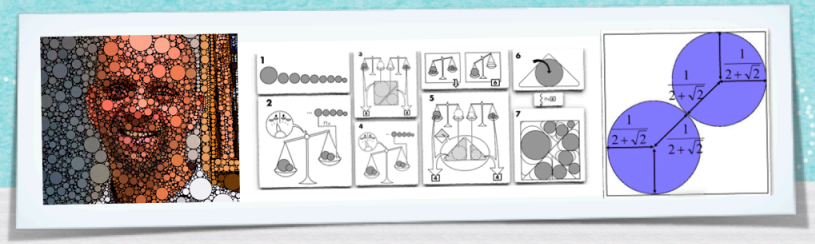


6 Mehrdimensionales Packen

Algorithmen und Datenstrukturen 2
Sommer 2024

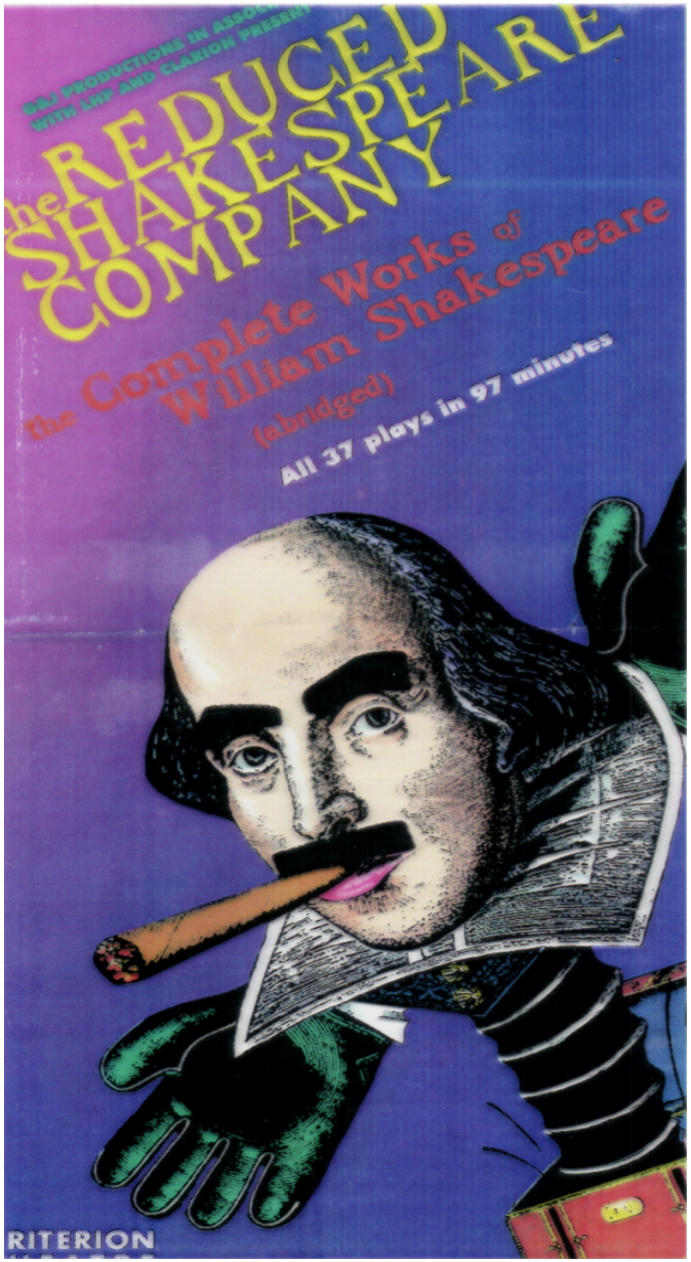
Prof. Dr. Sándor Fekete

Autos, Quader, Rechtecke

Reisezeit!

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

Wieviel passt in eine Vorlesung?



Wieviel passt in einen Kofferraum?

TEST KOFFERRAUM-VERGLEICH

VW Touran

Audi Q7

VAN GEGEN SUV

Touran fasst mehr als Q7

134 Liter zu wenig

Werksangabe: 695 Liter

Gemessen: 561 Liter

VW Touran

234 Liter zu wenig

Werksangabe: 775 Liter

Gemessen: 541 Liter

Audi Q7

Wenn das keine Überraschung ist: Audis 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 Litern. Wird seine Rückbank vorgeschoben, schluckt er sogar 716 Liter – wieder mehr als der Q7 (650 Liter mit vorgerückten Sitzen). Ein echter Raumriese, der Touran. Ganz anders der Audi Q7: Bei der Innenraumausnutzung enttäuscht der XXL-SUV. Das Werk nennt ein Volumen von 775 Litern hinter den Rücksitzen. Klingt 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 verschiebbare Rückbank eröffnet der Q7 mehrere Beladungsmöglichkeiten. Wenn das Heckrollo entfernt ist, nähert sich der Q7 seiner Prospektangabe und er erreicht immerhin 732 Liter. Bei dachhoher Beladung sind es bis zu 1070 Liter.

So sehen Testsieger aus: quadratisch, praktisch, gut. Als Lademeister erfüllt der Touran vorzügliche Dienste. Ob großes Reisegepäck oder Sportgeräte, er schluckt jede Menge Ladegut

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

CABRIO GEGEN SCHRÄGHECK

TwinTop verlädt Corsa

Die Preisliste des Tigra TwinTop prahlt mit 440 Liter Volumen bei geschlossenem Dach. Stark übertrieben: Tatsächlich passen nur 372 Liter rein. Opel hat sich nach eigener Aussage auf Computerdaten verlassen und das AUTO BILD-Ergebnis bestätigt. Bei echter Beladung von Hand schafften auch die Opel-Experten nur 381 Liter, wobei davon sieben Liter auf die Reserveradmulde 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 Rücksitzlehne 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

Opel Tigra Twintop

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-Nutzbarkeit ein. Seine Werksangabe verfehlt er klar

Opel Corsa

Werksangabe: 285 Liter

Gemessen: 238 Liter

47 Liter zu wenig

C-Klasse schluckt mehr als 3er-Touring

Dass der Lifestylelaster von BMW kein Raumwunder ist, war zu erwarten. Doch dass er nur 370 Liter einladen kann, statt der versprochenen 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 eine klassische Limousine besser. Beispiel: Mercedes C-Klasse. Zwar versprechen auch die Stuttgarter zu viel, aber mit 464 Litern liegt der Mercedes deutlich über dem BMW. 79 davon entfallen allerdings auf die Reserveradmulde.

464 Liter – 11 Liter weniger

Werksangabe: 475 Liter

Mercedes C-Klasse

370 Liter – 90 Liter weniger

Werksangabe: 460 Liter

BMW 3er-Touring

LIMO GEGEN KOMBI II

Mondeo größer als A6

Auch beim Vergleich Mondeo gegen A6 Avant hat das Stufenheck mehr Platz. Mit satten 523 Liter Stauvolumen gehört der große Ford zu den Platzkönigen im Vergleich. Er kann sich sogar ein Notrad erlauben, das ihm einige Liter kostet. Denn zur Werksangabe fehlen 27 Liter. Der Audi Kombi lädt nur 470 Liter ein – 75 Liter weniger als versprochen. Es wenn er ohne Jalousie (517 Liter) oder dachhoch (698 Liter) beladen wird, ist er konkurrenzfähig

523 Liter – 27 Liter weniger

Werksangabe: 550 Liter

Ford Mondeo

470 Liter – 75 Liter weniger

Werksangabe: 545 Liter

Audi A6 Avant

Golf Variant

VW Passat Variant

ZWEIMAL VARIANT

Golf fast so groß wie Passat!

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 gerüstet für Transportaufgaben aller Art. Die TÜV-Messung bestätigt den Werkswert bis auf fünf Liter

Der Passat-Laderaum 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 größerer Bruder? VW gibt zwar ein Ladevolumen von 603 Litern für den Passat an und wahrt damit einen Sicherheitsabstand von fast 100 Litern zum Golf, doch die TÜV-Messung enthüllt: Tatsächlich packen beide Variant nahezu gleich viel ein. Der Golf erreicht bis auf fünf Liter seine Werksangabe, der Passat liegt 95 Liter darunter. Der Kannibalisierungseffekt wird noch deutlicher bei Beladung ohne Gepäckraumrollo. 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 Reserveradmulde.

DIE KOFFERRAUM-KÖNIGE

1 VW Touran 561 Liter

Sieger ist der Touran. Keiner steckt mehr weg. Bei dachhoher Beladung sogar 928 Liter

2 Audi Q7 541 Liter

Obwohl der mächtige Audi-SUV seine Werksangabe extrem verfehlt, gehört er zu den Größten 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 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 Rücksitzbank – 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.

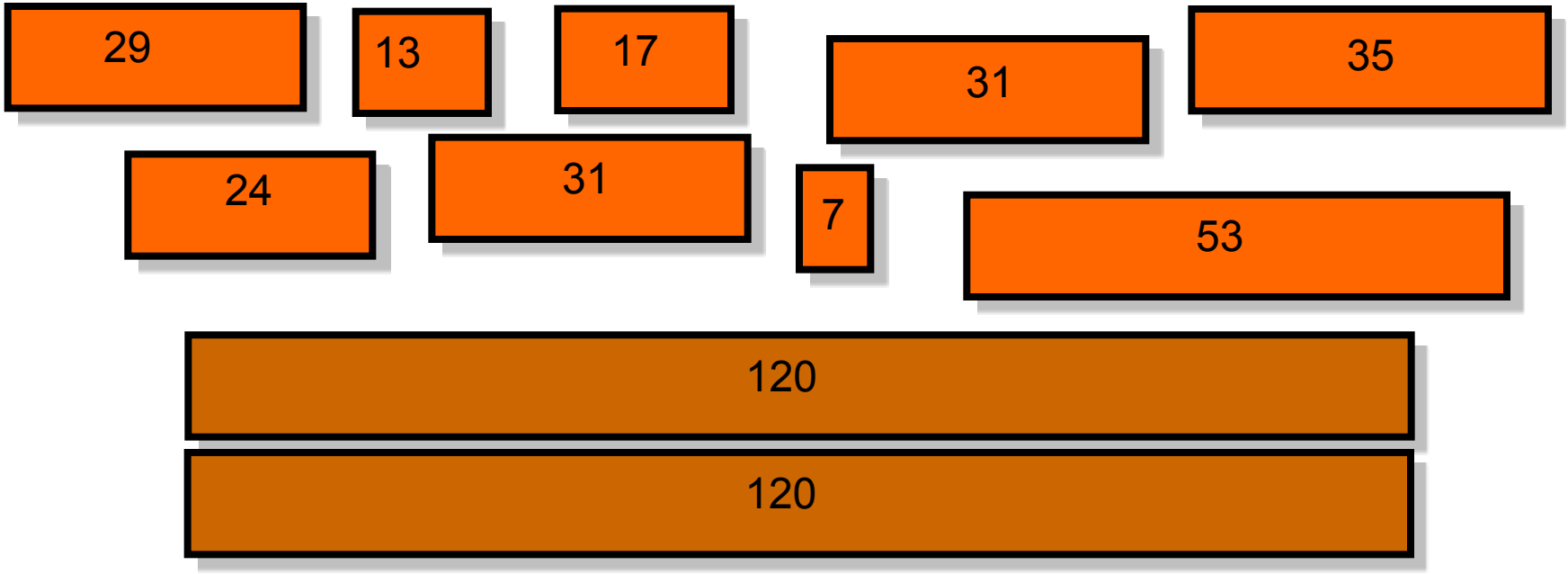
DIN 70020-1

Konzentration: Die N Quader (200 mal 100 m Millimeter) müssen g gestapelt und ausgezählt we

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.

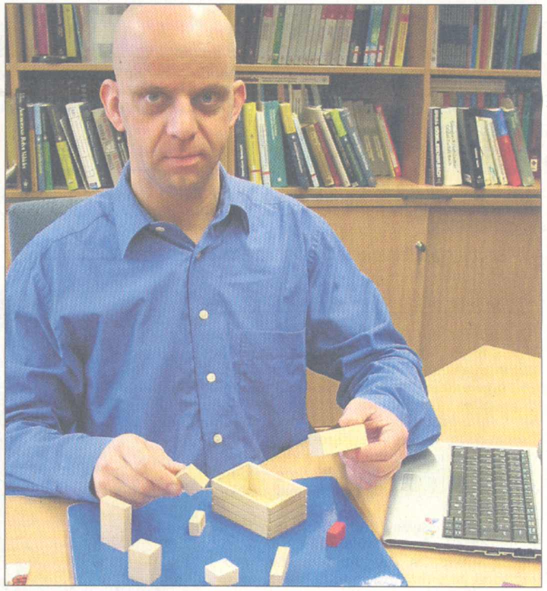
In jedem Fall gilt: „Man braucht immer einen Algorithmus, der unabhängig vom Einzelfall eine systematische Lösung anbietet", sagt Fekete. Für die Routenplanung bedeutet das: Ein einziger Algorithmus liefert sowohl den kürzesten Weg von

> „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

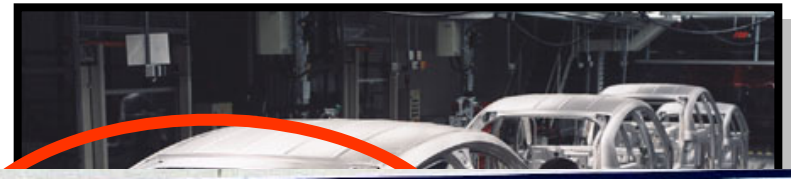


Professor Sándor Fekete zeigt auf einfache Weise die Probleme, die sich beim Packen und Beladen eines Kofferraumes ergeben können. Fotos (2): Steiner

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 selbststeuernden Robotern und de-

Kofferraumdesign



Klötzchen für Klötzchen: So hat der TÜV geteste

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 Styroporklötzchen in jedes Auto. Basierend auf DIN 70020-1 und ISO 3832, wurde nach TÜV-Spezifikation getestet. Dazu befanden sich die Sitze der Rücksitzbank – 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.

