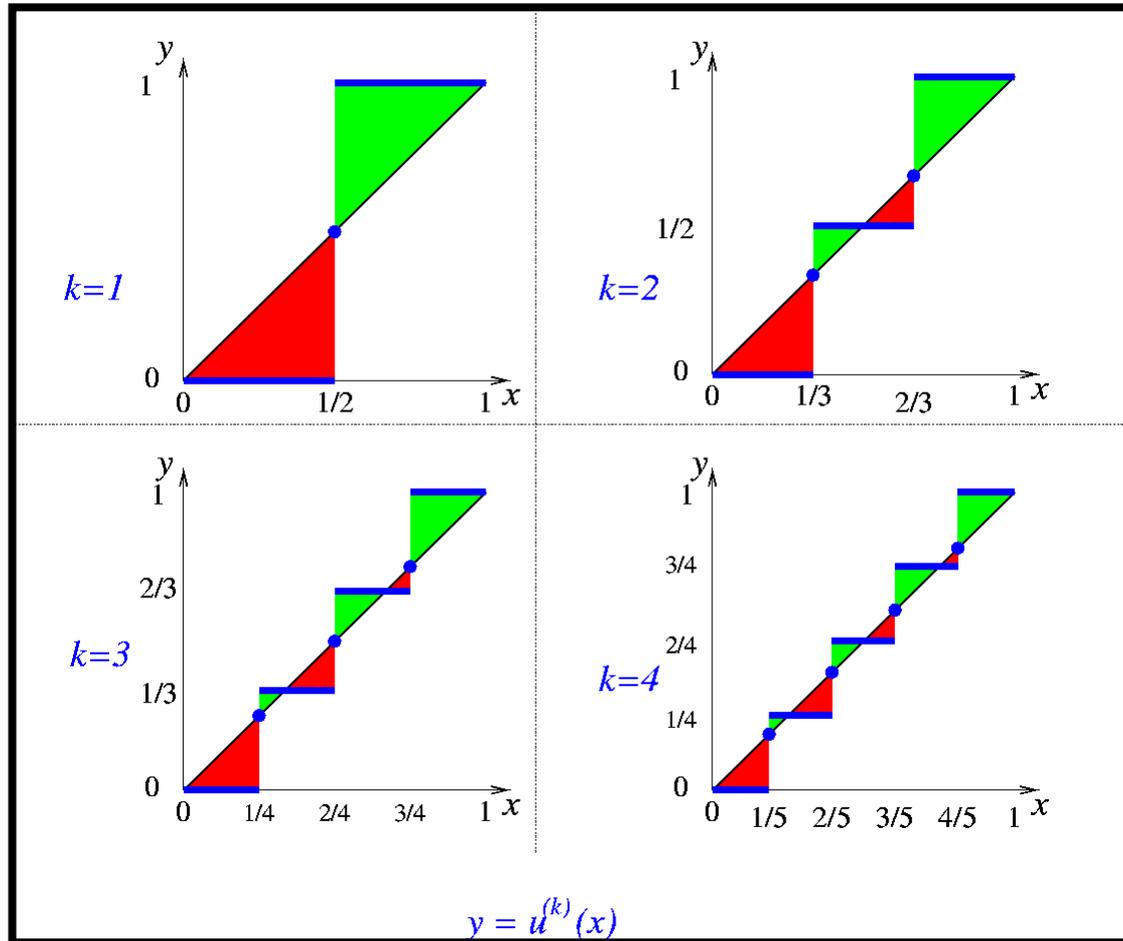


Gute und schnelle Schranken

Gute und schnelle Schranken

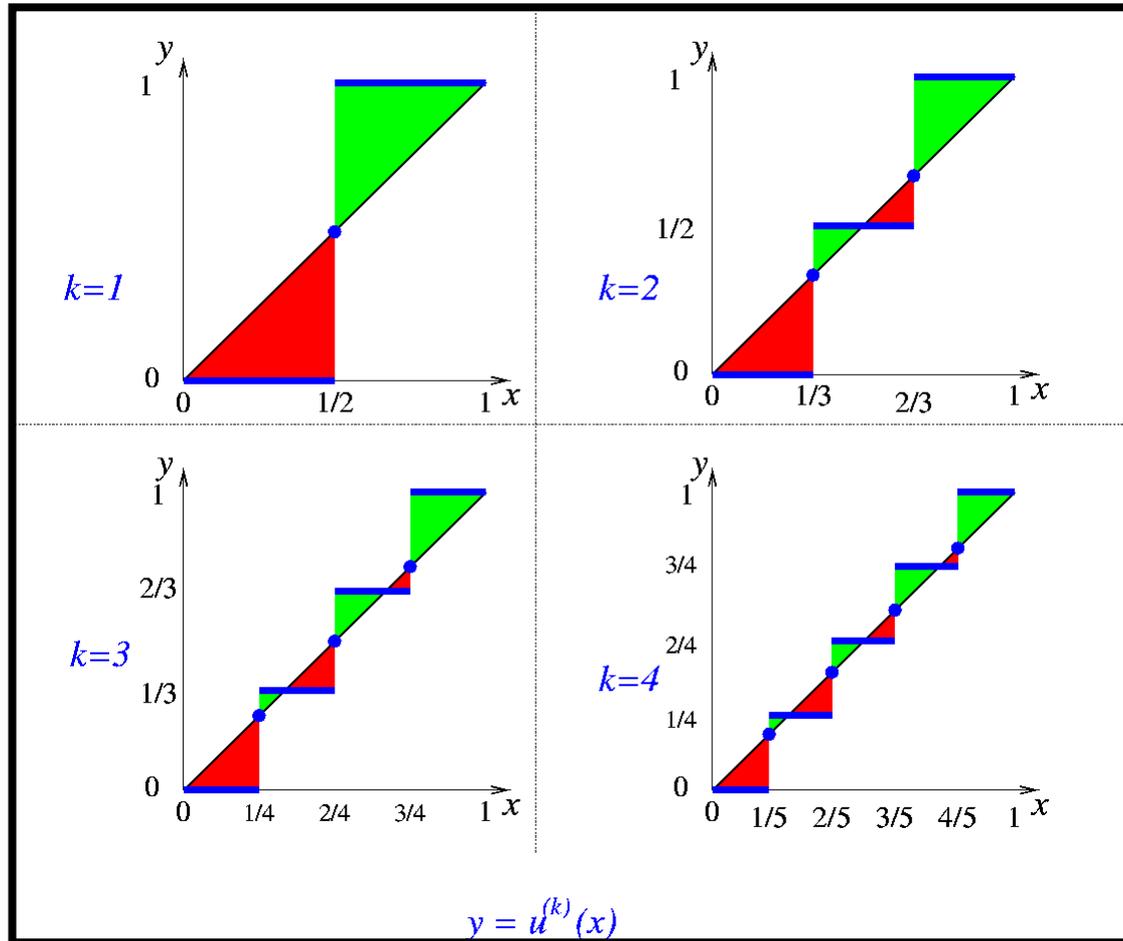


Gute und schnelle Schranken

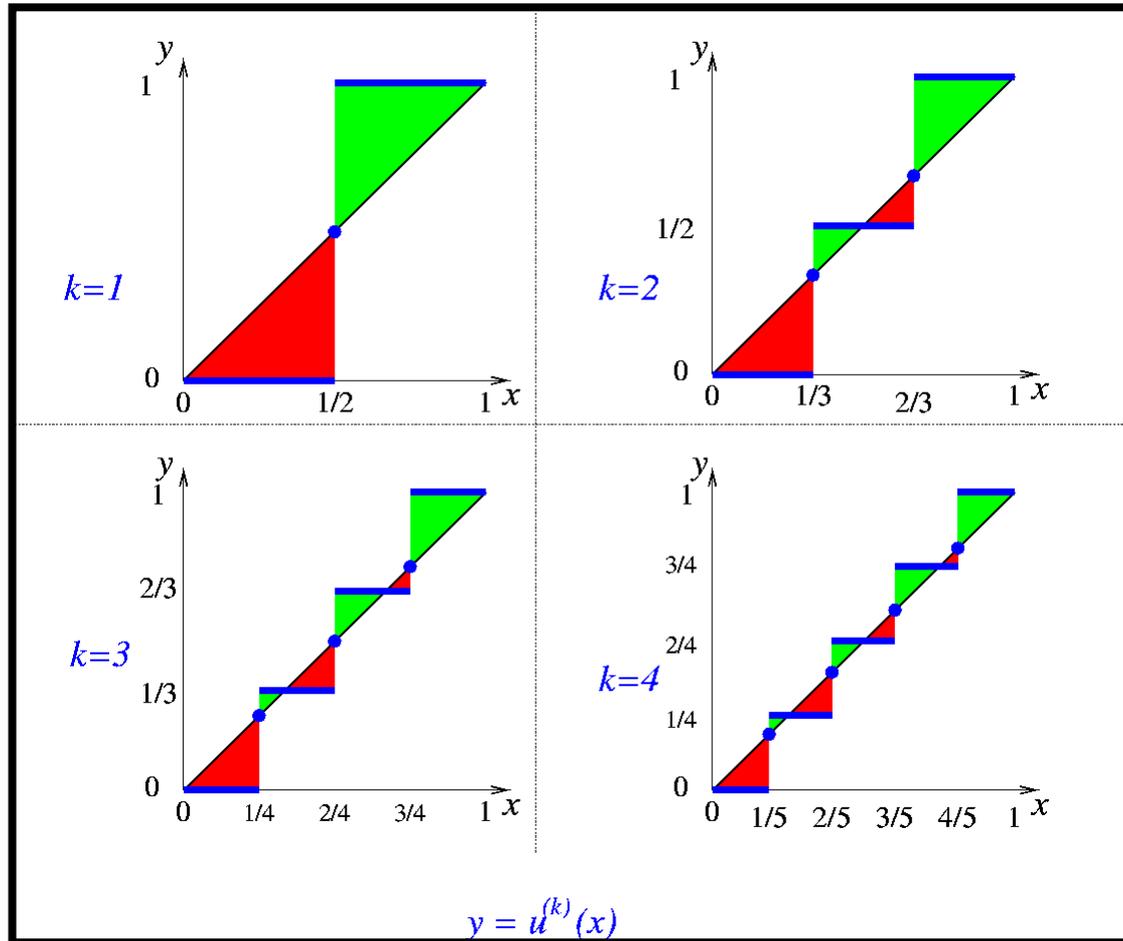


Konsequenz I: Die resultierenden Schranken sind in Linearzeit berechenbar und haben eine Gütegarantie von $\frac{3}{4}$.

Gute und schnelle Schranken

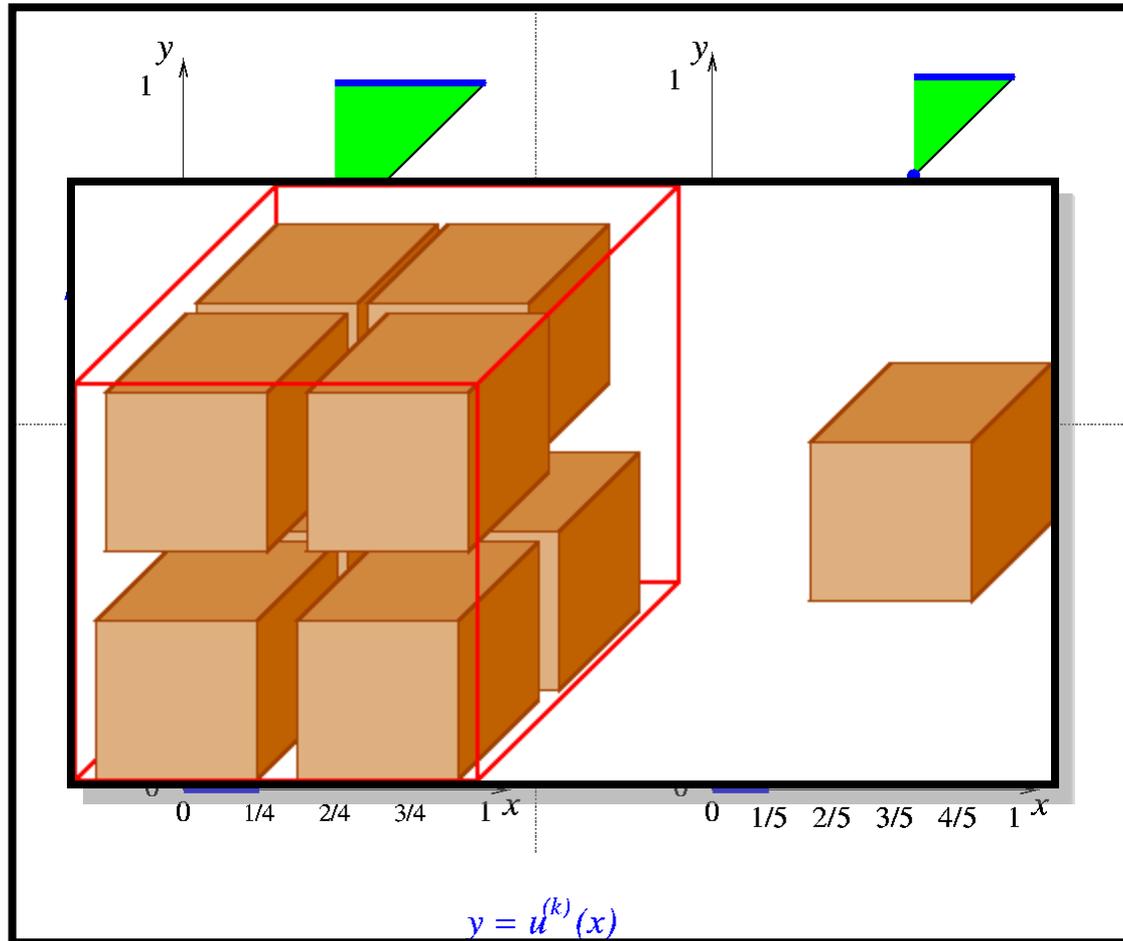


Gute und schnelle Schranken



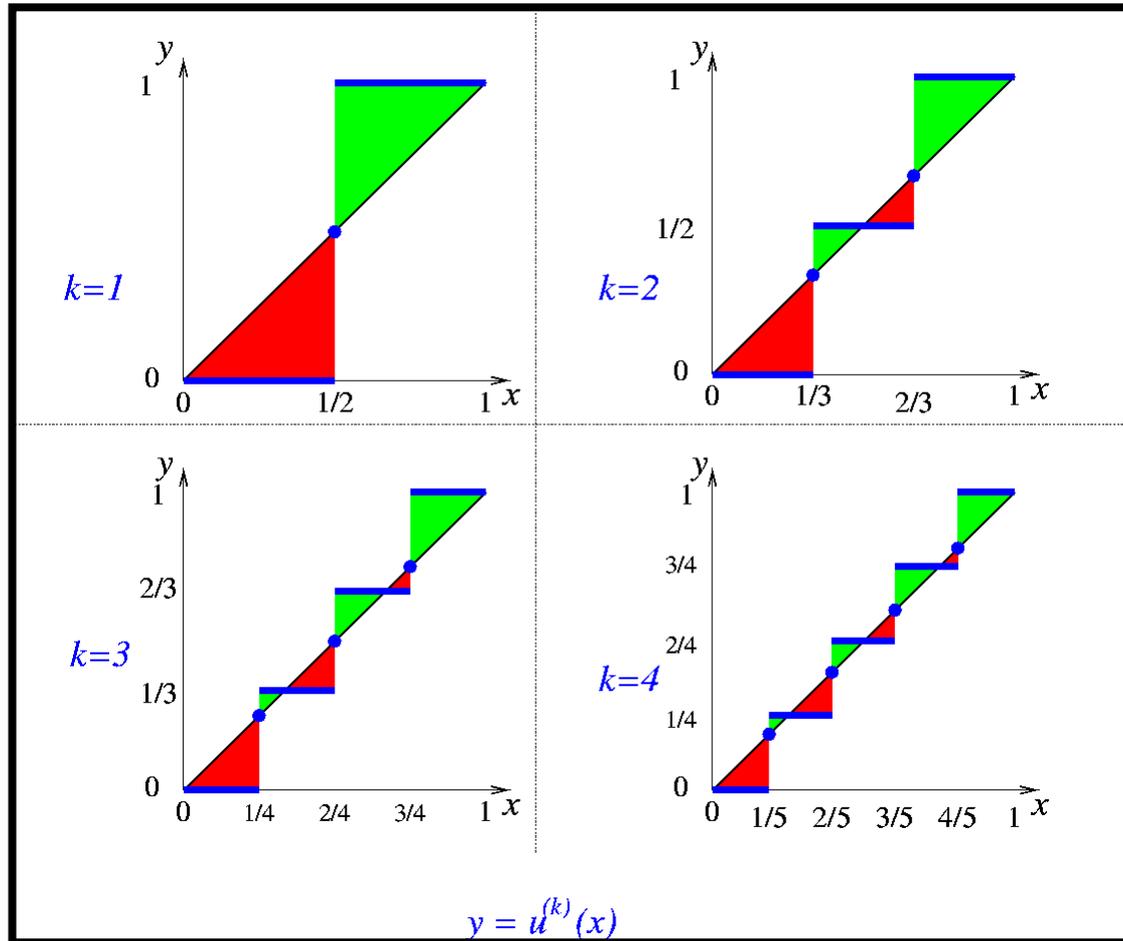
Konsequenz II: Framework für mehrdimensionale Schranken

Gute und schnelle Schranken



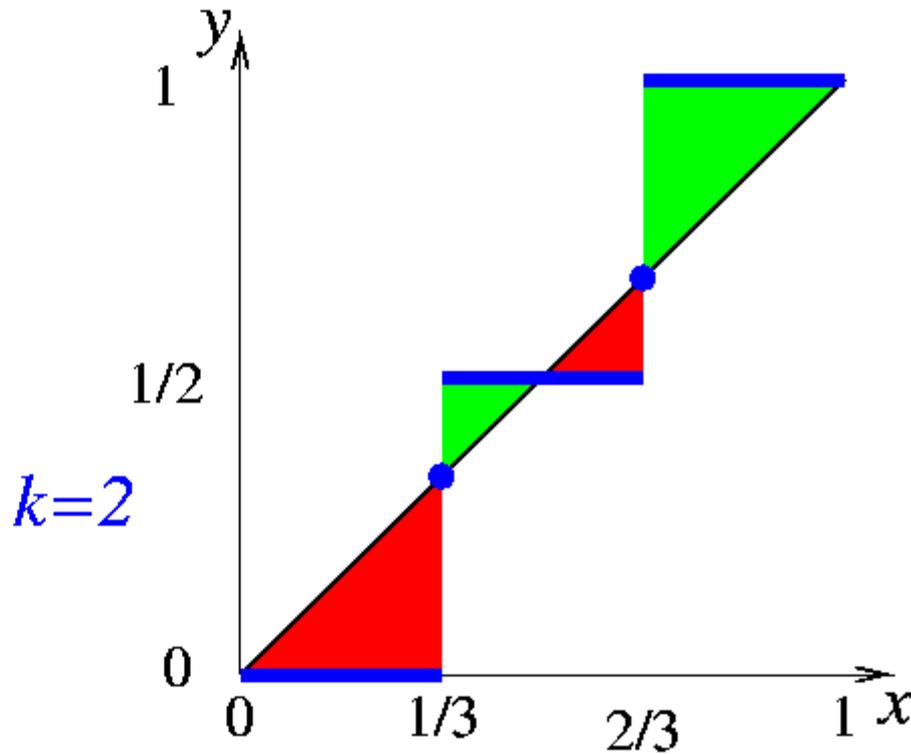
Konsequenz II: Framework für mehrdimensionale Schranken

Gute und schnelle Schranken



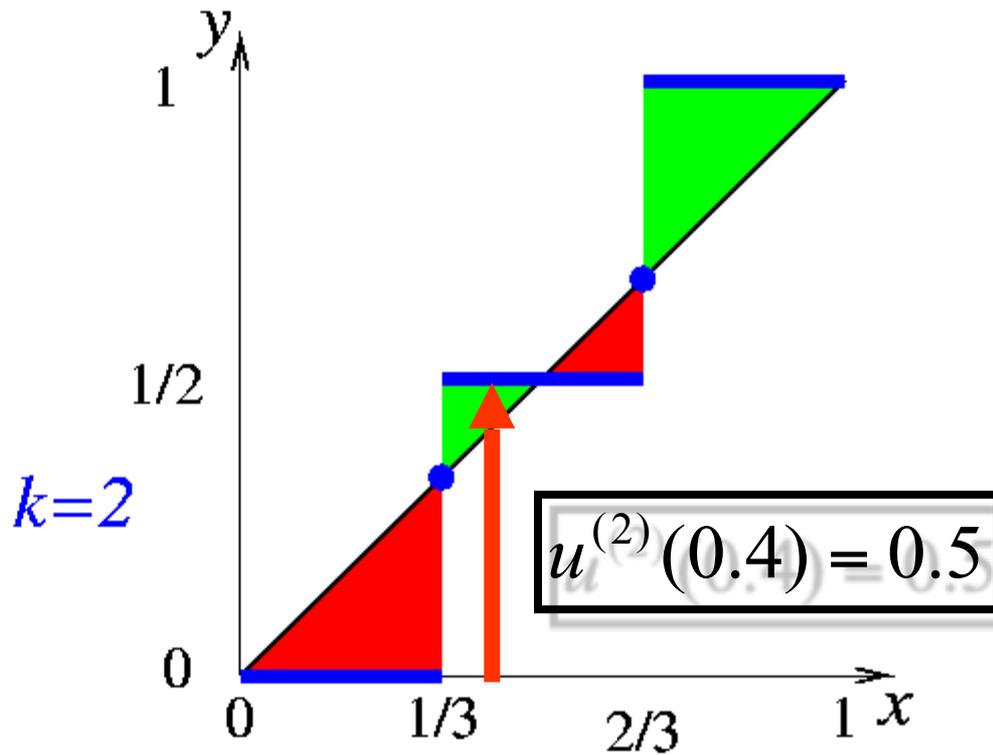
Konsequenz II: Framework für mehrdimensionale Schranken

Gute und schnelle Schranken



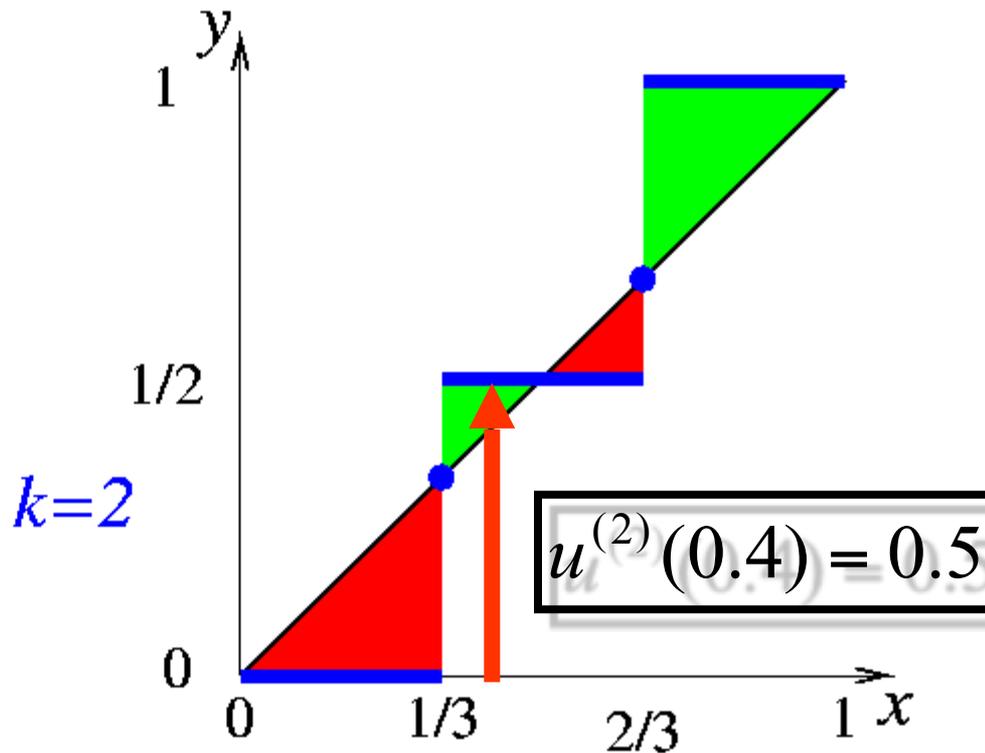
Konsequenz II: Framework für mehrdimensionale Schranken

Gute und schnelle Schranken



Konsequenz II: Framework für mehrdimensionale Schranken

Gute und schnelle Schranken



$$\text{Volumen: } 9 * (u^{(2)}(0.4))^3 = 9 * (0.5)^3 = 1.125$$

Konsequenz II: Framework für mehrdimensionale Schranken

Gute und schnelle Schranken

Table 7. Performance for instances without large or small items

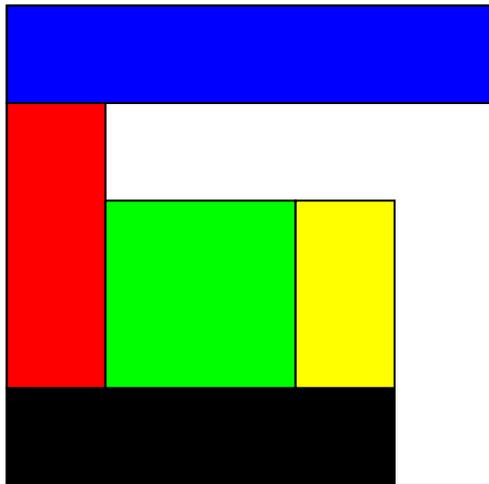
| $n = 100$ | Bound | L_1 | L_2 | $L_+^{(2)}$ | $L_+^{(3)}$ | $L_+^{(4)}$ | $L_+^{(5)}$ | $L_+^{(10)}$ | $L_+^{(20)}$ | $L_+^{(100)}$ |
|---------------|------------|-------|-------|-------------|-------------|-------------|-------------|--------------|--------------|---------------|
| Set | | | | | | | | | | |
| [20, ..., 80] | rel. error | 4.713 | 0.603 | 0.405 | 0.322 | 0.259 | 0.255 | 0.220 | 0.210 | 0.208 |
| | zero gap | 76 | 693 | 794 | 837 | 868 | 870 | 888 | 893 | 894 |
| | max gap | 13 | 2 | 1 | 1 | 1 | 1 | 1 | 1 | 1 |
| | total gap | 2598 | 311 | 206 | 163 | 132 | 130 | 112 | 107 | 106 |
| [20, ..., 70] | rel. error | 1.509 | 1.063 | 1.049 | 1.036 | 0.590 | 0.575 | 0.564 | 0.538 | 0.525 |
| | zero gap | 440 | 528 | 535 | 541 | 731 | 738 | 743 | 755 | 761 |
| | max gap | 7 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | total gap | 722 | 495 | 488 | 482 | 273 | 266 | 261 | 249 | 243 |
| [20, ..., 60] | rel. error | 1.340 | 1.340 | 1.340 | 1.324 | 1.324 | 1.324 | 1.324 | 1.324 | 1.324 |
| | zero gap | 483 | 483 | 483 | 484 | 484 | 484 | 484 | 484 | 484 |
| | max gap | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| | total gap | 556 | 556 | 556 | 549 | 549 | 549 | 549 | 549 | 549 |
| [20, ..., 50] | rel. error | 4.321 | 4.321 | 4.321 | 4.321 | 4.318 | 4.318 | 4.318 | 4.318 | 4.214 |
| | zero gap | 56 | 56 | 56 | 56 | 57 | 57 | 57 | 57 | 60 |
| | max gap | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| | total gap | 1615 | 1615 | 1615 | 1615 | 1614 | 1614 | 1614 | 1614 | 1575 |
| [20, ..., 40] | rel. error | 2.236 | 2.236 | 2.236 | 2.236 | 2.236 | 2.236 | 2.236 | 2.236 | 2.236 |
| | zero gap | 363 | 363 | 363 | 363 | 363 | 363 | 363 | 363 | 363 |
| | max gap | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| | total gap | 702 | 702 | 702 | 702 | 702 | 702 | 702 | 702 | 702 |
| [20, ..., 35] | rel. error | 6.020 | 6.020 | 6.020 | 6.020 | 6.020 | 6.020 | 6.020 | 6.020 | 6.016 |
| | zero gap | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| | max gap | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| | total gap | 1796 | 1796 | 1796 | 1796 | 1796 | 1796 | 1796 | 1796 | 1795 |
| [20, ..., 30] | rel. error | 2.597 | 2.597 | 2.597 | 2.597 | 2.597 | 2.597 | 2.597 | 2.597 | 2.594 |
| | zero gap | 603 | 603 | 603 | 603 | 603 | 603 | 603 | 603 | 603 |
| | max gap | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| | total gap | 727 | 727 | 727 | 727 | 727 | 727 | 727 | 727 | 726 |

Table 3. Zero gap (out of 1000 instances) for lower bounds for the BPP

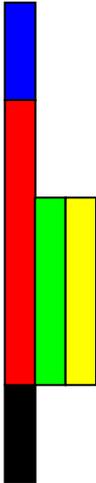
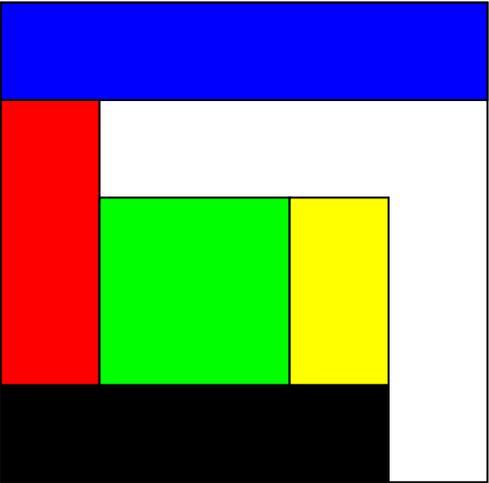
| | Bound | L_1 | L_2 | $L_*^{(2)}$ | $L_*^{(3)}$ | $L_*^{(4)}$ | $L_*^{(5)}$ | $L_*^{(10)}$ | $L_*^{(20)}$ | $L_*^{(100)}$ |
|---------------|-------|-------|-------|-------------|-------------|-------------|-------------|--------------|--------------|---------------|
| Set | n | | | | | | | | | |
| {1, ..., 100} | 32 | 286 | 908 | 918 | 928 | 940 | 942 | 953 | 956 | 959 |
| | 100 | 122 | 775 | 833 | 854 | 864 | 872 | 888 | 895 | 897 |
| | 316 | 29 | 574 | 706 | 747 | 764 | 774 | 811 | 826 | 828 |
| | 1000 | 2 | 382 | 579 | 613 | 632 | 638 | 667 | 678 | 681 |
| {1, ..., 90} | 32 | 347 | 924 | 937 | 945 | 949 | 952 | 963 | 965 | 968 |
| | 100 | 231 | 770 | 843 | 869 | 881 | 881 | 896 | 901 | 904 |
| | 316 | 275 | 679 | 796 | 826 | 939 | 841 | 855 | 864 | 865 |
| | 1000 | 410 | 703 | 854 | 872 | 878 | 878 | 888 | 888 | 888 |
| {1, ..., 80} | 32 | 596 | 938 | 952 | 959 | 963 | 963 | 967 | 967 | 969 |
| | 100 | 672 | 922 | 950 | 955 | 957 | 957 | 957 | 957 | 957 |
| | 316 | 862 | 951 | 971 | 971 | 971 | 971 | 971 | 971 | 971 |
| | 1000 | 982 | 989 | 989 | 989 | 989 | 989 | 989 | 989 | 989 |
| {20, ..., 80} | 32 | 187 | 836 | 866 | 882 | 902 | 906 | 921 | 928 | 931 |
| | 100 | 76 | 693 | 794 | 837 | 868 | 870 | 888 | 893 | 894 |
| | 316 | 21 | 474 | 676 | 720 | 755 | 760 | 795 | 808 | 814 |
| | 1000 | 1 | 354 | 607 | 646 | 666 | 666 | 687 | 698 | 699 |
| {20, ..., 70} | 32 | 541 | 764 | 777 | 793 | 867 | 877 | 887 | 893 | 902 |
| | 100 | 440 | 528 | 535 | 541 | 731 | 738 | 743 | 755 | 761 |
| | 316 | 9 | 13 | 13 | 13 | 104 | 105 | 105 | 105 | 105 |
| | 1000 | 0 | 0 | 0 | 0 | 6 | 6 | 6 | 6 | 6 |

Baumsuche: Packmusterklassen

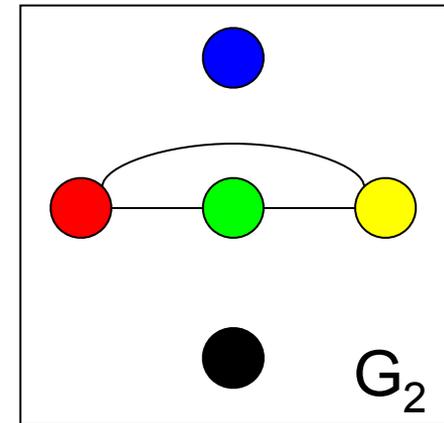
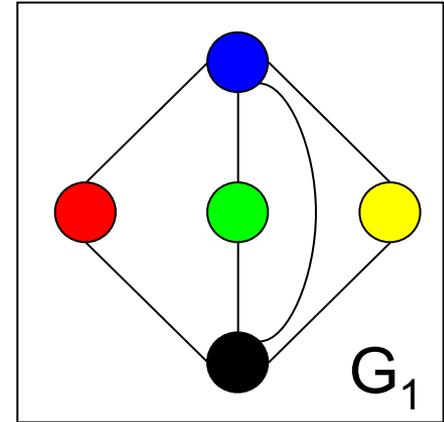
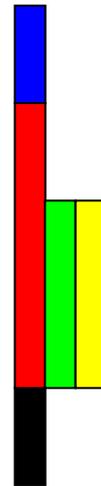
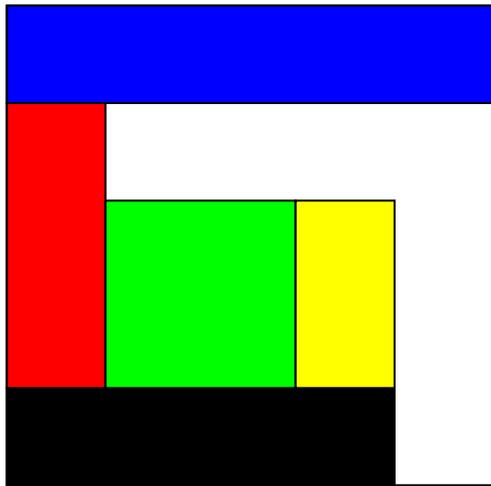
Baumsuche: Packmusterklassen



Baumsuche: Packmusterklassen



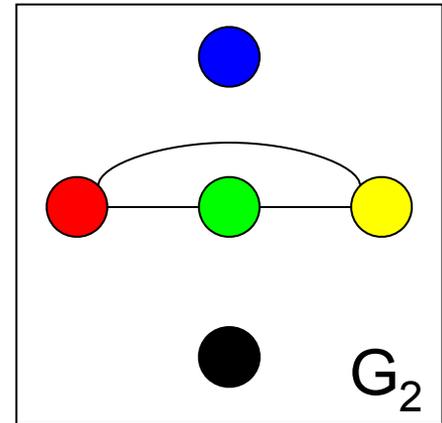
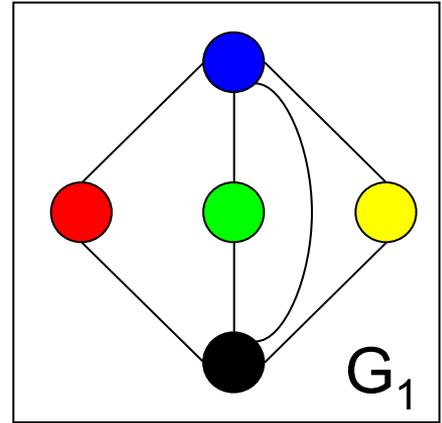
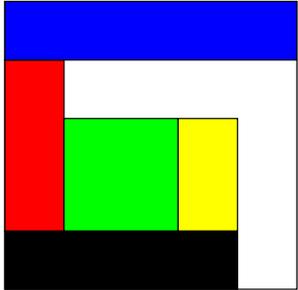
Baumsuche: Packmusterklassen



Baumsuche: Packmusterklassen

1.

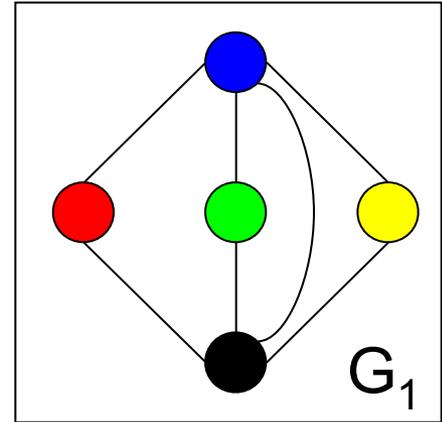
G_1 und G_2 sind Intervallgraphen



Baumsuche: Packmusterklassen

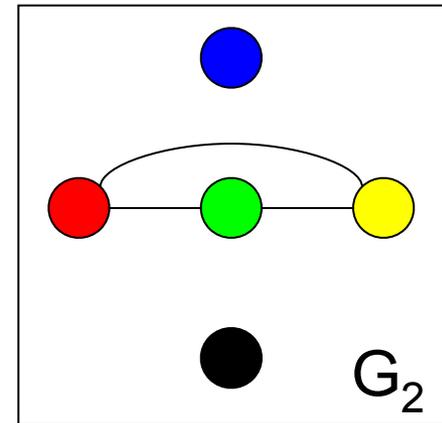
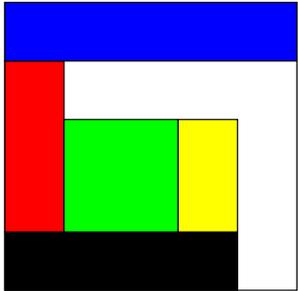
1.

