Russian Supercomputing Days
September 23 – 24, 2019, Moscow, Russia

Optimal Packings of Congruent Circles on a Square Flat Torus as Mixed-Integer Nonlinear Optimization Problem

Vladimir Voloshinov, *Sergey Smirnov*

Center for Distributed Computing, http://distcomp.ru,
Institute for Information Transmission Problems (Kharkevich Institute),
Russian Academy of Sciences

Supported by the Russian Science Foundation (project No. 16-11-10352)

Outlines

- **Packing Circles on a Square Flat Torus: challenge for specialists in combinatorial geometry; practical application for super-resolution imagery (space & aerophotography);**
- **Square Flat Torus Packing Problem (FTPP) as Mixed-Integer Nonlinear Problem (MINLP);**
- **Results of computing experiments (on three clusters from Russian Top50) numerical proof of a conjecture about optimal arrangement for 9 circles;**
- **Conclusions and Acknowledgements**

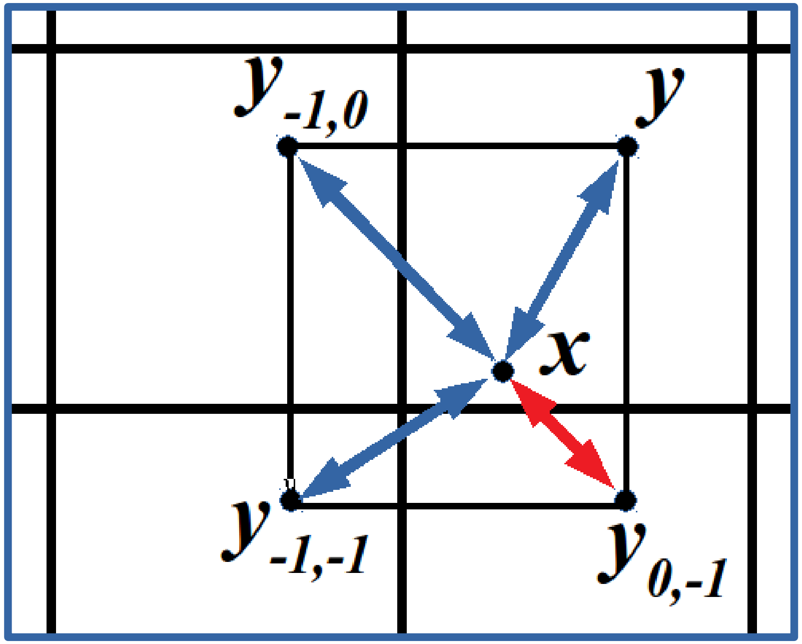
Flat Torus Packing Problem (FTPP), formulation

Metric on Flat Torus ($\mathbb{T}^2=\mathbb{R}^2/\mathbb{Z}^2$) is induced by Euclidean metric

$$x,y\in\mathbb{R}^2,\ d(x,y)\doteq\sum_{k=1:2}\left(\min\{|x_k-y_k|,1-|x_k-y_k|\}\right)^2$$

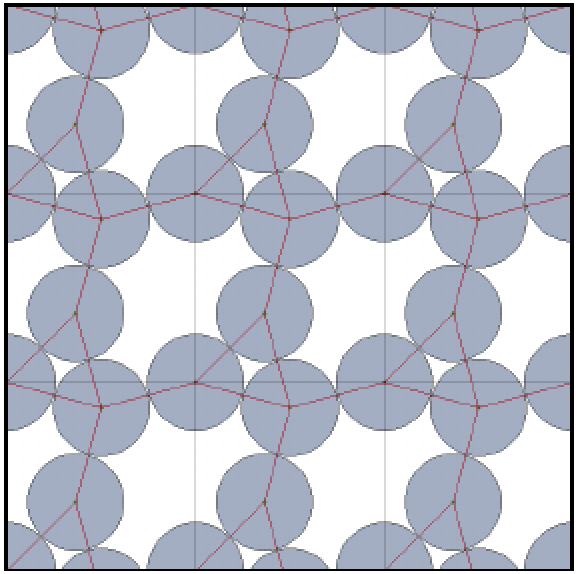
$$\{x_i: i\in 1:N\},\ x_i\in[0,1]\otimes[0,1]$$

$$\min_{1\leqslant i<j\leqslant N} d(x_i,x_j)\to\max_{\{x_i:i\in 1:N\}} : x_i\in[0,1]\otimes[0,1]\quad \text{(FTPP)}$$

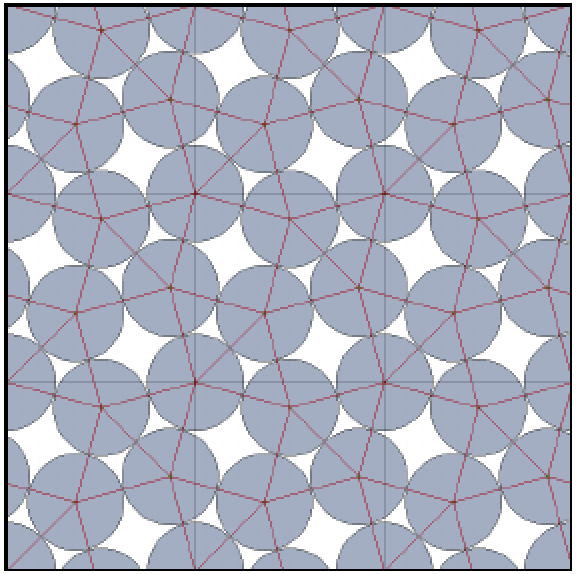


Examples of optimal solutions for

N = 3



N=4



Oleg Musin, Anton Nikitenko.