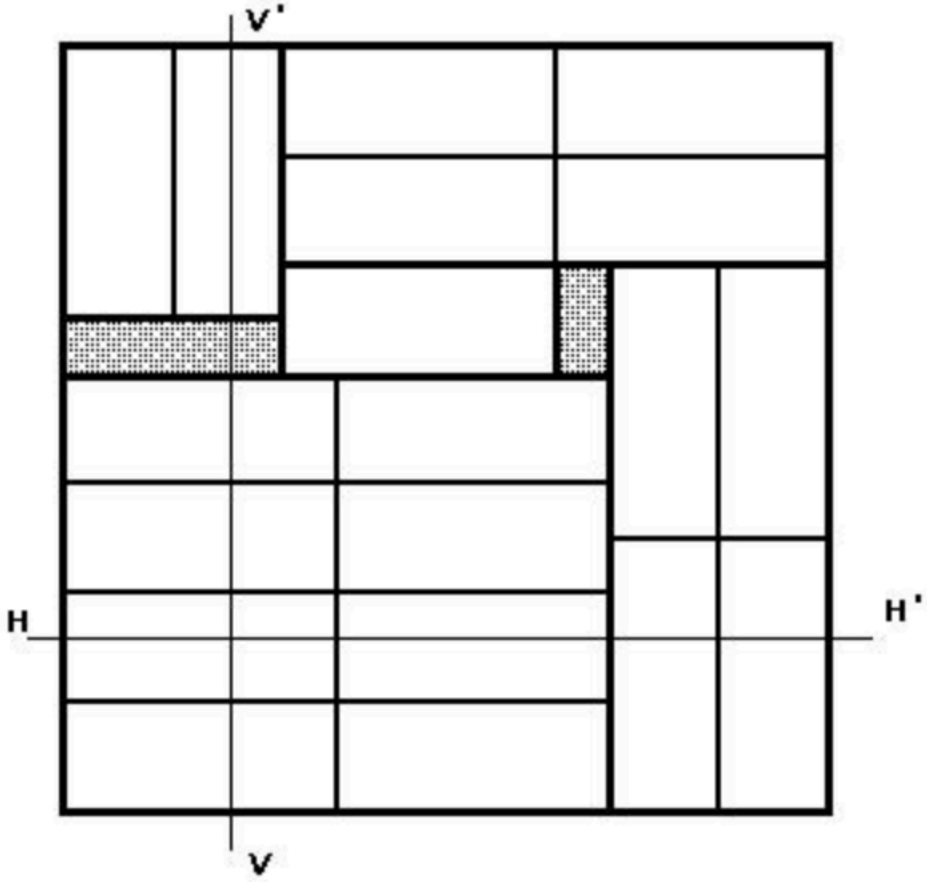
DIN 70020-1

Konzentration: Die K
Quader (200 mal 100 m
Millimeter) müssen g
gestapelt und ausgezählt we

DIN-Mess 20cm-
Quader ssen

Schwierigkeit: ken ist praktisch noch schwerer als eindimensionales!

Kofferraumdesign



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 project[1] 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

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.

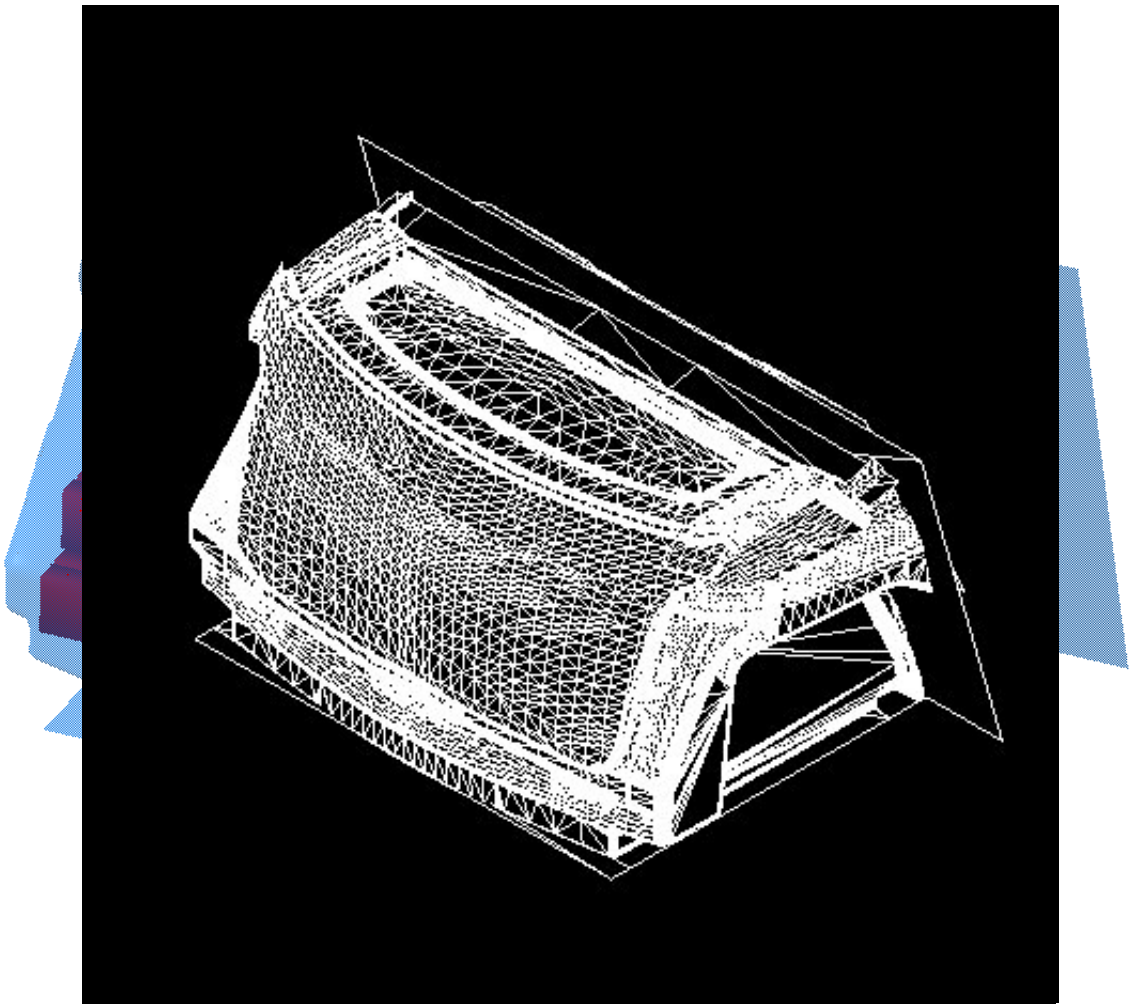
Related Open Problems

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

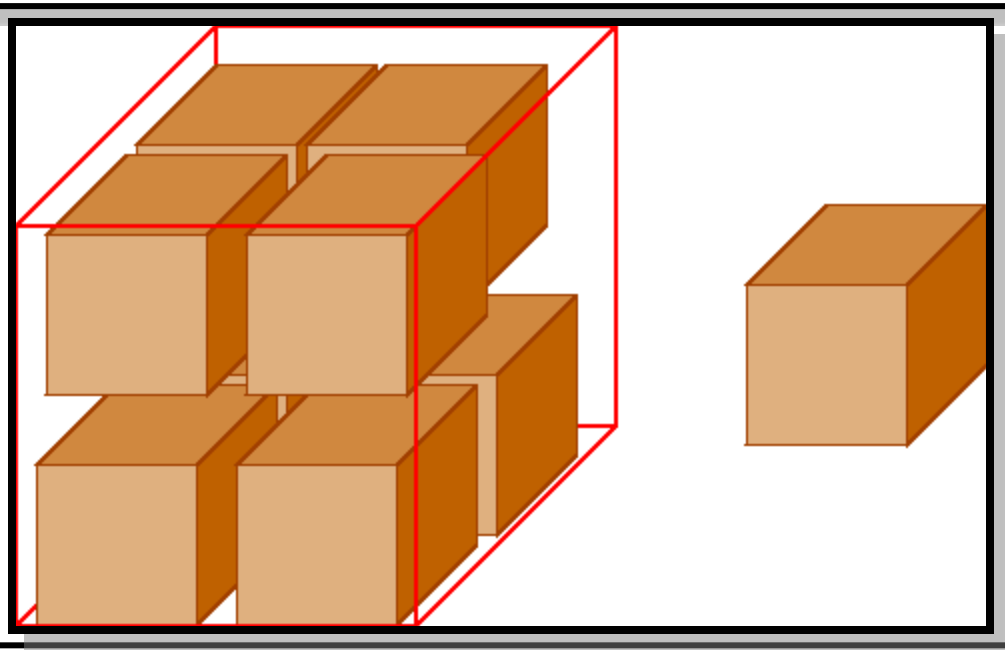
Ergebnisse



Mathematische Methoden

Ein einfaches Beispiel

Gegeben: 9 Würfel der Kantenlänge 0.4

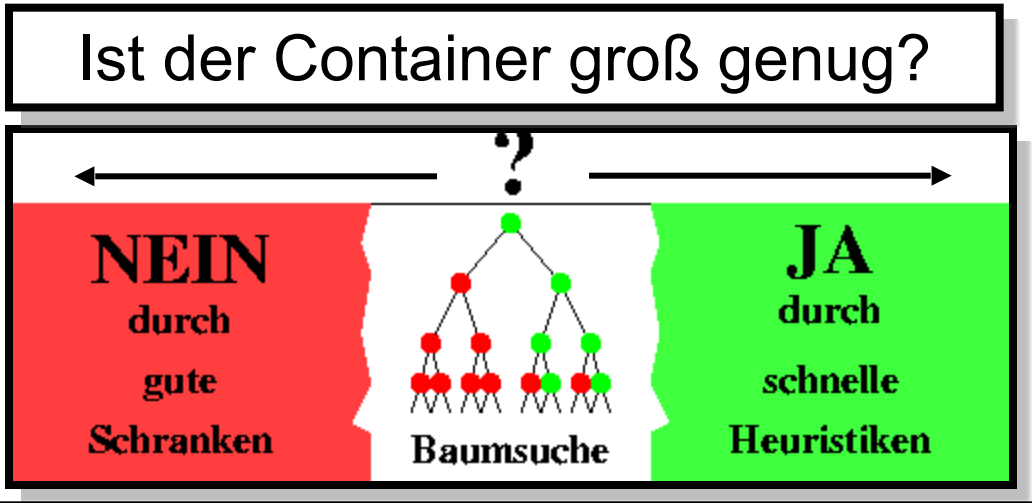


Frage: Passen die Würfel in einen würfelförmigen Container der Kantenlänge 1?

$$Volumen: 9 * 0.4^3 = 0.576$$

„Geht's nicht noch etwas schneller?“

Lösen schwerer Packungsprobleme



New classes of fast lower bounds for bin packing problems
SP **Fekete**... - Mathematical programming, 2001 - Springer
... xi∈T xi + ∑ xi∈**S**\T ⌊(k + 1)xi⌋ < (k + 1) ∑ xi∈**S** xi. Page 6. 16 Sándor P. **Fekete**, Jörg Schepers By definition of u (k) , the terms (k + 1) ∑ x∈T u (k)(x) and k ∑ x∈**S**\T u (k)(x) are in- teger, so by virtue of ∑ xi∈**S** xi ≤ 1, we have the inequality (k + 1) ∑ x∈T u (k)(x) + ...
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
... Sa"ndor P. Fekete1, JoØrg Schepersy,2 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, KoØln, 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
... Saindor P. **Fekete** J0"rg 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 Kiln, 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