G_1 und G_2 sind Intervallgraphen

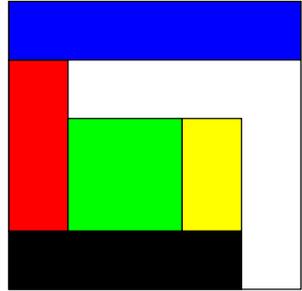


2.

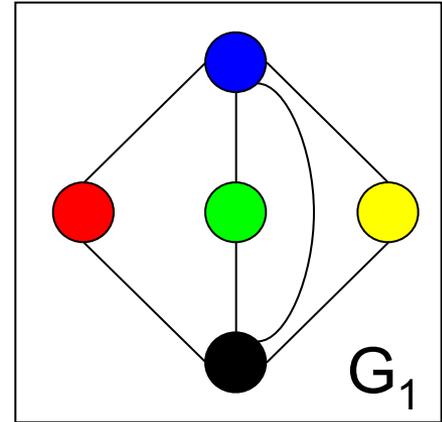
Unabhängige Mengen von Knoten in G_1 und G_2 sind i-zulässig



Baumsuche: Packmusterklassen

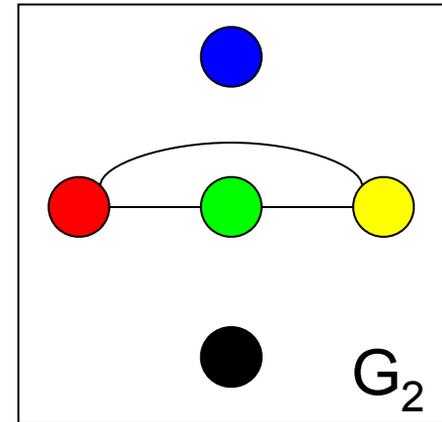


1. G_1 und G_2 sind Intervallgraphen



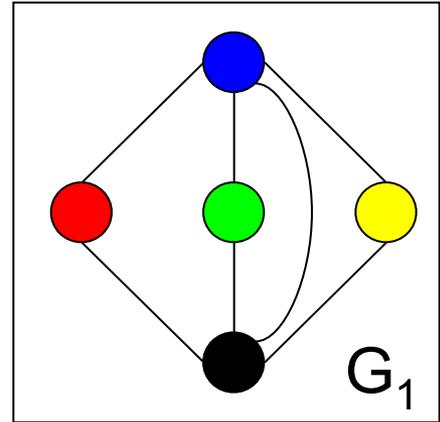
2. Unabhängige Mengen von Knoten in G_1 und G_2 sind i-zulässig

3. G_1 und G_2 haben keine Kanten gemeinsam

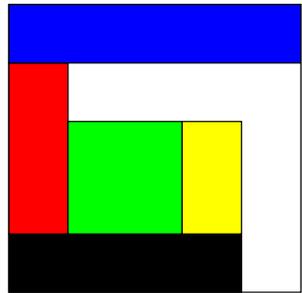


Baumsuche: Packmusterklassen

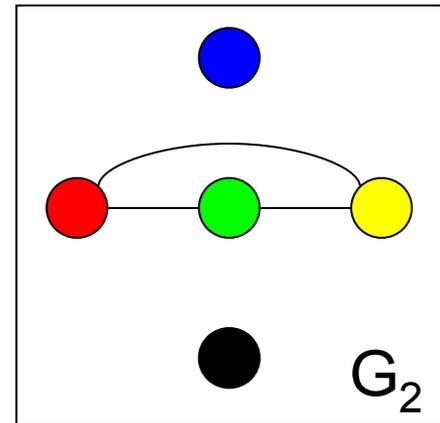
1. G_1 und G_2 sind Intervallgraphen



2. Unabhängige Mengen von Knoten in G_1 und G_2 sind i-zulässig



3. G_1 und G_2 haben keine Kanten gemeinsam



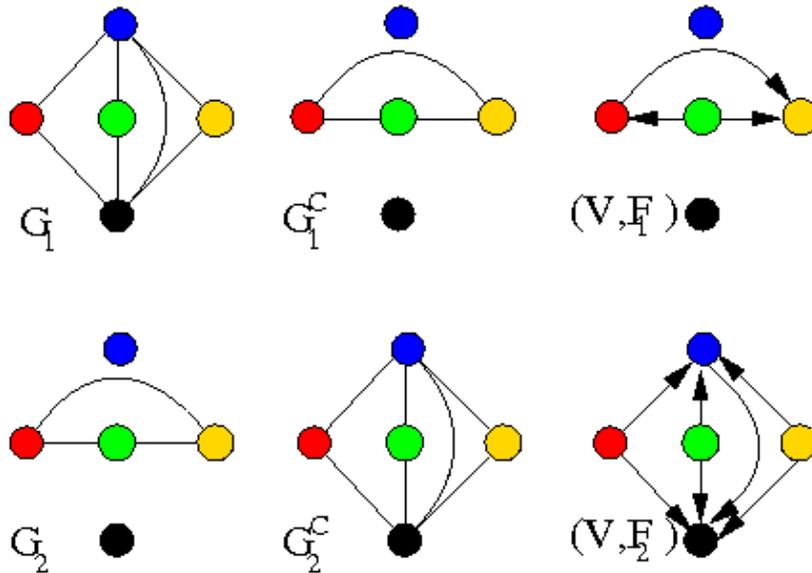
Satz (Fekete & Schepers): Eine Menge von Boxen lässt genau dann eine Packung zu, wenn es eine Packmusterklasse gibt.

Graphen statt Quader

Packmusterklassen (3)

Beweisskizze:

Wegen P1 sind die Komplementärgraphen transitiv orientierbar:



Diese Orientierung repräsentiert die Partialordnung der Objekte in der entsprechenden Koordinatenrichtung.

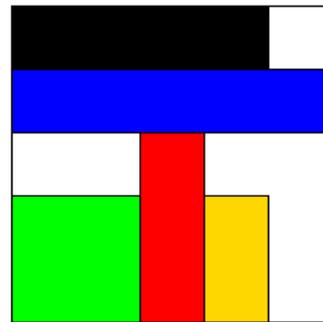
Graphen statt Quader

Packmusterklassen (4)

(Forts. Beweis)



Induktiv in aufsteigender Koordinatenordnung läßt sich dann eine Packung aufbauen:



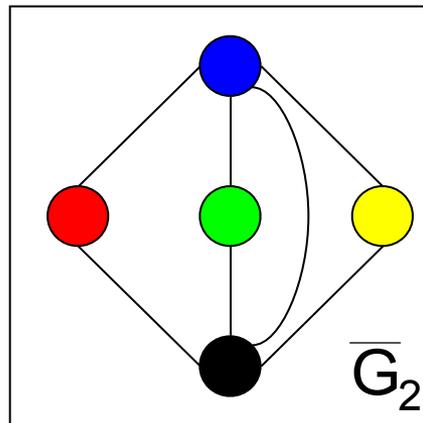
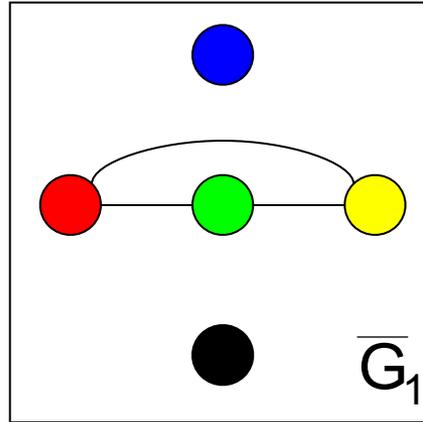
Wegen P3 gibt es keine Überlappungen.

Wegen P2 ragen keine Objekte aus dem Container.

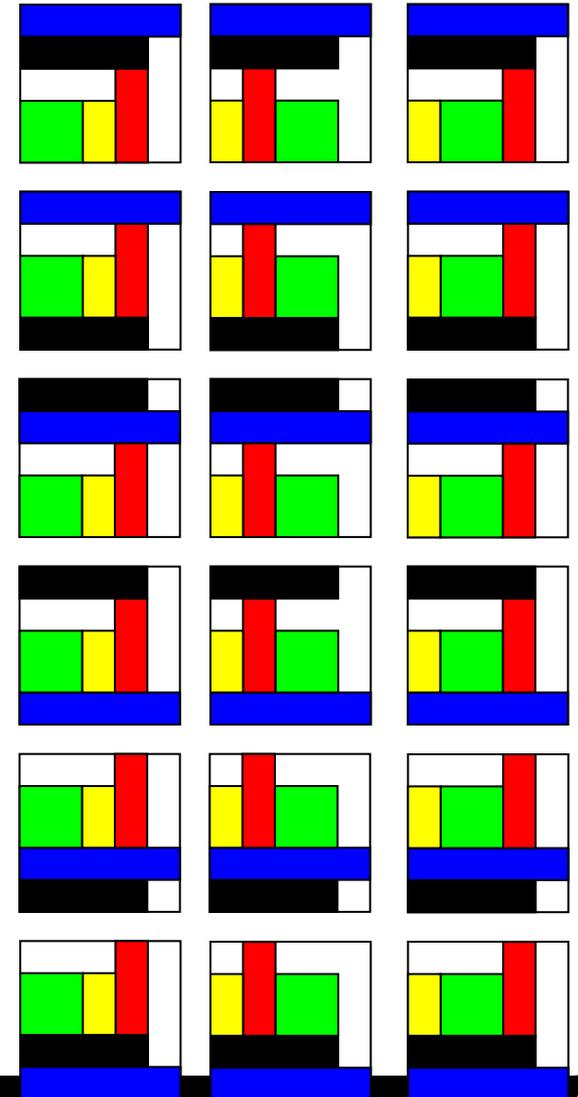
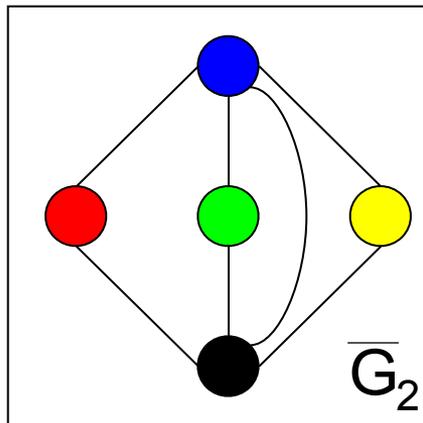
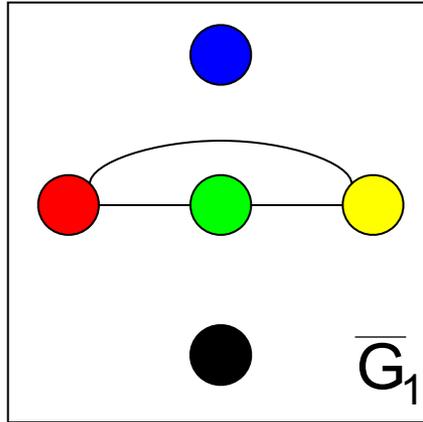
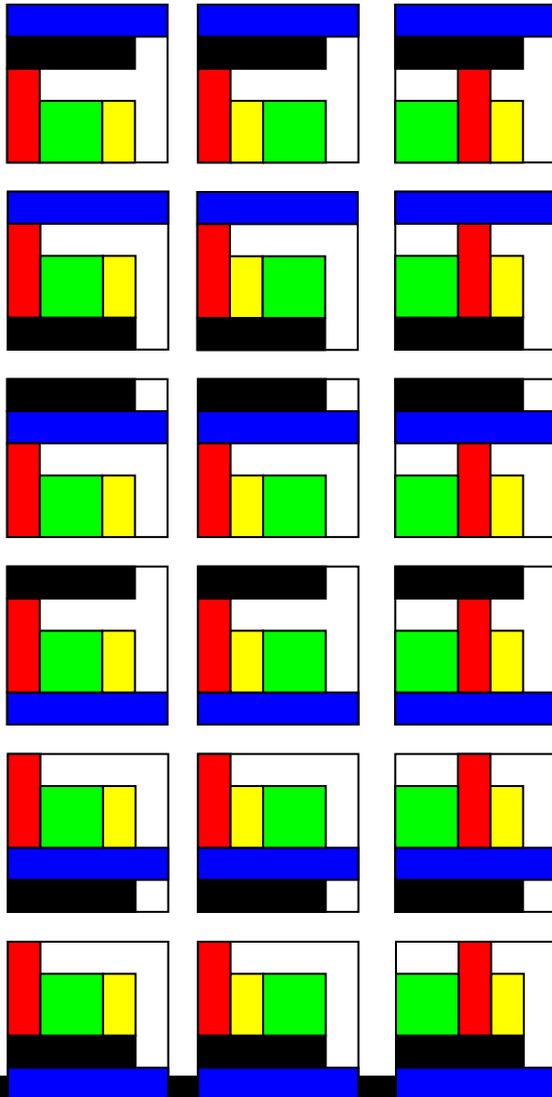
(Man beweist in der Induktion, daß es zu jedem Zeitpunkt der Konstruktion für jede Koordinate eine unabhängige Menge gibt, die die gesamte Ausdehnung der Packung in der Koordinate aufspannt.)



Packen und Packmusterklassen



Packen und Packmusterklassen



Ergebnisse

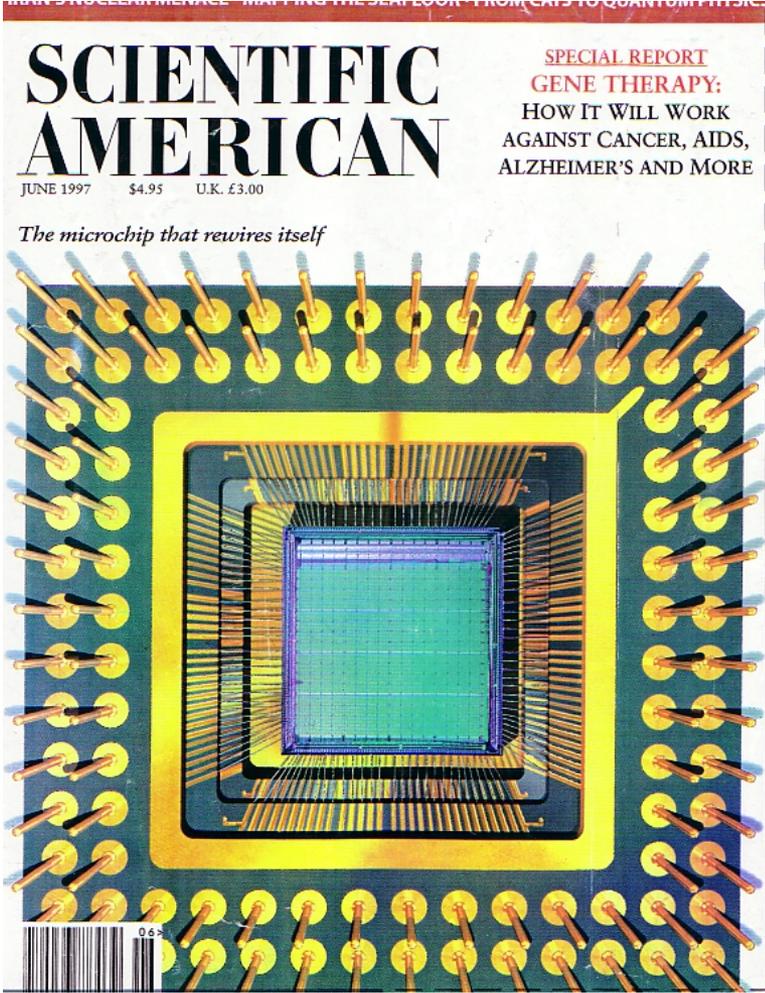
| Problem | Time/s | B85 | HC95 | CM04 | | | |
|-----------|---------|-------|-------|----------------|----------------|----------------|----------------|
| | | | | A ₀ | A ₁ | A ₂ | A ₃ |
| beasley1 | <0.01 | 0.9 | | | | | |
| beasley2 | <0.01 | 4.0 | | | | | |
| beasley3 | <0.01 | 10.5 | | | | | |
| beasley4 | <0.01 | 0.1 | 0.04 | | | | |
| beasley5 | <0.01 | 0.4 | | | | | |
| beasley6 | <0.01 | 55.2 | 45.20 | | | | |
| beasley7 | <0.01 | 0.5 | 0.04 | | | | |
| beasley8 | 0.02 | 218.6 | | | | | |
| beasley9 | <0.01 | 18.3 | 5.20 | | | | |
| beasley10 | <0.01 | 0.9 | | | | | |
| beasley11 | <0.01 | 79.1 | | | | | |
| beasley12 | 0.02 | 229.0 | > 800 | | | | |
| hadchr3 | <0.01 | | 532 | | | | |
| hadchr7 | 0.01 | | > 800 | | | | |
| hadchr8 | <0.01 | | > 800 | | | | |
| hadchr11 | <0.01 | | > 800 | | | | |
| hadchr12 | <0.01 | | 65.2 | | | | |
| wang20 | 0.67 | | | 6.75 | 6.31 | 17.84 | 2.72 |
| chrwhi62 | 0.54 | | | | | | |
| 3 | 0.54 | | | | | | |
| 3s | 0.46 | | | | | | |
| A1 | 1.12 | | | | | | |
| A1s | 1.51 | | | | | | |
| A2 | 1.62 | | | | | | |
| A2s | 8.35 | | | | | | |
| CHL2 | 10.36 | | | | | | |
| CHL2s | 6.84 | | | | | | |
| CHL3 | <0.01 | | | | | | |
| CHL3s | <0.01 | | | | | | |
| CHL4 | <0.01 | | | | | | |
| CHL4s | <0.01 | | | | | | |
| CHL5 | 4.66 | | | | | | |
| cgcut1 | <0.01 | | | 0.30 | 1.47 | 1.46 | 1.46 |
| cgcut2 | > 1,800 | | | > 1,800 | > 1,800 | 533.45 | 531.93 |
| cgcut3 | 0.54 | | | 23.76 | 23.68 | 4.59 | 4.58 |
| gcut1 | 0.01 | | | 0.00 | 0.00 | 0.01 | 0.01 |
| gcut2 | 0.47 | | | 0.52 | 0.19 | 25.75 | 0.22 |
| gcut3 | 4.34 | | | > 1,800 | 2.16 | 276.37 | 3.24 |
| gcut4 | 195.62 | | | > 1,800 | 346.99 | > 1,800 | 376.52 |
| gcut5 | 0.02 | | | 0.00 | 0.50 | 0.03 | 0.50 |
| gcut6 | 0.38 | | | 0.06 | 0.09 | 9.71 | 0.12 |
| gcut7 | 2.24 | | | 1.31 | 0.63 | 354.50 | 1.07 |
| gcut8 | 253.54 | | | 1,202.09 | 136.71 | > 1,800 | 168.50 |
| gcut9 | 0.01 | | | 0.01 | 0.09 | 0.05 | 0.08 |
| gcut10 | 0.67 | | | 0.01 | 0.13 | 6.49 | 0.14 |
| gcut11 | 8.82 | | | 16.72 | 14.76 | > 1,800 | 16.30 |
| gcut12 | 109.81 | | | 63.45 | 16.85 | > 1,800 | 25.39 |
| gcut13 | > 1,800 | | | > 1,800 | > 1,800 | > 1,800 | > 1,800 |
| okp1 | 10.82 | | | 24.06 | 25.46 | 72.20 | 35.84 |
| okp2 | 20.25 | | | > 1,800 | > 1,800 | 1,535.95 | 1,559.00 |
| okp3 | 5.98 | | | 21.36 | 1.91 | 465.57 | 10.63 |
| okp4 | 2.87 | | | 40.40 | 2.13 | 0.85 | 4.05 |
| okp5 | 11.78 | | | 40.14 | > 1,800 | 513.06 | 488.27 |

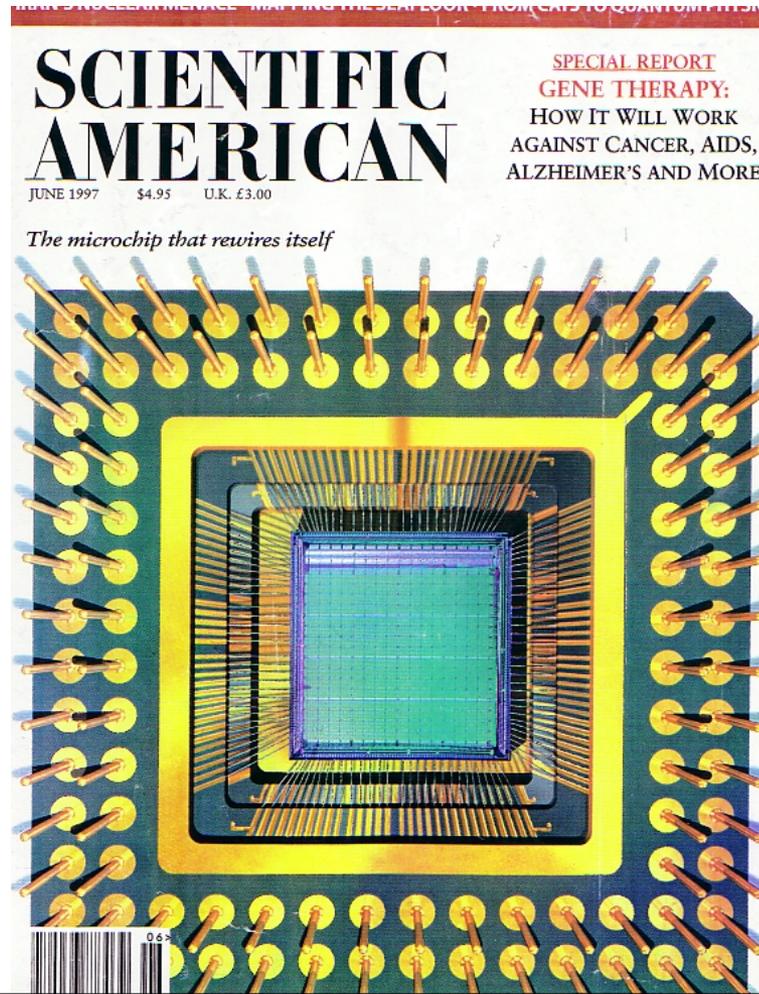
Packen in der Elektrotechnik

Leistungsfähige Methode für
mehrdimensionales orthogonales Packen

Packen in der Elektrotechnik

Packen in der Elektrotechnik

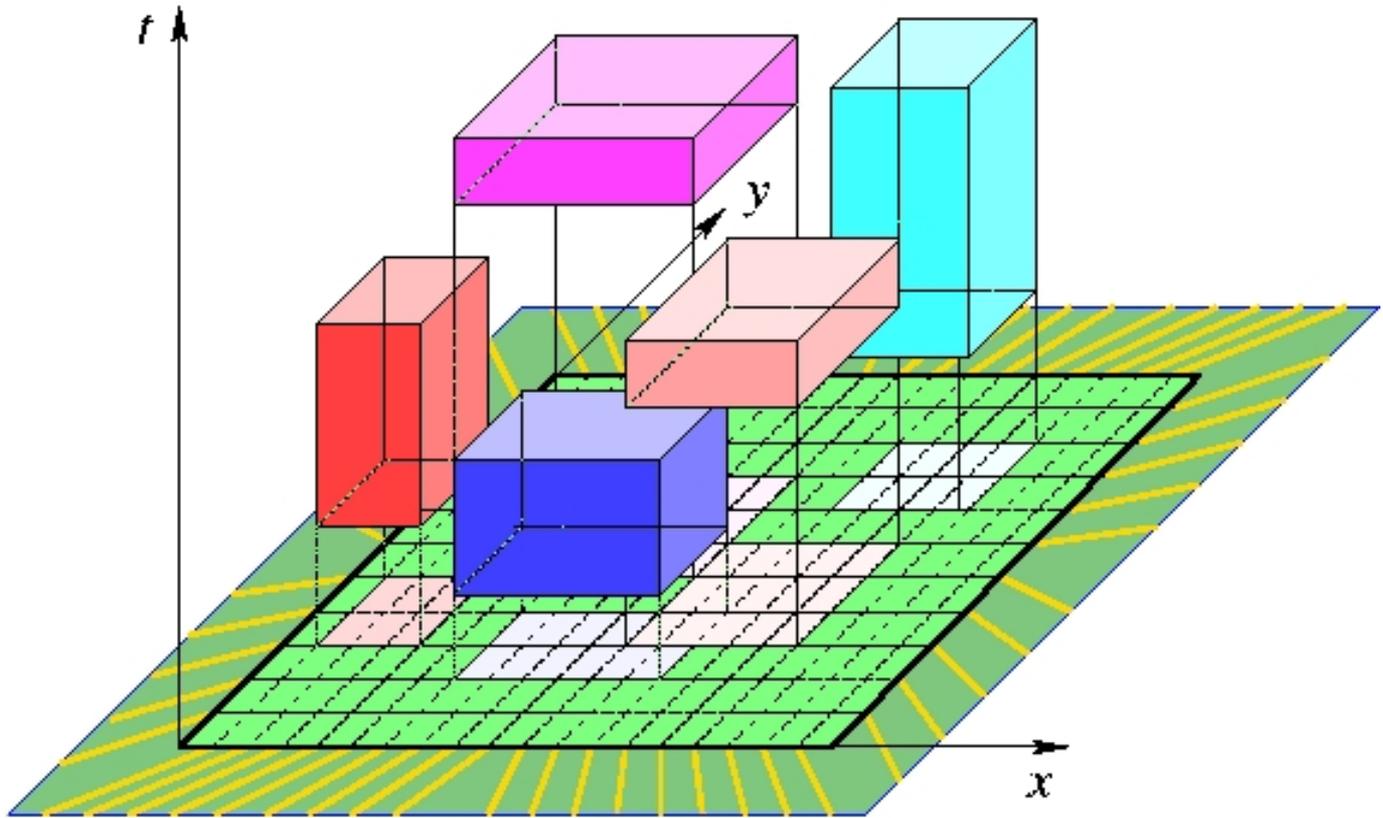




Anwendung: Reconfigurable Computing

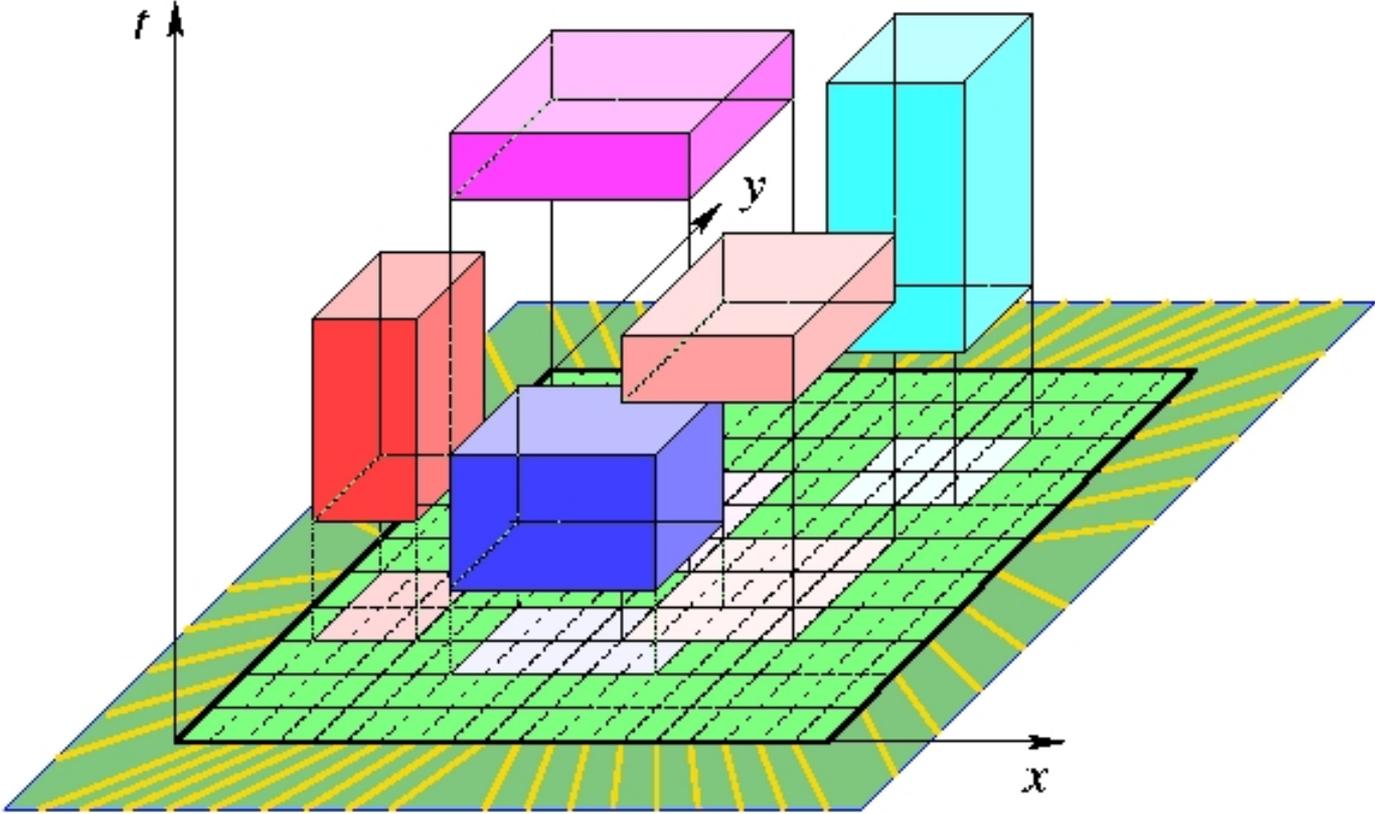
Anwendung: Reconfigurable Computing

Packen in der Elektrotechnik

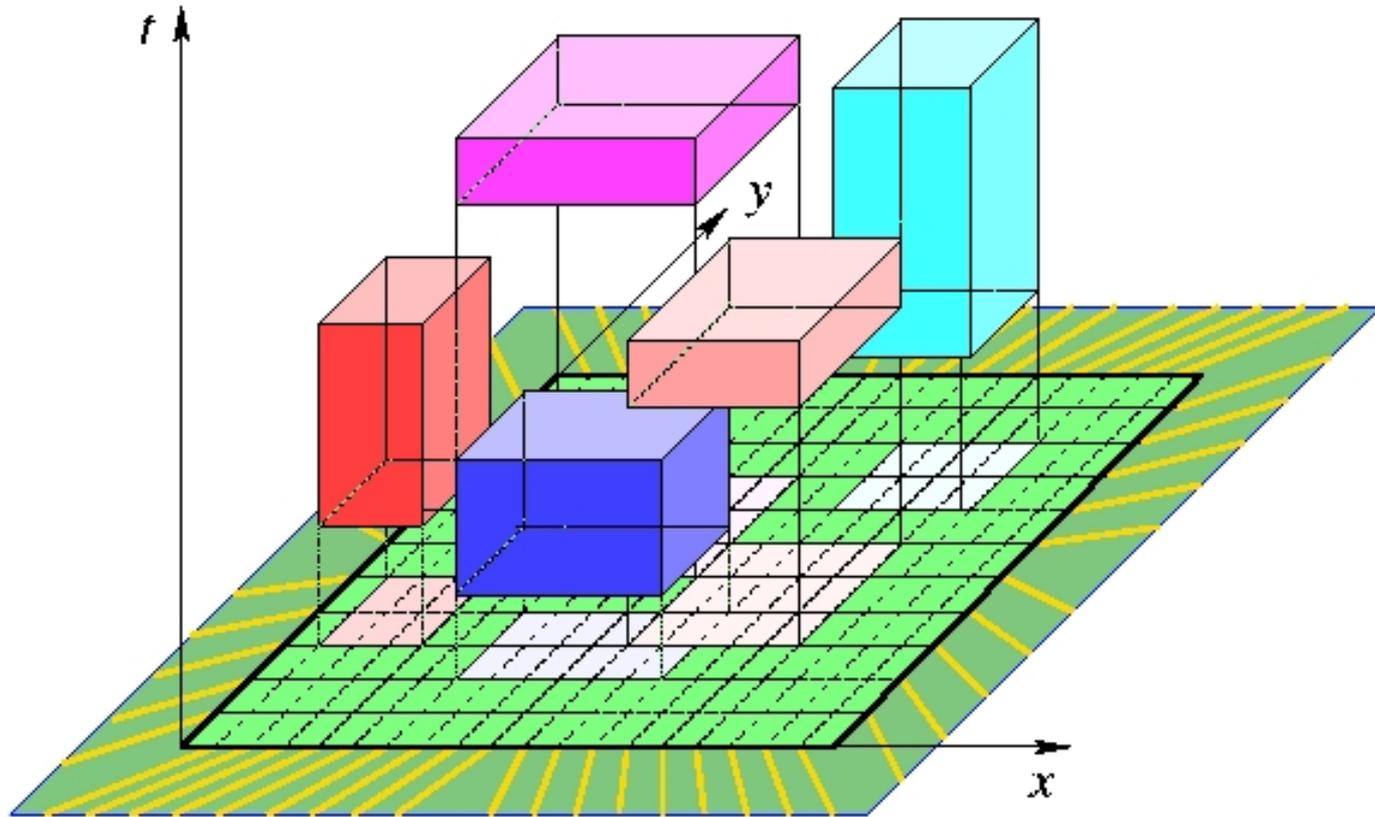


Anwendung: Reconfigurable Computing

Packen in der Elektrotechnik

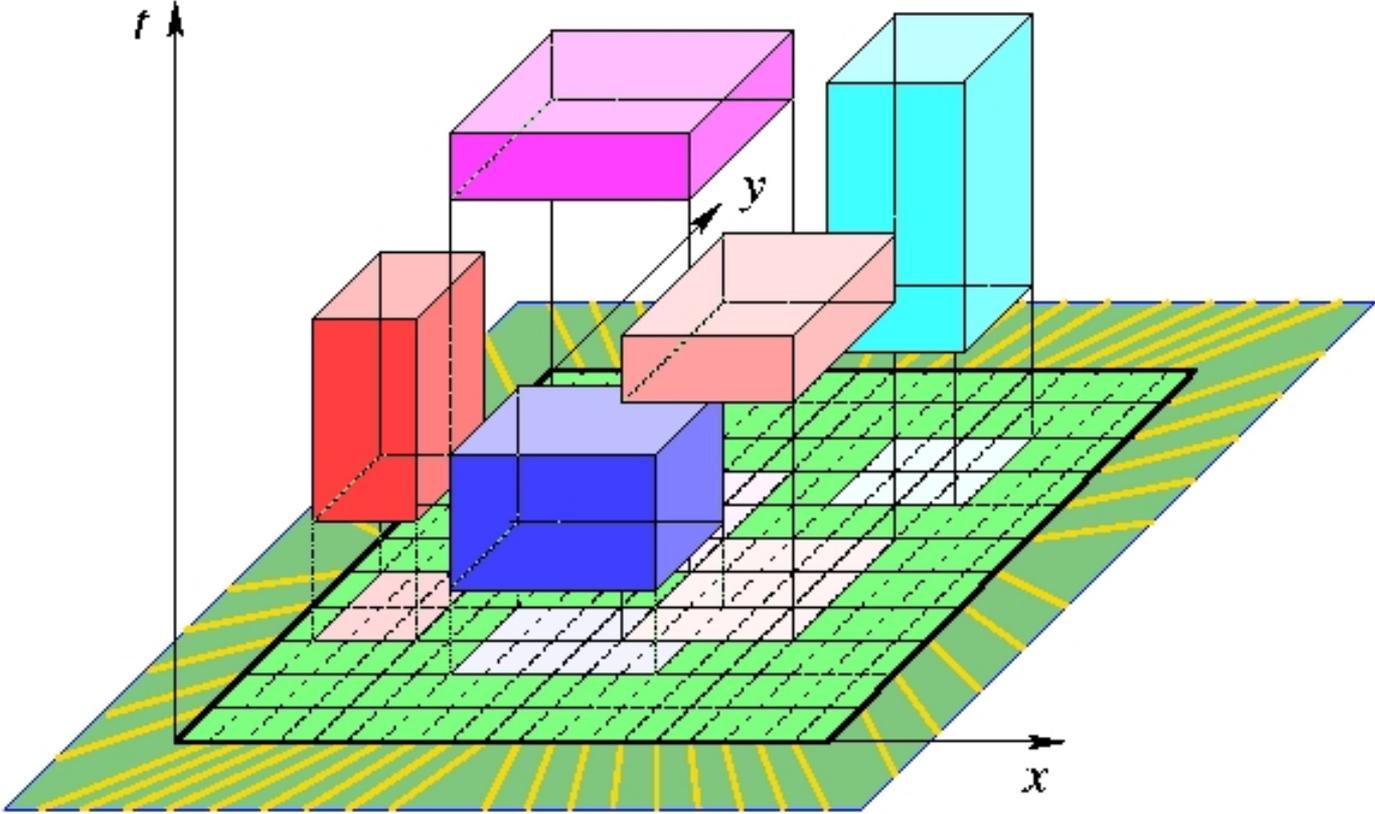


Packen in der Elektrotechnik

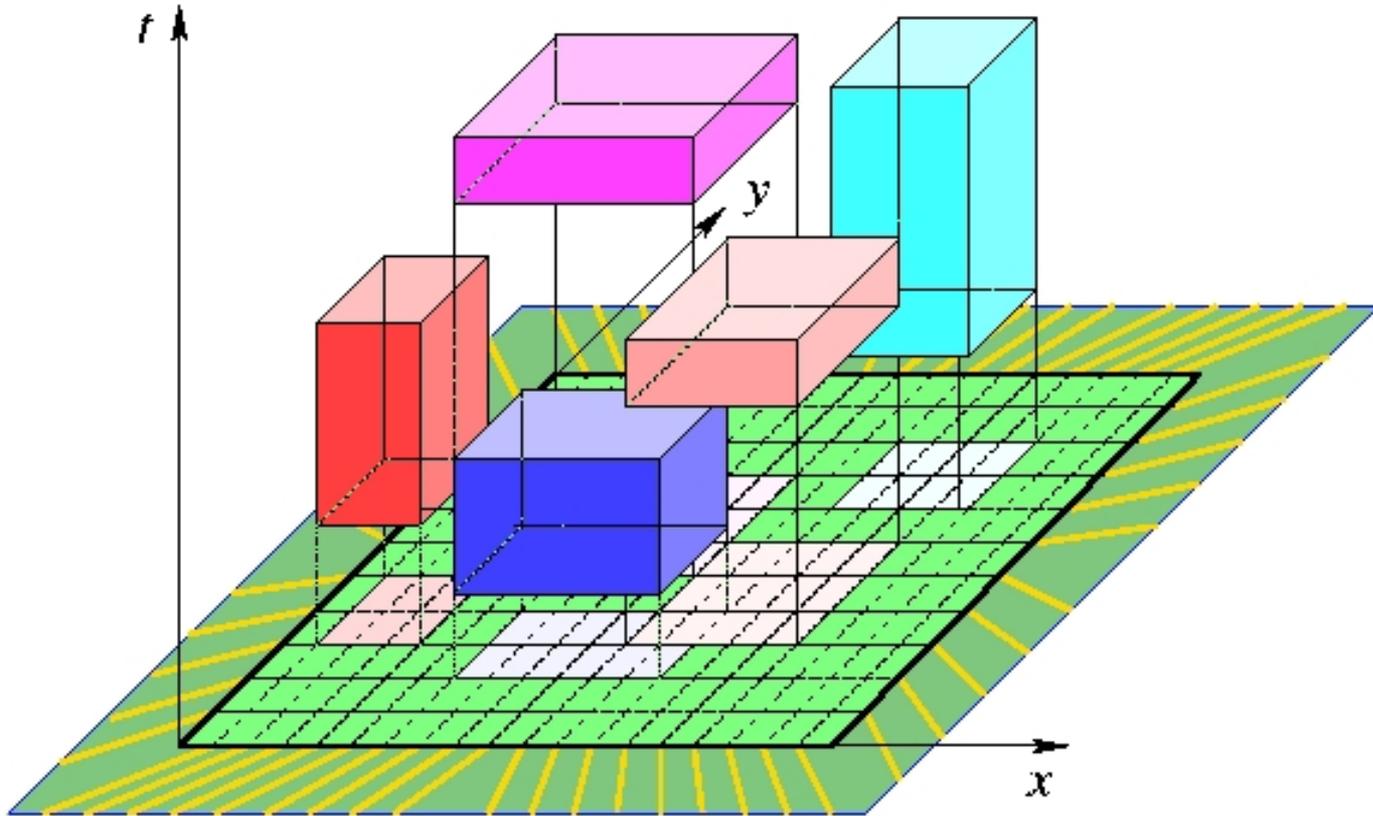


DFG-Projekt "ReCoNodes" 2003-2009,
mit Jürgen Teich (Software-Hardware-CoDesign,
Erlangen)

Packen in der Elektrotechnik

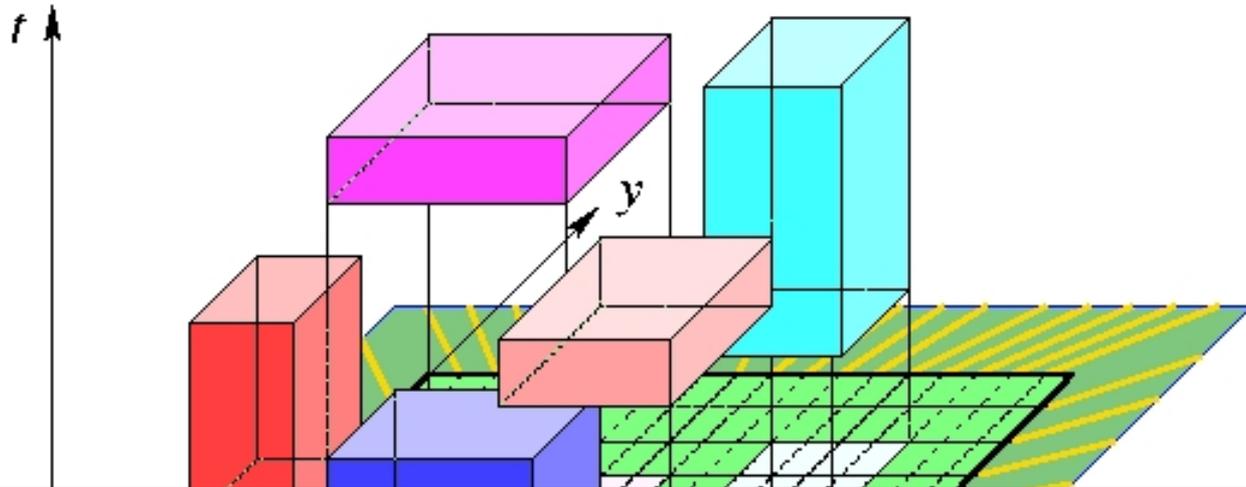


Packen in der Elektrotechnik



Erweiterung: Packen mit Anordnungsrestriktionen
(Fekete, Köhler, Teich, *SIAM J. Disc. Math.* 2006)

Packen in der Elektrotechnik



[Optimal FPGA module placement with temporal precedence constraints](#)

[S Fekete, E Kohler...](#) - Design, Automation and Test in ..., 2002 - ieeexplore.ieee.org

Optimal FPGA Module Placement with Temporal Precedence Constraints ... Siindor P.