Optimal packings … arxiv.org, 2012

Flat Torus Packing Problem (FTPP), history

Max-min pairwise distances for N points placed somewhere – well-known problems in combinatorial geometry

IMHO, most popular is Tammes problem: points on a sphere

J.B.M Melissen. **Packing and covering with circles, Thesis, 1997 (Packing problem formulation for $\mathbb{T}^2$ and solutions for $N \leqslant 4$)**

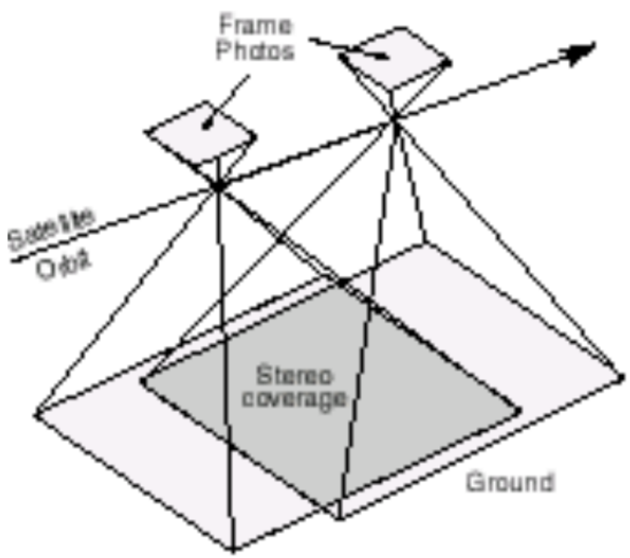
A. Heppes. **Densest Circle Packing on the Flat Torus, 2000 (solutions for rectangular torus $\mathbb{T}^2(a,b)=[0,a)\otimes[0,b)$ for $N \leqslant 4$)**

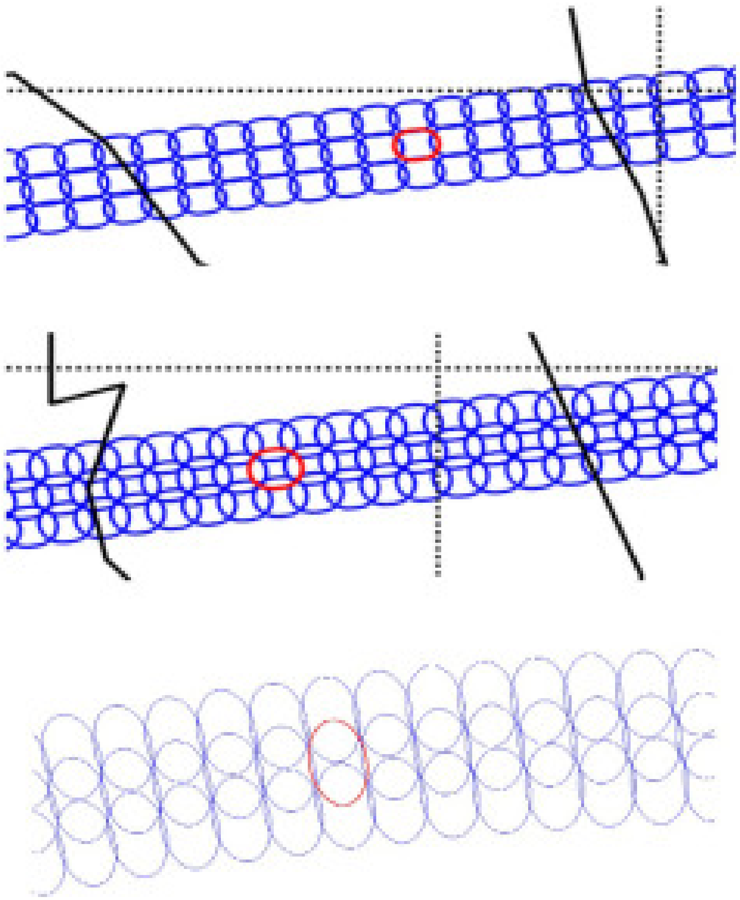
Oleg Musin, Anton Nikitenko. **Optimal packings of congruent circles on a square flat torus. arxiv.org, 2012;**
Discrete & Computational Geometry, 2016.

(Solutions for $\mathbb{T}^2$, $N \leqslant 8$ and <u>only conjecture for N=9</u>)

Practical application for super-resolution imagery (space & aerophotography).

Musin & Nikitenko told that their interest on the subject had been inspired by that subject...