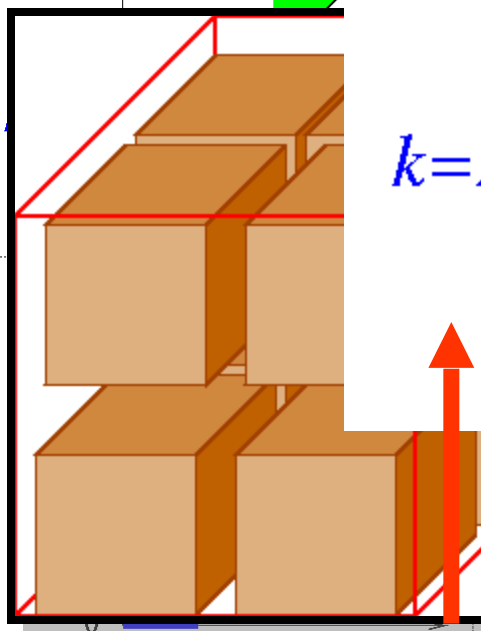
Gute und schnelle Schrank

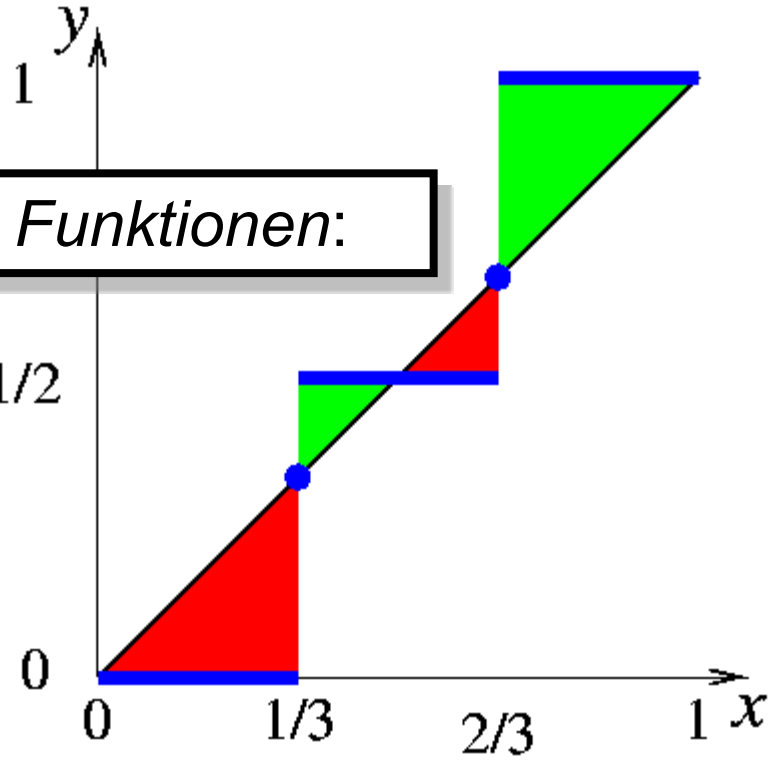
Idee: Betrachte *dualzulässige Funktionen*:

(1) f:

(2) $\sum_{i \in S}$

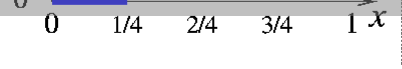
Satz (Fek
Ist f eine
eine unter

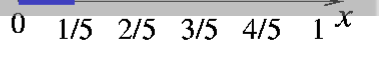




$u^{(2)}(0.4) = 0.5$

(w_i)





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

Konsequenz II: Framework für mehrdimensionale Schranken berechenbar und haben eine Gütegarantie von ¾.

Ergebnisse

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

Ergebnisse

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

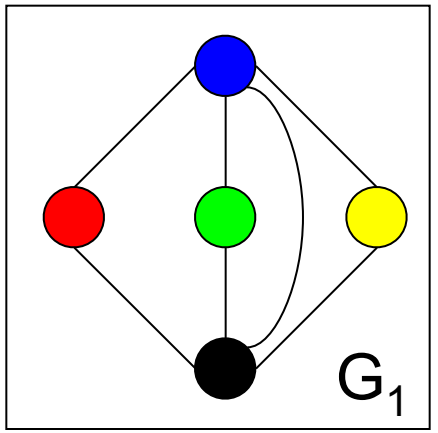
	Bound	L_1	L_2	$L_*^{(2)}$	$L_*^{(3)}$	$L_*^{(4)}$	$L_*^{(5)}$	$L_*^{(10)}$	$L_*^{(20)}$	$L_*^{(100)}$
Set	n									
{1, . . . , 100}	32	286	908	918	928	940	942	953	956	959
	100	122	775	833	854	864	872	888	895	897
	316	29	574	706	747	764	774	811	826	828
	1000	2	382	579	613	632	638	667	678	681
{1, . . . , 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

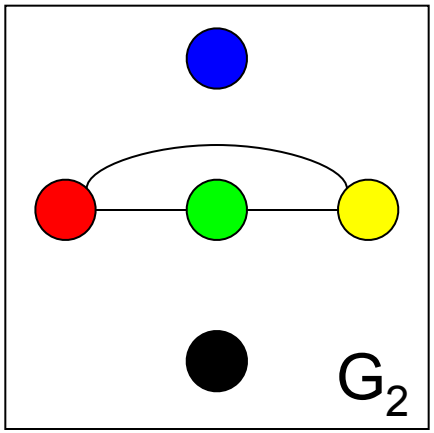
Baumsuche: Packmusterklassen

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



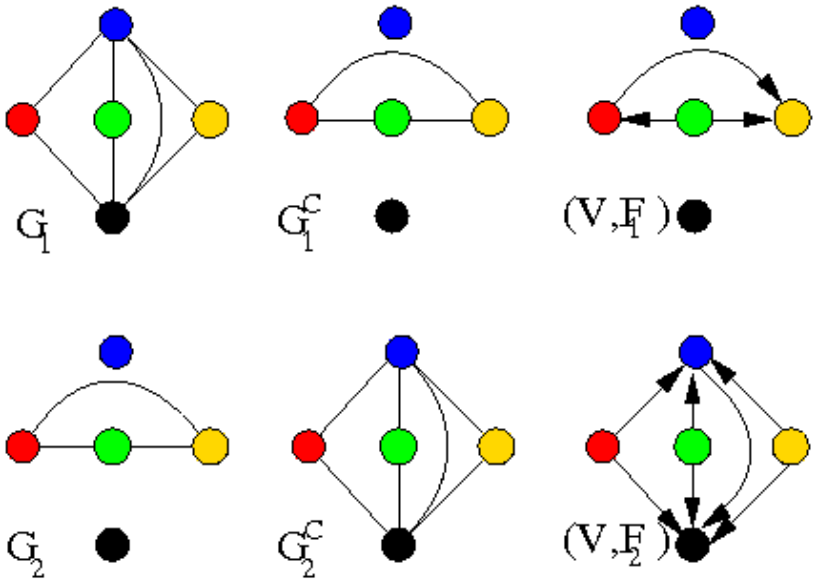


Satz (Fekete & Schepers): Eine Menge von Boxen lässt genau dann eine Packung zu, wenn es eine Packmusterklasse gibt.

Packmusterklassen (3)

Beweisskizze:

Wegen P1 sind die Komplementärgraphen transitiv orientierbar:

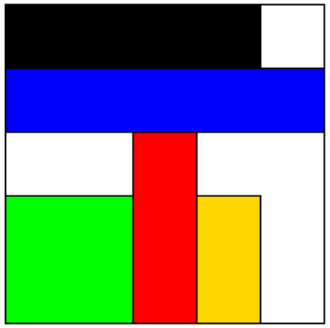


Diese Orientierung repräsentiert die Partialordnung der Objekte in der entsprechenden Koordinatenrichtung.

Packmusterklassen (4)

(Forts. Beweis)

Induktiv in aufsteigender Koordinatenordnung läßt sich dann eine Packung aufbauen:

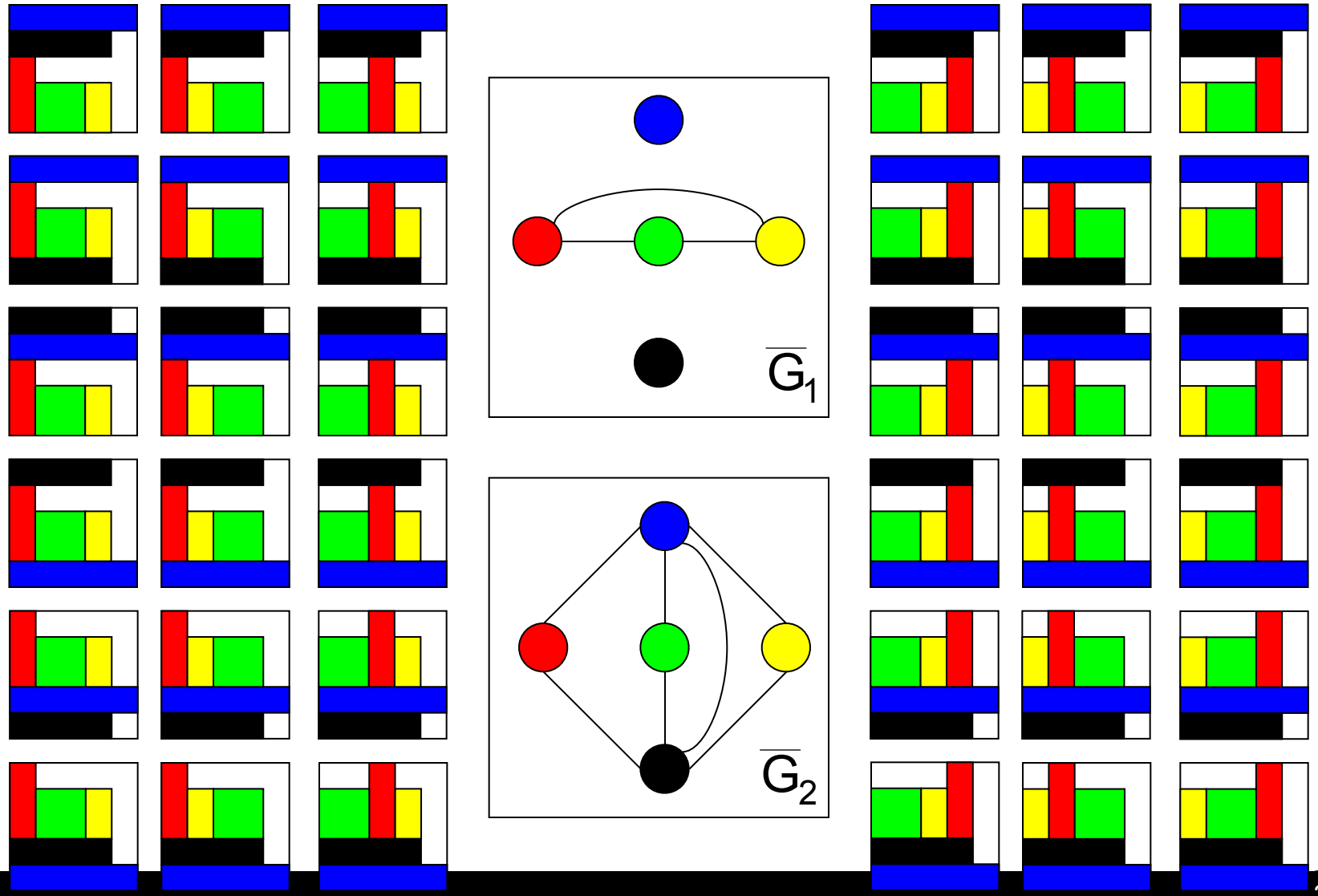


Wegen P3 gibt es keine Überlappungen.