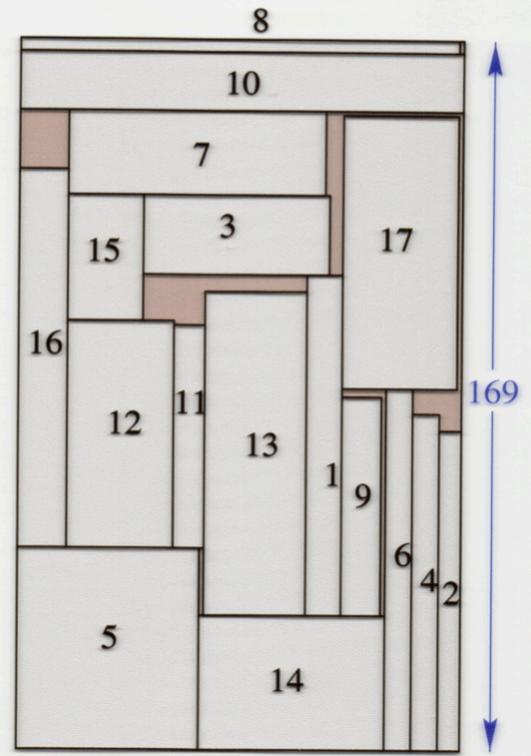
Fekete* Ekkehard Kohler+ Jiirgen Teich4 ... We consider the optimal placement of hardware modules in space and time for FPGA architectures ,with re-configuration capabilities, ...

[Cited by 97](#) - [Related articles](#) - [All 25 versions](#) - [Import into BibTeX](#)

Erweiterung: Packen mit Anordnungsrestriktionen

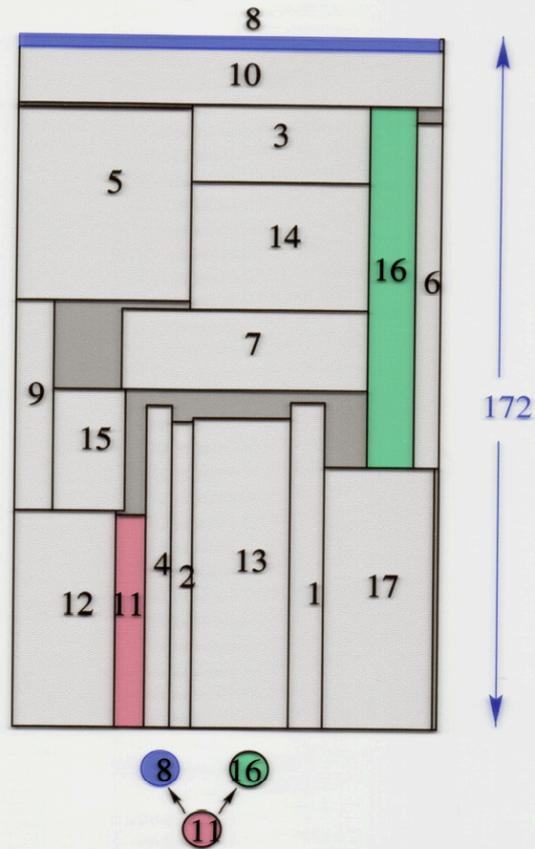
(Fekete, Köhler, Teich, *SIAM J. Disc. Math.* 2006)

Optimale Packungen für Benchmarks



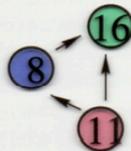
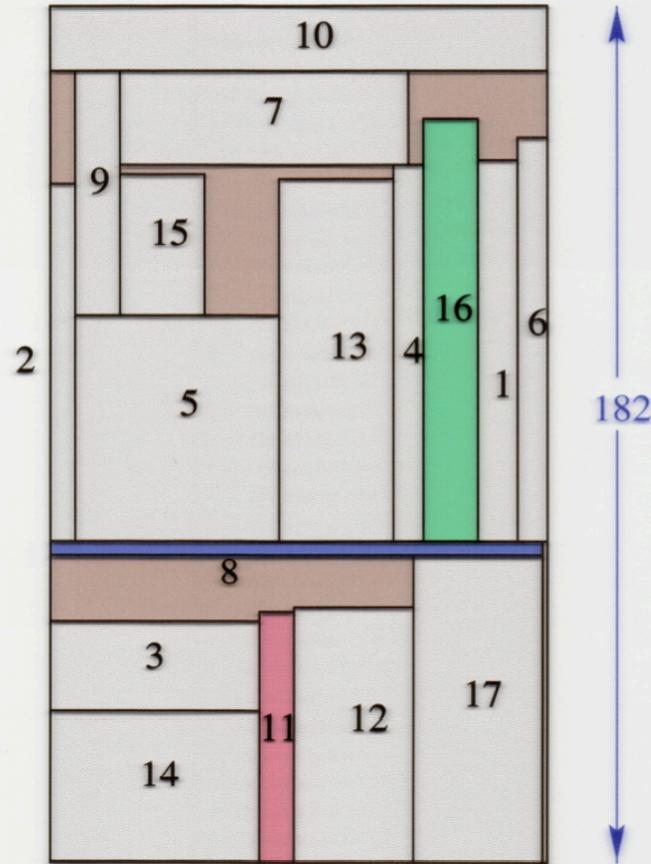
Rechenzeit: 7,29 s für Lösung,
179 s für untere Schranke

Optimale Packungen für Benchmarks (II)



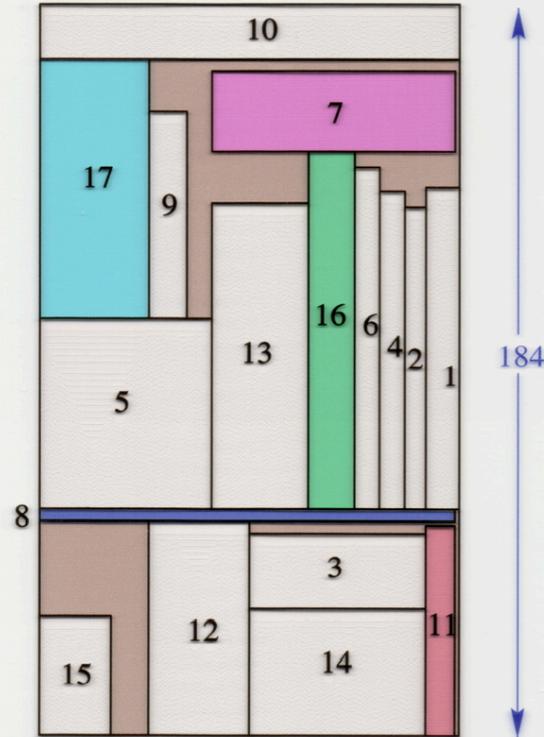
Rechenzeit: 6,73 s für Lösung,
1102 s für untere Schranke

Optimale Packungen für Benchmarks (III)



Rechenzeit: 5,39 s für Lösung,
330 s für untere Schranke

Optimale Packungen für Benchmarks (IV)



Rechenzeit: 236 s für Lösung,
553 s für untere Schranke

Origami

Falter unter sich

Falter unter sich



Falter unter sich

Erik Demaine



Falter unter sich

Erik Demaine

Robert Lang



Falter unter sich

Origami

ROBERT J. LANG ORIGAMI



art

science

artist

publications

info



ROBERT J. LANG ORIGAMI

art

compositions

Insects & arthropods

birds

mammals

plants & flowers

sea life & mollusks

human figures

reptiles & amphibians

dinosaurs & mythical

objects

geometrics & tessellations

currency

bronzes

crease patterns

commercial

exhibitions

lectures

monumental

challenge

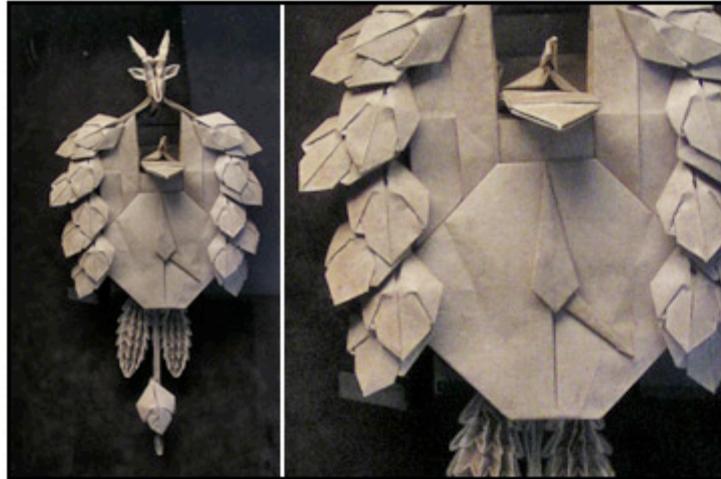
science

artist

publications

info

Objects



Black Forest Cuckoo Clock, opus 182

Medium: One uncut 1x10 rectangle of Zanders 'elefantenhaut' paper

Composed: 1987

Folded: 1987

Size: 15"

Diagrams: *Origami Design Secrets*

Comments: This was for a time my most well-known composition and the most complex origami figures around. My first cuckoo clock was fairly plain; my second (unpublished) one was composed after a visit to the Black Forest. For the third design, I came up with this figure, composed and folded in Ludwigsburg, Germany.



ROBERT J. LANG ORIGAMI

art

compositions

insects & arthropods

birds

mammals

plants & flowers

sea life & mollusks

human figures

reptiles & amphibians

dinosaurs & mythical

objects

geometrics & tessellations

currency

bronzes

crease patterns

commercial

exhibitions

lectures

monumental

challenge

science

artist

publications

info

Dinosaurs & Mythological



New
Entry



Redpath Pteranodon, opus 506

Medium: One uncut square of St.-Armand paper

Composed: 2007

Folded: 2007

Size: 15"

Diagrams: *License to Fold*

Comments: This model was commissioned by the Redpath Museum of Natural History in Montreal, Quebec, for folding in life size (3 meter wingspan). The full size version is currently installed at the Museum.



ROBERT J. LANG ORIGAMI

art

compositions

Insects & arthropods

birds

mammals

plants & flowers

sea life & mollusks

human figures

reptiles & amphibians

dinosaurs & mythical

objects

geometrics & tessellations

currency

bronzes

crease patterns

commercial

exhibitions

lectures

monumental

challenge

science

artist

publications

info

Insects & Arthropods



Mt Diablo Tarantula, opus 481

Medium: One uncut square of Korean hanji

Composed: 2006

Folded: 2006

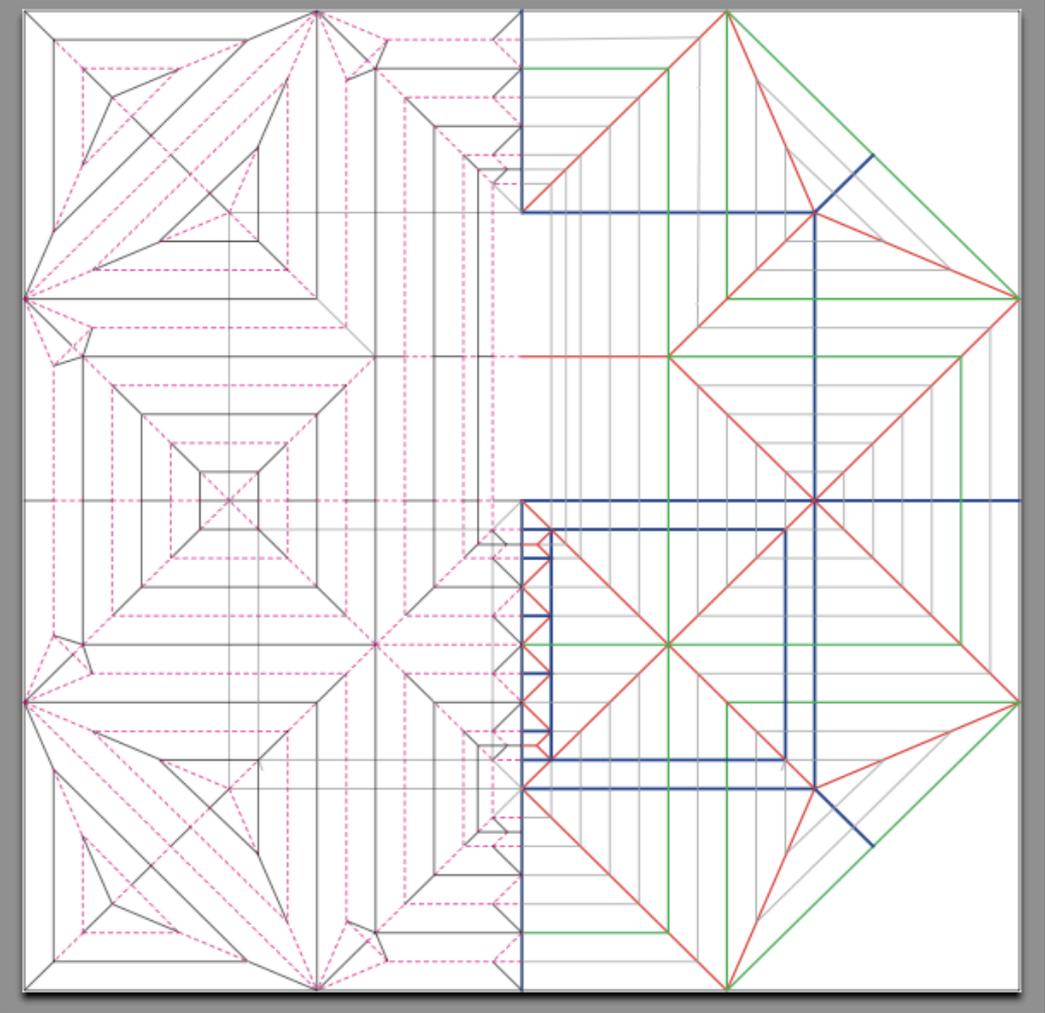
Size: 3.5"

Diagrams: Not available

Crease Pattern: [here](#)



Origami



Origami

username ***** [Register](#) [Forgot Pass](#) [ACCOUNT](#) | [EVENTS](#) | [RESOURCES](#) | [SUBSCRIBE](#)

English | en Español | auf Deutsch | in Italiano | 中文 | in India

technology review
Published by MIT

The Future of Energy Webcast at EmTech 2010
Watch On Demand Now.

technology review

BLOGS

[HOME](#) | [COMPUTING](#) | [WEB](#) | [COMMUNICATIONS](#) | [ENERGY](#) | [MATERIALS](#) | [BIOMEDICINE](#) | [BUSINESS](#) | **BLOGS** | [VIDEO](#)

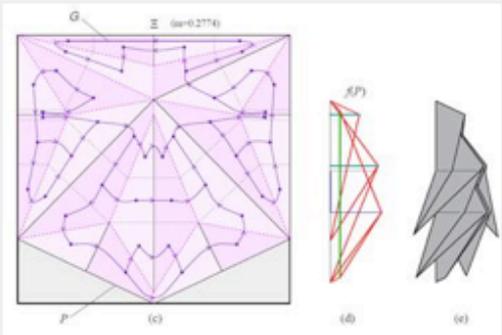
[MAGAZINE](#) | [NEWSLETTERS](#)

 **The Physics arXiv Blog**

Origami Crease Pattern Design Proved NP-Hard

Origamists have long suspected their art is computationally hard. Now they've proved it.

KFC 08/11/2010
30 COMMENTS



Some 20 years or so, various individuals recognised that the problem of folding a square sheet of paper into an arbitrary 3D shape had many similarities to problems in computational geometry. These practitioners began developing algorithms that automatically generate the crease patterns that turn a flat sheet into an intricate shape of your choice. Thanks to this and the magical power of modern computing

BIO

The Physics arXiv Blog produces daily coverage of the best new ideas from an online forum called the **Physics arXiv** on which scientists post early versions of their latest ideas. Contact me at KentuckyFC@arxivblog.com

Advertisement



RECENT POSTS FROM ARXIV BLOG

- The Frightening Link Between Phase Changes And Terrorism
- Galactic GPS Put Through Its Paces

Circle Packing for Origami Design Is Hard

Erik D. Demaine* Sándor P. Fekete† Robert J. Lang‡

1 Introduction

Over the last 20 years, the world of origami has been changed by the introduction of design algorithms that bear a close relationship to, if not outright ancestry from, computational geometry. One of the first robust algorithms for origami design was the circle/river method (also called the tree method) developed independently by Lang [7,9] and Meguro [12,13]. This algorithm and its variants provide a systematic method for folding any structure that topologically resembles a graph theoretic weighted tree. Other algorithms followed, notably one by Tachi [15] that gives the crease pattern to fold an arbitrary 3D surface.

Hopes of a general approach for efficiently solving all origami design problems were dashed early on, when Bern and Hayes showed in 1996 that the general problem of crease assignment — given an arbitrary crease pattern, determine whether each fold is mountain or valley — was NP-complete [1]. In fact, they showed more: given a complete crease assignment, simply determining the stacking order of the layers of paper was also NP-complete. Fortunately, while crease assignment in the general case is hard, the crease patterns generated by the various design algorithms carry with them significant extra information associated with each crease, enough extra information that the problem of crease assignment is typically only polynomial in difficulty. This is certainly the case for the tree method of design [3].

Designing a model using the tree method (or one of its variants) is a two-step process: the first step involves solving an optimization problem where one solves for certain key vertices of the crease pattern. The second step constructs creases following a geometric prescription and assigns their status as mountain, valley, or unfolded. The process of constructing the creases and assigning them is definitely polynomial in complexity; but, up to now, the computational complexity of the optimization was not established.

There were reasons for believing that the optimization was, in principle, computationally intractable. The conditions on the vertex coordinates in the

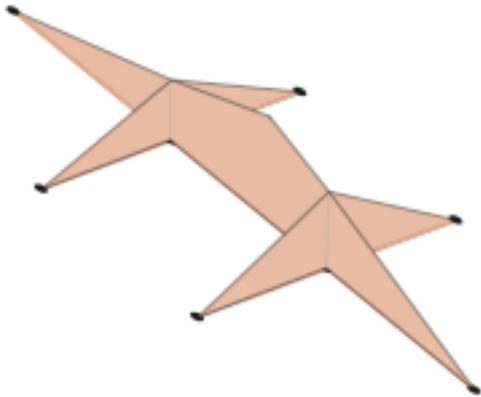
*edemaine@mit.edu.

†s.fekete@tu-bs.de.

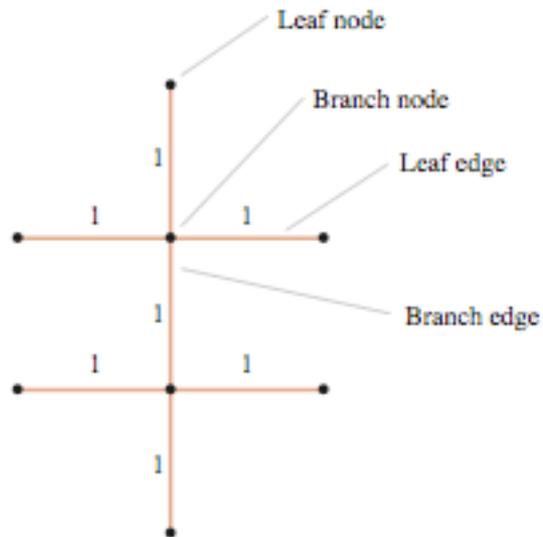
‡robert@langorigami.com.

Schritt 1 - Baumkonstruktion

Hypothetical Base

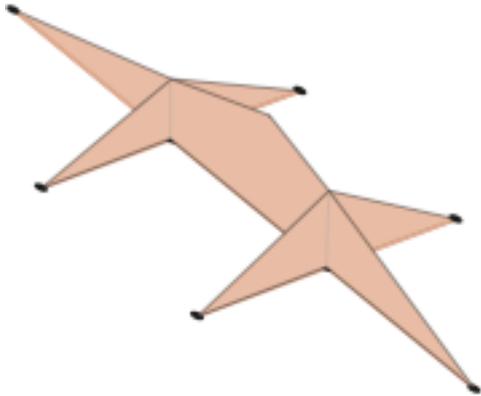


Stick Figure (tree)

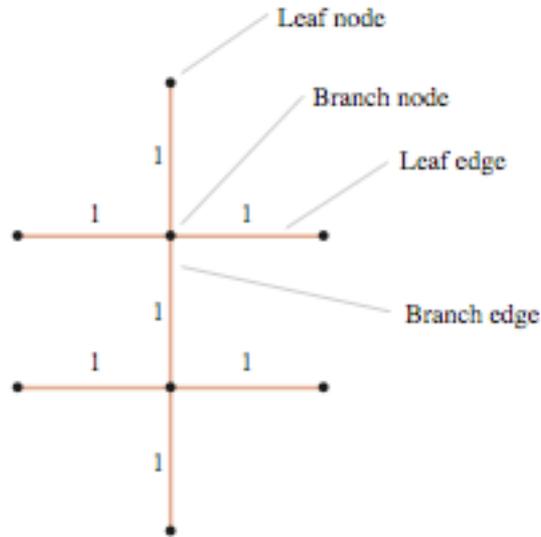


Schritt 1 - Baumkonstruktion

Hypothetical Base



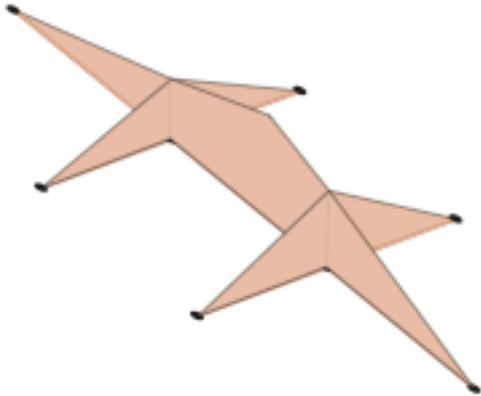
Stick Figure (tree)



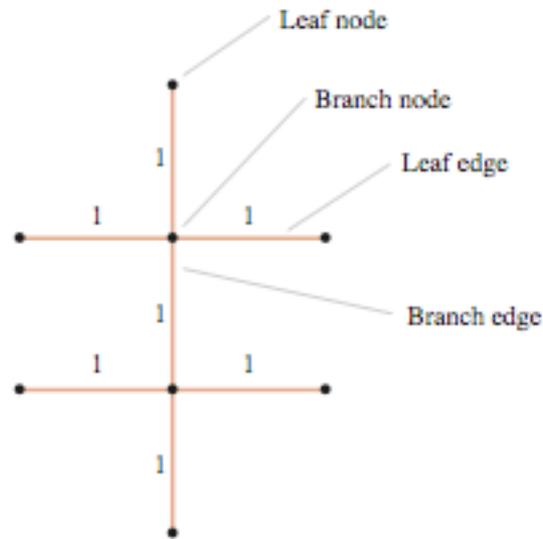
- Jede Kante repräsentiert eine Klappe.

Schritt 1 - Baumkonstruktion

Hypothetical Base



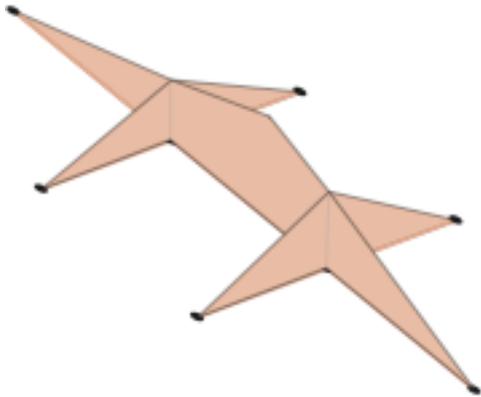
Stick Figure (tree)



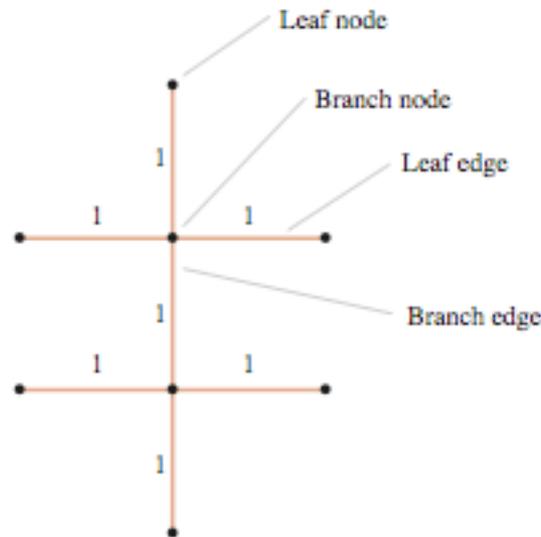
- Jede Kante repräsentiert eine Klappe.
- Jeder Knoten ist eine Klappenspitze oder eine Verbindung von Klappen.

Schritt 1 - Baumkonstruktion

Hypothetical Base

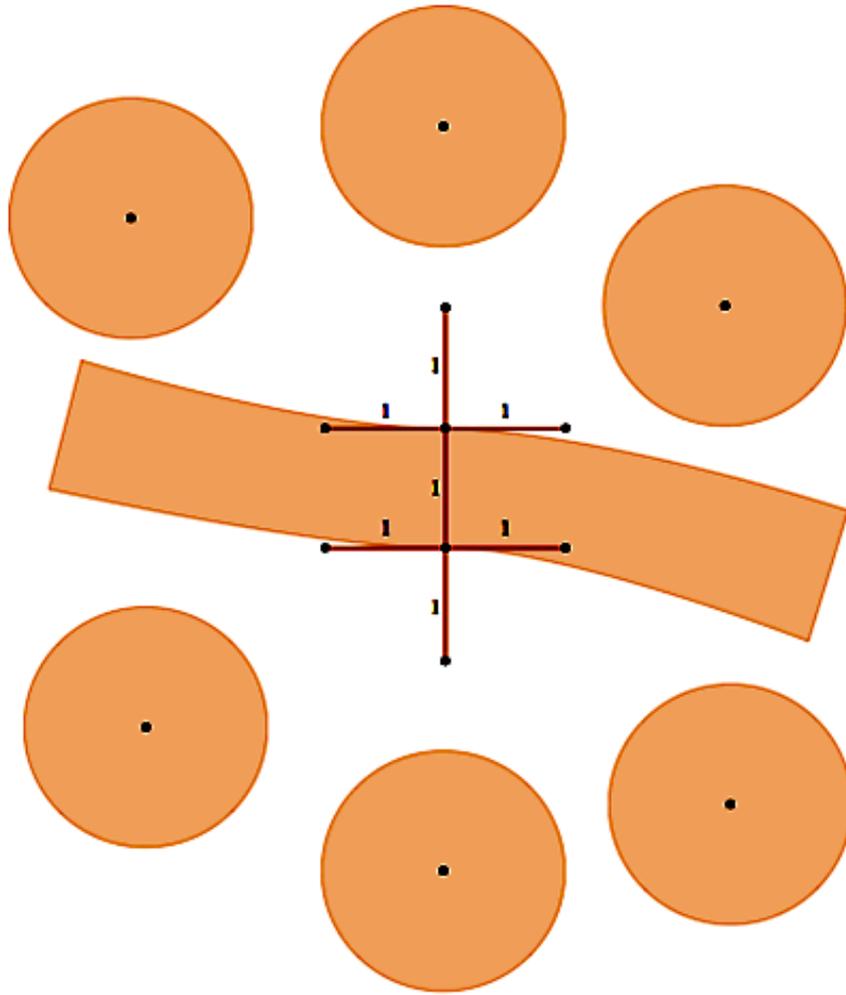


Stick Figure (tree)

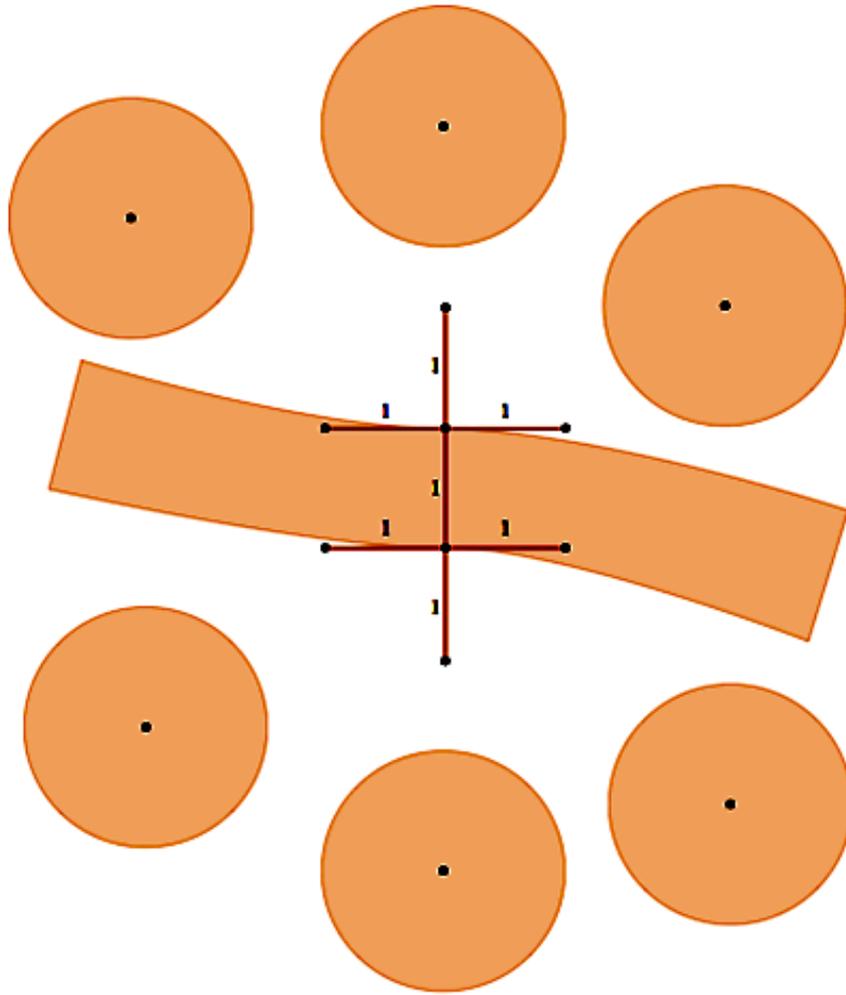


- Jede Kante repräsentiert eine Klappe.
- Jeder Knoten ist eine Klappenspitze oder eine Verbindung von Klappen.
- Endklappen entsprechen Blattkanten im Baum.