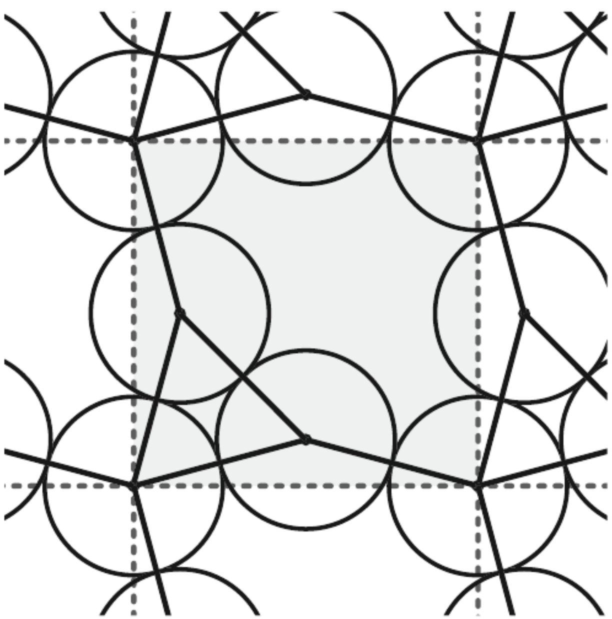


Combinatorial geometry approach to the problem

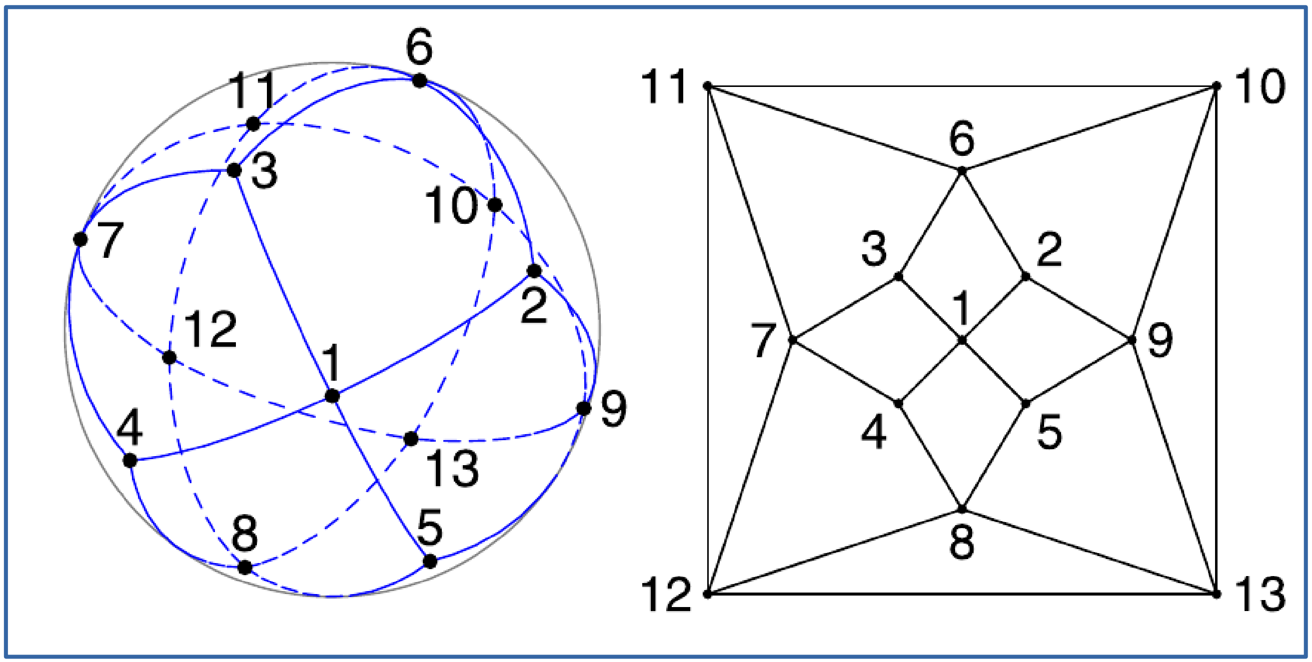
Necessary optimality condition as irreducible (rigid) contact graph and computer aided enumeration of all such graphs.

Irreducible contact graph: **if it is *rigid* in a sense that there are *no possible slight motions of vertices that change the contact graph or increase the edge lengths*.**

It is rather general approach used for various packing problems, e.g. Tammes problem N⩽14 (Musin, Tarasov, 2012-14).



FTPP, N=3



Tammes, N=13

Reasons and approach

To extend our experience in optimization modeling in distributed computing environment:

- **<u>Reduce</u> the problem <u>to Mixed-Integer Nonlinear Math. Programming (MINLP)</u> of canonical form**
- **<u>Try Branch-and-Bound solver</u> (<u>parallel</u>, if possible)**

Various problems

Universal approach

B&B solver

MINLP problems

FTPP as NLP with Mixed-Integer variables (e.g. see Dantzig, 1960)

$i=1{:}N,\ \ k=1{:}2,\ \ \mathrm{IJ}\doteq\{(i,j):1\leqslant i<j\leqslant N\}$,

$\boxed{y_{ijk}\doteq\min\{|x_{ik}-x_{jk}|,1-|x_{ik}-x_{jk}|\}},\ \Big(k=1{:}2,(i,j)\in\mathrm{IJ}\Big)$,

$\boxed{z_{ijk}\doteq-|x_{ik}-x_{jk}|}=\min\{x_{jk}-x_{ik},x_{ik}-x_{jk}\}\ \Big(k=1{:}2,(i,j)\in\mathrm{IJ}\Big)$,

$\eta_{ijk}\in\{0,1\},\ \zeta_{ijk}\in\{0,1\}\ \ \Big(k=1{:}2,\ (i,j)\in\mathrm{IJ}\Big)$.

***$2N^2$* continuous variables x_{ik}, y_{ijk}, z_{ijk} and *$2N(N-1)$* binary η_{ijk}, ζ_{ijk}.**

$$D\to\max\big(\text{with vars. } x_{ik},y_{ijk},z_{ijk},\eta_{ijk},\zeta_{ijk}\big),\text{s.t.}:$$

$$\boxed{D\leqslant\sum_{k=1:2}y_{ijk}^2,}$$

non-convex quadratic, B&B alg.

$$-y_{ijk}-\eta_{ijk}\leqslant x_{ik}-x_{jk}\leqslant 1-y_{ijk},$$
$$-1+y_{ijk}\leqslant x_{ik}-x_{jk}\leqslant y_{ijk}+\eta_{ijk},$$
$$z_{ijk}+\eta_{ijk}\leqslant y_{ijk}\leqslant -z_{ijk}, \qquad \text{(FTPP)}$$
$$z_{ijk}\leqslant x_{ik}-x_{jk}\leqslant z_{ijk}+2\zeta_{ijk},$$
$$-z_{ijk}-2\Big(1-\zeta_{ijk}\Big)\leqslant x_{ik}-x_{jk}\leqslant -z_{ijk},$$
$$0\leqslant x_{ik}\leqslant 1,\ y_{ijk}\in\mathbb{R},\ z_{ijk}\in\mathbb{R},\ \eta_{ijk}\in\{0,1\},\ \zeta_{ijk}\in\{0,1\}.$$

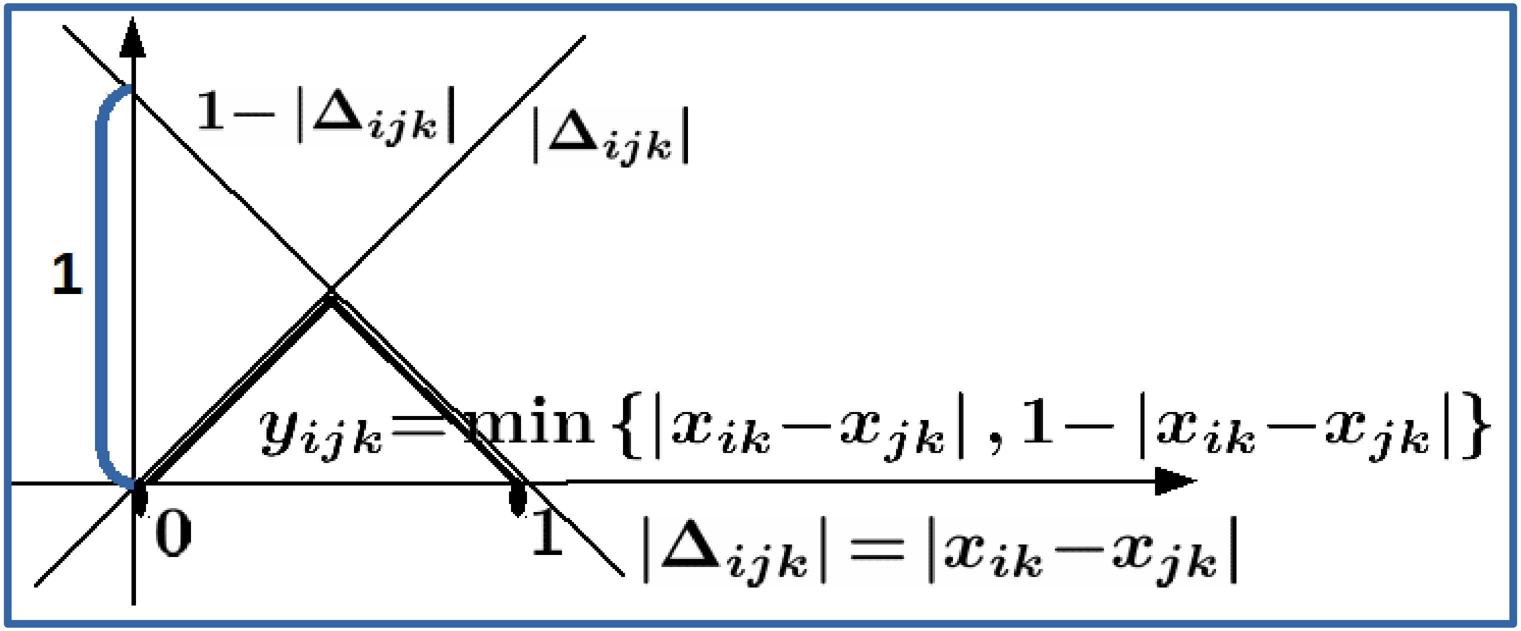
Non-convex piecewise-linear constraints ==> MILP