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

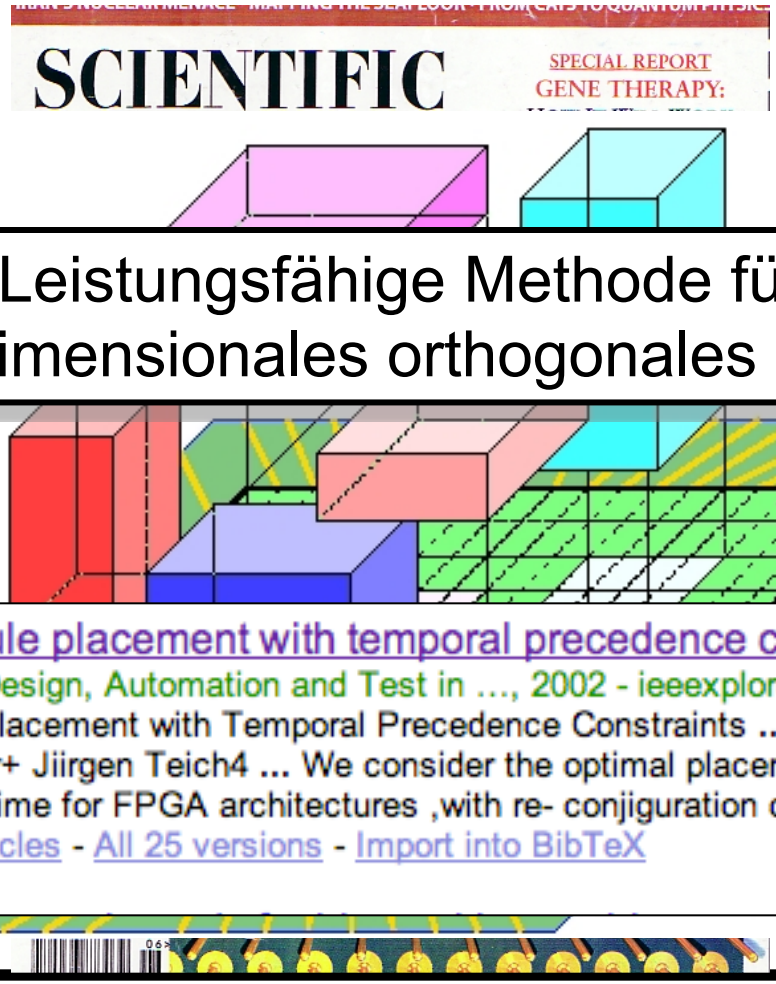


23

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

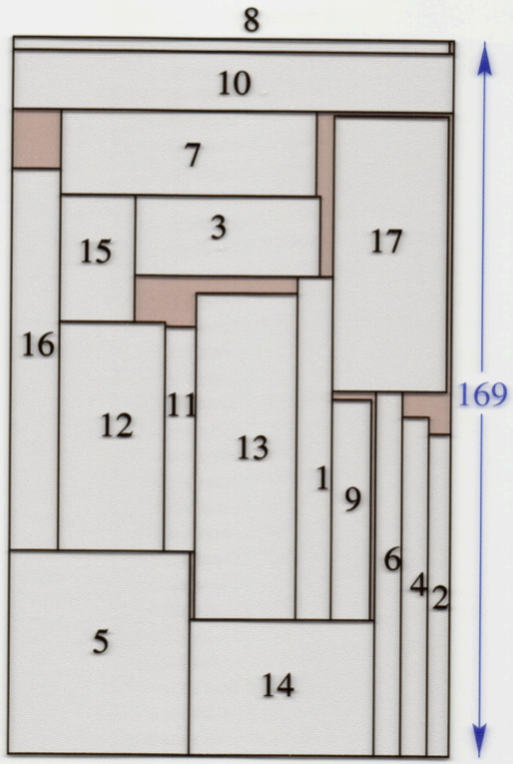
Optimal FPGA module placement with temporal precedence constraints
S Fekete, E Kohler... - Design, Automation and Test in ..., 2002 - ieeexplore.ieee.org
Optimal FPGA Module Placement with Temporal Precedence Constraints ... Siindor P. **Fekete*** Ekkehard Kohler+ Jiirgen Teich4 ... We consider the optimal placement of hardware mod- ules in space and time for FPGA architectures ,with re- conjiguration capabilities, ...
Cited by 97 - Related articles - All 25 versions - Import into BibTeX

Er

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

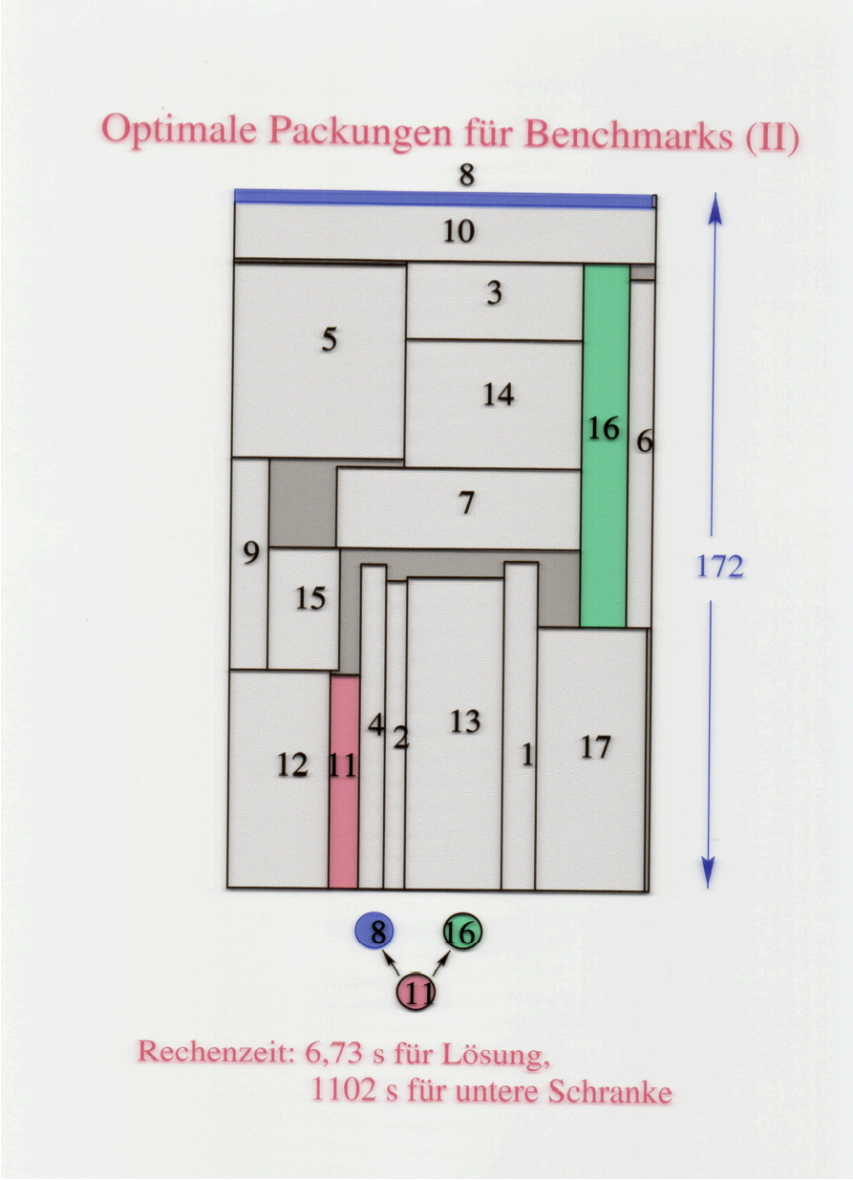
Ergebnisse

Optimale Packungen für Benchmarks

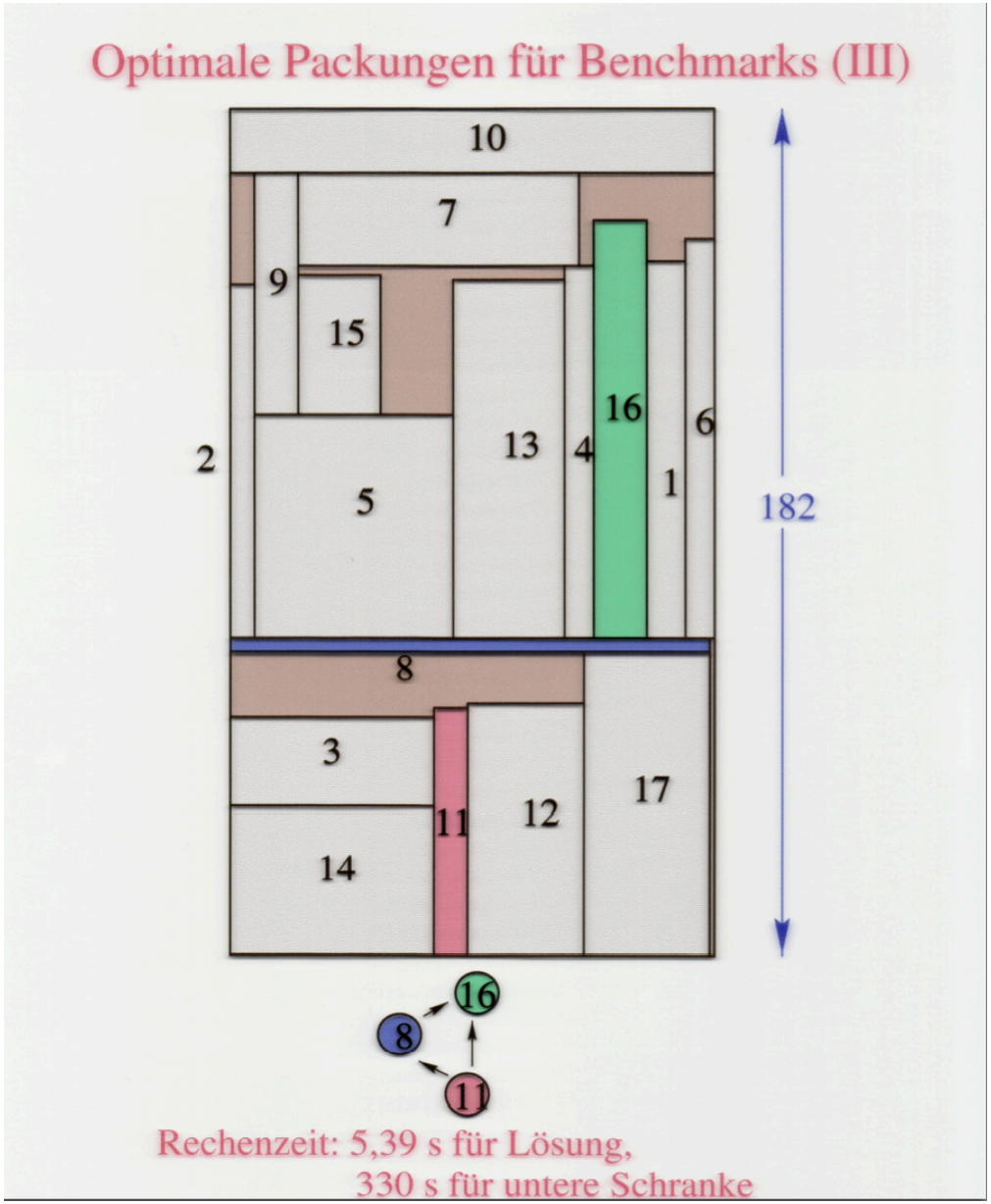


Rechenzeit: 7,29 s für Lösung,
179 s für untere Schranke

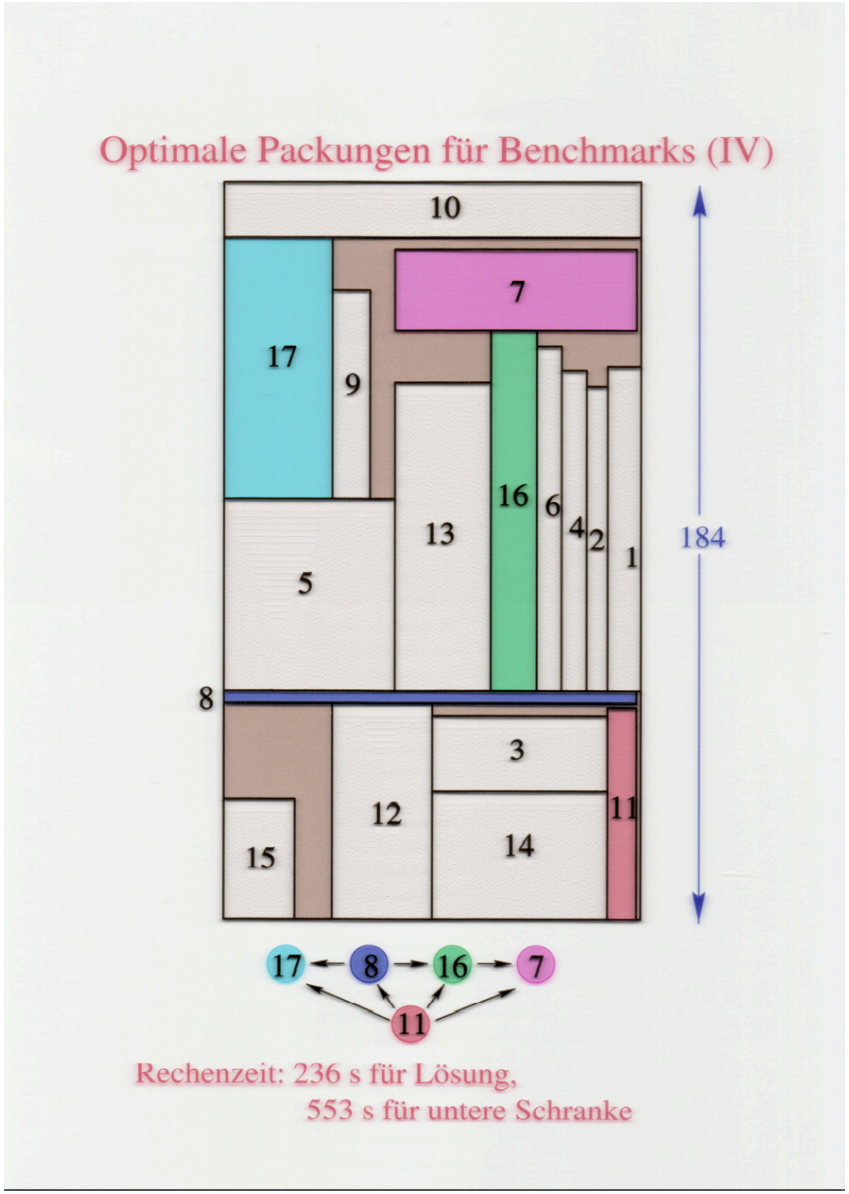
Ergebnisse


Optimale Packungen für Benchmarks (II)
8
10
5
3
14
16
6
7
9
15
172
12
11
4
2
13
1
17
8
16
11
Rechenzeit: 6,73 s für Lösung,
1102 s für untere Schranke