Schritt 2 – Von Bäumen zu Flüssen

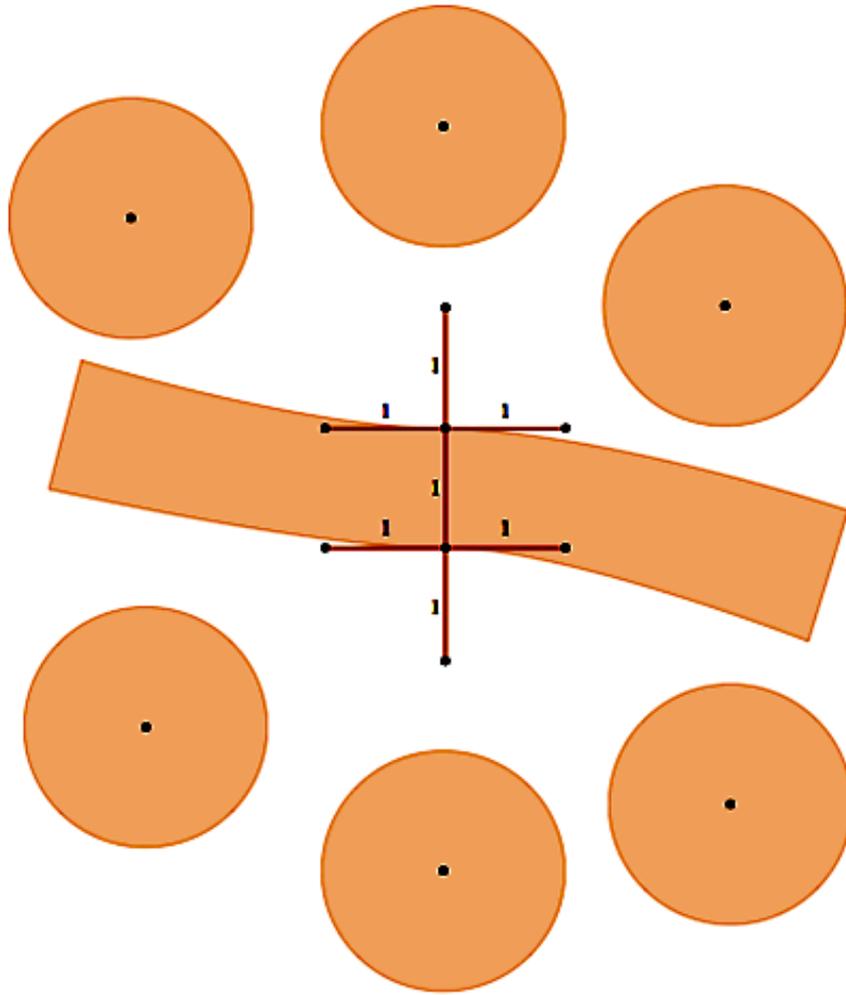


Schritt 2 – Von Bäumen zu Flüssen



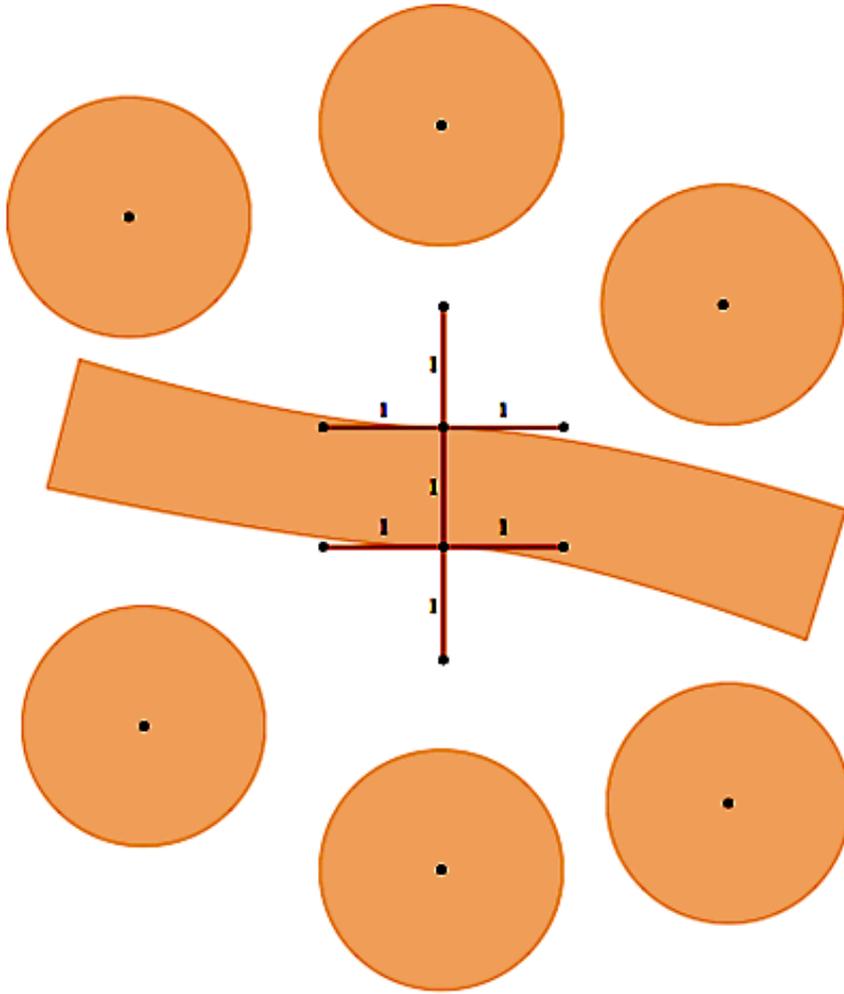
- Endklappen werden zu Kreisen

Schritt 2 – Von Bäumen zu Flüssen



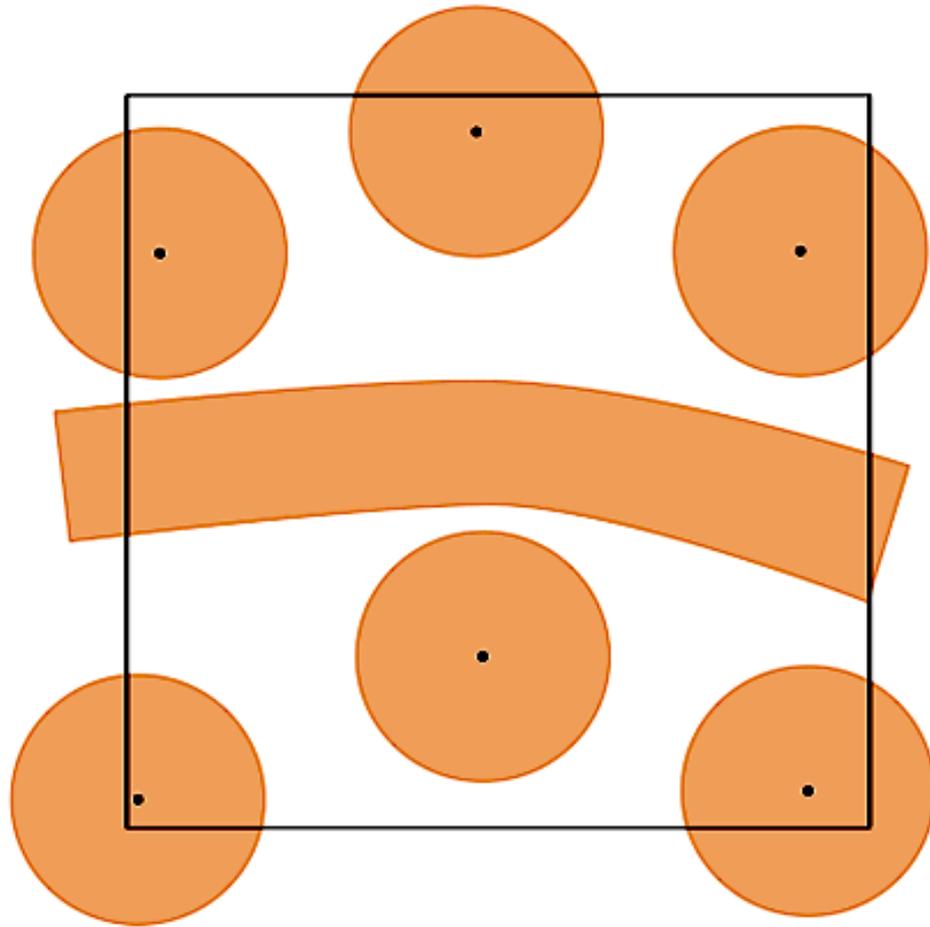
- Endklappen werden zu Kreisen
- Verzweigungskanten werden zu Flüssen

Schritt 2 – Von Bäumen zu Flüssen

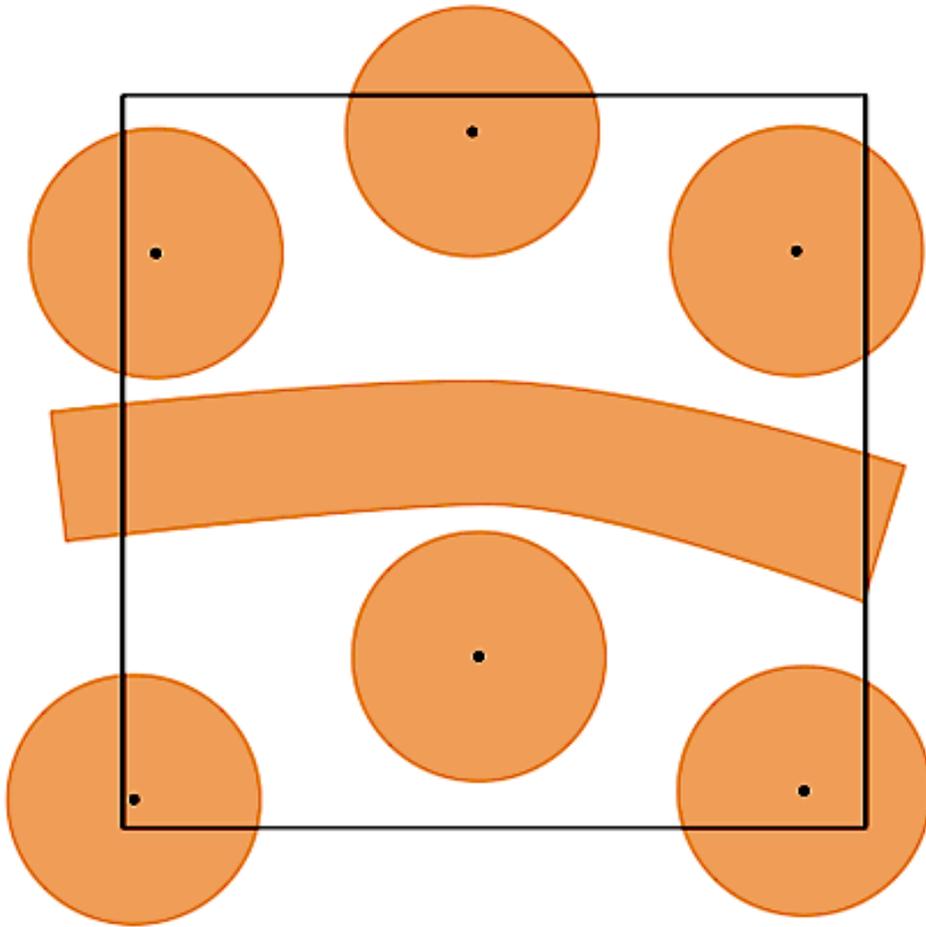


- Endklappen werden zu Kreisen
- Verzweigungskanten werden zu Flüssen
- Radius/Breite eines Kreises/Flusses ist die Länge einer Klappe.

Schritt 3 – Kreispackung in Quadrat

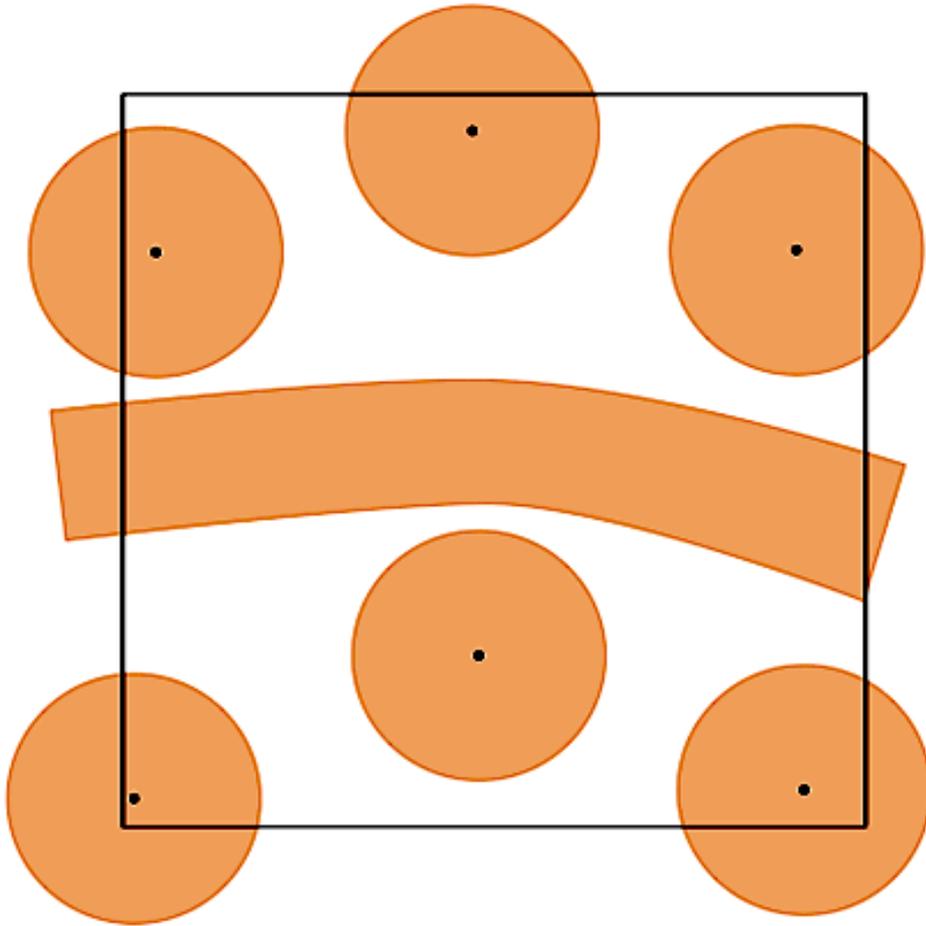


Schritt 3 – Kreispackung in Quadrat



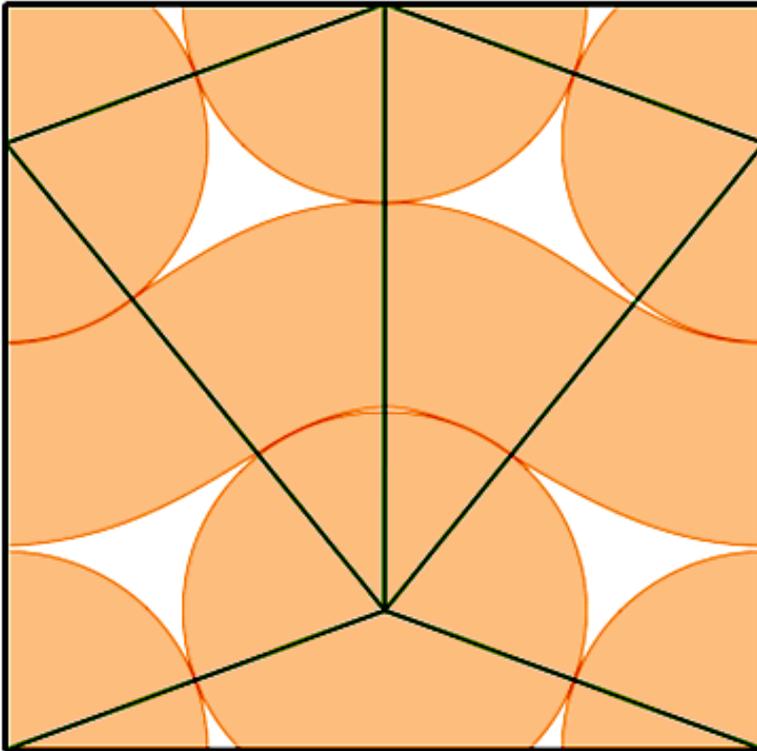
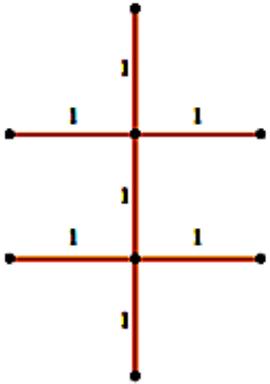
- Packe Mittelpunkte in ein Quadrat, dann vergrößere die Objekte (oder schrumpfe das Quadrat).

Schritt 3 – Kreispackung in Quadrat

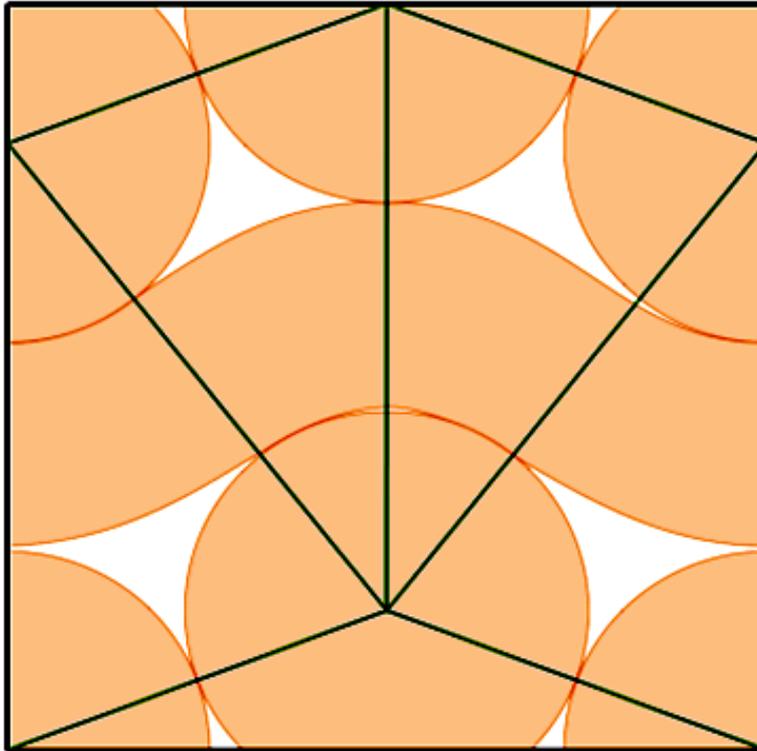
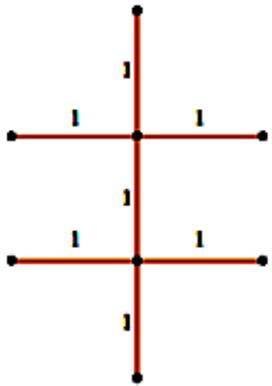


- Packe Mittelpunkte in ein Quadrat, dann vergrößere die Objekte (oder schrumpfe das Quadrat).
- Kreismittelpunkte müssen innerhalb des Quadrates bleiben (Mittelpunkte sind Klappenspitzen).

Schritt 4 – Dualgraph bilden

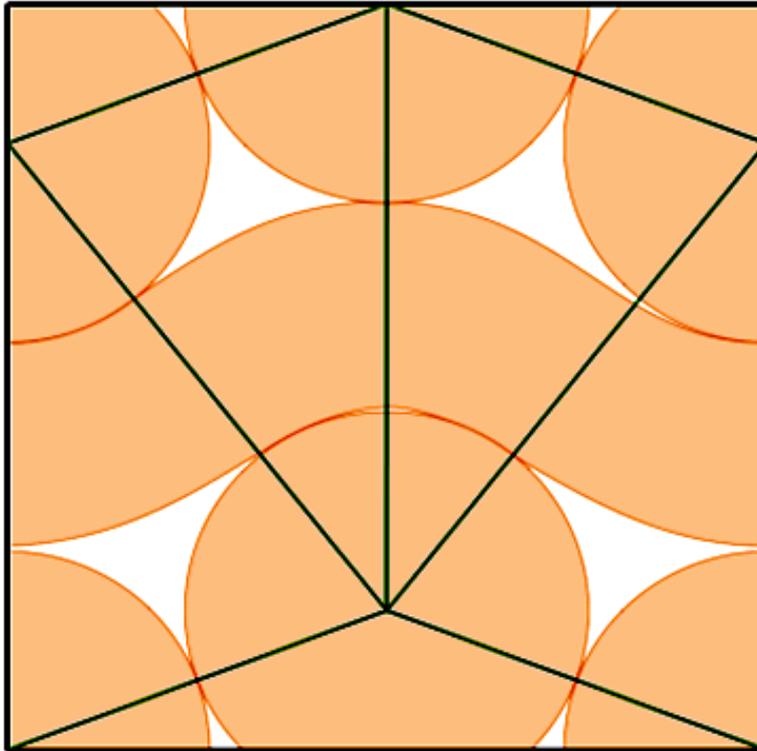
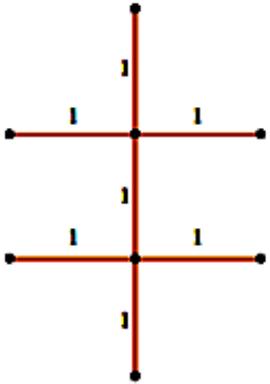


Schritt 4 – Dualgraph bilden



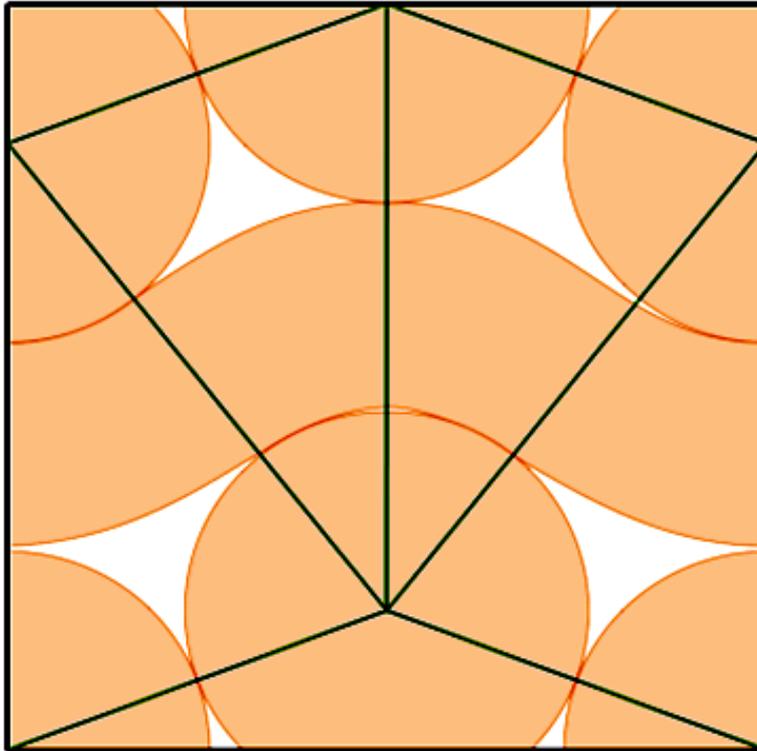
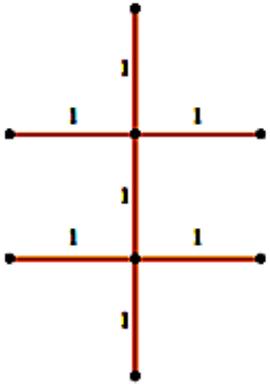
- Resultierende Polygone sind alle konvex.

Schritt 4 – Dualgraph bilden



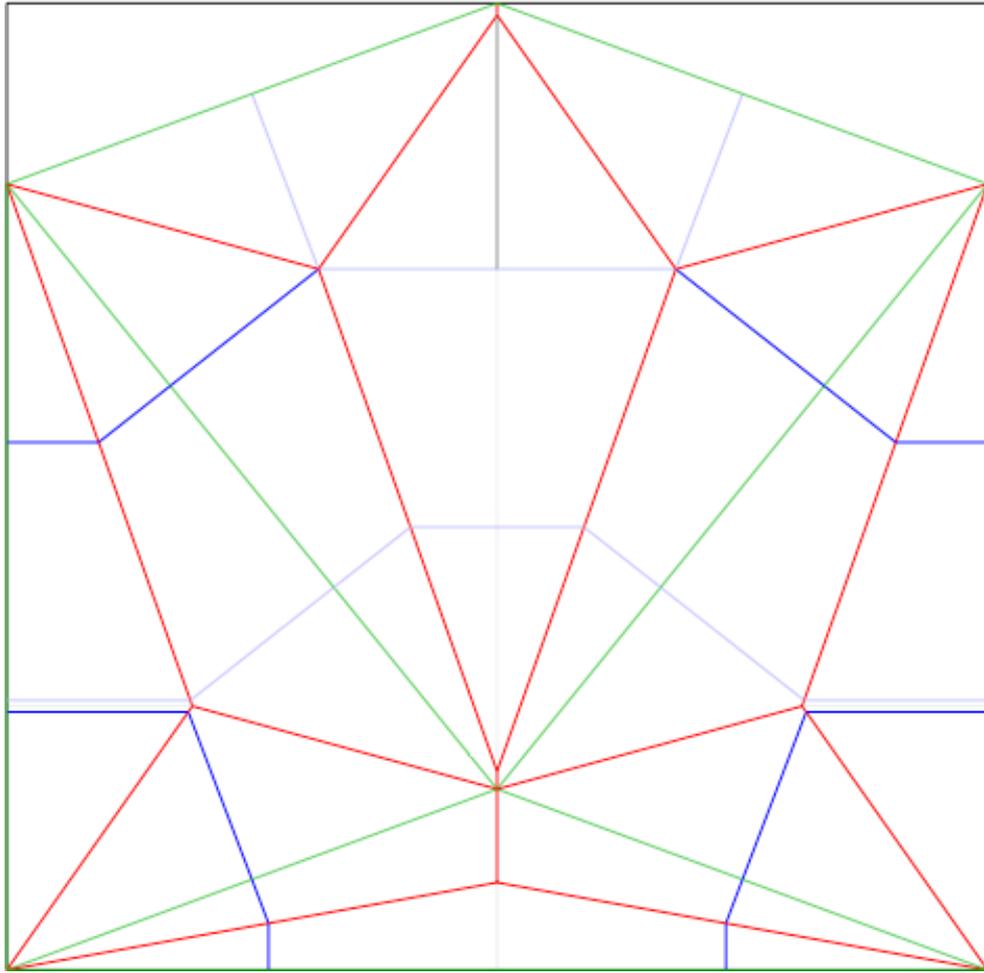
- Resultierende Polygone sind alle konvex.
- Keine “losen” Kreise erlaubt

Schritt 4 – Dualgraph bilden

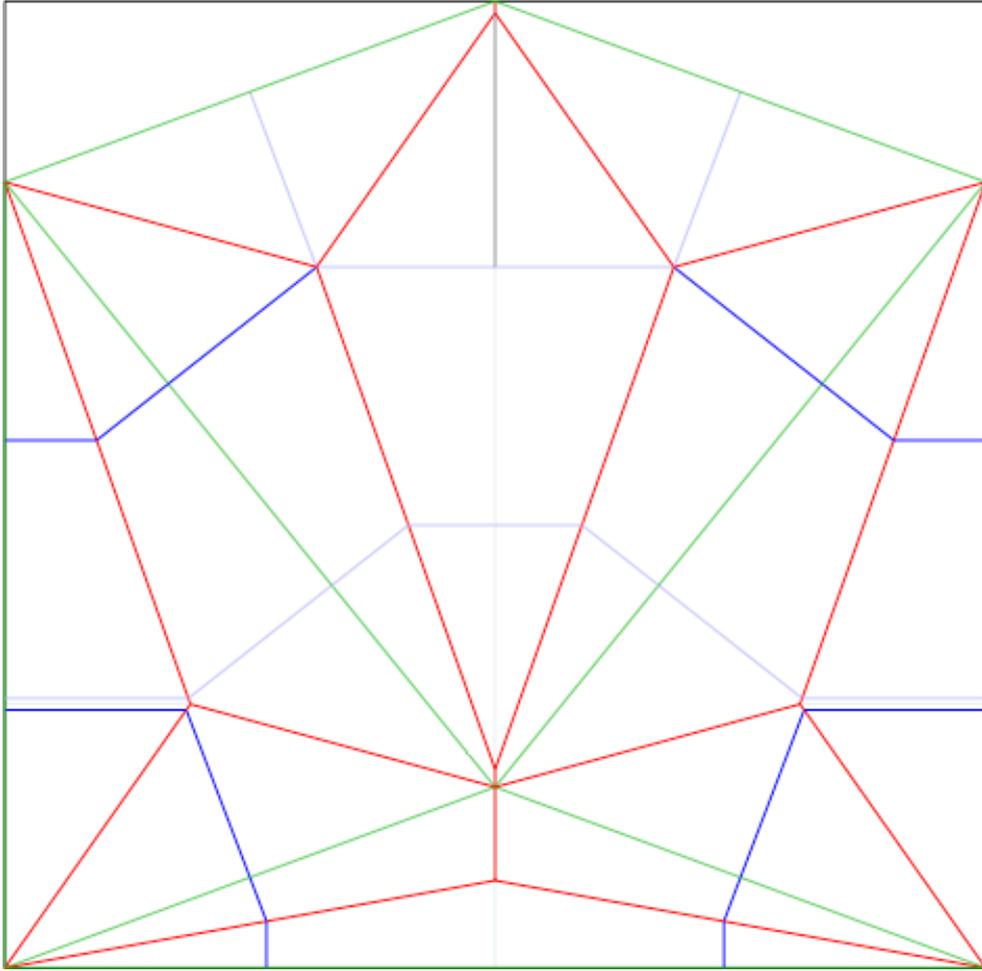


- Resultierende Polygone sind alle konvex.
- Keine “losen” Kreise erlaubt
- Verbindungen sind axiale Faltungen

Schritt 5 – Quadrat mit Faltungen füllen

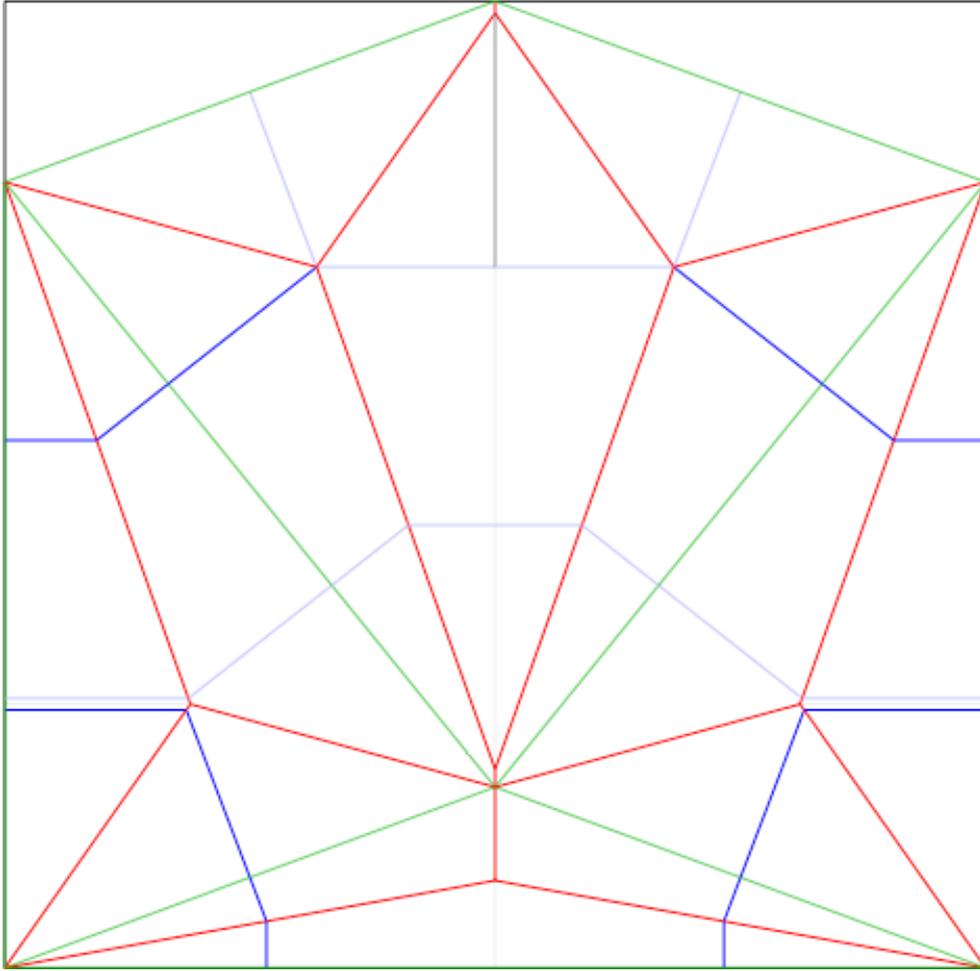


Schritt 5 – Quadrat mit Faltungen füllen



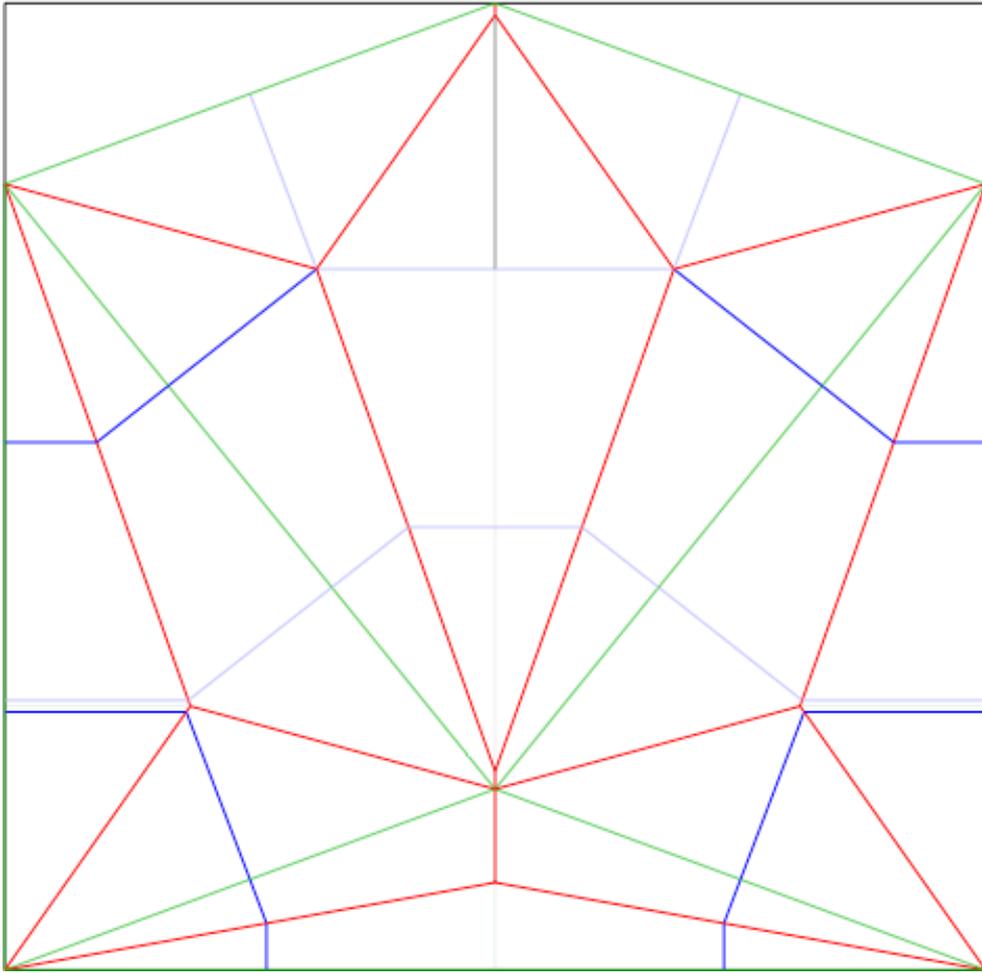
- Axial (grün)

Schritt 5 – Quadrat mit Faltungen füllen



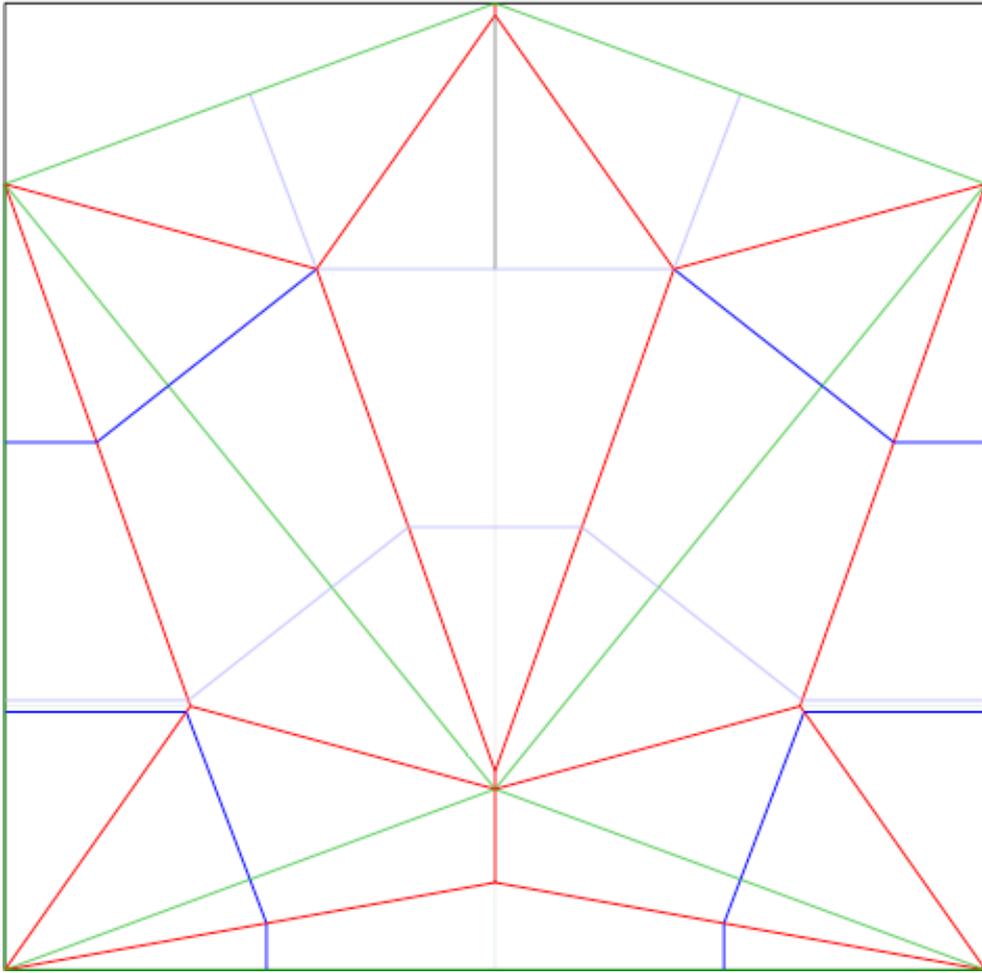
- Axial (grün)
- Kante (rot)