Replacement piecewise-linear equalities with linear inequalities with binary variables (it seems that Dantzig, 1960, was the first)

$$y_{ijk} \doteq \min\{|x_{ik}-x_{jk}|, 1-|x_{ik}-x_{jk}|\},\quad \bigl(k=1{:}2,\ (i,j)\in \mathrm{IJ}\bigr)$$

is equivalent to system of linear inequalities

$$y_{ijk} \leqslant |x_{ik}-x_{jk}|,\quad y_{ijk} \leqslant 1-|x_{ik}-x_{jk}|,\quad y_{ijk} \geqslant |x_{ik}-x_{jk}| - \eta_{ijk}$$
$$y_{ijk} \geqslant 1-|x_{ik}-x_{jk}| - 1 + \eta_{ijk} = -|x_{ik}-x_{jk}| + \eta_{ijk},\ \eta_{ijk} \in \{0,1\}.$$



Similar approach – to "exclude" $|x_{ik}-x_{jk}|$

B&B solver: SCIP and ParaSCIP, Zuse Institute Berlin

Parallel implementation of B&B solver **SCIP**, **http://scip.zib.de**, and MPI for HPC environments, **http://ug.zib.de**

SCIP is distributed under the ZIB Academic License: all sources are available; free use SCIP&ParaSCIP for research purposes for members of non-commercial and academic institution.

UG (**Ubiquity Generator**) is a framework to parallelize B&B solvers in a distributed or shared memory computing environment.

ParaSCIP = UG[SCIP, MPI], *FiberSCIP=UG[SCIP, **Pthreads**]*, *ParaXpress*=UG[Xpress, MPI],…

Yuji Shinano, Tobias Achterberg, Timo Berthold, Stefan Heinz, Thorsten Koch, *ParaSCIP -- a parallel extension of SCIP*, **2012**

OAK Ridge National Laboratory, USA

- Titan, Cray XK7, ~500000 cores, http://www.olcf.ornl.gov/titan

Successful run of ParaSCIP with 80 000 cores.

Memory consumption – one of the main SCIP bottleneck

For problems with polynomials SCIP consumes a lot of memory to store nodes of B&B search tree

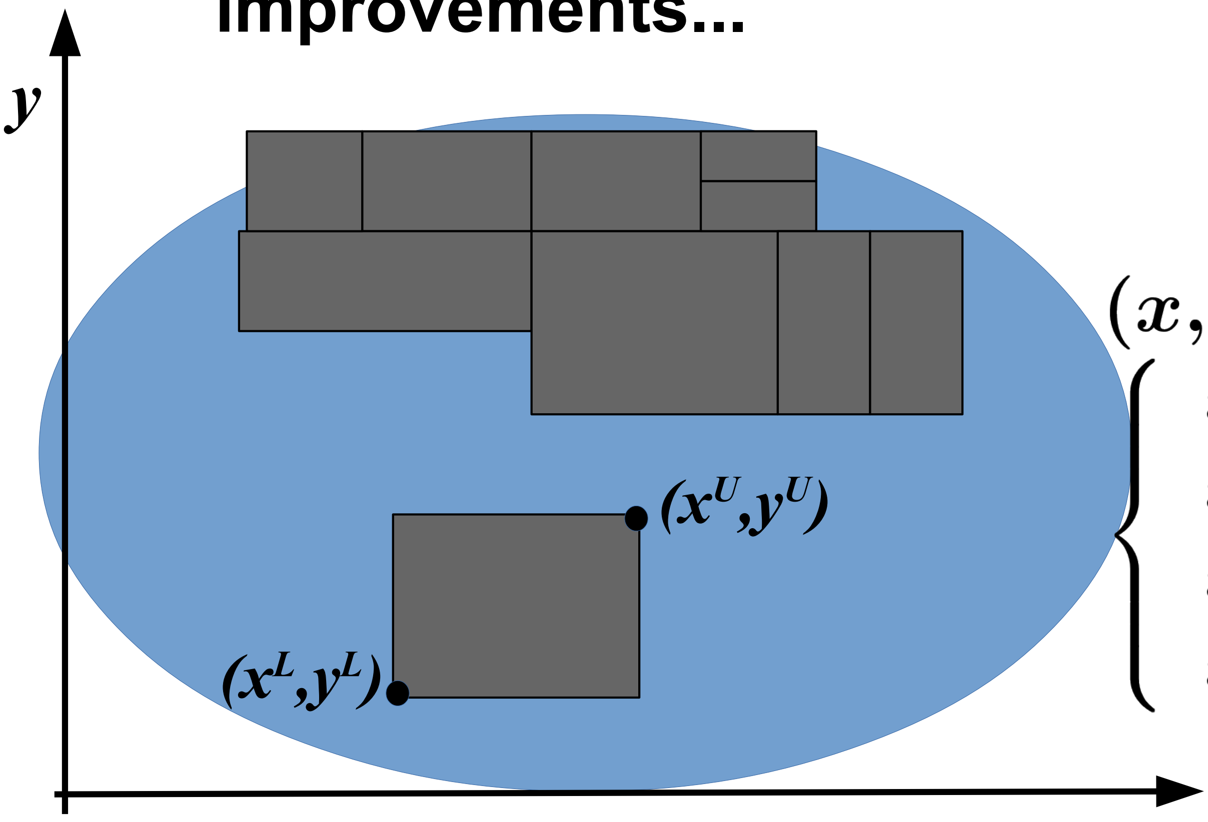
```
 time | node   | left  |LP iter|**| MEM |****|  dualbound    | primalbound   | gap
****************************************************************************
****************************************************************************
1010m| 57590k|  1885k|222559k|**|  15G|****|-1.623906e-01 |-1.339752e-01 |21.21%
1011m| 57620k|  1881k|222644k|**|  15G|****|-1.621946e-01 |-1.339752e-01 |21.06%
******************************************************************************
1012m| 57730k|  1868k|222951k|**|  15G|****|-1.617647e-01 |-1.339752e-01 |20.74%
1013m| 57760k|  1856k|223021k|**|  15G|****|-1.615385e-01 |-1.339752e-01 |20.57%
******************************************************************************
******************************************************************************
1171m| 69120k|  6371 |252397k|**|  15G ****|-1.406250e-01 |-1.339752e-01 | 4.96%
******************************************************************************

SCIP Status        : problem is solved [optimal solution found]
Solving Time (sec) : 70241.18
Solving Nodes      : 69133338
Primal Bound       : -1.33975222726390e-01 (41 solutions)
Dual Bound         : -1.33975222726390e-01
Gap                : 0.00 %
```