Ergebnisse


Optimale Packungen für Benchmarks (III)
10
7
9
15
2
5
13
4
16
1
6
182
8
3
11
12
17
14
16
8
11
Rechenzeit: 5,39 s für Lösung,
330 s für untere Schranke

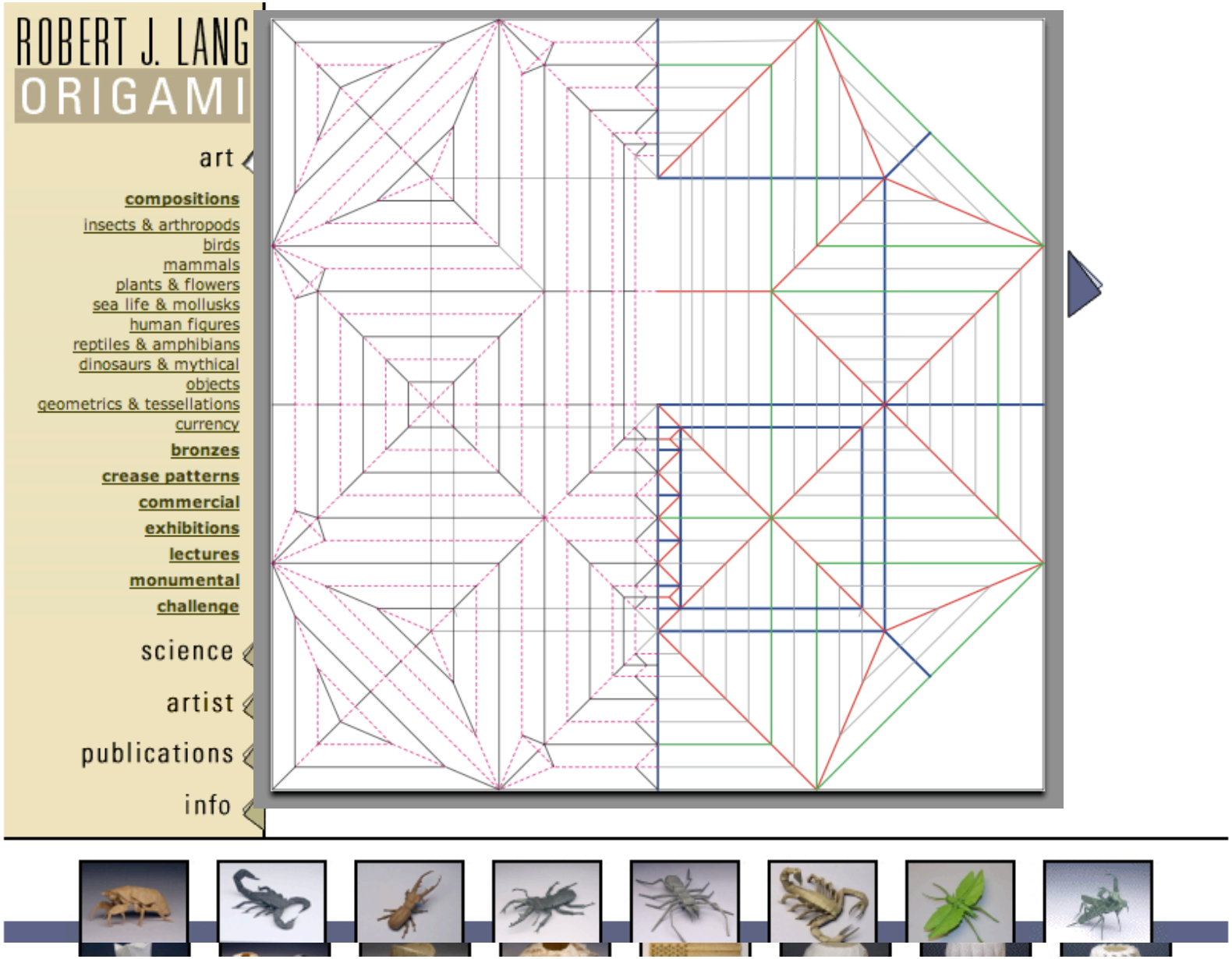
Ergebnisse


Optimale Packungen für Benchmarks (IV)
10
7
17
9
16
6
4
2
1
13
5
184
8
3
12
11
14
15
17
8
16
7
11
Rechenzeit: 236 s für Lösung,
553 s für untere Schranke

Origami

Falter unter sich

Origami


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

Origami

arXiv:1008.1224v2 [cs.CG] 20 Sep 2010

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.

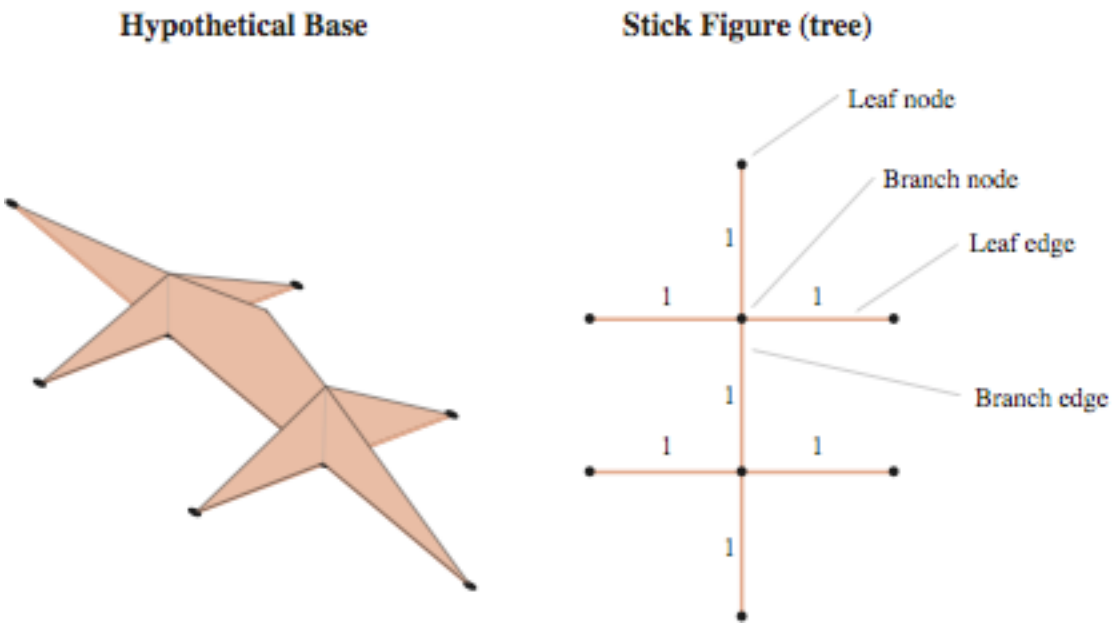
Hopes of a general approach for efficiently solving all origami design problems were dashed early on, when Bern and Hayes showed in 1996 that the general problem of crease assignment — given an arbitrary crease pattern, determine whether each fold is mountain or valley — was NP-complete [1]. In fact, 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

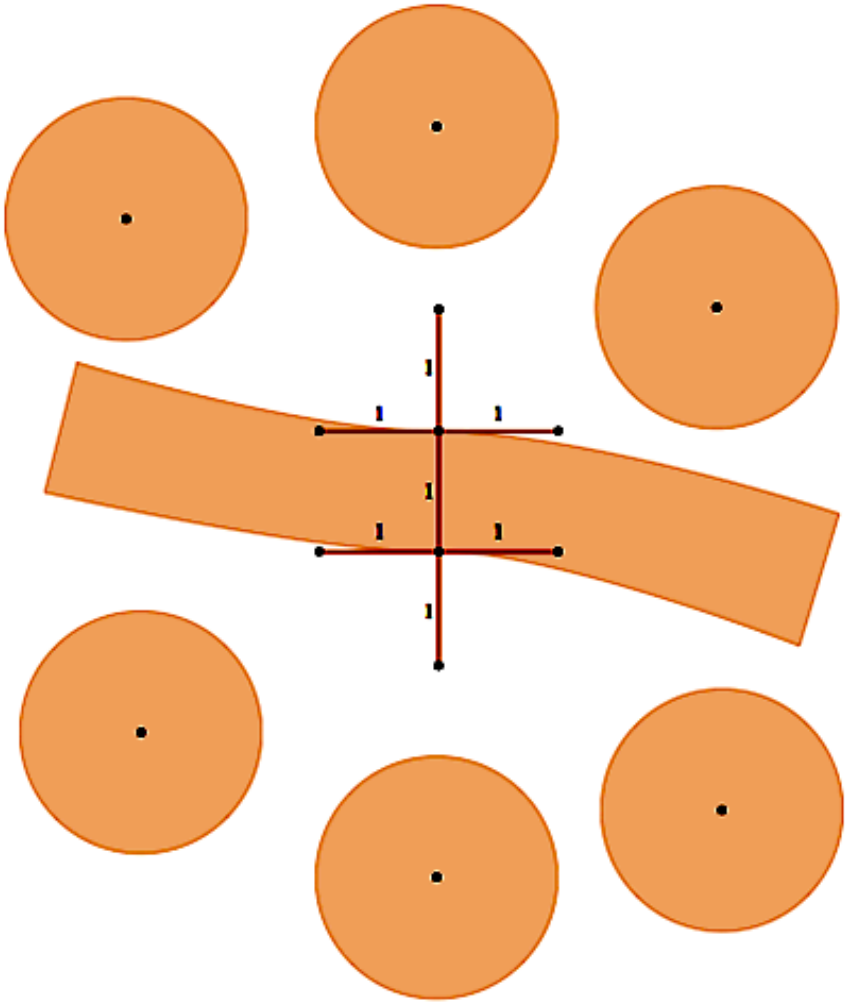
*edemaine@mit.edu.
†s.fekete@tu-bs.de.
‡robert@langorigami.com.

Schritt 1 - Baumkonstruktion



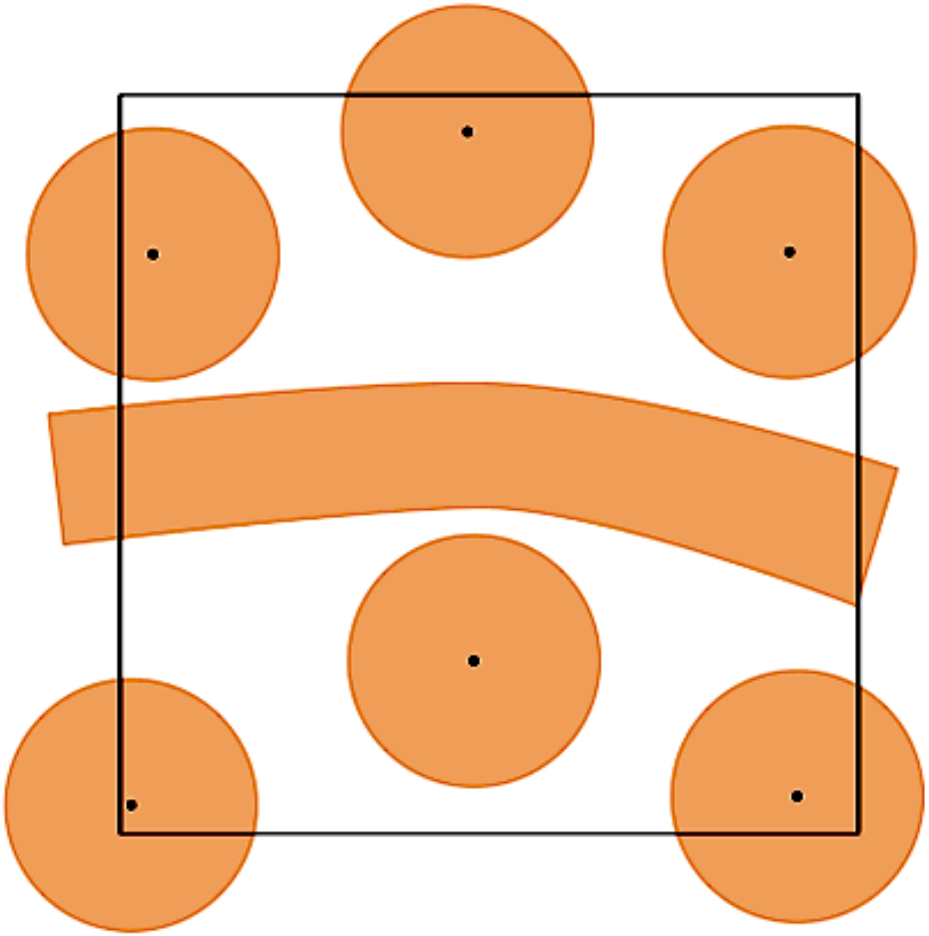
- Jede Kante repräsentiert eine Klappe.
- Jeder Knoten ist eine Klappenspitze oder eine Verbindung von Klappen.
- Endklappen entsprechen Blattkanten im Baum.