Schritt 5 – Quadrat mit Faltungen füllen



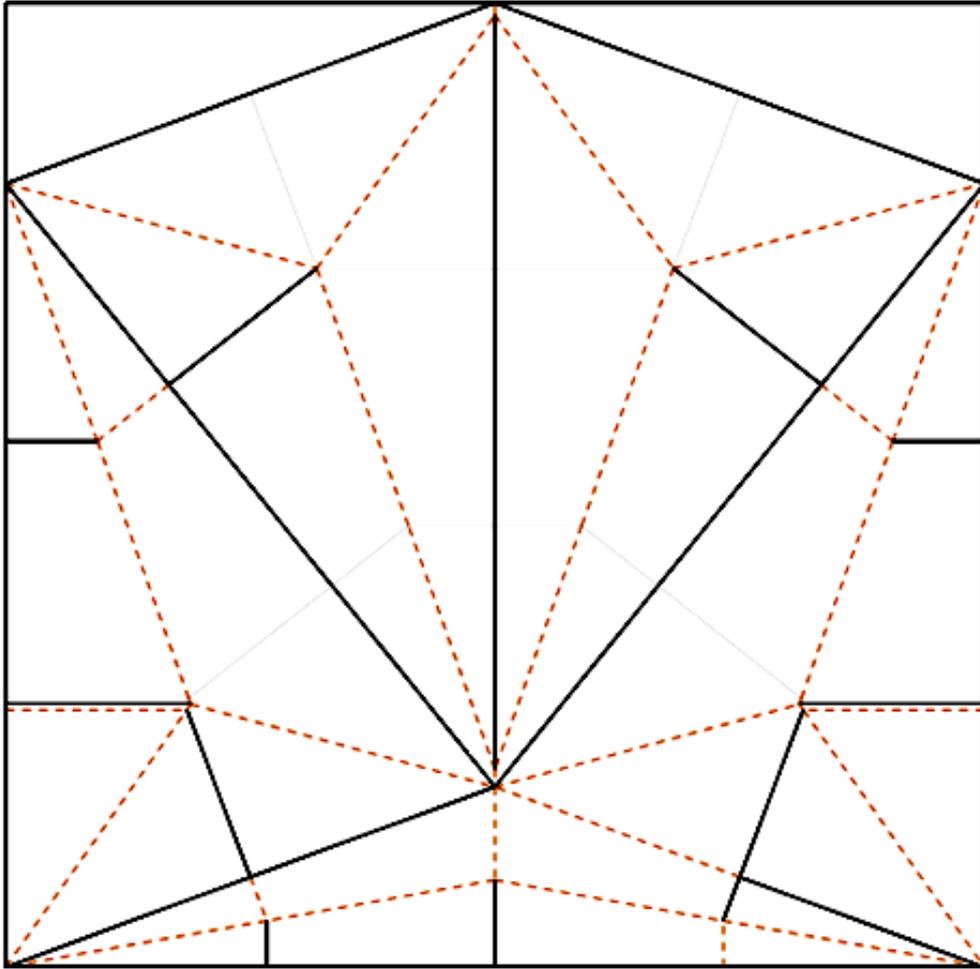
- Axial (grün)
- Kante (rot)
- Zwickel (grau)

Schritt 5 – Quadrat mit Faltungen füllen

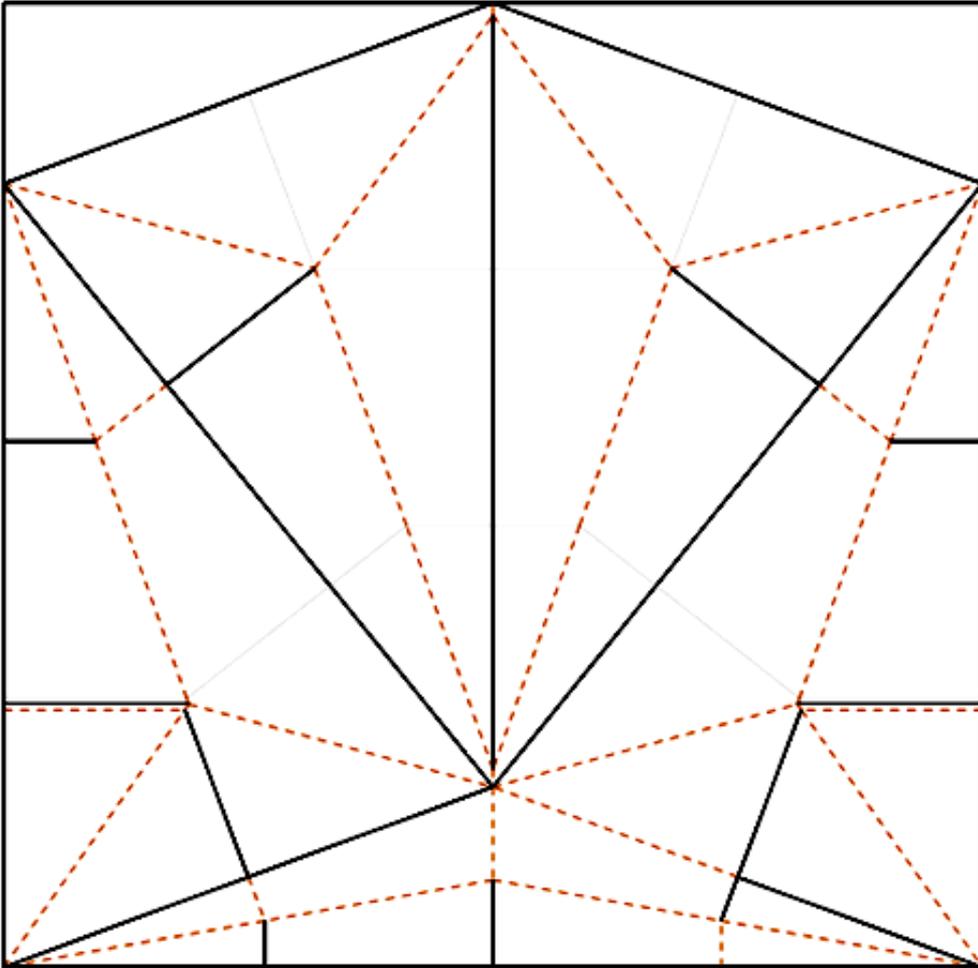


- Axial (grün)
- Kante (rot)
- Zwickel (grau)
- Scharnier (blau)

Schritt 6 – Faltungen zuordnen



Schritt 6 – Faltungen zuordnen



- Ein exakter Algorithmus existiert
 - Implementiert in TreeMaker
 - Schneller; Einfache Regeln anwenden und die Ausnahmen behandeln
 - Axial = fast immer Berg
 - Kante = immer Tal
 - Zwickel = immer Berg
 - Scharnier = Berg, Tal, flach, abhängig von Klappenrichtung

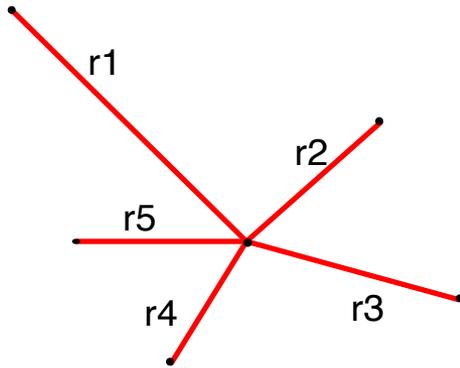
Sterne und Kreispackungen



Sterne und Kreispackungen

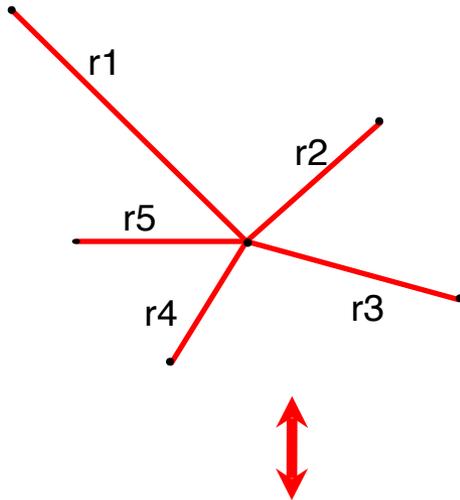
- Die Pfadbedingungen für einen Stern sind äquivalent zum Problem, die Mittelpunkte einer Menge von Kreisen in ein Quadrat zu packen.

Sterne und Kreispackungen



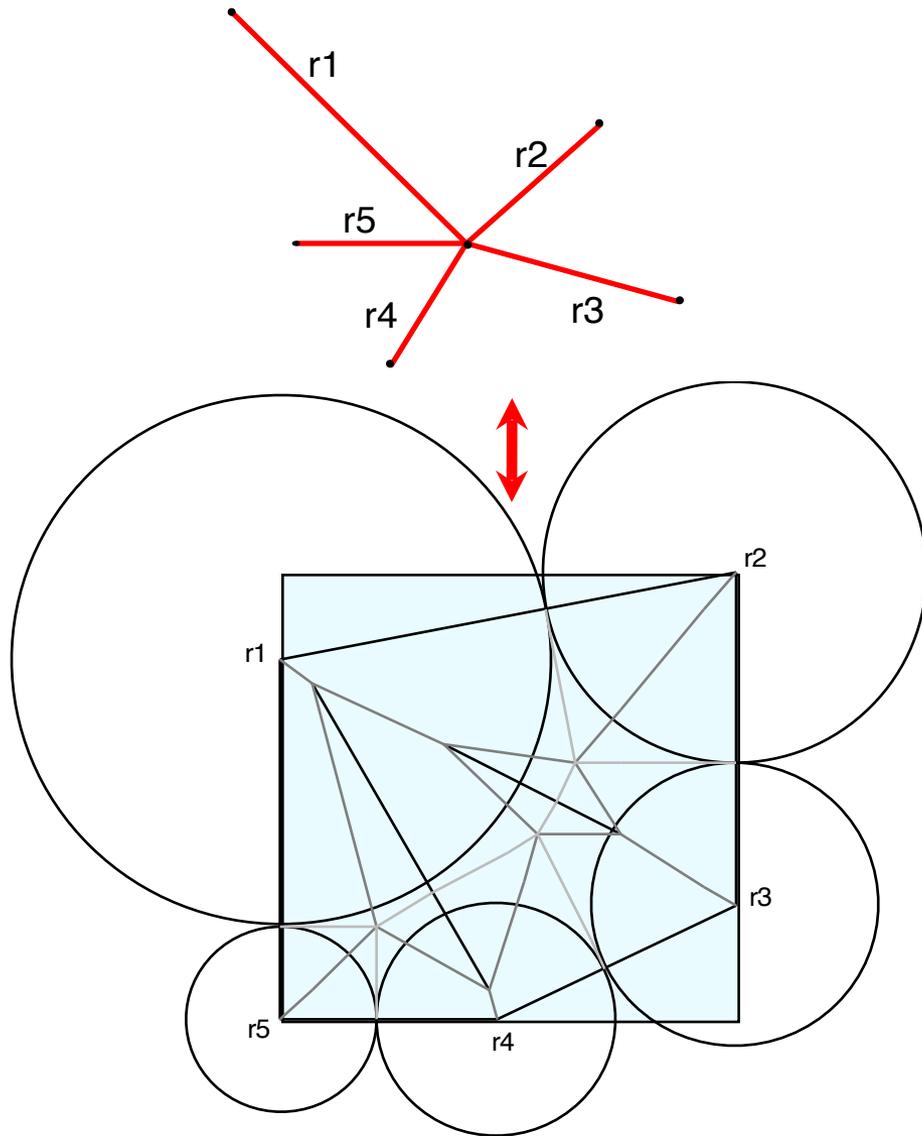
- Die Pfadbedingungen für einen Stern sind äquivalent zum Problem, die Mittelpunkte einer Menge von Kreisen in ein Quadrat zu packen.

Sterne und Kreispackungen



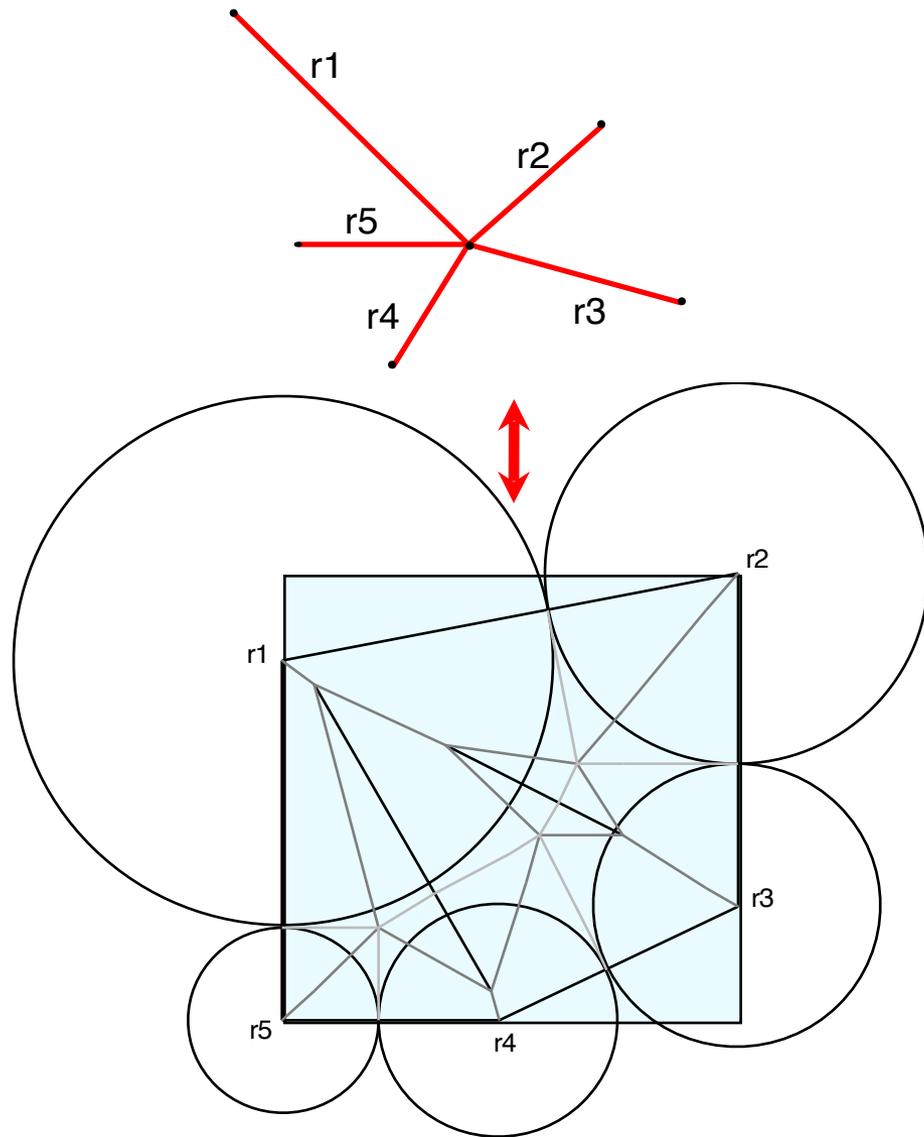
- Die Pfadbedingungen für einen Stern sind äquivalent zum Problem, die Mittelpunkte einer Menge von Kreisen in ein Quadrat zu packen.

Sterne und Kreispackungen



- Die Pfadbedingungen für einen Stern sind äquivalent zum Problem, die Mittelpunkte einer Menge von Kreisen in ein Quadrat zu packen.

Sterne und Kreispackungen



- Die Pfadbedingungen für einen Stern sind äquivalent zum Problem, die Mittelpunkte einer Menge von Kreisen in ein Quadrat zu packen.
- Der Kreisradius entspricht jeweils der Kantenlänge.

Komplexität: 3-Partition



Komplexität: 3-Partition

Gegeben:

$3k$ Zahlen $\frac{1}{4} < \boxed{?} < \frac{1}{2}$

Komplexität: 3-Partition



Gegeben:

3k Zahlen

$$\frac{1}{4} < \boxed{?} < \frac{1}{2}$$

Komplexität: 3-Partition



Gegeben:

3k Zahlen

$$\frac{1}{4} < \boxed{?} < \frac{1}{2}$$

Komplexität: 3-Partition



Gegeben:

$3k$ Zahlen $\frac{1}{4} < \boxed{?} < \frac{1}{2}$

Gesucht:

Eine Partition in gleichgroße
Tripel – d.h., eine
Packung in k
Einheitscontainer.

Komplexität: 3-Partition



Gegeben:

$3k$ Zahlen $\frac{1}{4} < \boxed{?} < \frac{1}{2}$

Gesucht:

Eine Partition in gleichgroße
Tripel – d.h., eine
Packung in k
Einheitscontainer.

Komplexität: 3-Partition



Gegeben:

$3k$ Zahlen $\frac{1}{4} < \boxed{?} < \frac{1}{2}$

Gesucht:

Eine Partition in gleichgroße
Tripel – d.h., eine
Packung in k
Einheitscontainer.

Bekannt:

3-Partition ist NP-schwer.

Komplexität: 3-Partition



Gegeben:

$3k$ Zahlen $\frac{1}{4} < \boxed{?} < \frac{1}{2}$

Gesucht:

Eine Partition in gleichgroße
Tripel – d.h., eine
Packung in k
Einheitscontainer.

Bekannt:

3-Partition ist NP-schwer.

Komplexität: Beweis von NP-Schwere



Komplexität: Beweis von NP-Schwere

Gegeben:

$3k$ Zahlen $\frac{1}{4} < \boxed{?} < \frac{1}{2}$

Komplexität: Beweis von NP-Schwere



Gegeben:

3k Zahlen $\frac{1}{4} < \boxed{?} < \frac{1}{2}$

Komplexität: Beweis von NP-Schwere



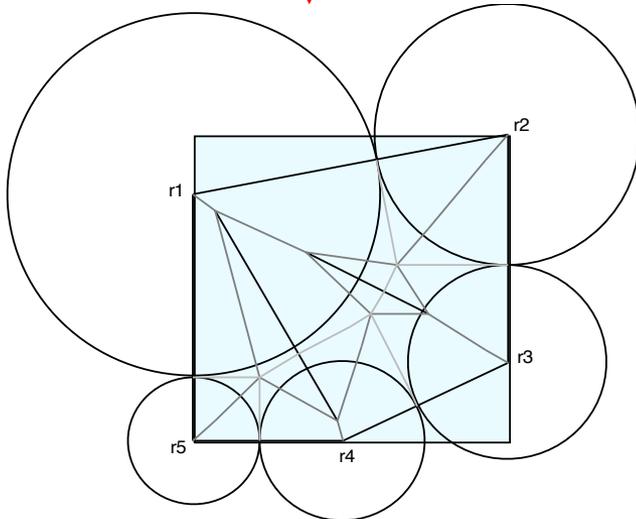
Gegeben:

3k Zahlen $\frac{1}{4} < \boxed{?} < \frac{1}{2}$

Transformiere in:

eine Kreispackungsinstanz die gelöst werden kann, gdw die 3-Partitioninstanz lösbar ist.

Komplexität: Beweis von NP-Schwere



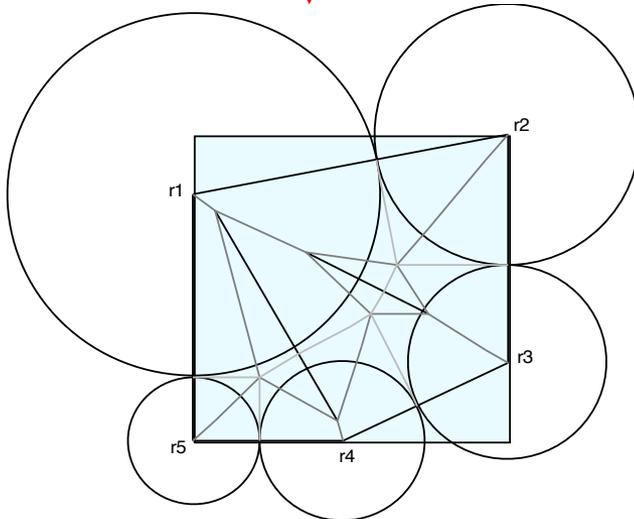
Gegeben:

3k Zahlen $\frac{1}{4} < \boxed{?} < \frac{1}{2}$

Transformiere in:

eine Kreispackungsinstanz die gelöst werden kann, gdw die 3-Partitioninstanz lösbar ist.

Komplexität: Beweis von NP-Schwere



Gegeben:

3k Zahlen $\frac{1}{4} < \boxed{?} < \frac{1}{2}$

Transformiere in:

eine Kreispackungsinstanz die gelöst werden kann, gdw die 3-Partitioninstanz lösbar ist.

Konsequenz 1:

Falls es einen polynomiellen Algorithmus für Kreispacken gibt, dann ist $P=NP$.

Konsequenz 2:

Falls es einen effizienten Algorithmus für Origami-Design gibt, ist $P=NP$

Untere Schranke für Quadrate

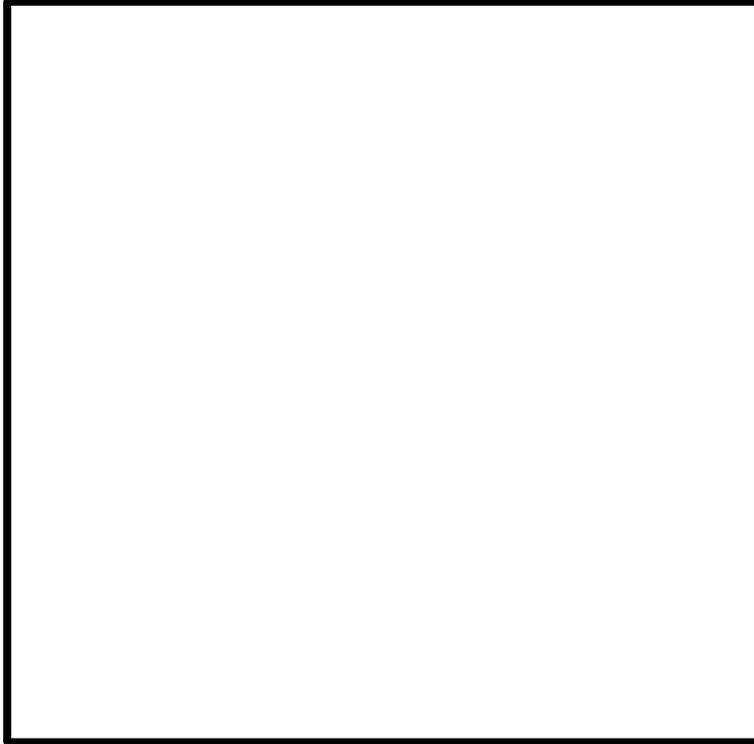
Satz:

Jede Menge von
Quadraten mit
Gesamtfläche
höchstens 0.5 lässt sich
in ein Einheitsquadrat
packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



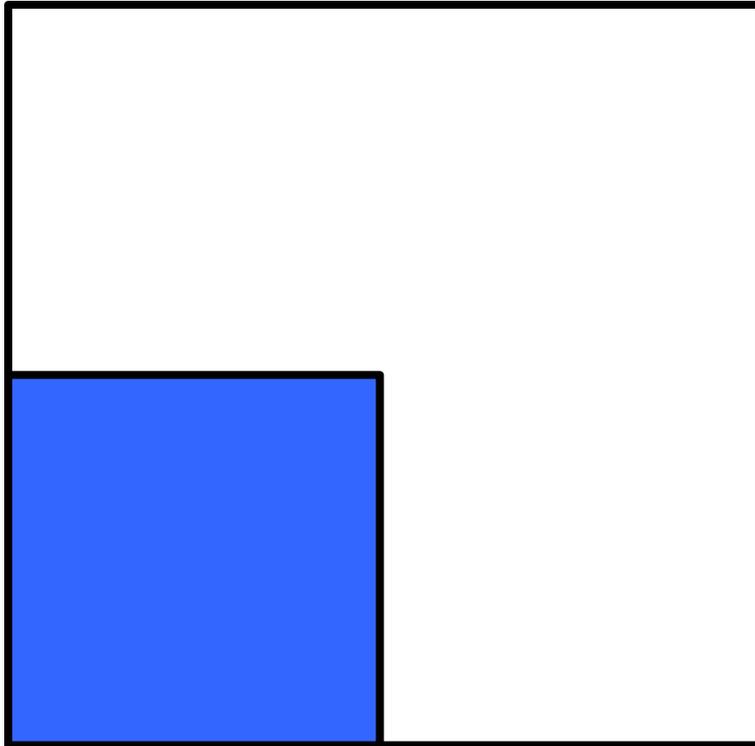
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



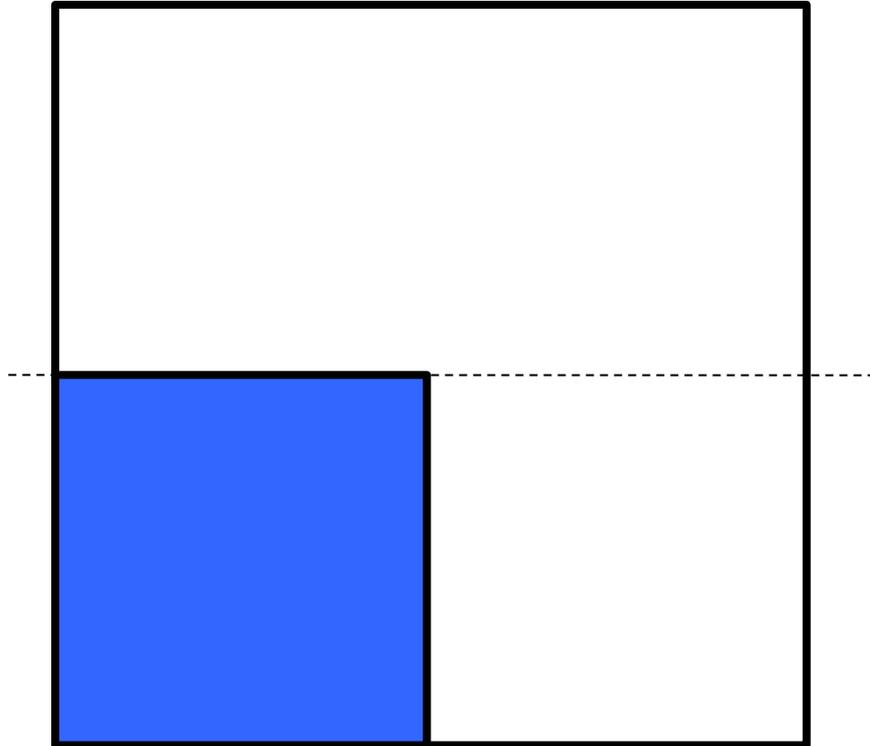
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



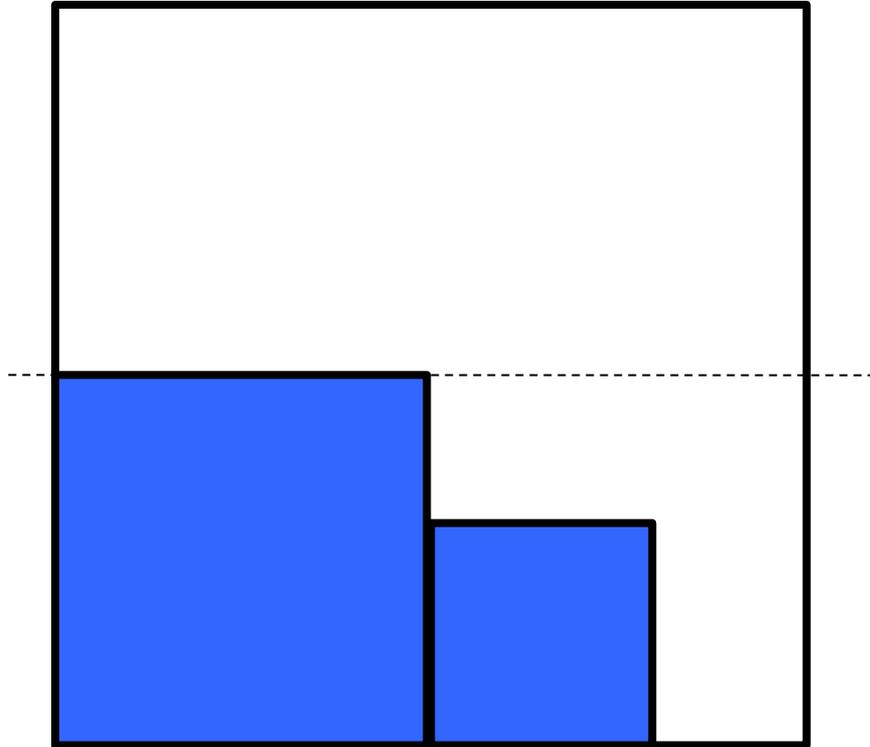
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



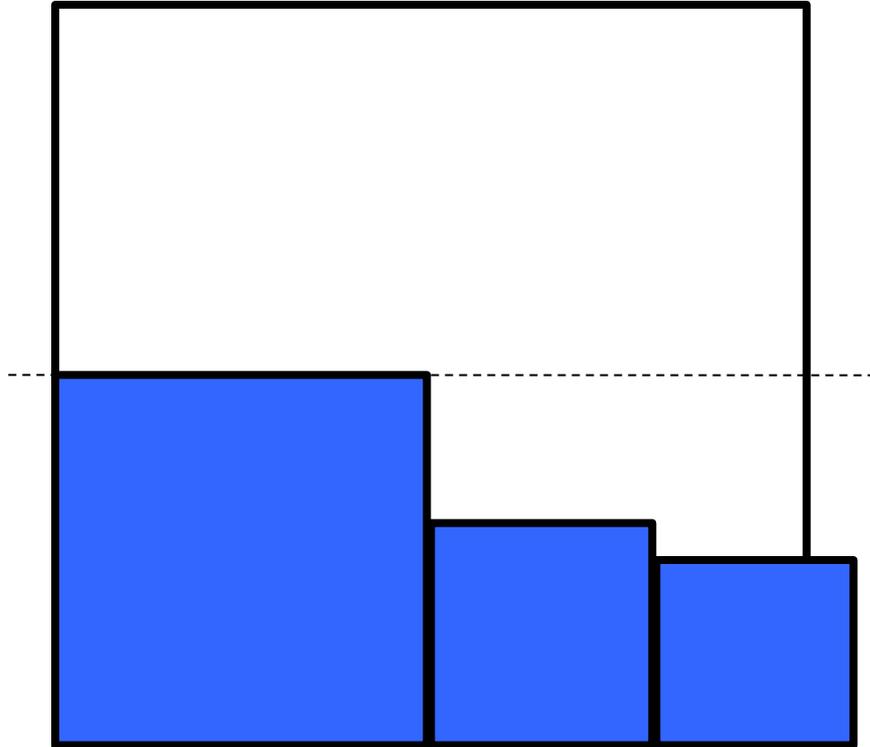
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



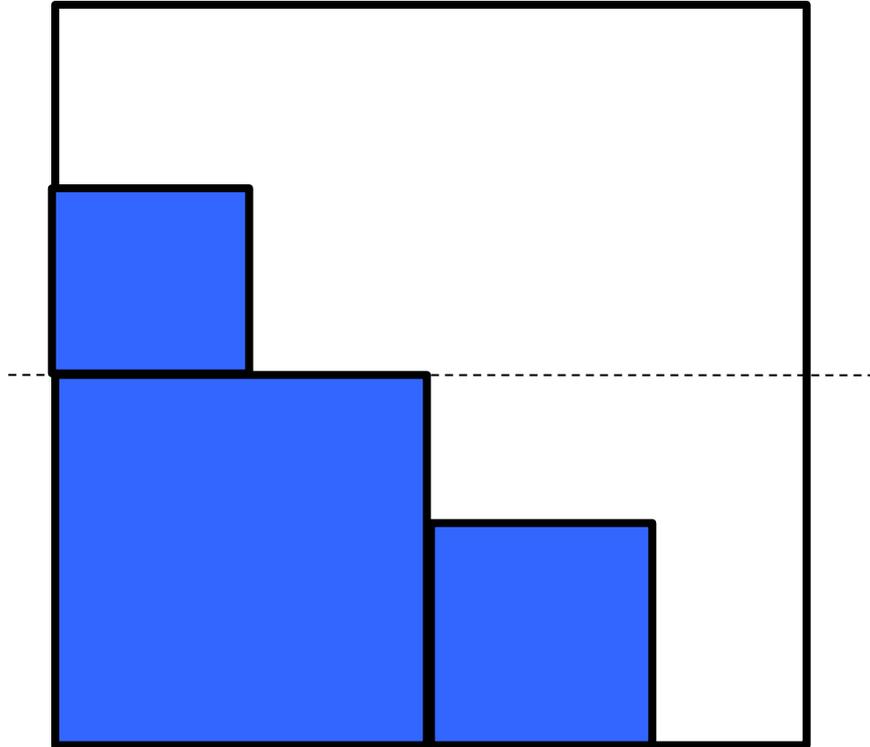
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



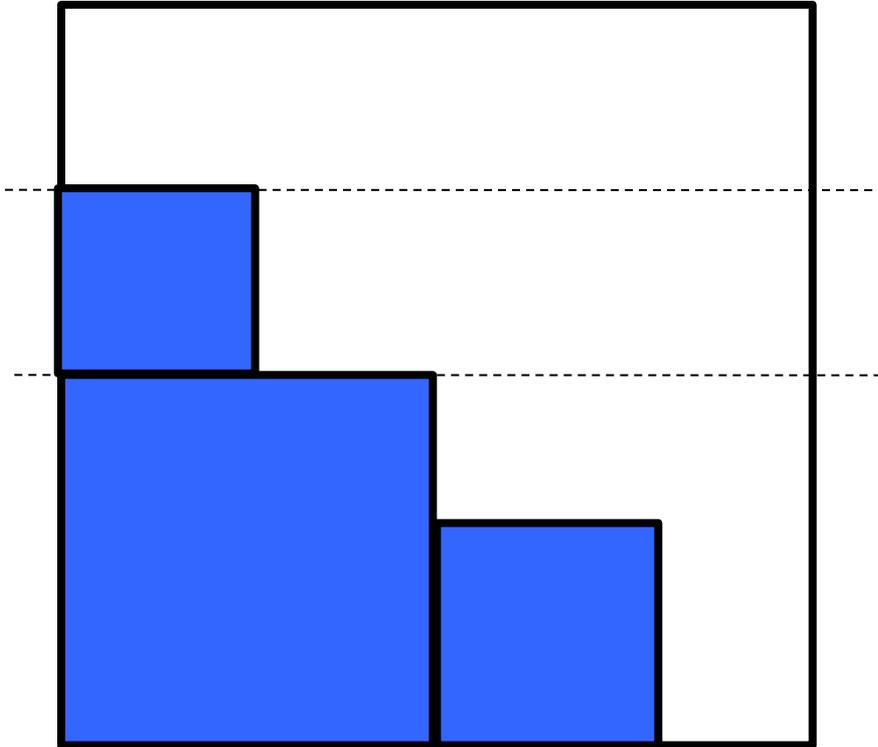
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



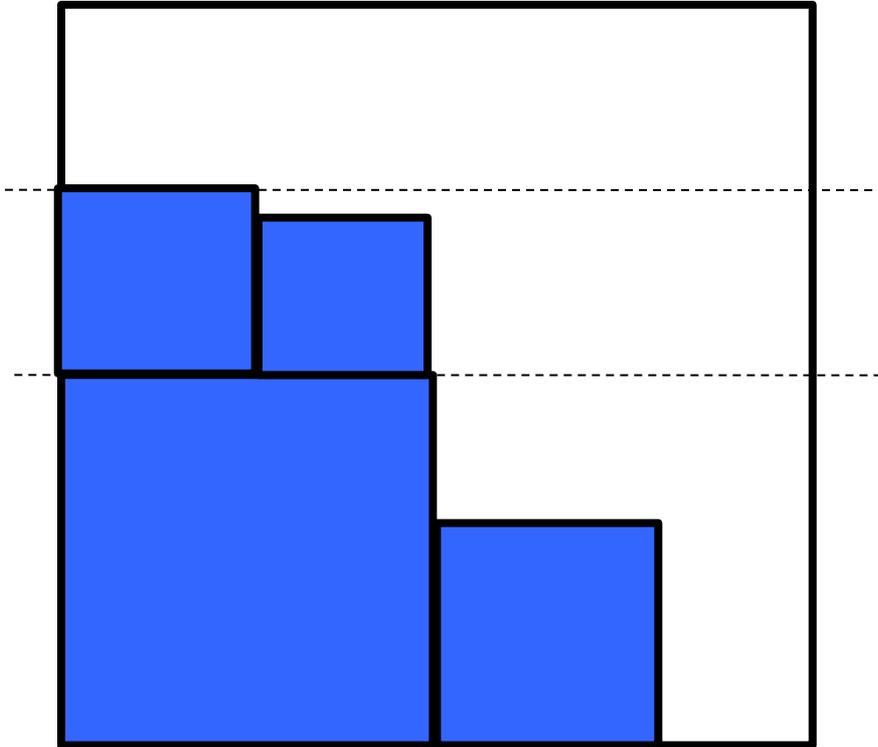
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



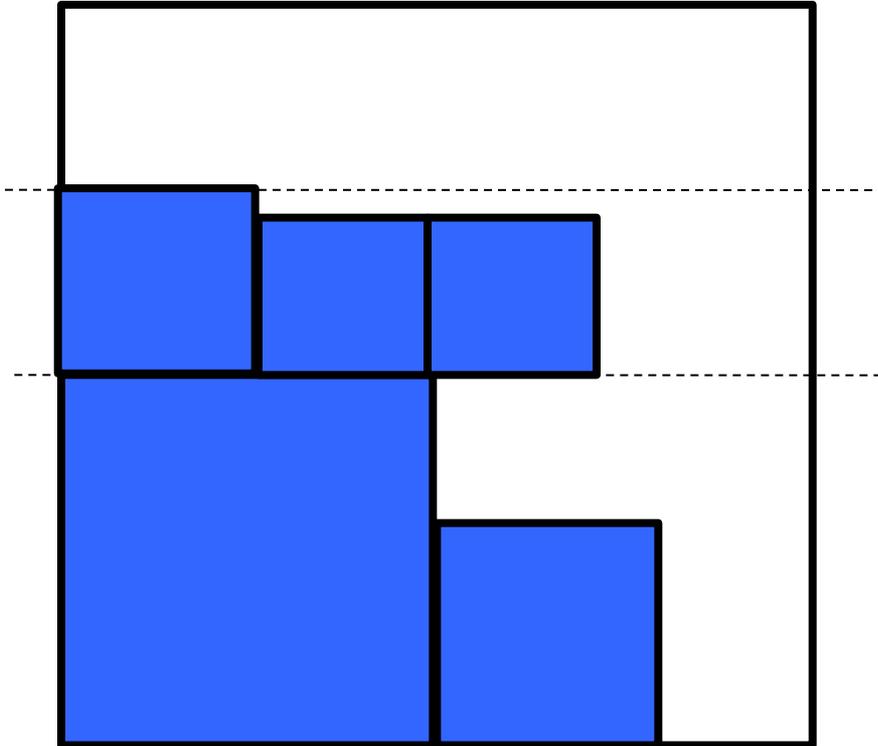
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