Excerpt of SCIP log when solving FTPP problem with N=8.

Convex relaxation to get lower bound on “hyper-rectangles”

Garth P. McCormik envelopes (1976) and their improvements...



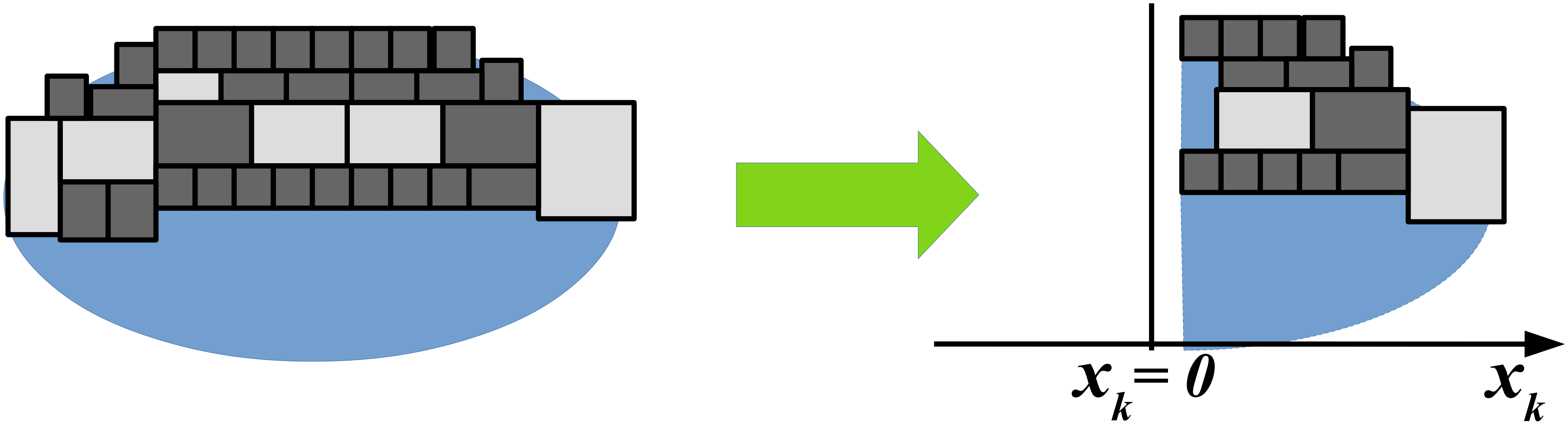
$$(x,y)\in[x^L,x^U]\times[y^L,y^U]:\quad \begin{cases} xy \geqslant x^L y + y^L x - x^L y^L \\ xy \geqslant x^U y + y^U x - x^U y^U \\ xy \leqslant x^L y + y^U x - x^L y^U \\ xy \leqslant x^U y + y^L x - x^U y^L \end{cases}$$

The more rectangles (smaller ones!), the tighter approximation we have, the less the BnB gap!

The more rectangles – the more memory to store them!

Extra constraints to reduce feasible domain volume

Reducing the volume domain by half gives "twice" less BnB search tree (rectangle partitions covering domain)



$x_{11}=0.5, x_{12}=0,$ - **anti-shifting, the 1st point is fixed (0.5,0)**

$x_{i2} \geqslant x_{(i-1)2} (i=2:N),$ - **anti-renumbering, 2nd coordinate increases**

$x_{11} \leqslant x_{21}.$ **(**) - anti-mirroring w.r.t. vertical axis**

Addition of above simple constraints give reasonable speed-up of Branch-and-Bound algorithm. E.g. last inequality gives almost double acceleration (some detailes further).