Schritt 2 – Von Bäumen zu Flüssen



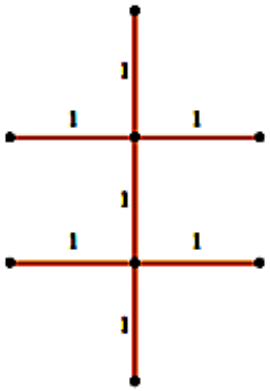
- Endklappen werden zu Kreisen
- Verzweigungskanten werden zu Flüssen
- Radius/Breite eines Kreises/Flusses ist die Länge einer Klappe.

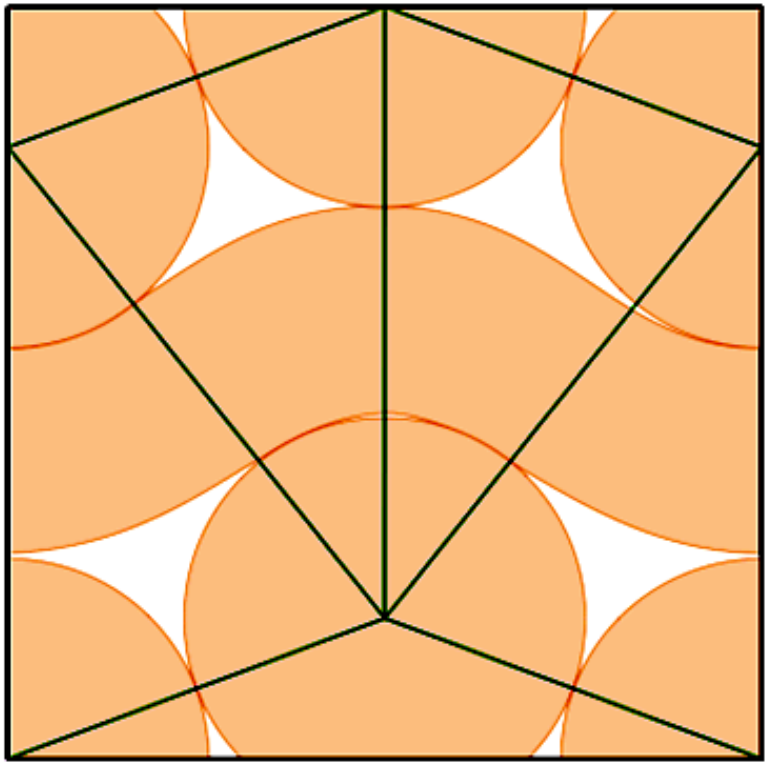
Schritt 3 – Kreispackung in Quadrat



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

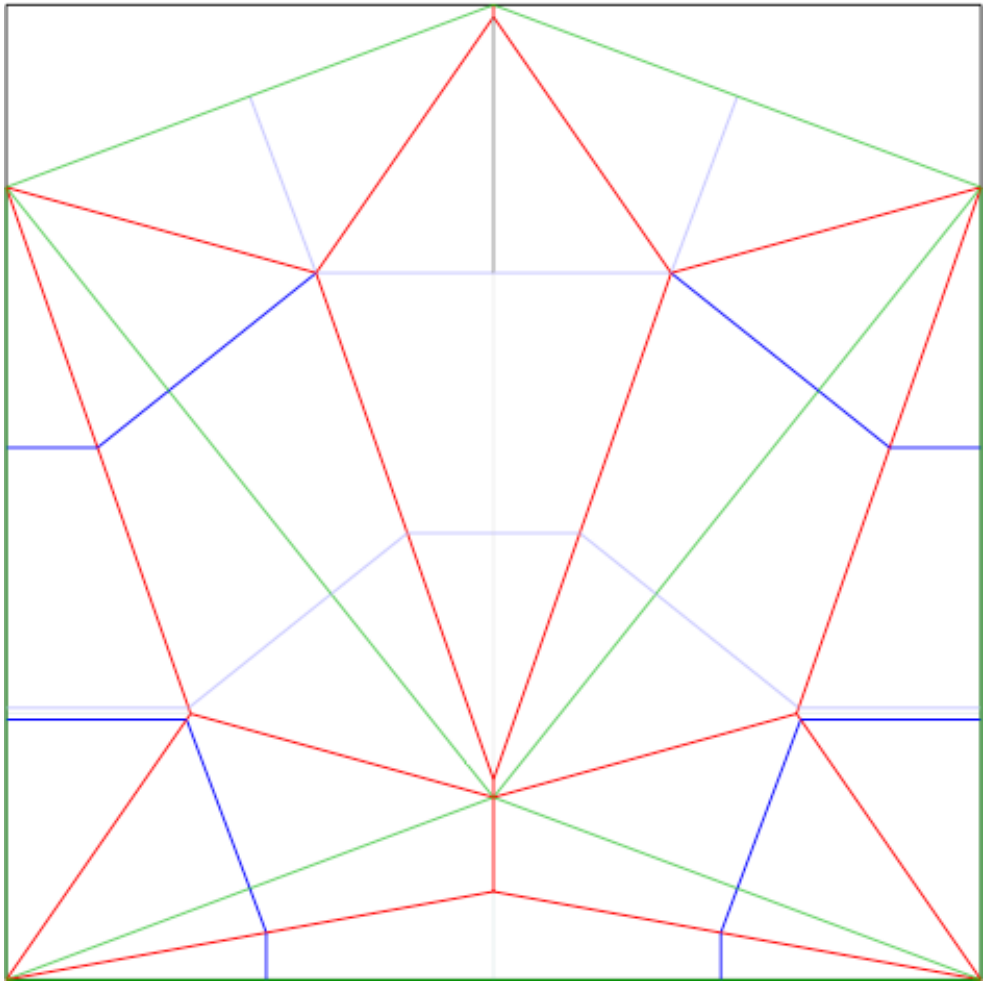
Schritt 4 – Dualgraph bilden





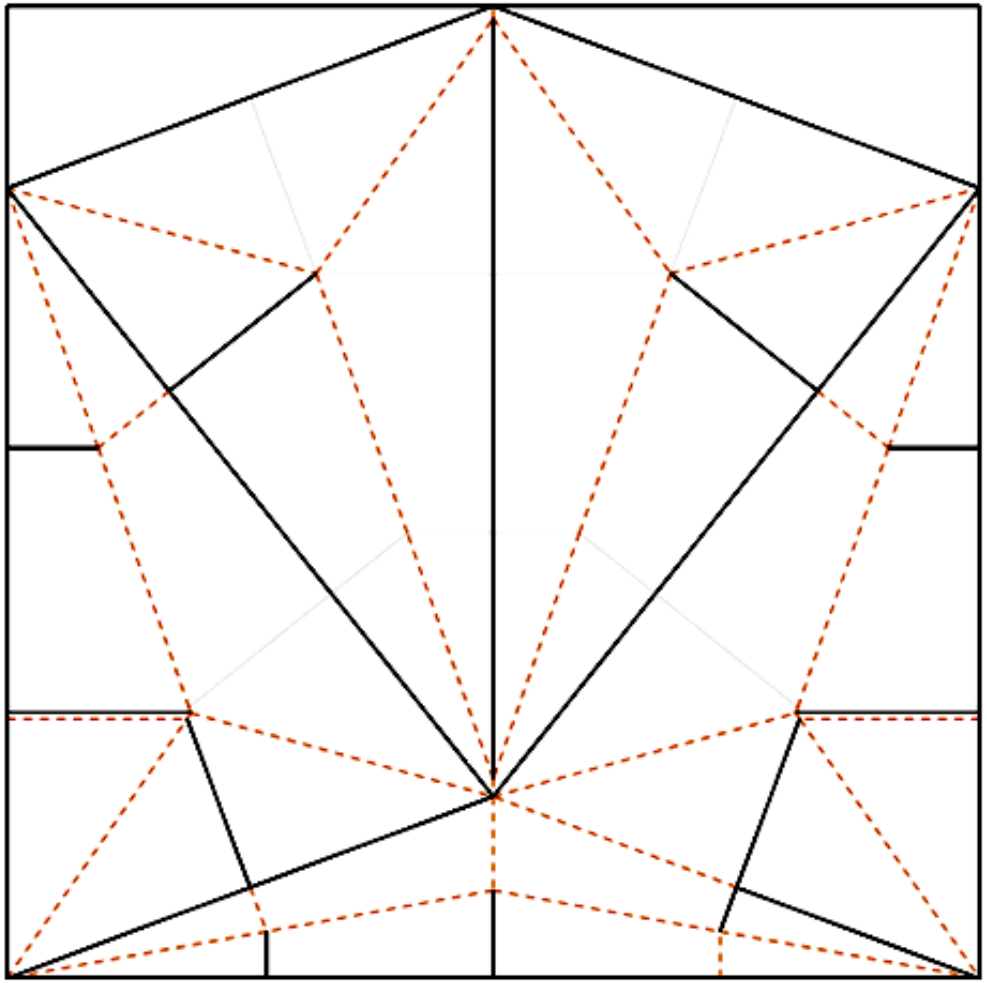
- Resultierende Polygone sind alle konvex.
- Keine “losen” Kreise erlaubt
- Verbindungen sind axiale Faltungen

Schritt 5 – Quadrat mit Faltungen füllen



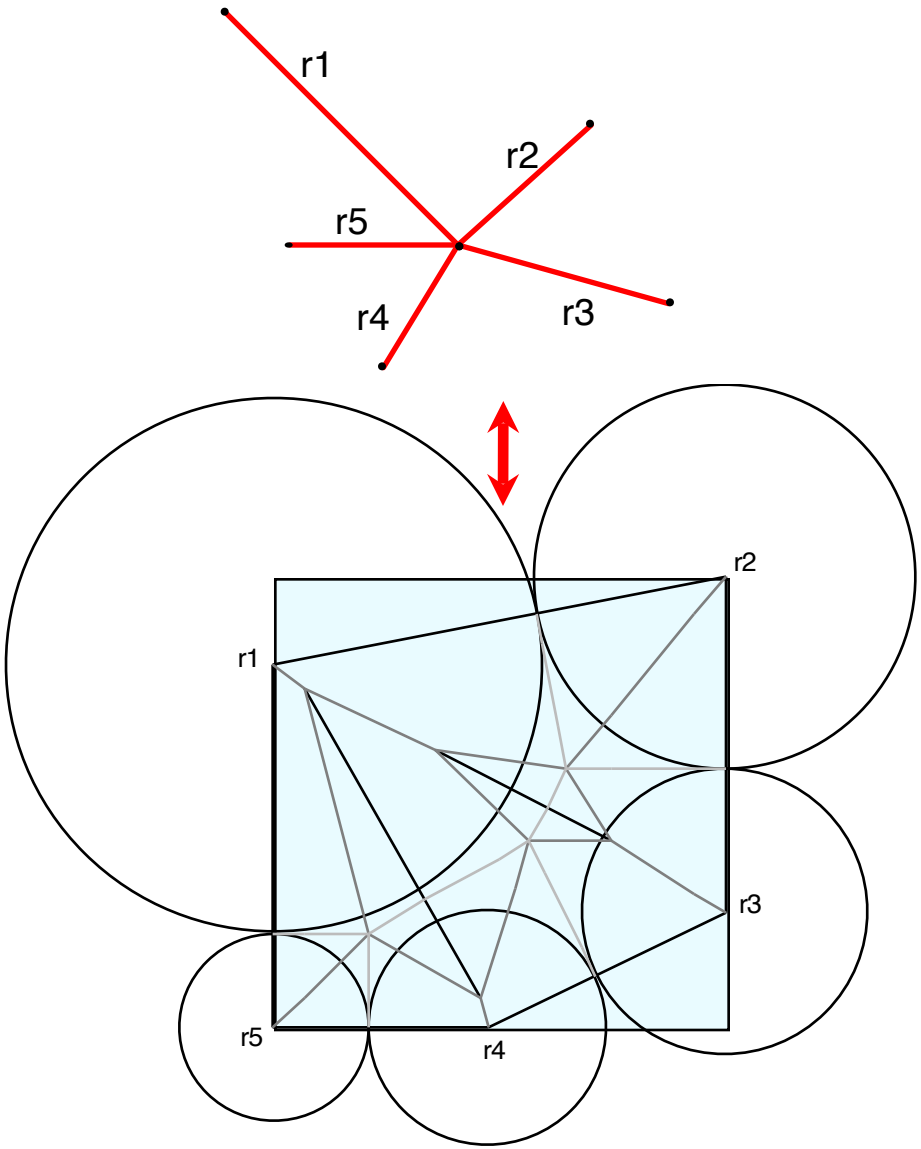
- Axial (grün)
- Kante (rot)
- Zwickel (grau)
- Scharnier (blau)

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



- Die Pfadbedinungen 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

Gegeben:

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

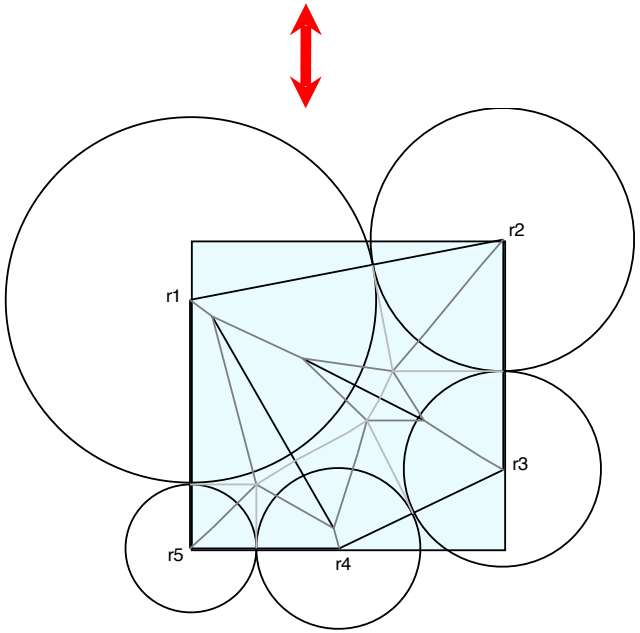
Gesucht:

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

Bekannt:

3-Partition ist NP-schwer.