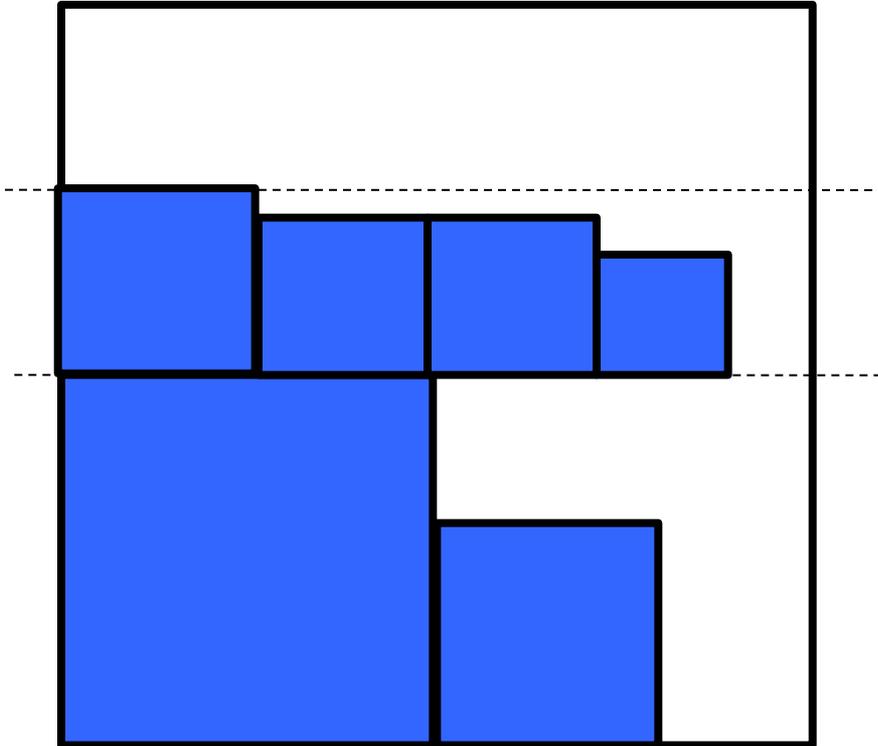
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



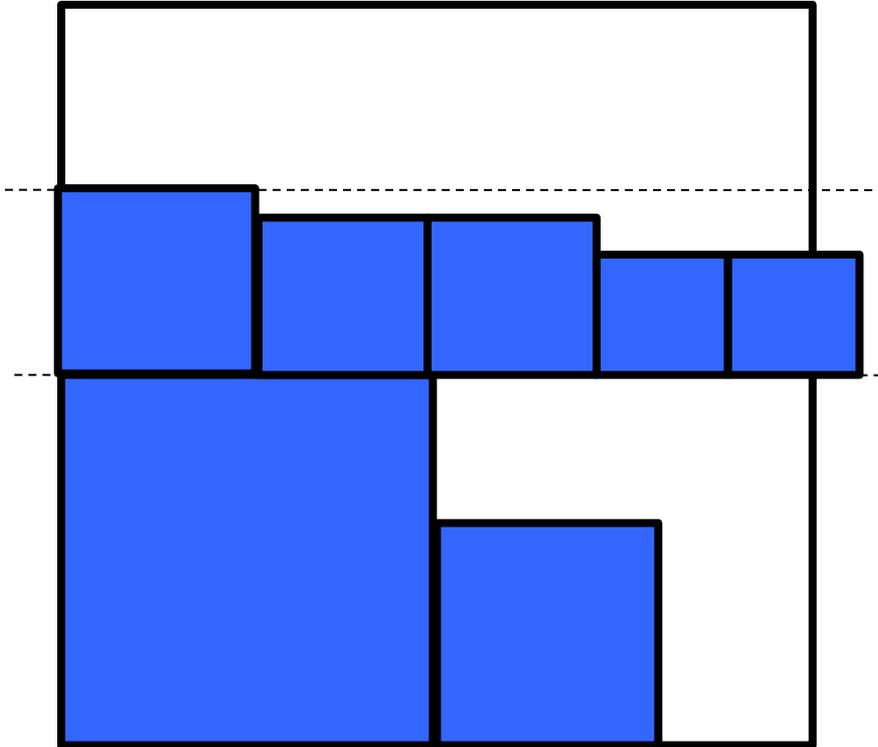
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



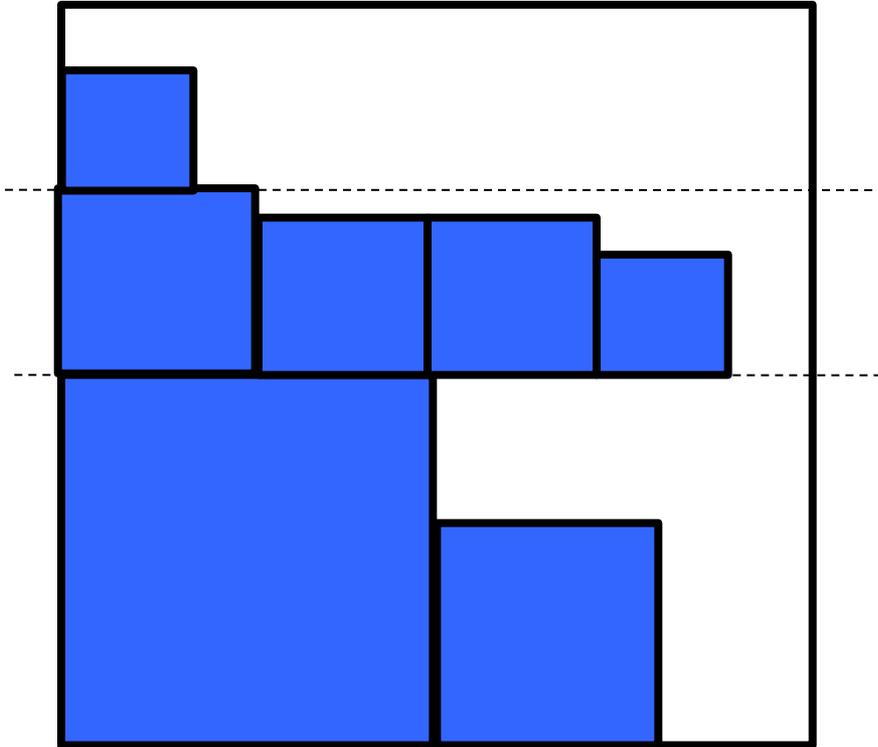
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



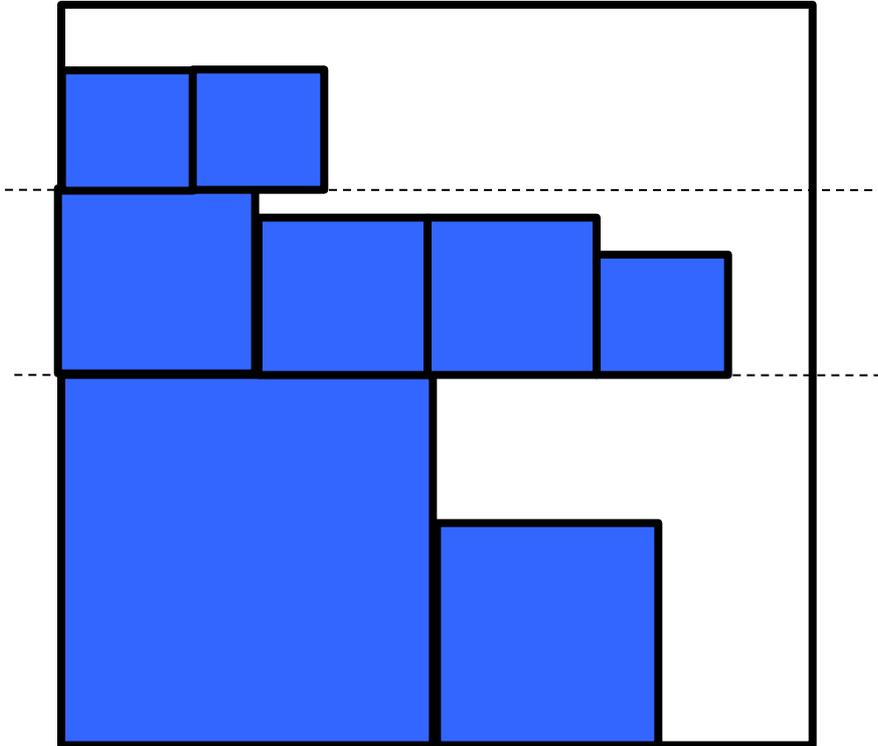
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



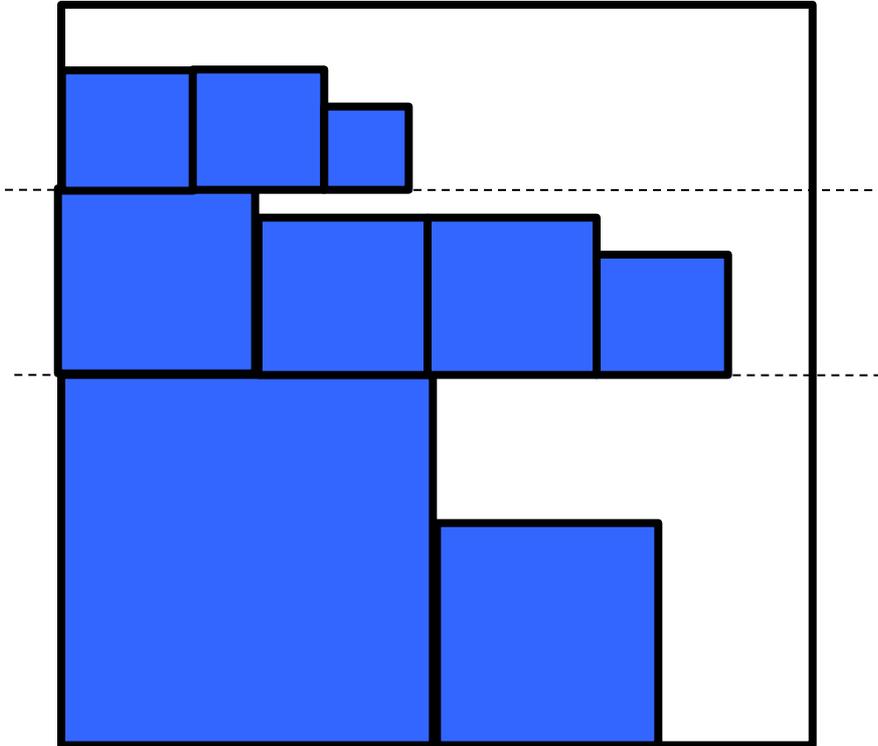
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



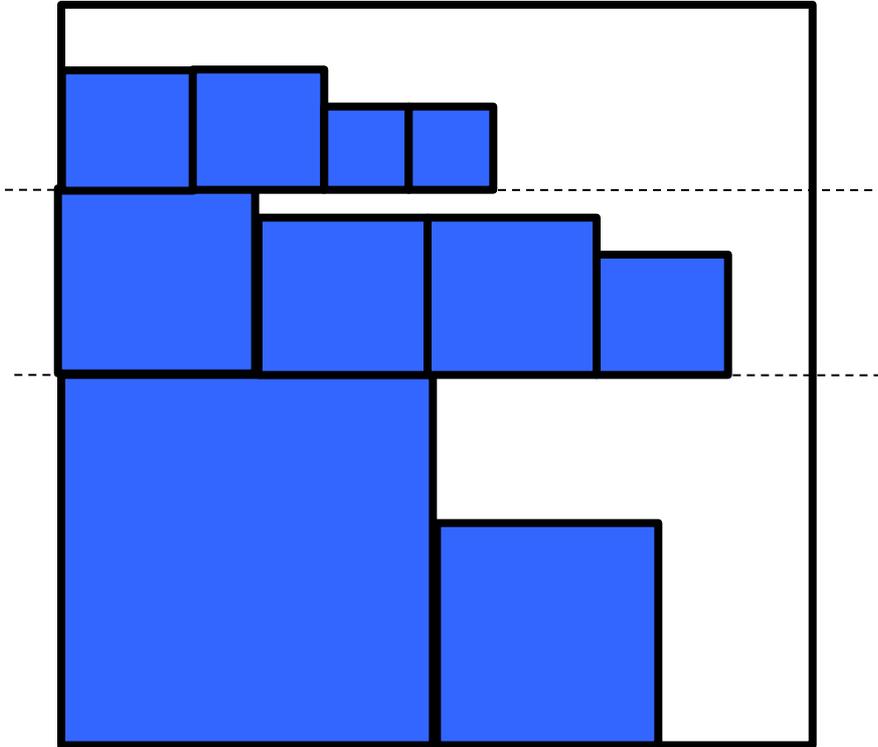
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



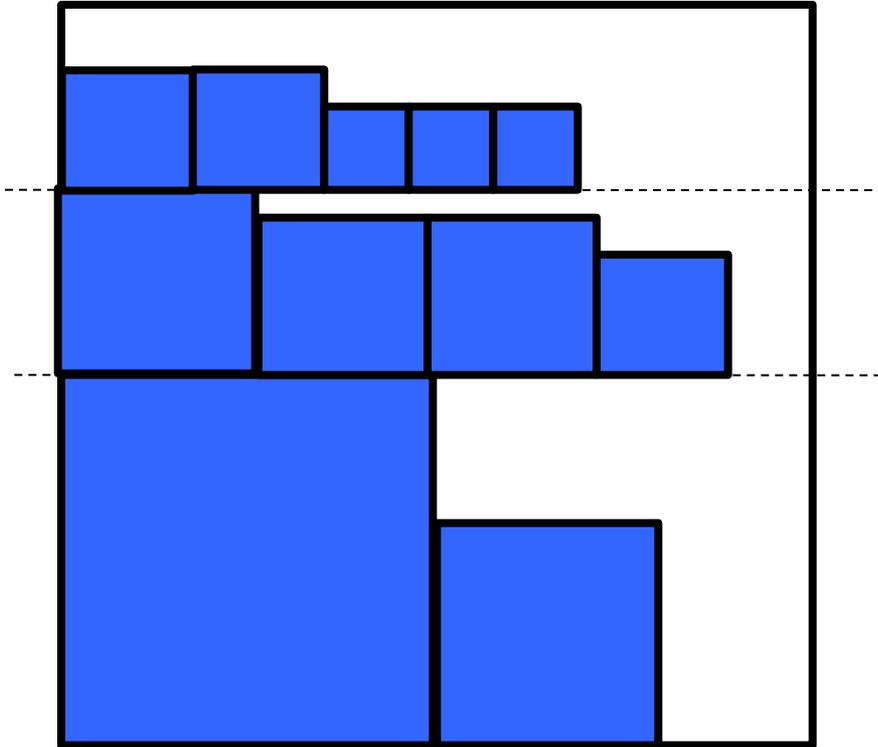
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



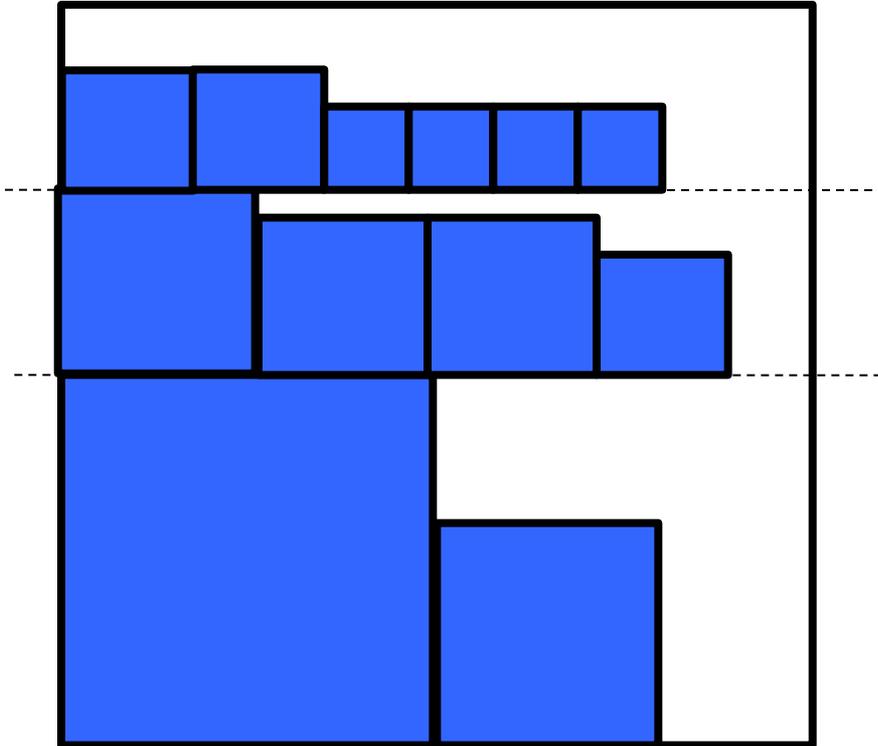
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



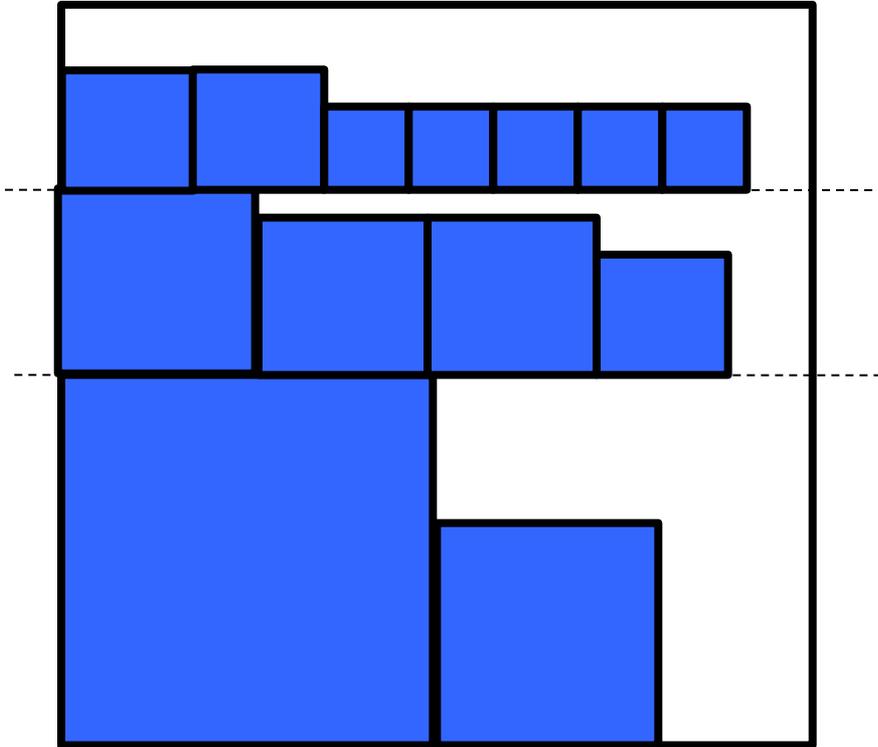
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



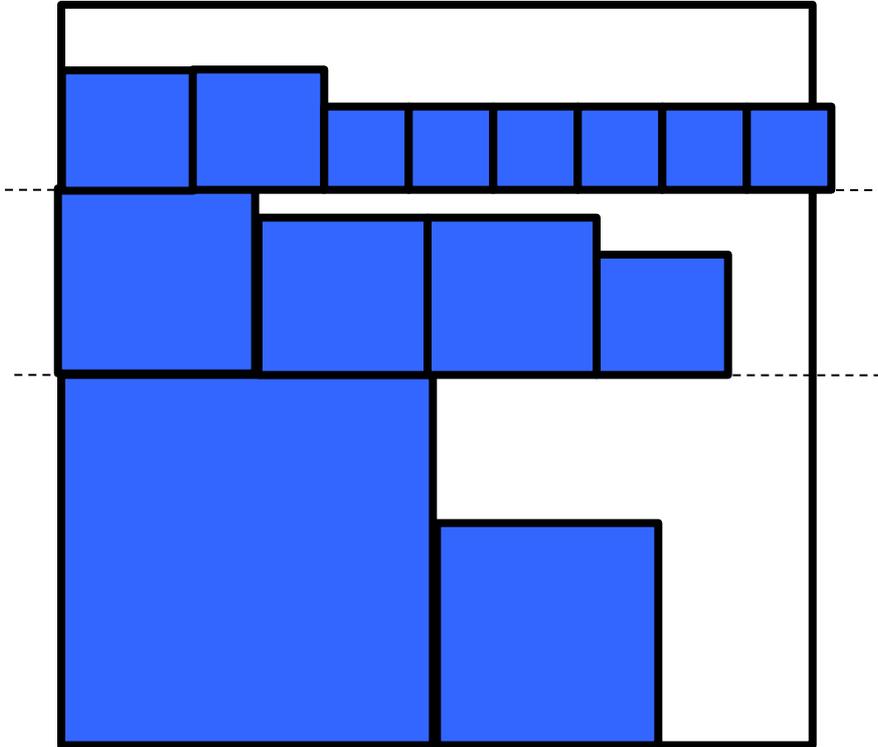
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



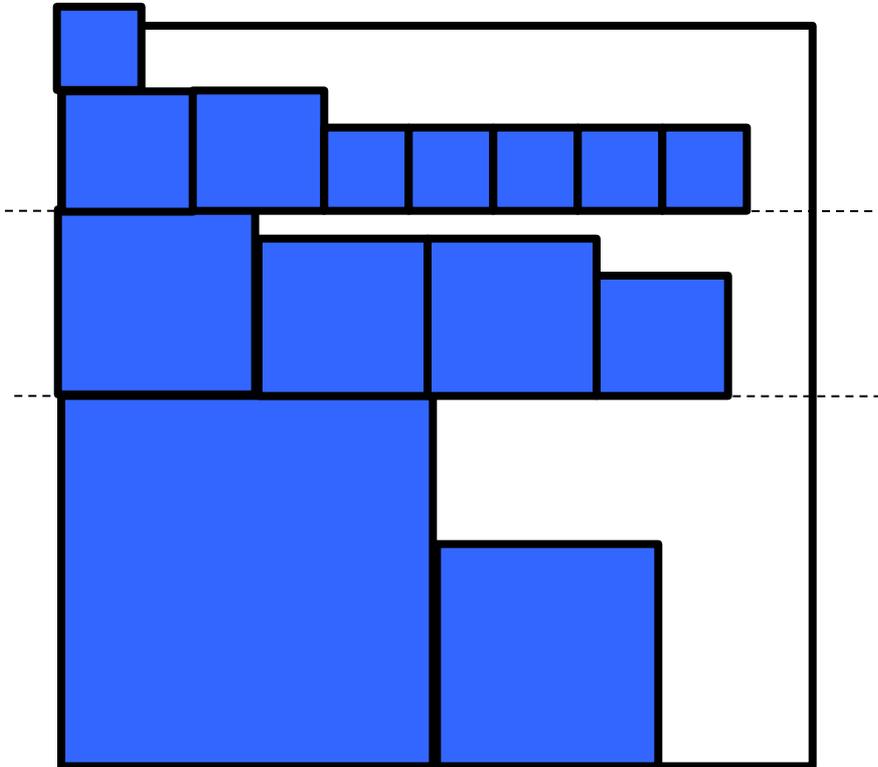
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate



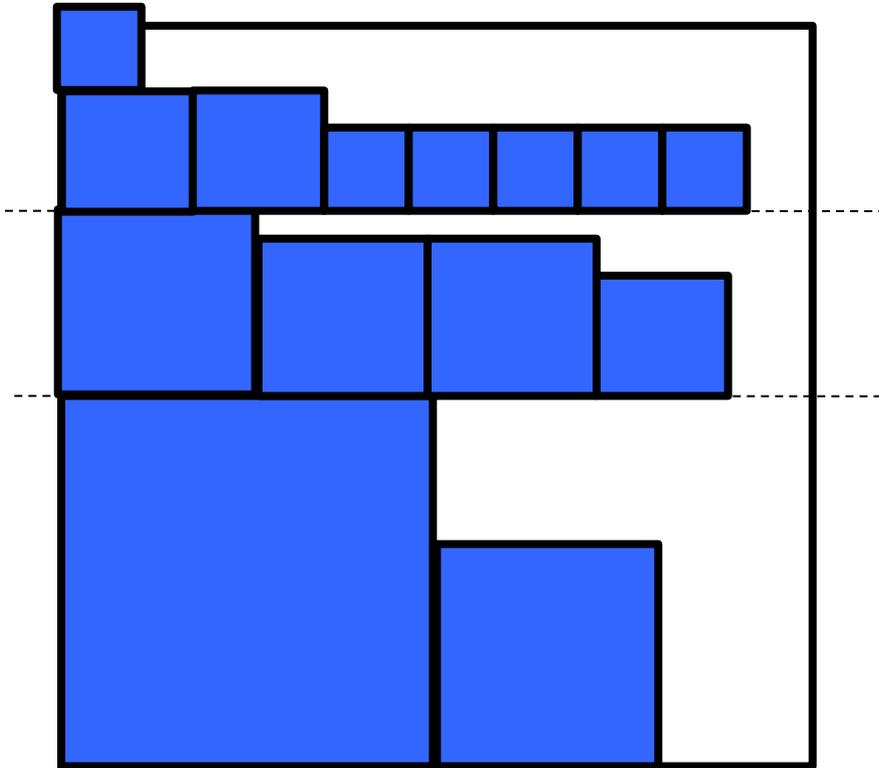
Satz:

Jede Menge von Quadraten mit Gesamtfläche höchstens 0.5 lässt sich in ein Einheitsquadrat packen.

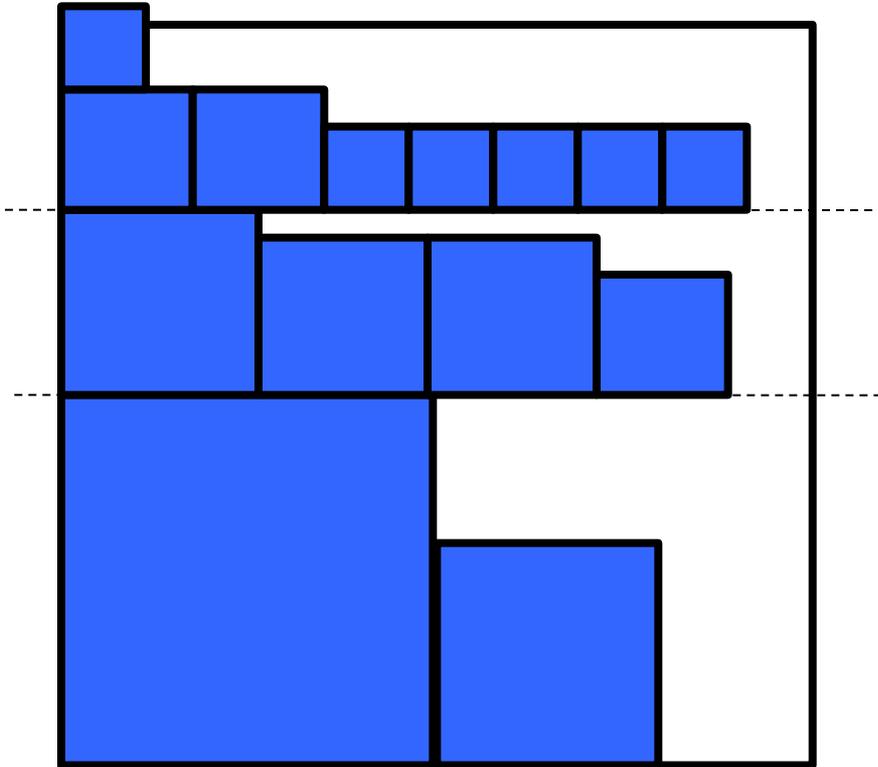
Beweisidee:

- (1) Sortiere die Quadrate nach absteigender Größe.
- (2) Verwende Shelf Packing.
- (3) Betrachte erste unzulässige Packung.

Untere Schranke für Quadrate

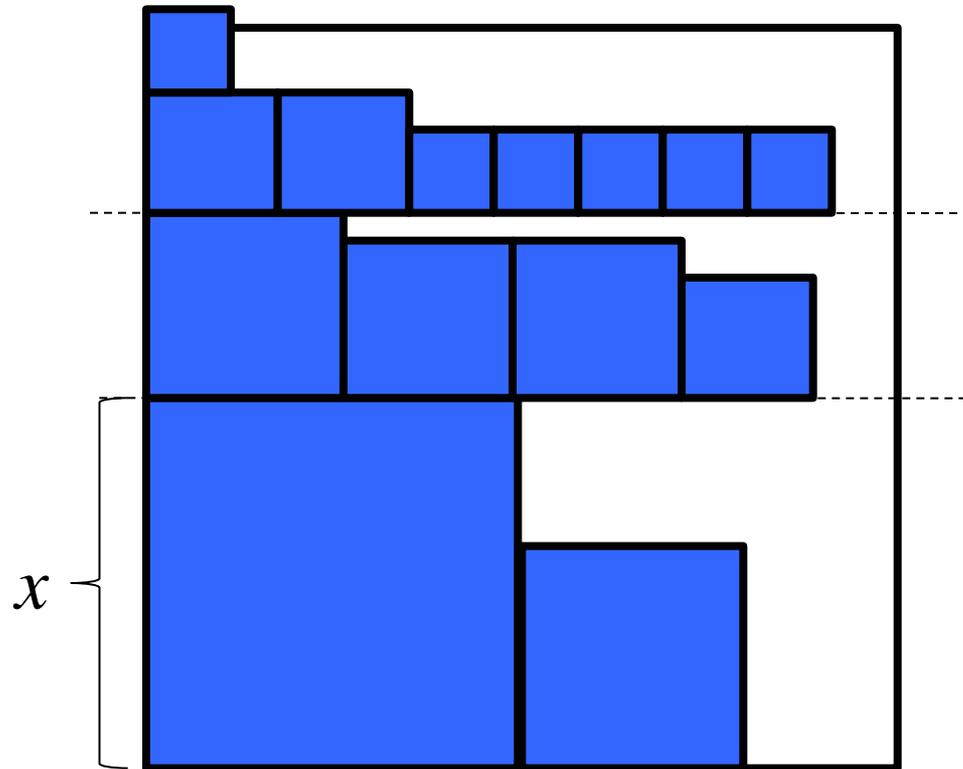


Untere Schranke für Quadrate



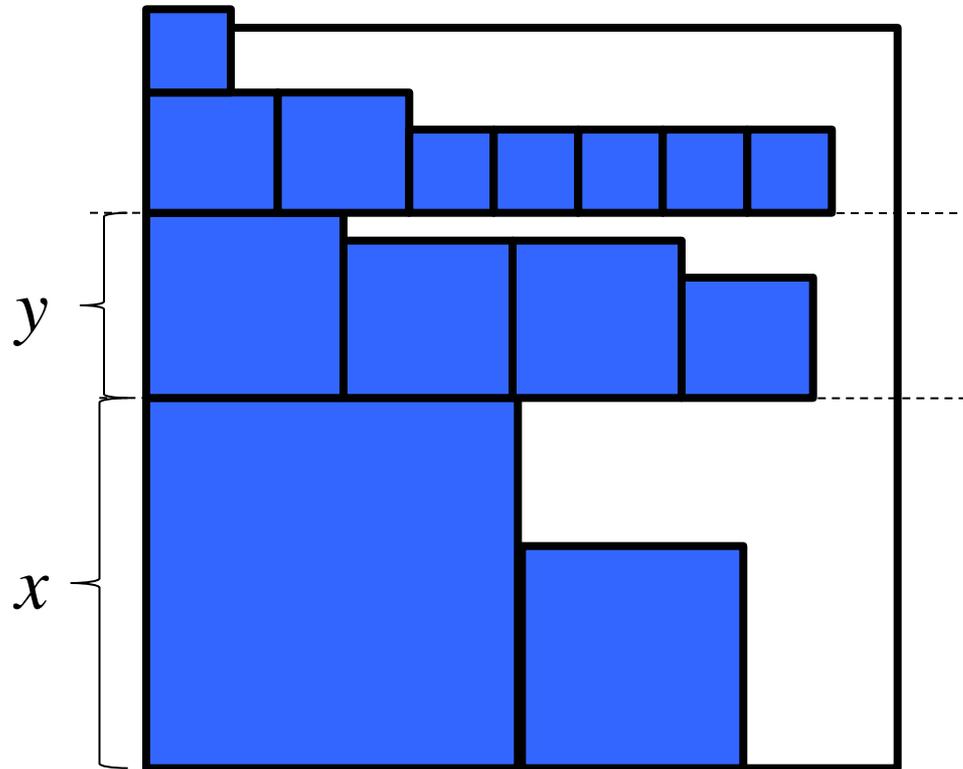
**Bilanziere gepackte
Fläche:**

Untere Schranke für Quadrate



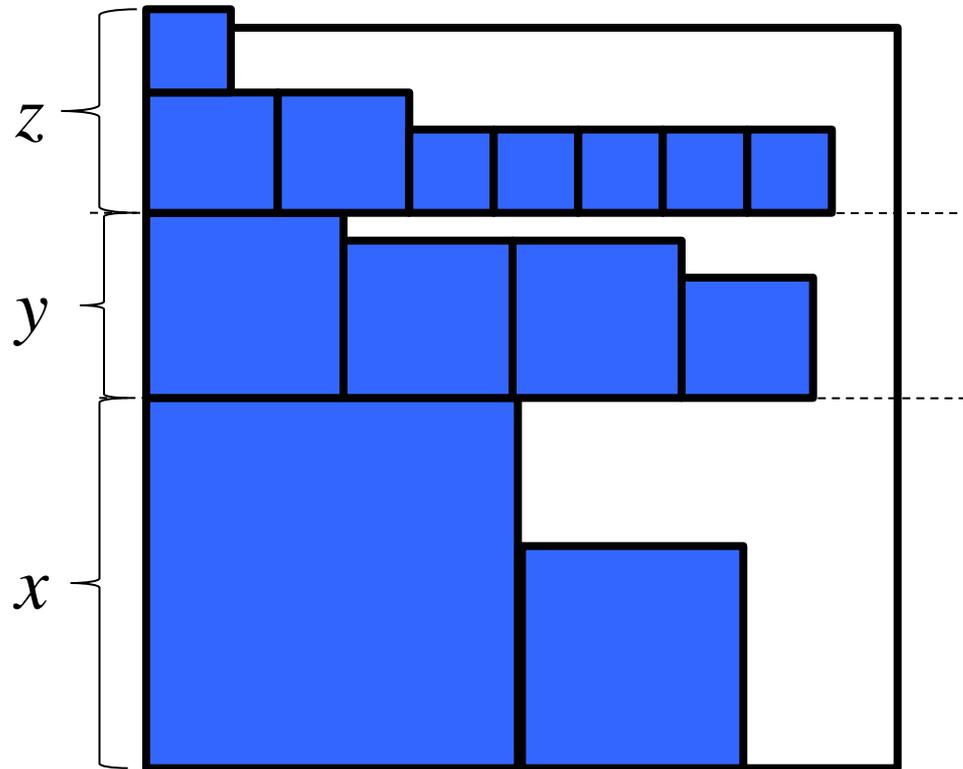
Bilanziere gepackte
Fläche:

Untere Schranke für Quadrate



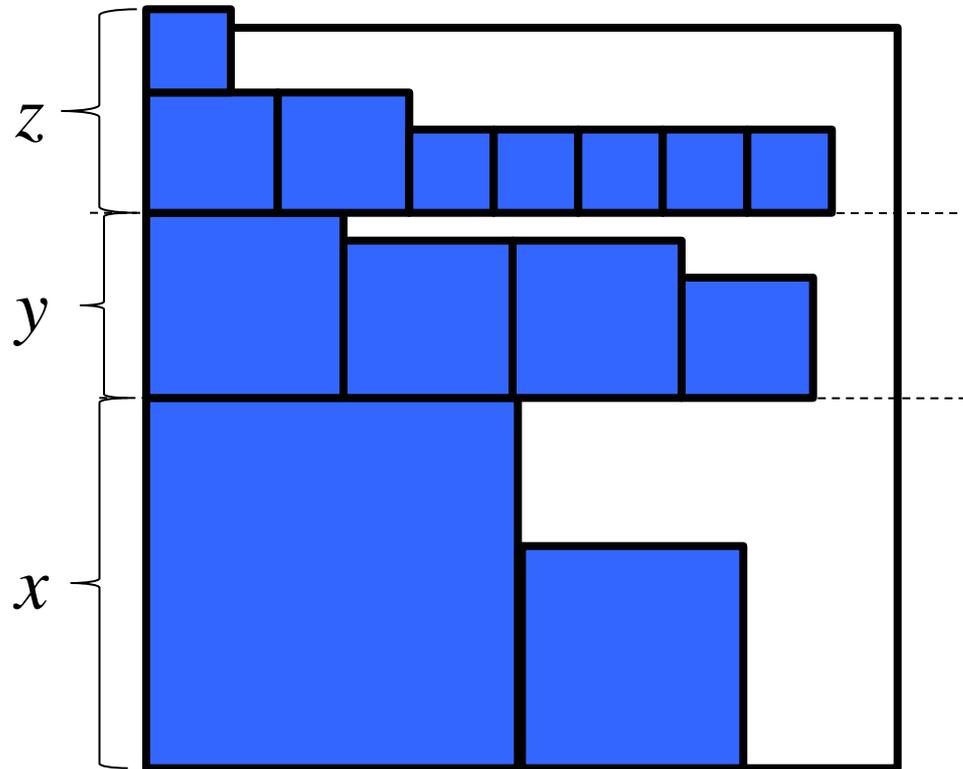
Bilanzieren gepackte
Fläche:

Untere Schranke für Quadrate



Bilanziere gepackte
Fläche:

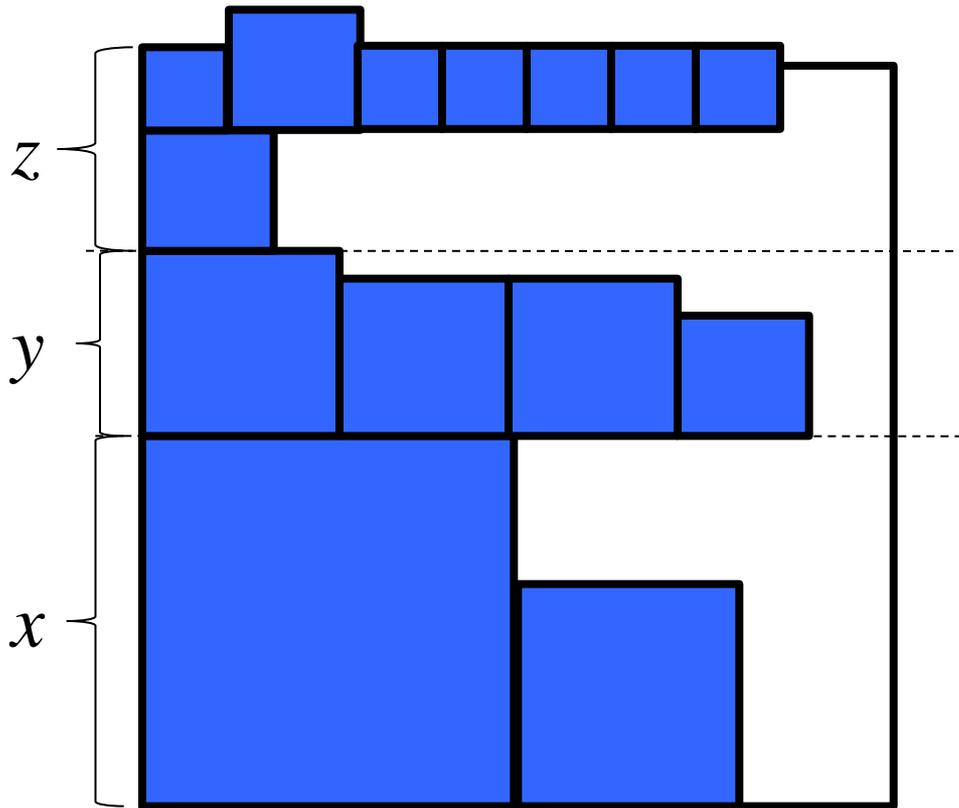
Untere Schranke für Quadrate



Bilanziere gepackte
Fläche:

$$x + y + z > 1$$

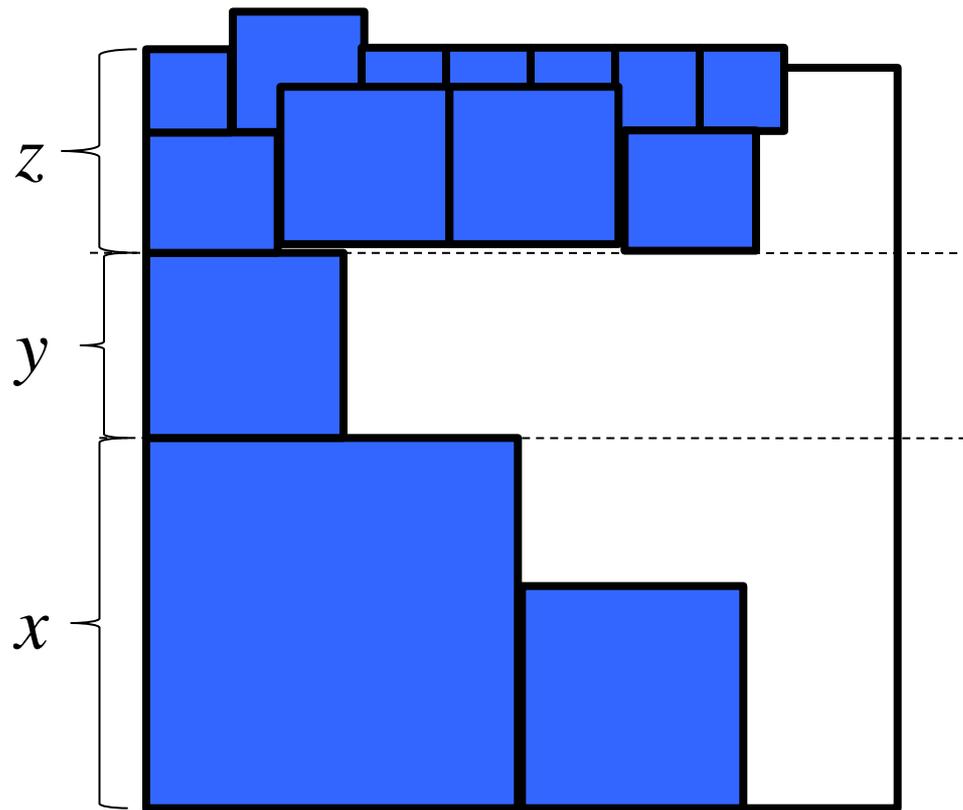
Untere Schranke für Quadrate



Bilanzieren gepackte
Fläche:

$$x + y + z > 1$$

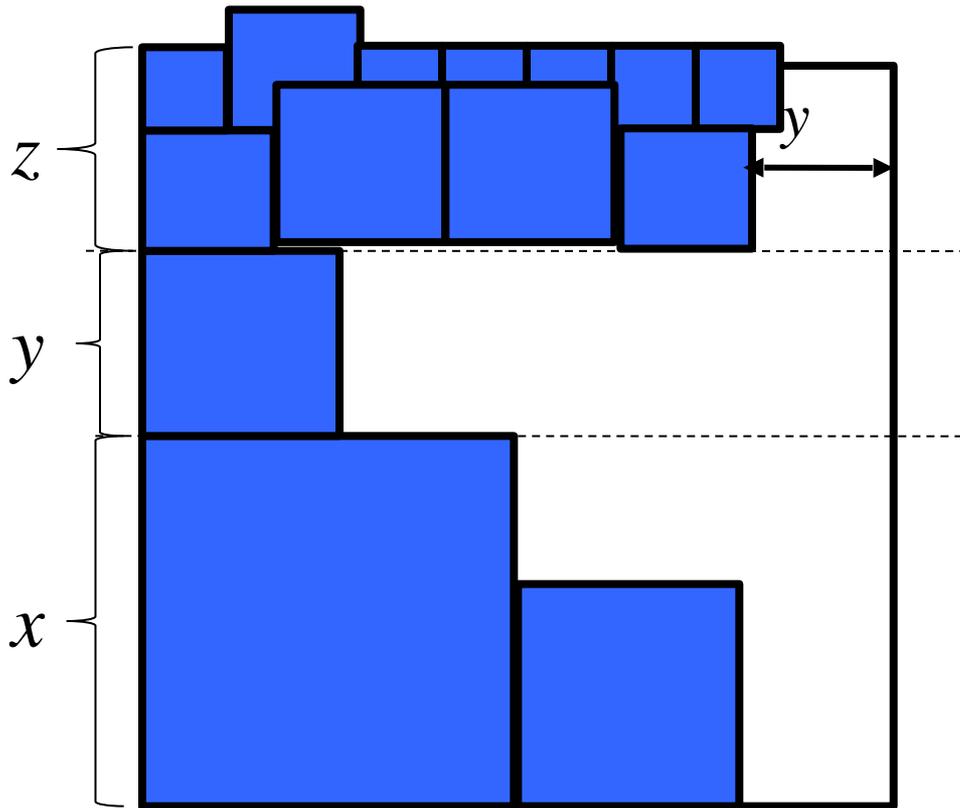
Untere Schranke für Quadrate



Bilanziere gepackte
Fläche:

$$x + y + z > 1$$

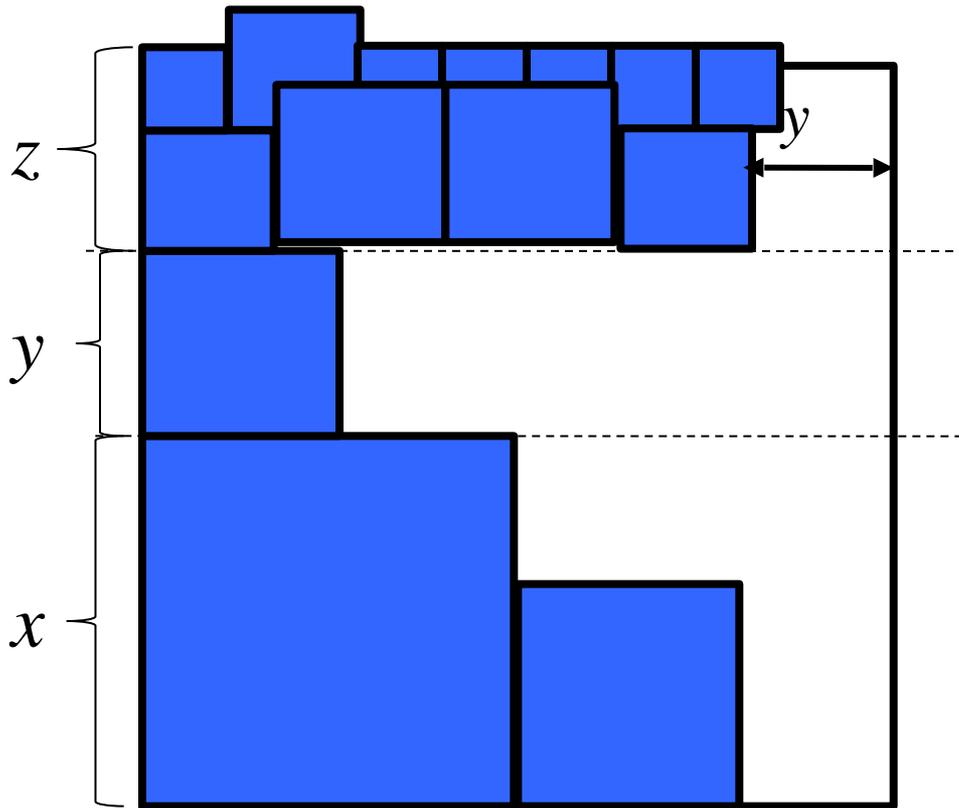
Untere Schranke für Quadrate



Bilanziere gepackte
Fläche:

$$x + y + z > 1$$

Untere Schranke für Quadrate

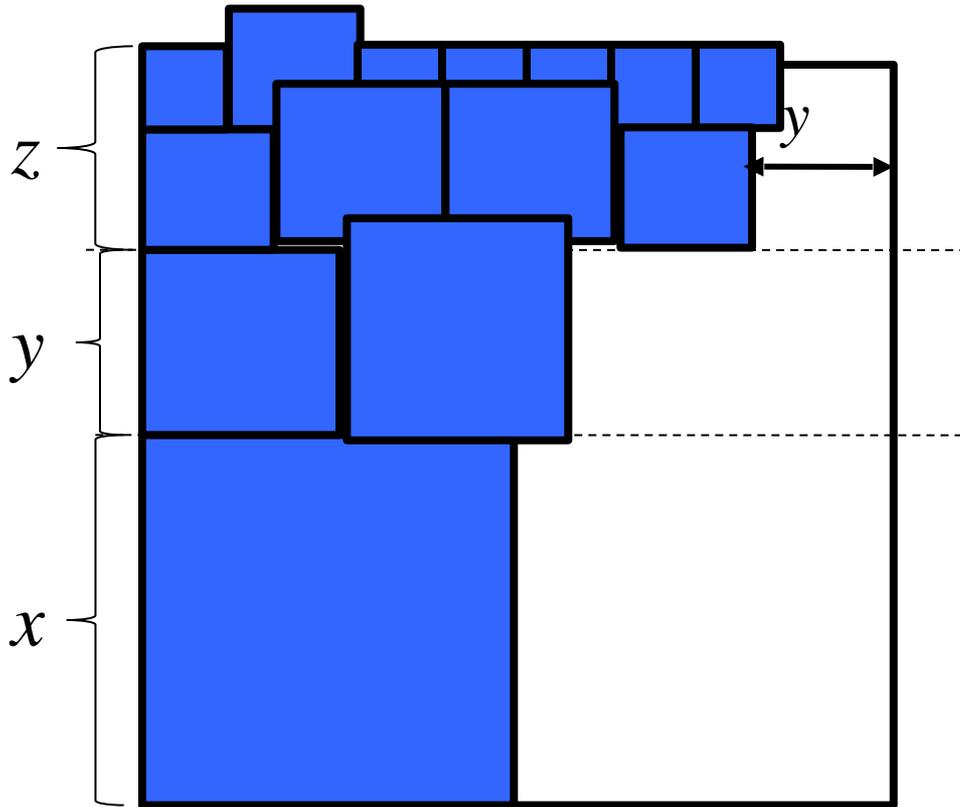


Bilanziere gepackte Fläche:

$$x + y + z > 1$$

$$F > z(1 - y)$$

Untere Schranke für Quadrate

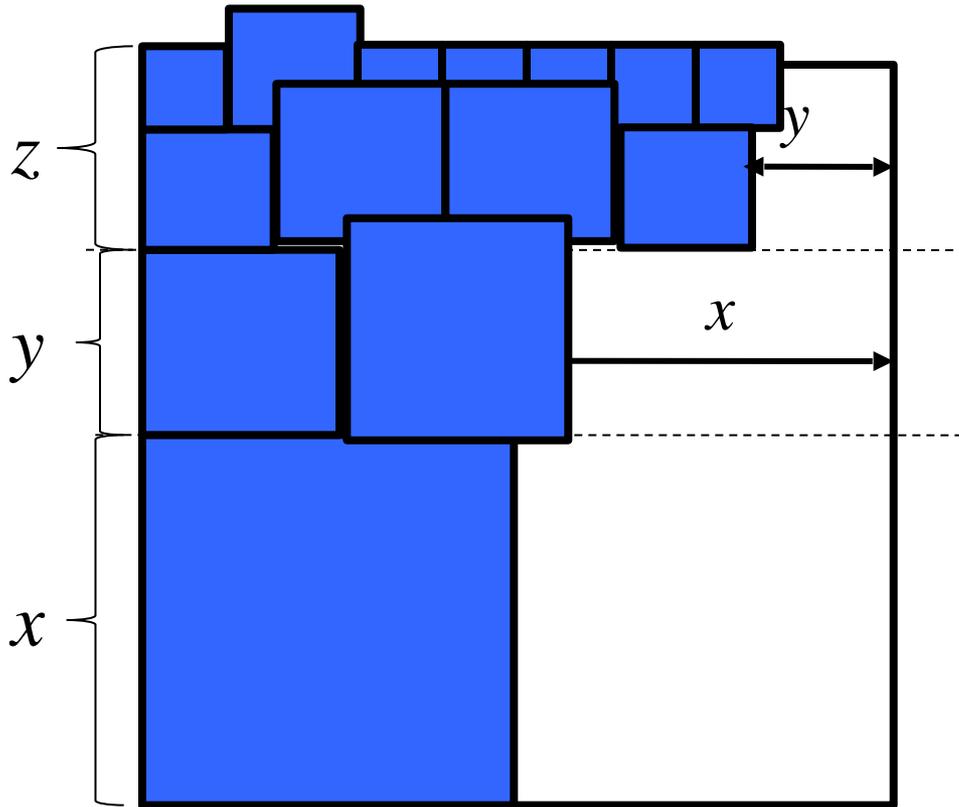


**Bilanziere gepackte
Fläche:**

$$x + y + z > 1$$

$$F > z(1 - y)$$

Untere Schranke für Quadrate

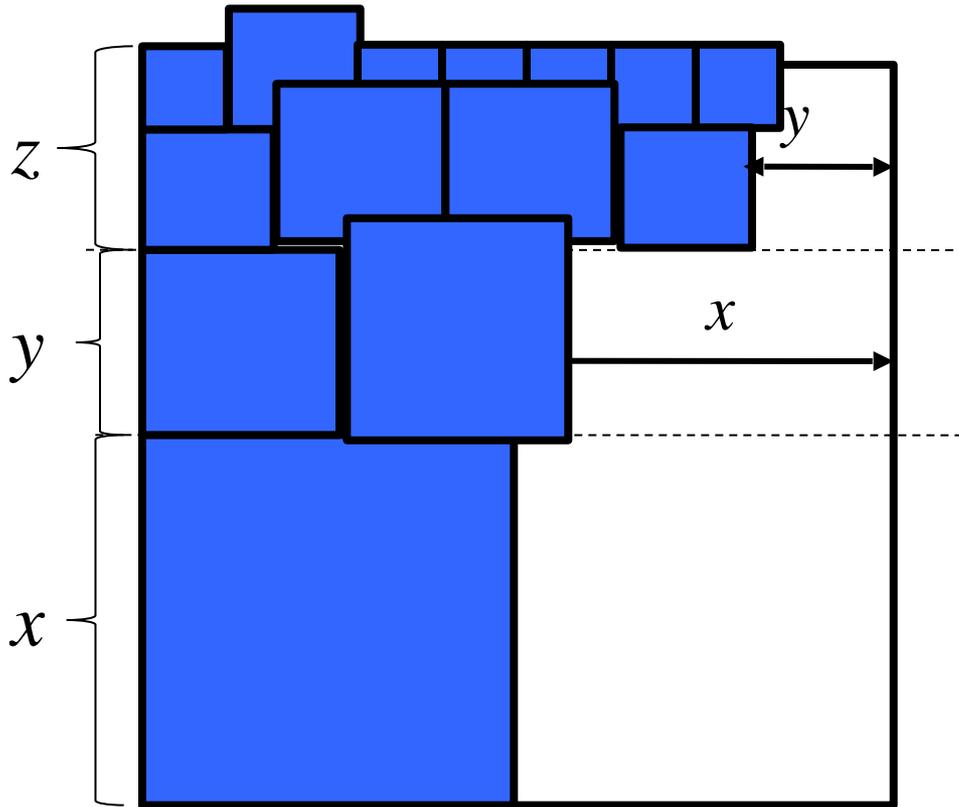


Bilanziere gepackte
Fläche:

$$x + y + z > 1$$

$$F > z(1 - y)$$

Untere Schranke für Quadrate

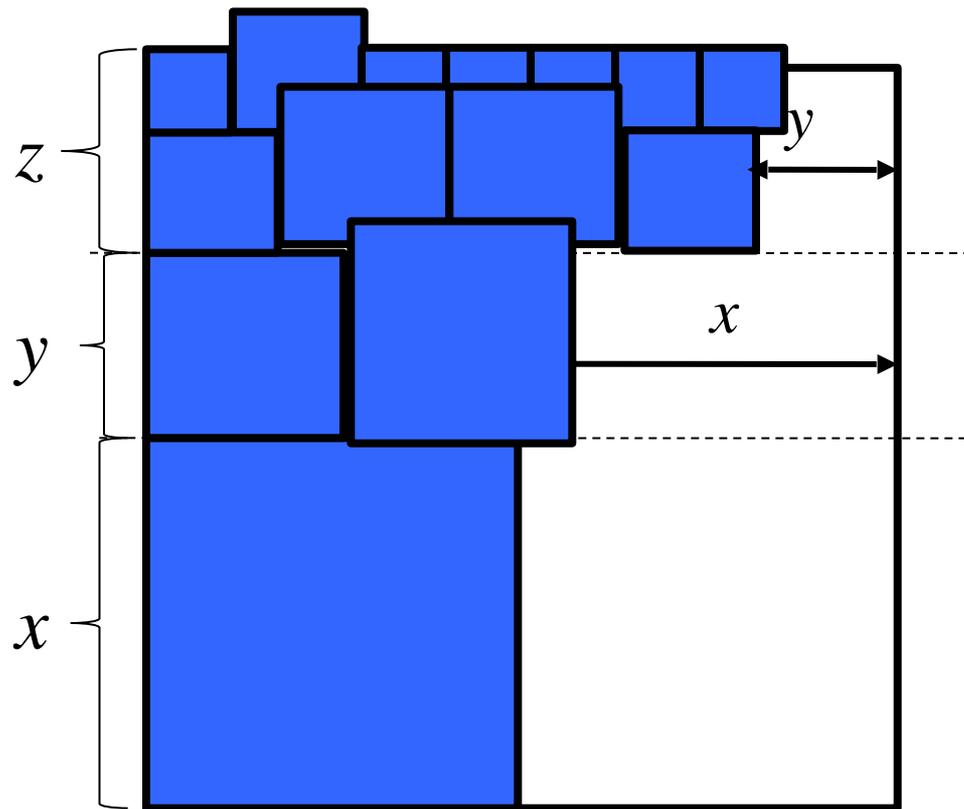


**Bilanziere gepackte
Fläche:**

$$x + y + z > 1$$

$$F > z(1 - y) \\ + y(1 - x)$$

Untere Schranke für Quadrate

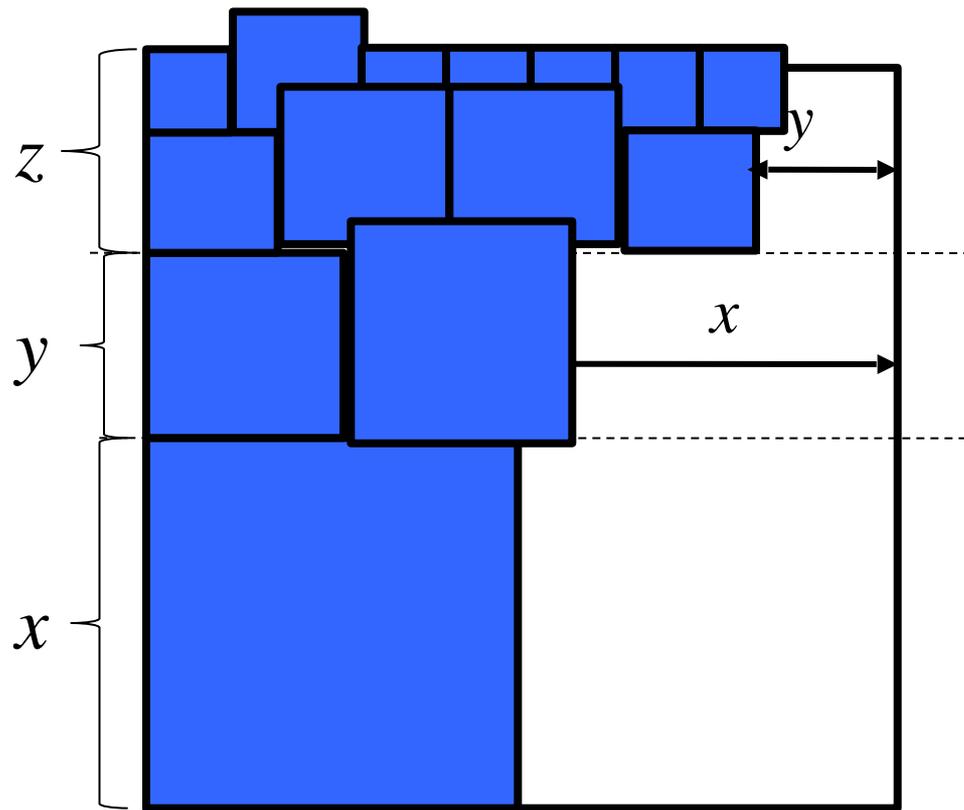


Bilanziere gepackte
Fläche:

$$x + y + z > 1$$

$$F > z(1 - y) \\ + y(1 - x) \\ + x^2$$

Untere Schranke für Quadrate



Bilanziere gepackte
Fläche:

$$x + y + z > 1$$

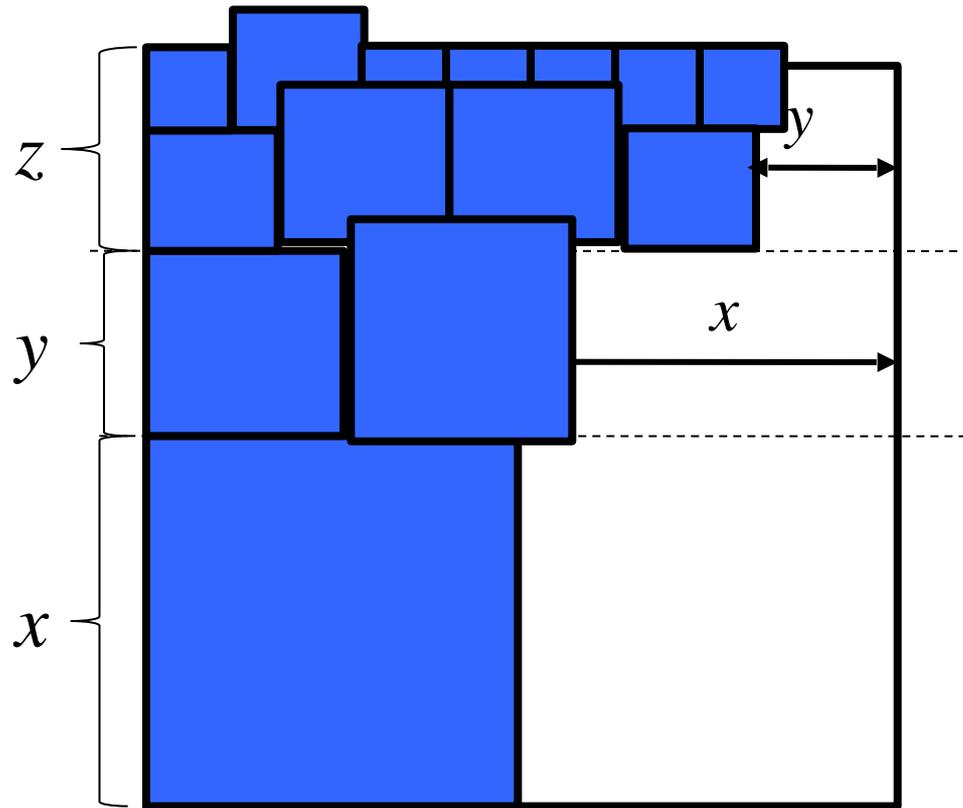
$$F > z(1 - y)$$

$$+ y(1 - x)$$

$$+ x^2$$

$$> (1 - x - y)(1 - y) + y(1 - x) + x^2$$

Untere Schranke für Quadrate



**Bilanziere gepackte
Fläche:**

$$x + y + z > 1$$

$$F > z(1 - y)$$

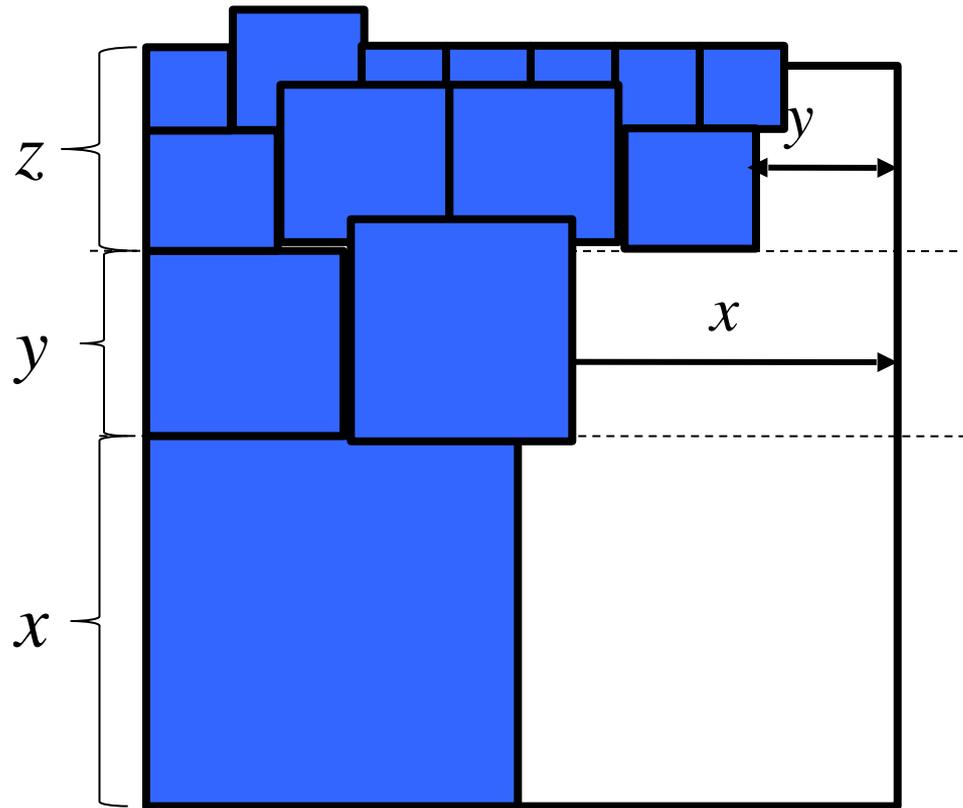
$$+ y(1 - x)$$

$$+ x^2$$

$$> (1 - x - y)(1 - y) + y(1 - x) + x^2$$

$$= 1 - x - 2y + xy + y^2 + y - xy + x^2$$

Untere Schranke für Quadrate



Bilanzierbare gepackte Fläche:

$$x + y + z > 1$$

$$F > z(1 - y)$$

$$+ y(1 - x)$$

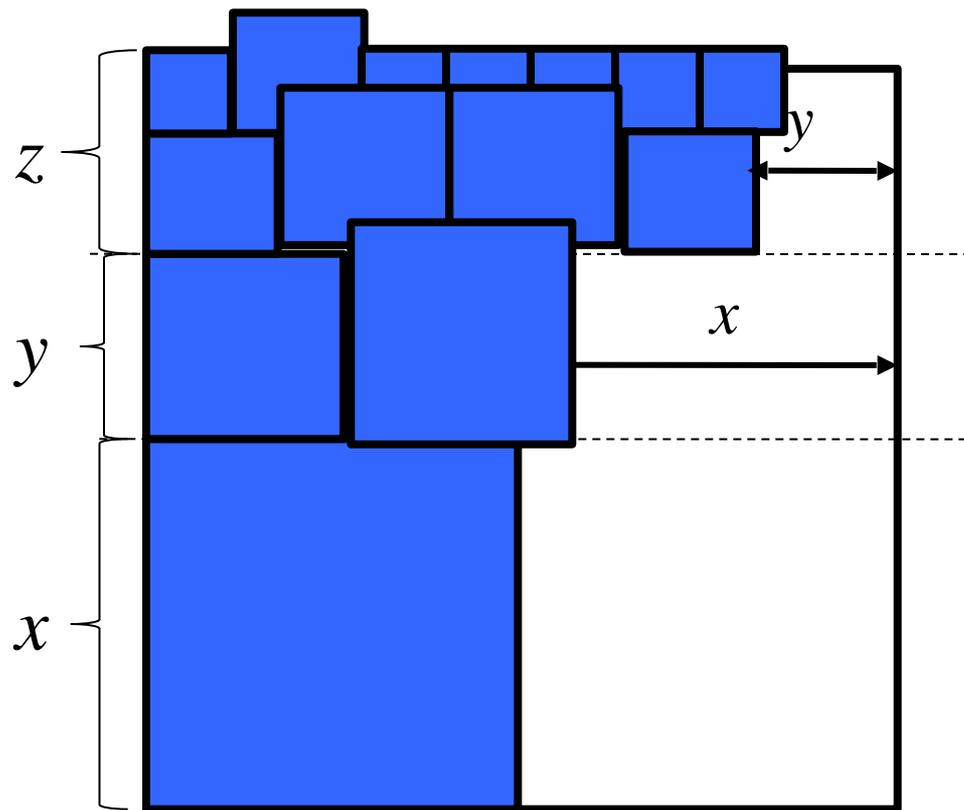
$$+ x^2$$

$$> (1 - x - y)(1 - y) + y(1 - x) + x^2$$

$$= 1 - x - 2y + xy + y^2 + y - xy + x^2$$

$$= \left(x^2 - x + \frac{1}{4}\right) + \left(y^2 - y + \frac{1}{4}\right) + \frac{1}{2}$$

Untere Schranke für Quadrate



Bilanziere gepackte Fläche:

$$x + y + z > 1$$

$$F > z(1 - y)$$

$$+ y(1 - x)$$

$$+ x^2$$

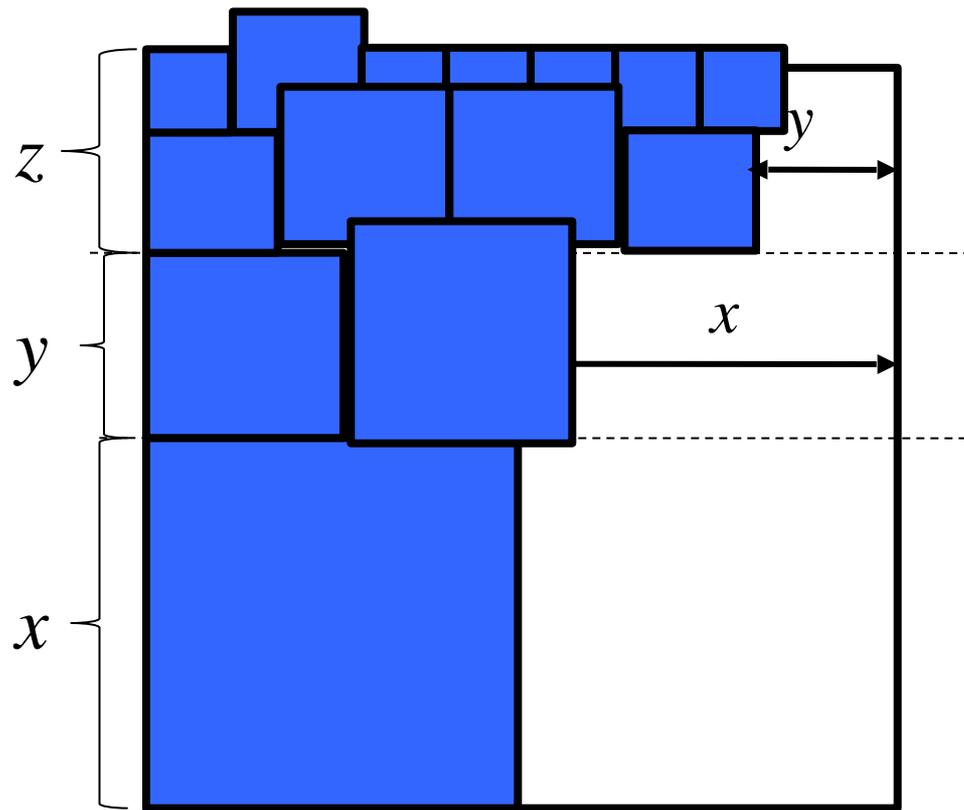
$$> (1 - x - y)(1 - y) + y(1 - x) + x^2$$

$$= 1 - x - 2y + xy + y^2 + y - xy + x^2$$

$$= (x^2 - x + \frac{1}{4}) + (y^2 - y + \frac{1}{4}) + \frac{1}{2}$$

$$= (x - \frac{1}{2})^2 + (y - \frac{1}{2})^2 + \frac{1}{2}$$

Untere Schranke für Quadrate



Bilanziere gepackte Fläche:

$$x + y + z > 1$$

$$F > z(1 - y)$$

$$+ y(1 - x)$$

$$+ x^2$$

$$> (1 - x - y)(1 - y) + y(1 - x) + x^2$$

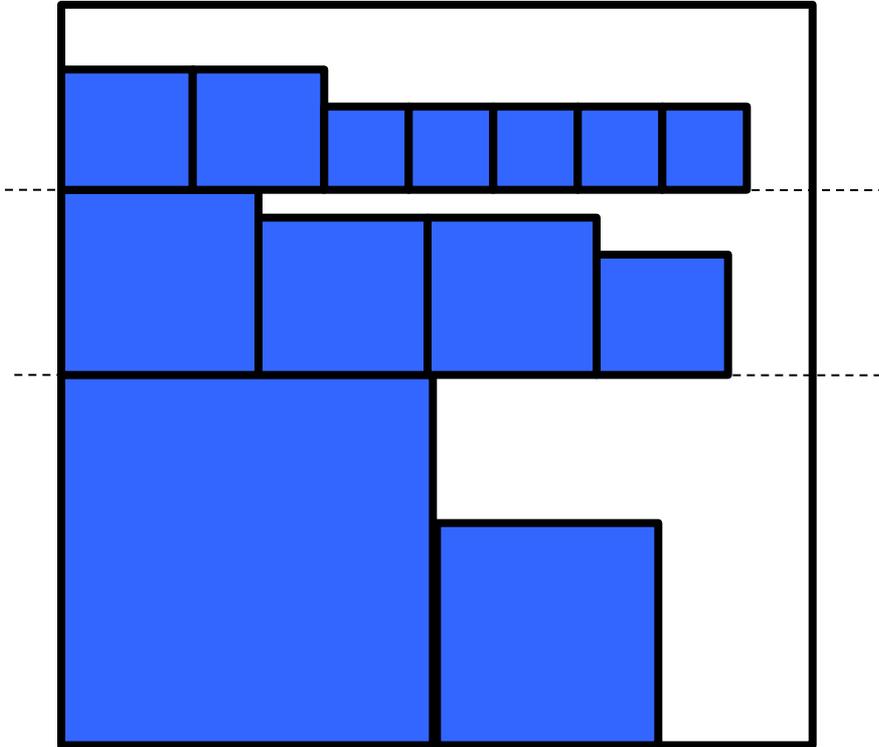
$$= 1 - x - 2y + xy + y^2 + y - xy + x^2$$

$$= \left(x^2 - x + \frac{1}{4}\right) + \left(y^2 - y + \frac{1}{4}\right) + \frac{1}{2}$$