Clusters where we run ParaSCIP

From Russian Top50

Lomonosov, MSU, https://users.parallel.ru/wiki/pages/22-config

HPC4, NRC "Kurchatov Institute", http://computing.nrcki.ru

Govorun, JINR, Dubna, http://hlit.jinr.ru/about_govorun/cpu_govorun

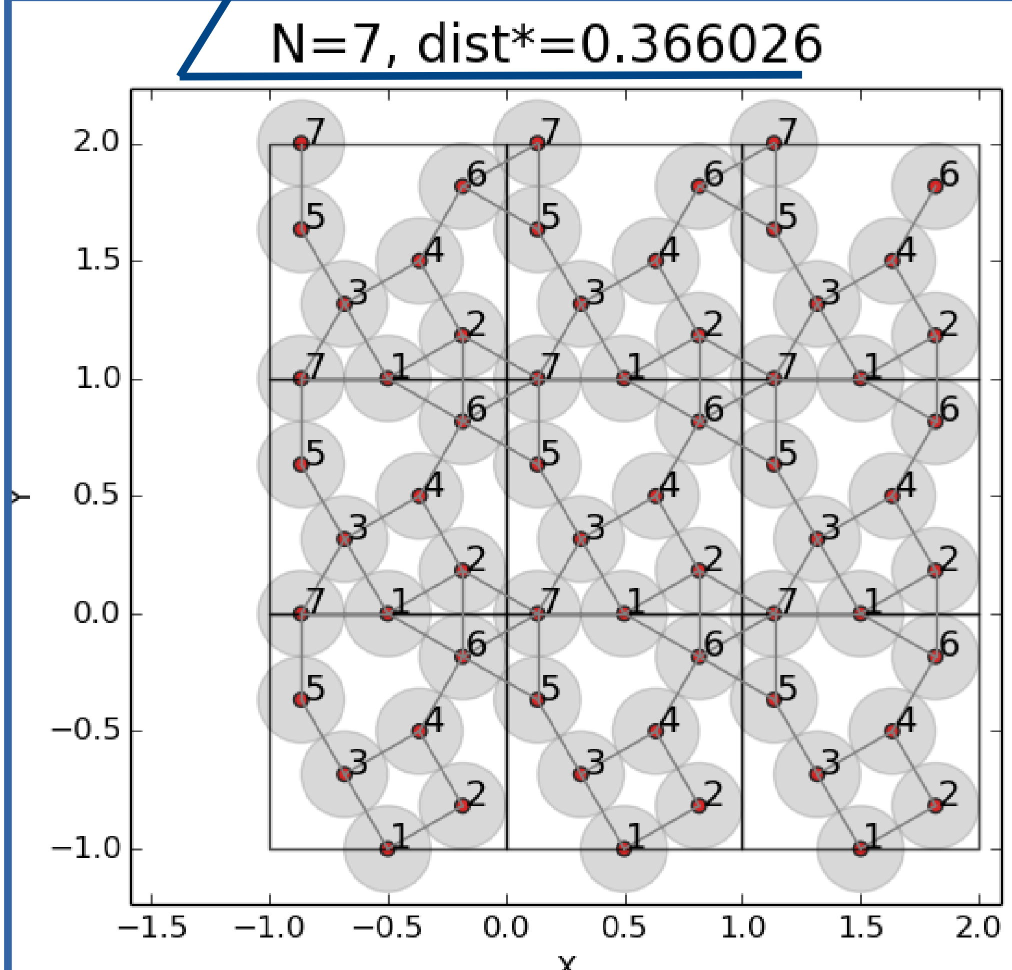
Cluster	Partition	CPU (Intel Xeon)	CPU cores/node	mem/node
Lomonosov, CentOS 6.9	regular4	X5570 2.93 GHz	8	12 Gb
HPC4 CentOS 6.7	hpc4-3d	E5-2680 v3 2.5 GHz	24	128 Gb
Govorun Sci. Linux 7.6	skylake	Gold 6154 3.7 GHz	36	192 Gb

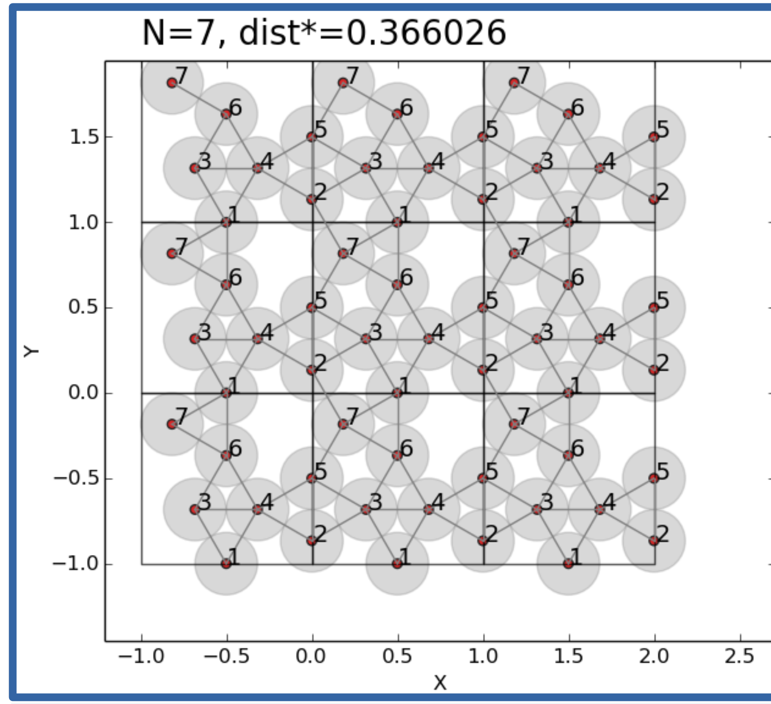
FTPP N=7, three different optimal arrangements

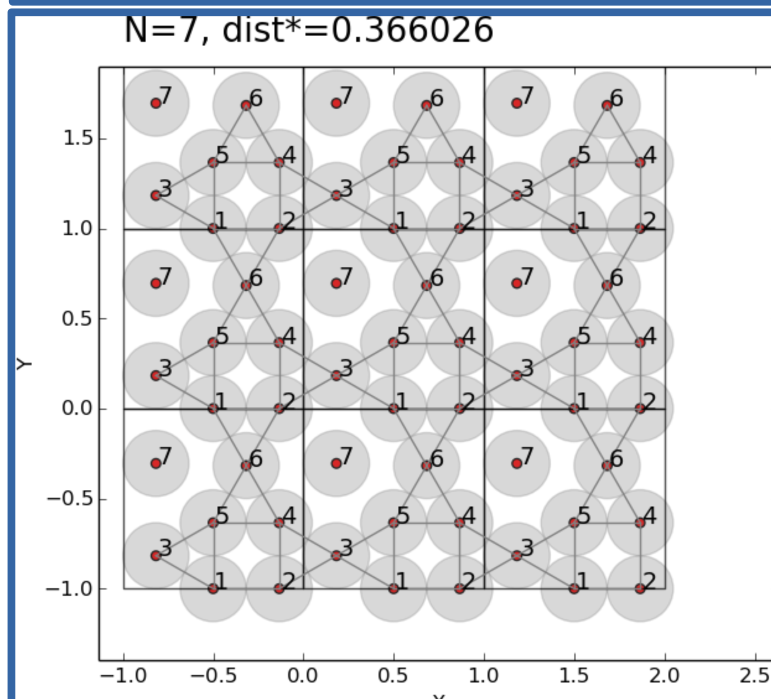
The case **N=7, 2550 sec (43 min) by standalone SCIP**, gives three different (up to isometric transformation) solutions

0.366026...