Komplexität: Beweis von NP-Schwere



Gegeben:

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

Transformiere in:

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

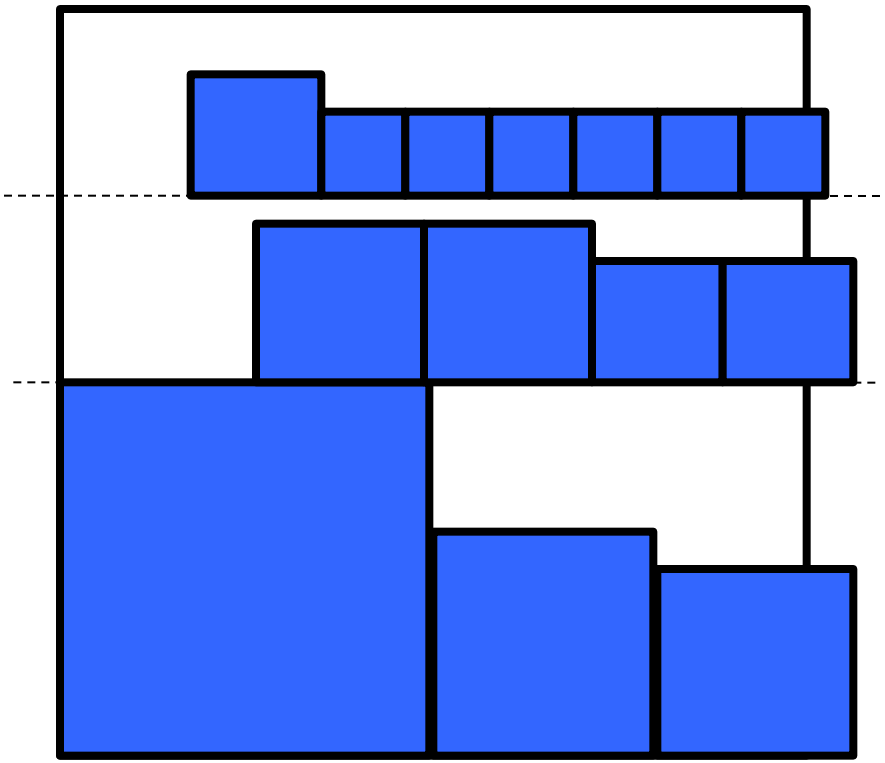
Konsequenz 1:

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

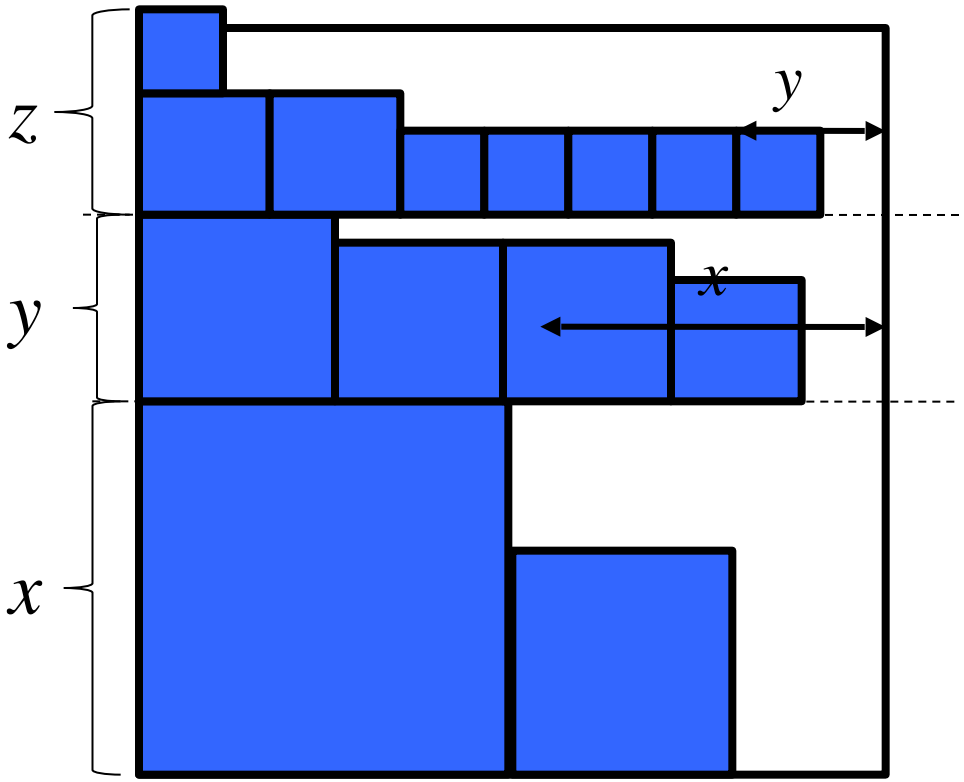
Untere Schranke für Quadrate



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



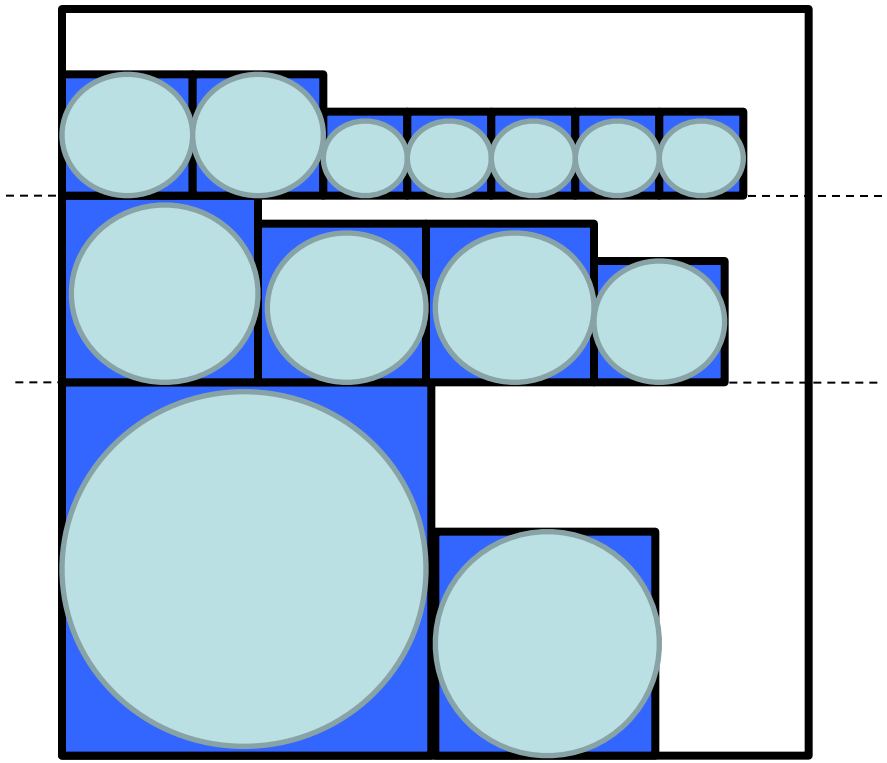
Bilanziere gepackte Fläche:

$$x + y + z > 1$$

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

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

Untere Schranke für Kreise

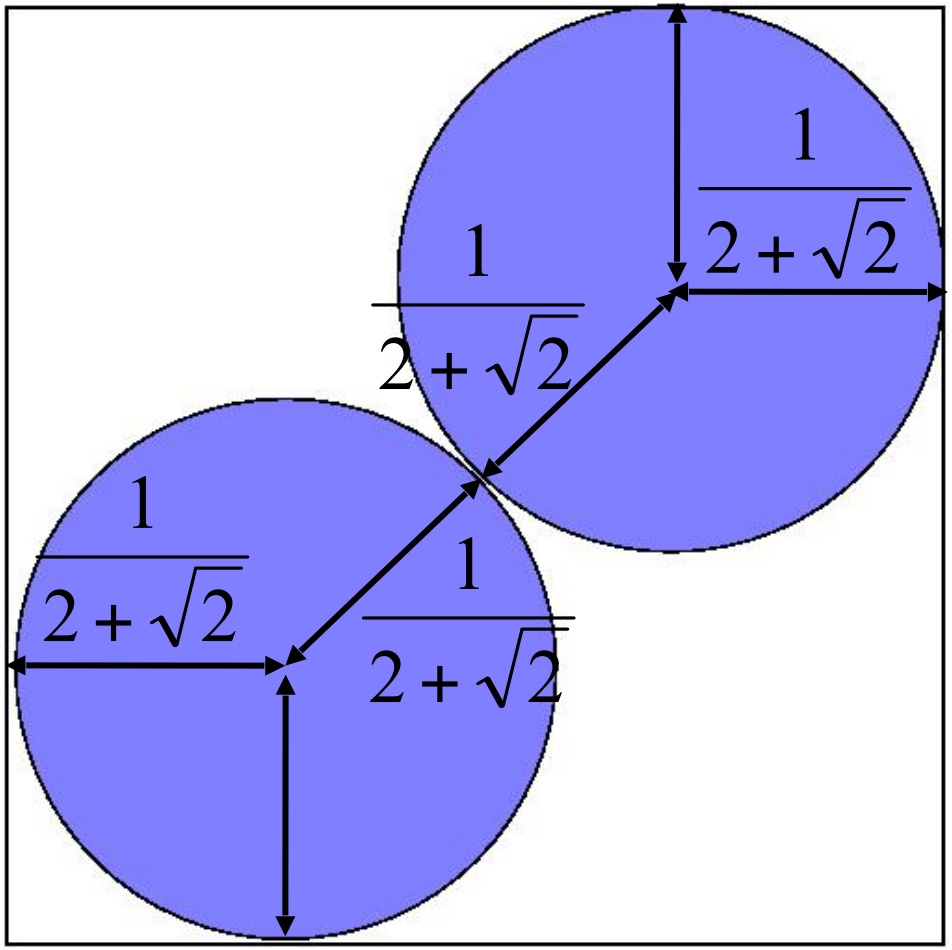


Korollar:
Jede Menge von Kreisen mit Gesamtfläche höchstens

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

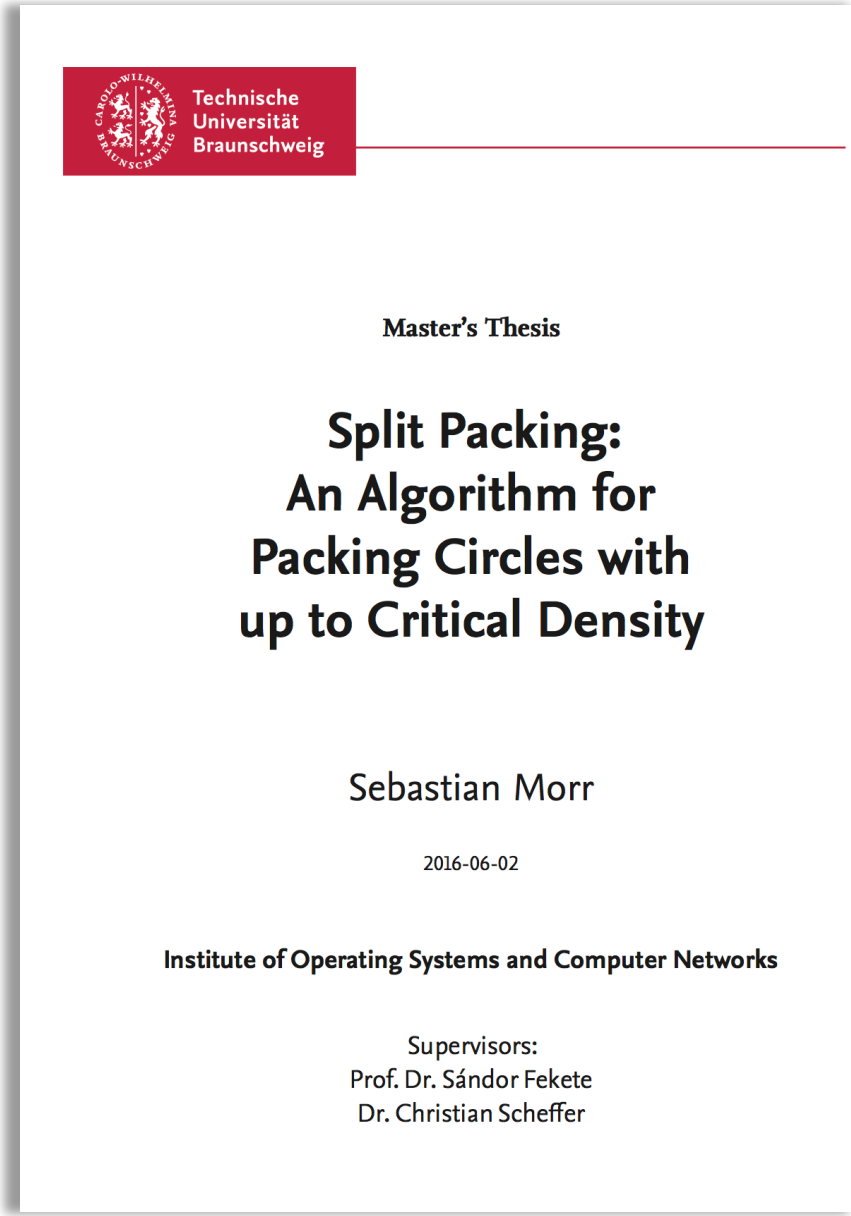
lässt sich in ein Einheitsquadrat packen.