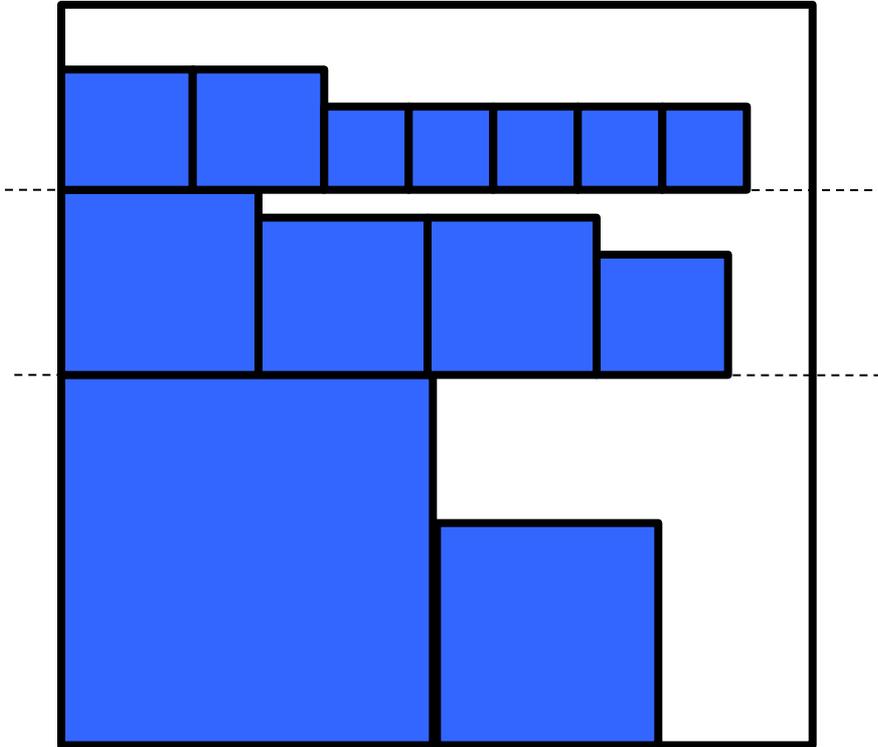
$$= \left(x - \frac{1}{2}\right)^2 + \left(y - \frac{1}{2}\right)^2 + \frac{1}{2}$$



Untere Schranke für Kreise

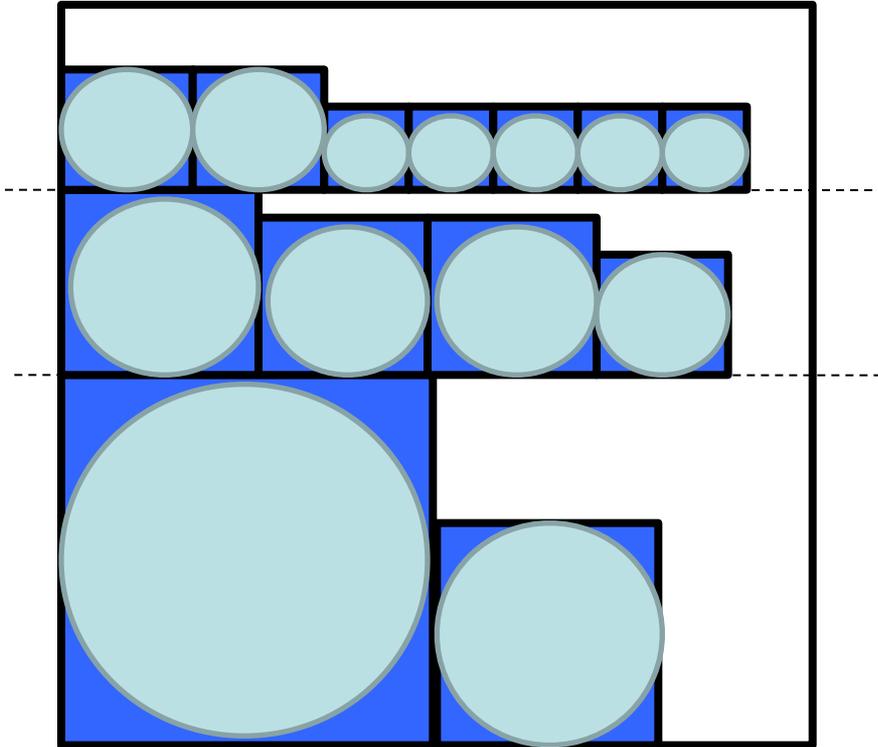


Untere Schranke für Kreise



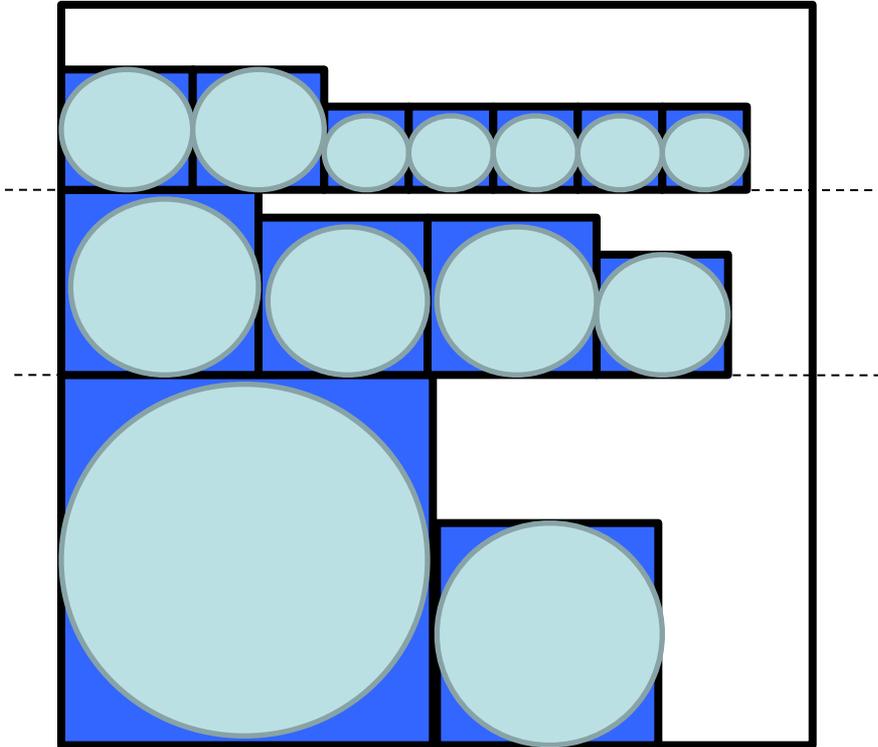
Korollar:

Untere Schranke für Kreise



Korollar:

Untere Schranke für Kreise



Korollar:

Jede Menge von Kreisen
mit Gesamtfläche
höchstens

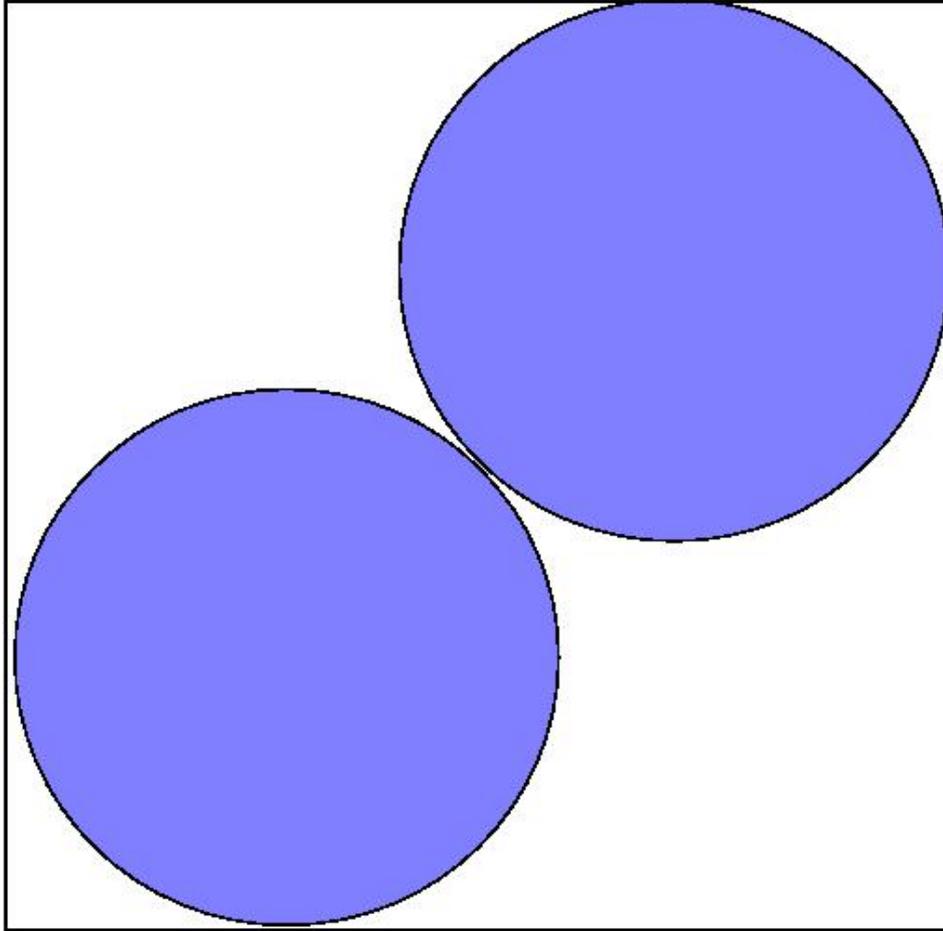
$$\frac{\pi}{8} = 0.392699\dots$$

lässt sich in ein
Einheitsquadrat packen.

Obere Schranke

Kritische Konfiguration
für Kreise, Dichte:

Obere Schranke



Kritische Konfiguration
für Kreise, Dichte:

Obere Schranke

Kritische Konfiguration
für Kreise, Dichte:

Obere Schranke

Kritische Konfiguration
für Kreise, Dichte:

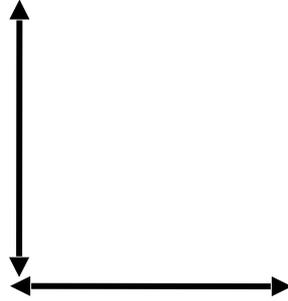


Obere Schranke

Kritische Konfiguration
für Kreise, Dichte:

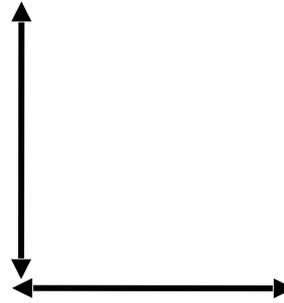
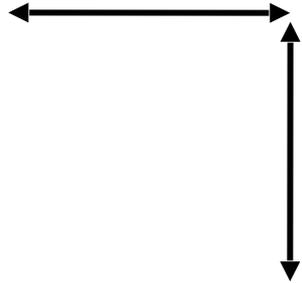


Obere Schranke



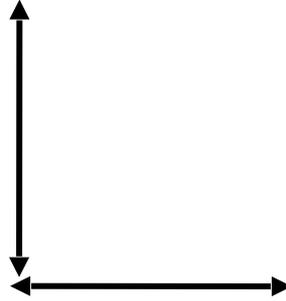
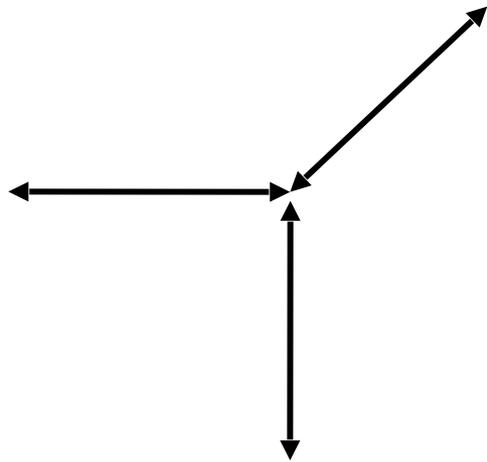
Kritische Konfiguration
für Kreise, Dichte:

Obere Schranke



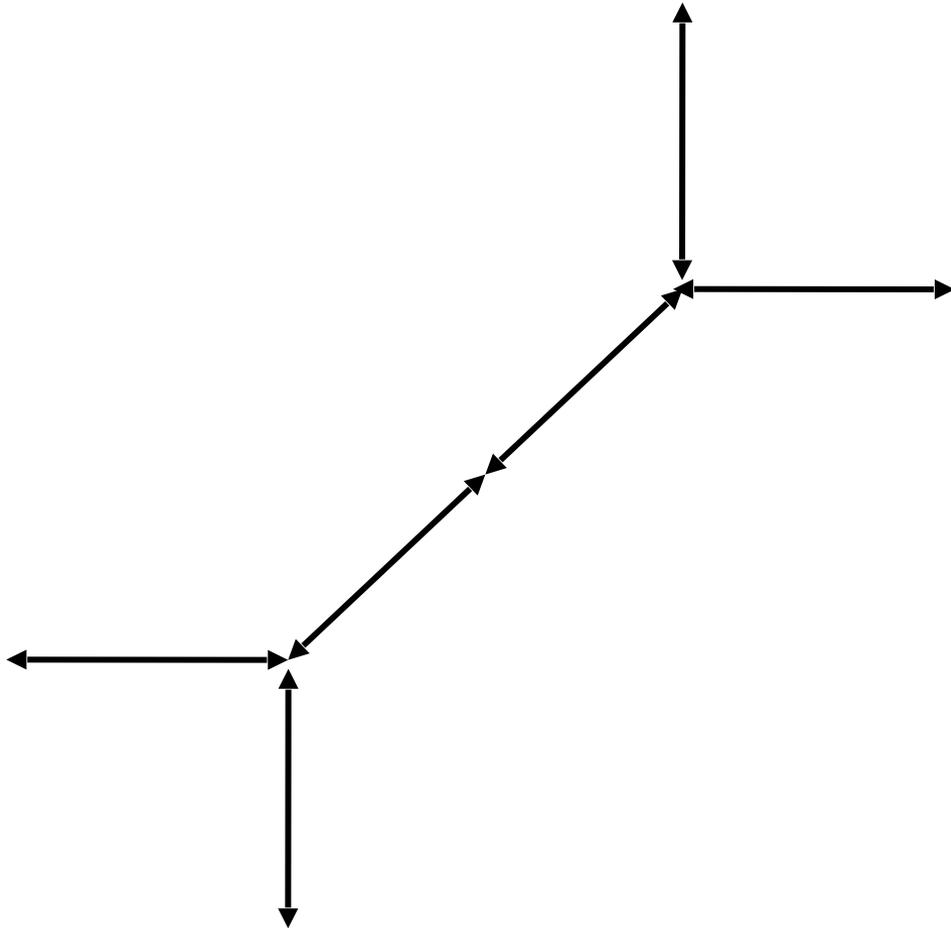
Kritische Konfiguration
für Kreise, Dichte:

Obere Schranke



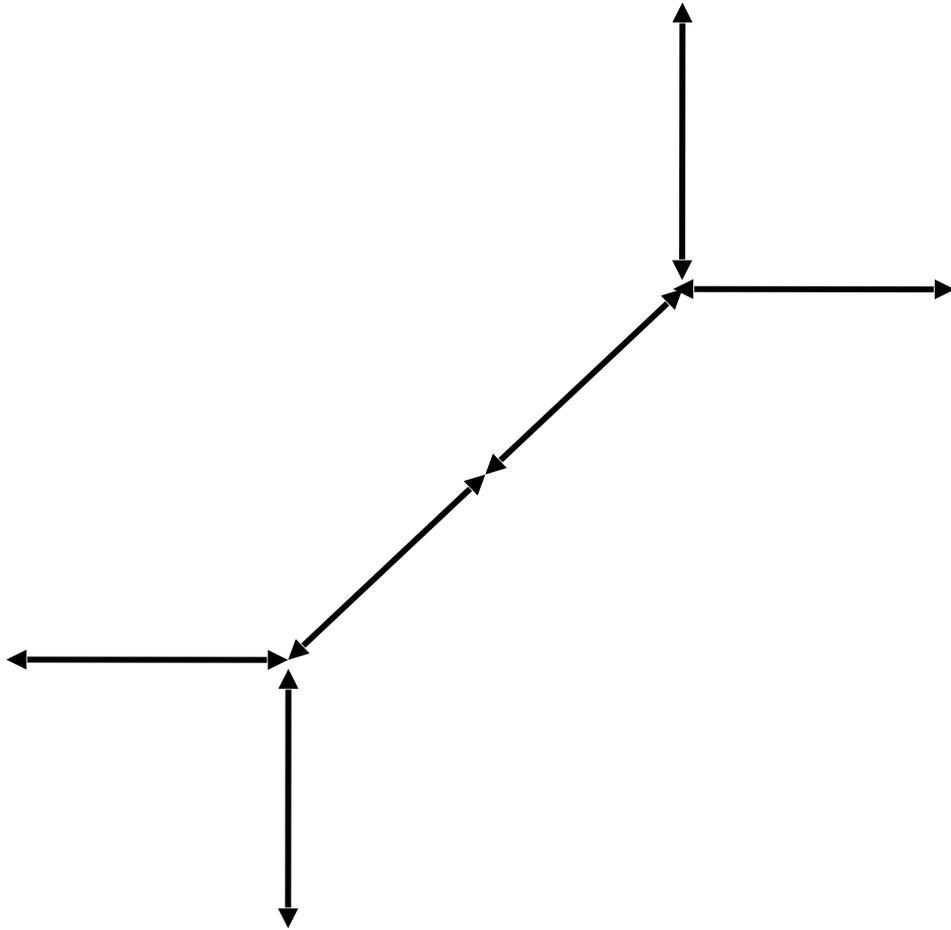
Kritische Konfiguration
für Kreise, Dichte:

Obere Schranke



Kritische Konfiguration
für Kreise, Dichte:

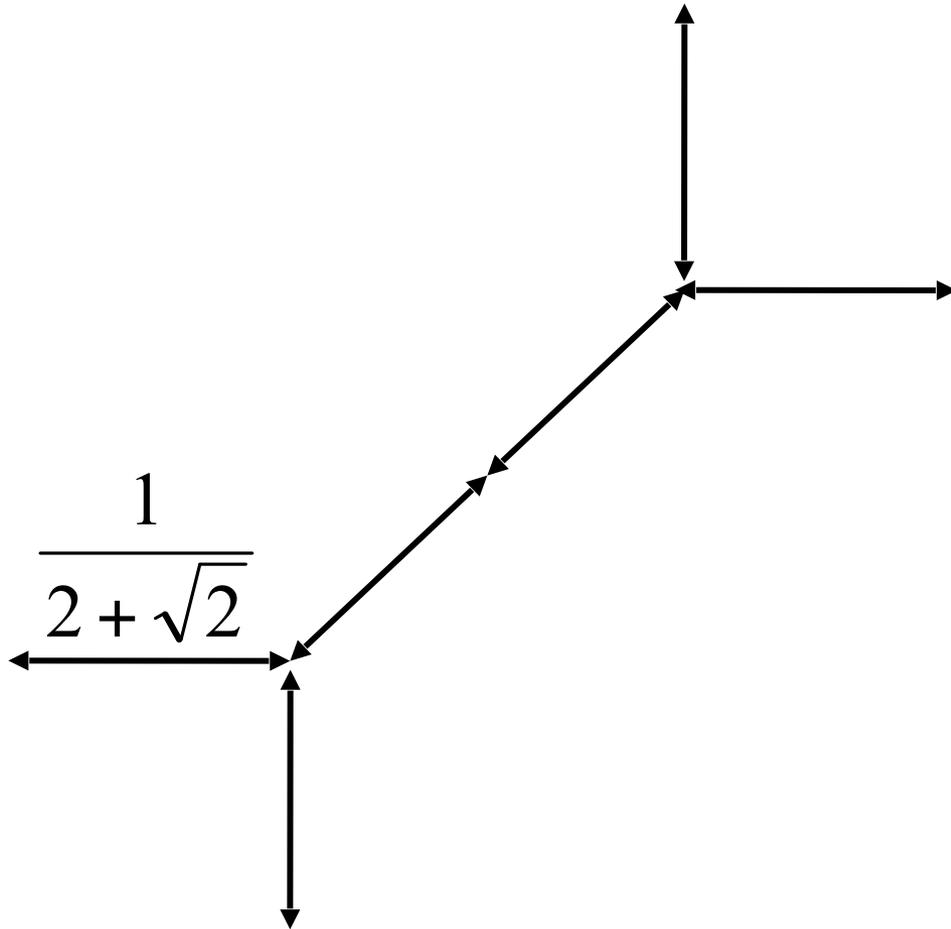
Obere Schranke



Kritische Konfiguration
für Kreise, Dichte:

$$\frac{2\pi}{6 + 4\sqrt{2}} = 0.53901\dots$$

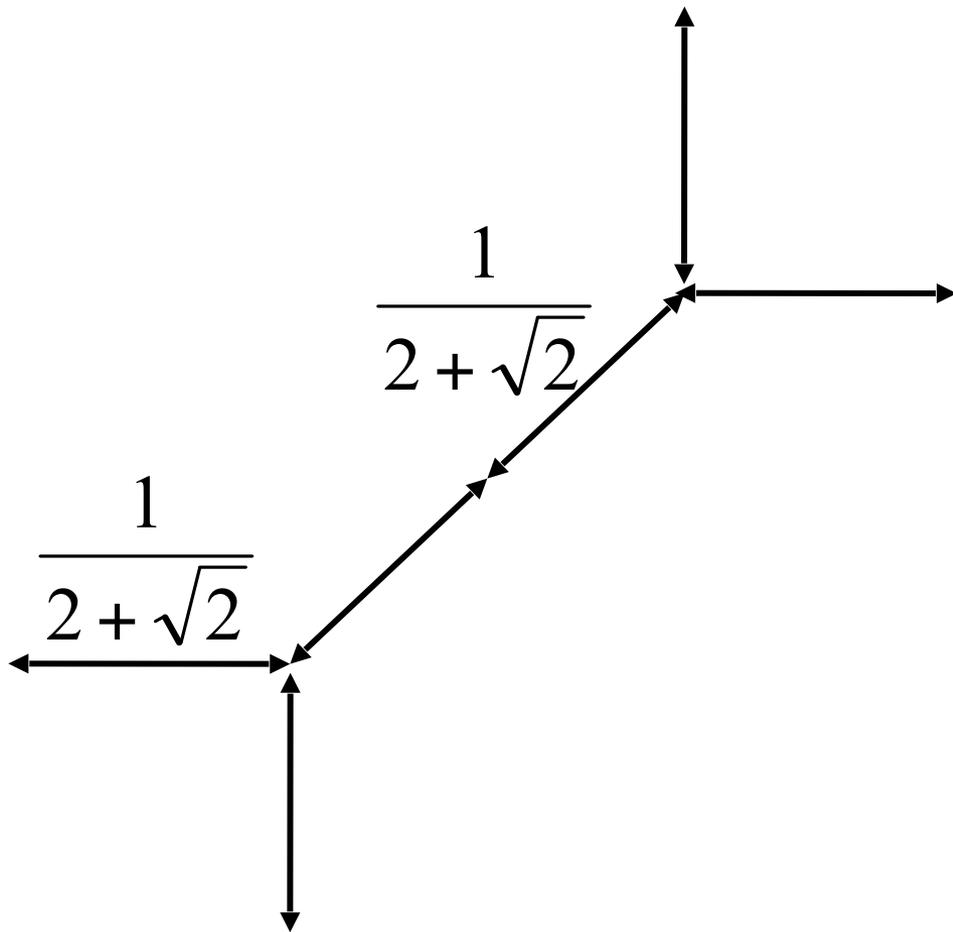
Obere Schranke



Kritische Konfiguration
für Kreise, Dichte:

$$\frac{2\pi}{6 + 4\sqrt{2}} = 0.53901\dots$$

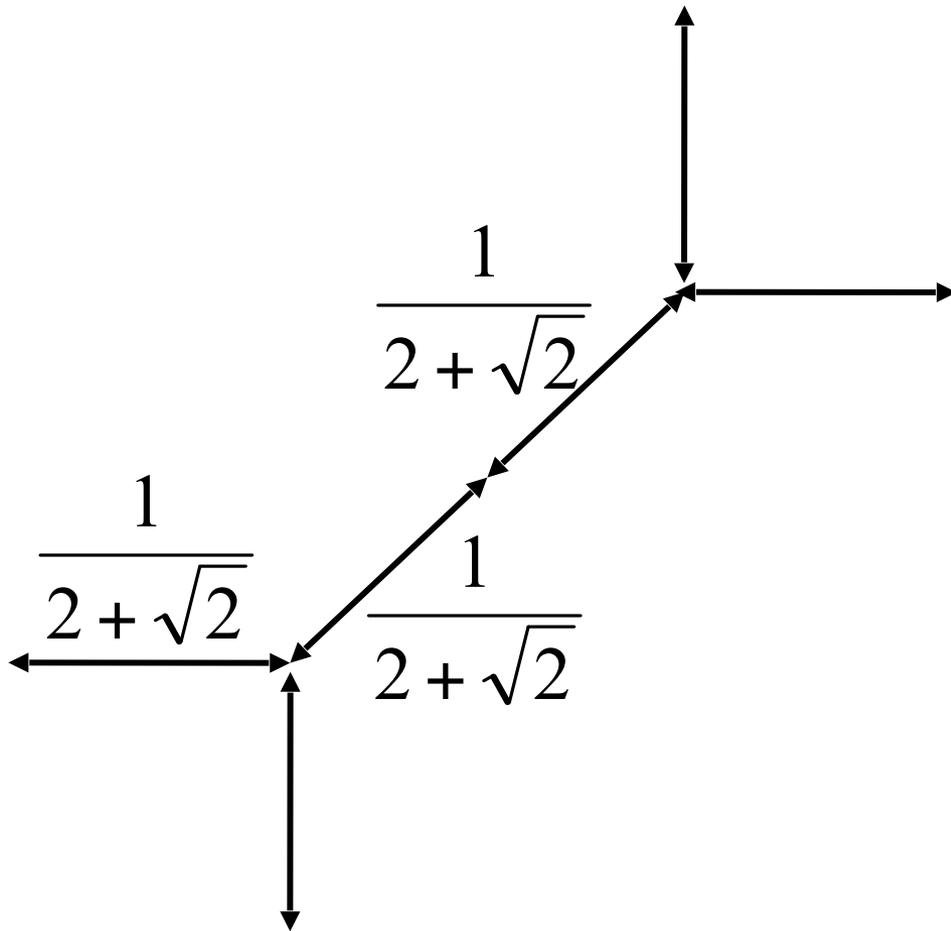
Obere Schranke



Kritische Konfiguration
für Kreise, Dichte:

$$\frac{2\pi}{6 + 4\sqrt{2}} = 0.53901\dots$$

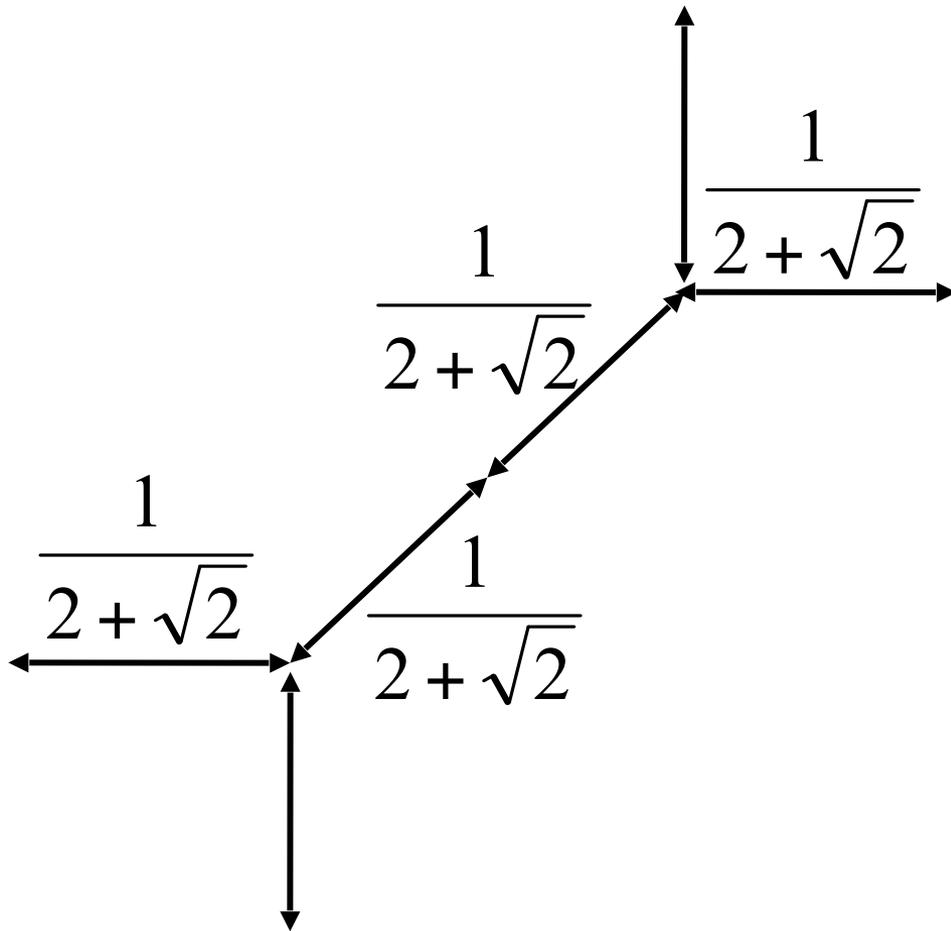
Obere Schranke



Kritische Konfiguration
für Kreise, Dichte:

$$\frac{2\pi}{6 + 4\sqrt{2}} = 0.53901\dots$$

Obere Schranke



Kritische Konfiguration
für Kreise, Dichte:

$$\frac{2\pi}{6 + 4\sqrt{2}} = 0.53901\dots$$

Master's Thesis

Split Packing: An Algorithm for Packing Circles with up to Critical Density

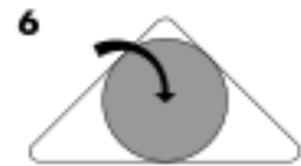
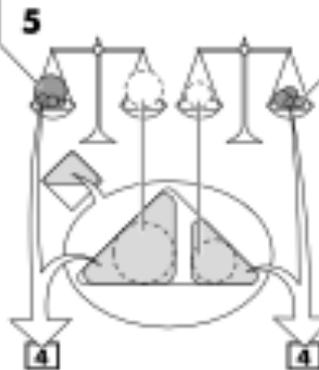
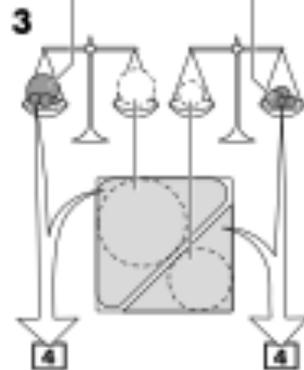
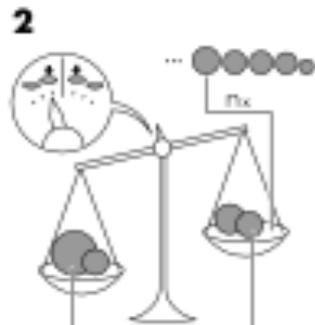
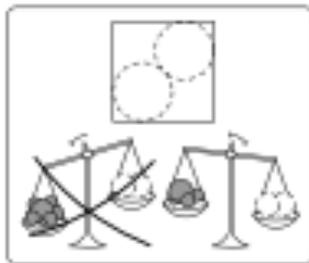
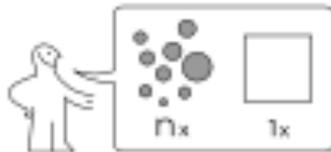
Sebastian Morr

2016-06-02

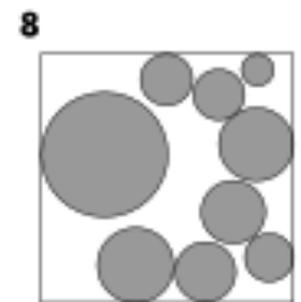
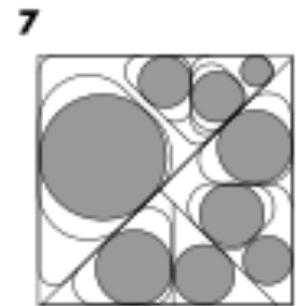
Institute of Operating Systems and Computer Networks

Supervisors:
Prof. Dr. Sándor Fekete
Dr. Christian Scheffer

SPLIT-PÄCK



$n \times 6$



<https://www.cs.cmu.edu/~sreth/packing/>

Online-Packen

Ergebnisse

Left arrow-move left
Right arrow-move right
Down arrow-drop fast
Up arrow-instant drop
Spacebar, X-rotate right
Z-rotate left

Next: 

Level: 4
Lines: 11
- 1

Press P to pause

Online Square Packing with Gravity

Sándor P. Fekete · Tom Kamphans · Nils Schweer

Received: 21 October 2010 / Accepted: 9 November 2012
© Springer Science+Business Media New York 2012

Abstract We analyze the problem of packing squares in an online fashion: Given a semi-infinite strip of width 1 and an unknown sequence of squares of side length in $[0, 1]$ that arrive from above, one at a time. The objective is to pack these items as they arrive, minimizing the resulting height. Just like in the classical game of Tetris, each square must be moved along a collision-free path to its final destination. In addition, we account for gravity in both motion (squares must never move up) and position (any final destination must be supported from below). A similar problem has been considered before; the best previous result is by Azar and Epstein, who gave a 4-competitive algorithm in a setting without gravity (i.e., with the possibility of letting squares “hang in the air”) based on ideas of shelf packing: Squares are assigned to different horizontal levels, allowing an analysis that is reminiscent of some bin-packing arguments. We apply a geometric analysis to establish a competitive factor of 3.5 for the bottom-left heuristic and present a $\frac{34}{13} \approx 2.6154$ -competitive algorithm.

Keywords Online packing · Strip packing · Squares · Gravity · Tetris

1 Introduction

1.1 Packing Problems

Packing problems arise in many different situations, either concrete (where actual physical objects have to be packed), or abstract (where the space is virtual, e.g., in

Tom Kamphans was supported by DFG grant FE 407/8-3, project “ReCoNoder”. A preliminary extended abstract summarizing the results of this paper appeared in [15].

S.P. Fekete (✉) · T. Kamphans · N. Schweer
Department of Computer Science, Algorithms Group, Braunschweig University of Technology,
Mühlentorstrasse 23, 38106 Braunschweig, Germany
e-mail: s.fekete@tu-bs.de

Published online: 16 November 2012

 Springer

Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund

Master Thesis

Online Packing Into Squares

by

Hella-Franziska Hoffmann

Supervisor:

Prof. Dr. Sándor P. Fekete

Online Square-into-Square Packing

Sándor P. Fekete¹ and Hella-Franziska Hoffmann²

¹ Department of Computer Science, TU Braunschweig, Germany
s.fekete@tu-bs.de

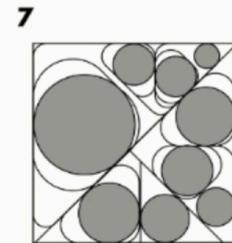
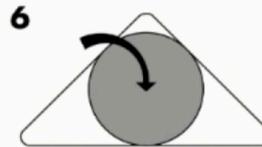
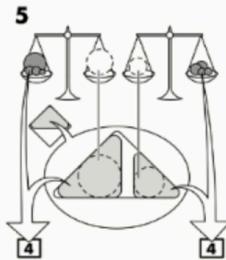
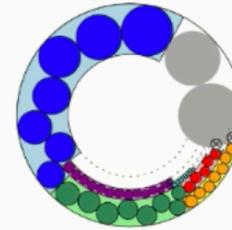
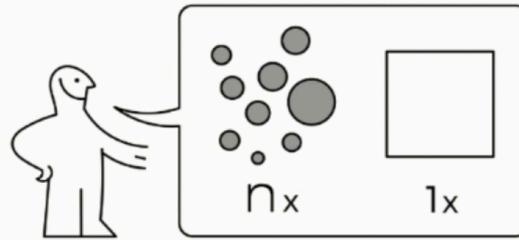
² Cheriton School of Computer Science, University of Waterloo, Canada
hrhoffma@uwaterloo.ca

Abstract. In 1967, Moon and Moser proved a tight bound on the critical density of squares in squares: any set of squares with a total area of at most $1/2$ can be packed into a unit square, which is tight. The proof requires full knowledge of the set, as the algorithmic solution consists in sorting the objects by decreasing size, and packing them greedily into shelves. Since then, the online version of the problem has remained open; the best upper bound is still $1/2$, while the currently best lower bound is $1/3$, due to Han et al. (2008). In this paper, we present a new lower bound of $11/32$, based on a dynamic shelf allocation scheme, which may be interesting in itself.

We also give results for the closely related problem in which the size of the square container is not fixed, but must be dynamically increased in order to accommodate online sequences of objects. For this variant, we establish an upper bound of $3/7$ for the critical density, and a lower bound of $1/8$. When aiming for accommodating an online sequence of squares, this corresponds to a $2.82\dots$ -competitive method for minimizing the required container size, and a lower bound of $1.33\dots$ for the achievable factor.

Keywords: Packing, online problems, packing squares, critical density.

Ergebnisse



Packing Geometric Objects with Optimal Worst-Case Density

Aaron Becker, Sándor P. Fekete, Phillip Keldenich,
Sebastian Morr, Christian Scheffer

Vielen Dank!

Vielen Dank!