$$d^{opt} = \frac{1}{1+\sqrt{3}}$$

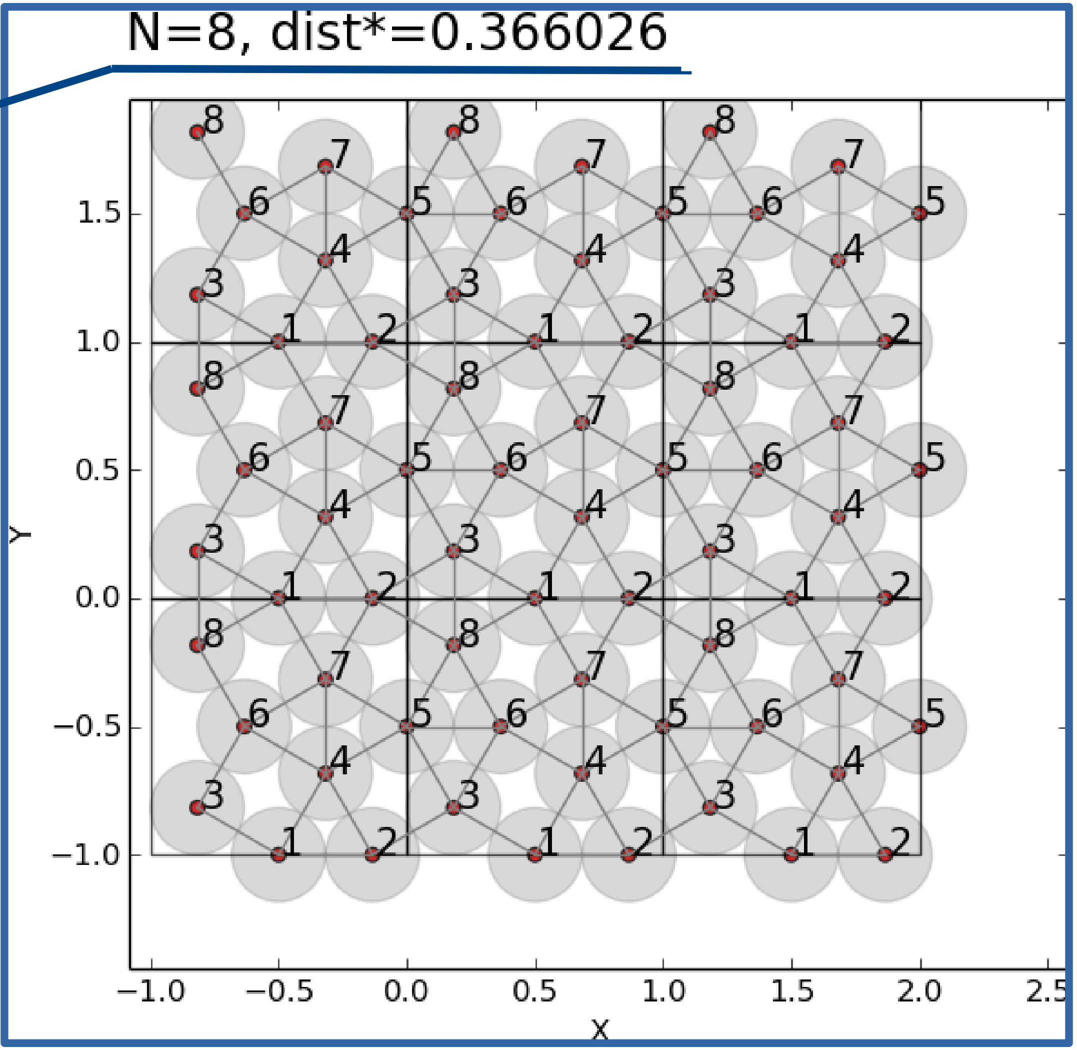






FTPP, N=8, SCIP vs ParaSCIP NRC “Kurchatov institute”

0.366026...



$$d^{opt} = \frac{1}{1+\sqrt{3}}$$

SCIP, 1CPU, 1 solver, 780 min

ParaSCIP, 8 CPUs, 7 solvers, 126 min
Efficiency (CPU): 780/126/8 = 0.77
Efficiency (solvers): 780/126/7 = 0.88

More experiments N⩽8

Dramatic growth of solving time, sec., when N increases (SCIP).

	$N\!=\!5$	$N\!=\!6$	$N\!=\!7$	$N\!=\!8$
FTPP	30, desk.	118, desk.	2552, desk.	70240, serv.
FTPP with (**)	–	66, lapt.	1940, lapt.	27230, serv.

desktop (CPU Intel i7-6700 3.40GHz, 32Gb);
laptop (CPU Intel i5-8250U 3.40GHz, 8Gb;
server (CPU 2xIntelXeon 5620 2.4Ghz, 32Gb)

Solving times for N=8, for SCIP and ParaSCIP at
HPC4, NRC "Kurchatov Institute"; efficiency (CPU) 0.77
Lomonosov, MSU 2.3 times ($=\frac{19}{65}\times\frac{64}{8}$) "slower" than HPC4

	CPU cores	solving time, min
SCIP, HPC4	1	780
ParaSCIP, HPC4	8 (7 solvers)	126
ParaSCIP, HPC4, constraint (**)	8 (7 solvers)	65
ParaSCIP, Lomonosov, –"– (**)	64 (63 solvers)	19

(**) means presence of additional inequality $x_{11} \leqslant x_{21}$ (only one!)

FTPP N=9, ParaSCIP, HPC4, Govorun

Cluster	with (**)	nodes	CPU cores	time, min
HPC4	no	8	128	956
HPC4	yes	8	128	480
Govorun	yes	6	216	108

"with (**)" means presence of additional inequality $x_{11} \leqslant x_{21}$

Flat Torus Packing Problem as MINLP, complexity, N=9:

HPC4: $128 \times 480/60 = 1024$ CPU×hours

Govorun: $216 \times 108/60 = 389$ CPU×hours

Numerical proof of conjecture from the article (as analytic formula): Oleg Musin, Anton Nikitenko. Optimal packings of congruent circles on a square flat torus. *Discrete & Computational Geometry*, 55(1):1–20, 2016.