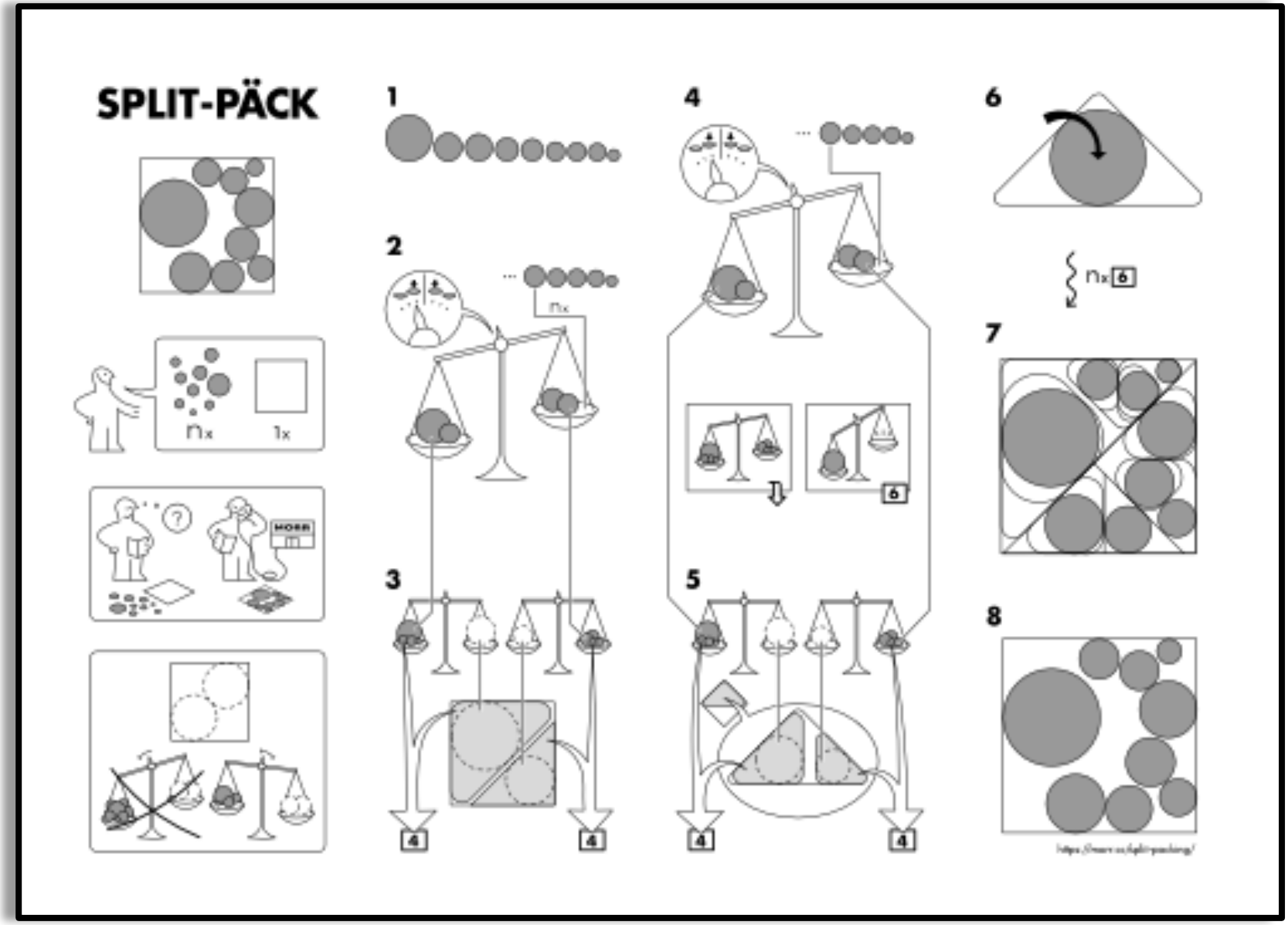
Obere Schranke



Kritische Konfiguration für Kreise, Dichte:

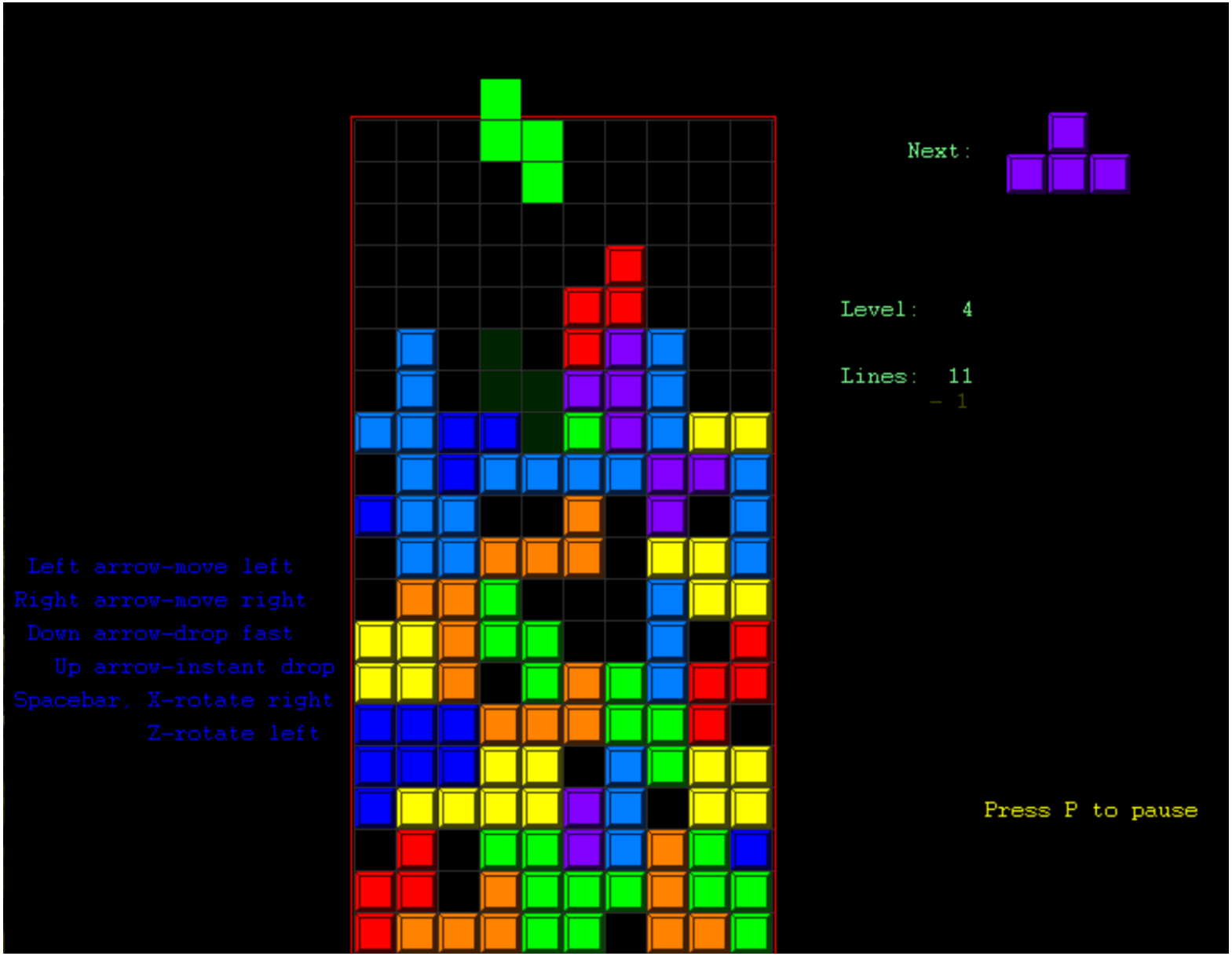
$$\frac{2\pi}{6+4\sqrt{2}} = 0.53901\ldots$$


Technische
Universität
Braunschweig
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
1
2
3
4
5
6
7
8
nx
1x
6
4

Online-Packen

Ergebnisse


Next:
Level: 4
Lines: 11
- 1
Left arrow-move left
Right arrow-move right
Down arrow-drop fast
Up arrow-instant drop
Spacebar, X-rotate right
Z-rotate left
Press P to pause

Ergebnisse

Algorithmica
DOI 10.1007/s00453-012-9713-8

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 "ReCoNodes". 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, Mühlenpfordtstrasse 23, 38106 Braunschweig, Germany
e-mail: s.fekete@tu-bs.de

Published online: 16 November 2012

Technische Universität Braunschweig
Institut für Betriebssysteme und Rechnerverbund

Master Thesis

Online Packing Into Squares

by
Hella-Franziska Hoffmann

Supervisor:
Prof. Dr. Sándor P. Fekete

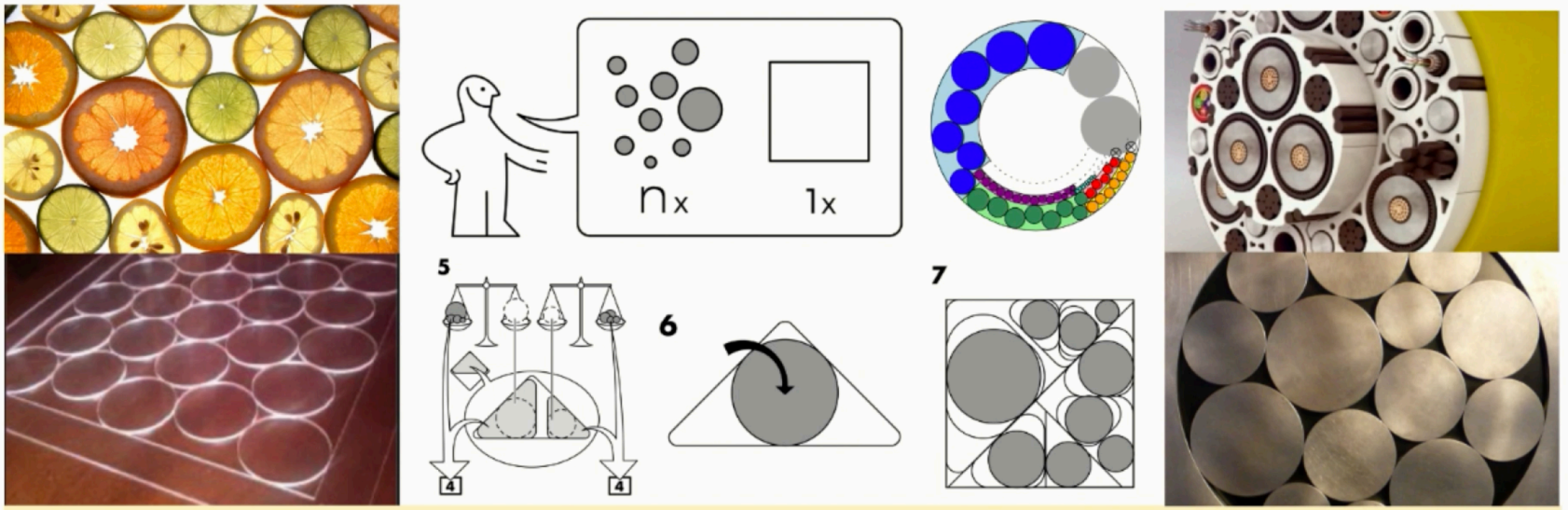
Online Square-into-Square Packing

Sándor P. Fekete[1] and Hella-Franziska Hoffmann[2]

[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...-competitive method for minimizing the required container size, and a lower bound of 1.33... for the achievable factor.

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



Packing Geometric Objects with Optimal Worst-Case Density

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

Vielen Dank!