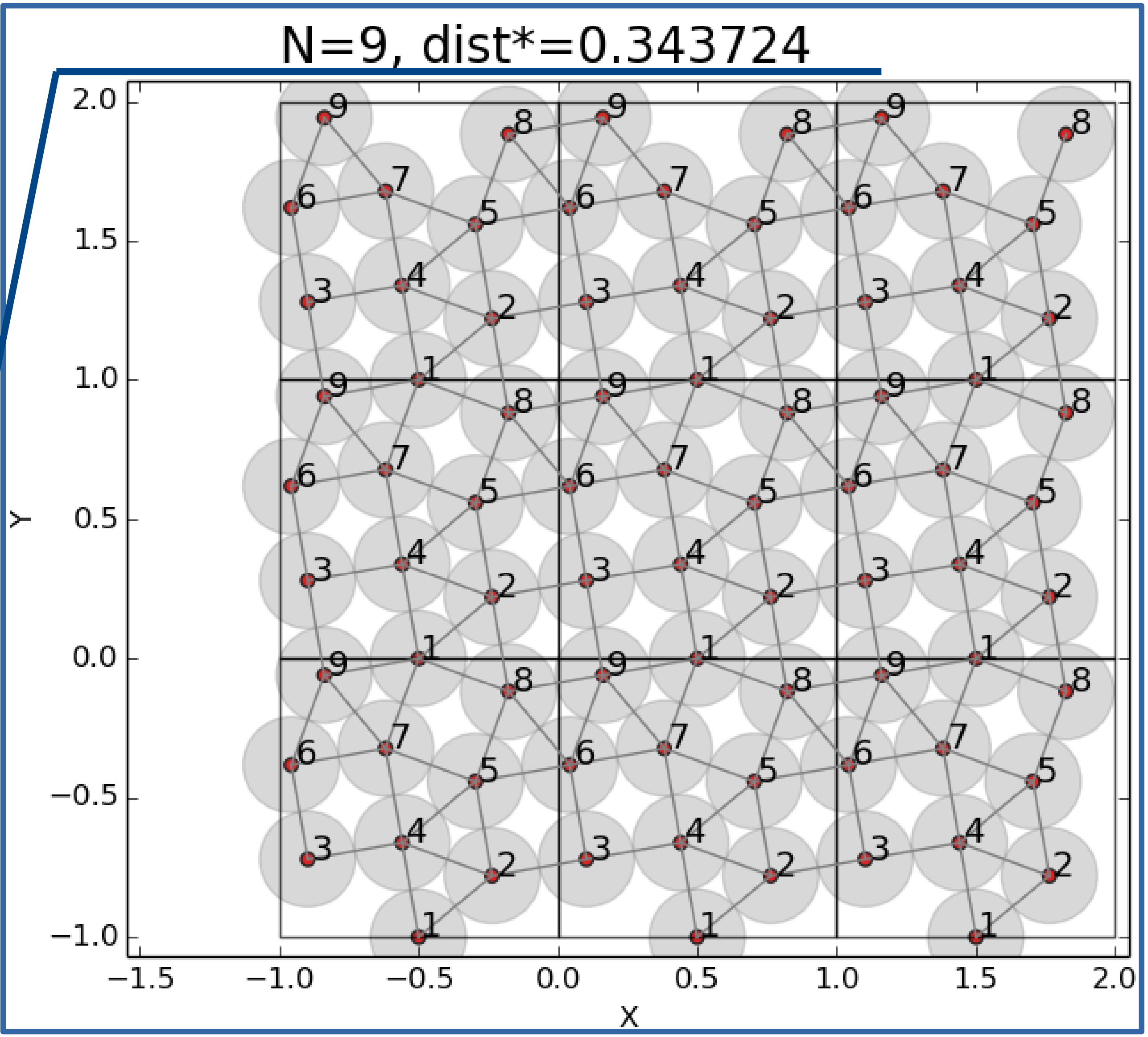
FTPP N=9, ParaSCIP

HPC4:
1024 CPU×hours

Govorun:
389 CPU×hours

0.3437238385...

$$d^{conj} = \frac{1}{\sqrt{5+2\sqrt{3}}}$$



Numerical proof of conjecture from the article (as analytic formula): it is irreducible contact graph and there is no any arrangement with distance by 1.e-6 % better. Constraints may be violated by 1.e-6

Conclusion remarks

- **General purpose branch-and-bound solver may help with hard problems in combinatorial geometry formulated as MINLP** (we also have some experience with Tammes and Thomson problems)
- **Confirmation of semi-empirical rule: simple auxiliary constraints reducing the volume of feasible domain reduce solving time of NLP by B&B-algorithm that uses McCormik envelopes** (for spatial arrangements optimization)
- **Rather moderate solving time of FTPP with 9 circles (389 CPU×hours on Govorun cluster) gives hope for solving this problem for N=10 (approx ~4000 CPU×hours) not in far future.**
- **MINLP solver SCIP and its parallel implementation, ParaSCIP, have been successfully built and used on three clusters from Russian Top50** (Lomonosov MSU, HPC4 KIAE, Govorun JINR)

At the end

- ParaSCIP 6.0.0 is installed at Lomonosov as a package, everyone can try (module av; module add ParaSCIP)

Moderate future plan (regarding FTPP)

- To try Flat Torus Packing problem with circles of different diameters (model is implemented already); N=10 !?

Authors thank:

- Alexey Tarasov for the advice to try Flat Torus Packing as an example of global optimization problem;
- Stefan Vigerske and Yuji Shinano, ZIB.DE, for consultations on the use of SCIP and ParaSCIP solvers.

Thank you for your attention.

Questions?

Tammes problem (2)

Formulated as global optimization with bilinear functions

$$z \to \min_{x_i \in \mathbb{R}^3,\ i=1:N},$$

$$x_i^\mathsf{T} x_j = \sum_{k=1:3} x_{i,k} x_{j,k} \leqslant z, \quad (1 \leqslant i < j \leqslant N)$$

$$\|x_i\|^2 = \sum_{k=1:3} (x_{i,k})^2 = 1, \quad (i=1:N)$$

$$x_{1,1}=x_{1,2}=0,\ x_{1,3}=1,$$
$$x_{2,1}=0,\ x_{2,1} \geqslant 0,$$
$$x_{i+1,3} \leqslant x_{i,3} \ (1 \leqslant i \leqslant N-1)$$

Auxiliary, antisymmetric, constraints:
- set **1st** point equal to a “pole” (0,0,1);
- **2nd** point lies in ZOY plane, “positively”
- “anti-renumeration”, **3d** coordinate in ascending order.

Non-convex problem; no integer variables, many local optimums.

SCIP solver supports non-convex problems with polynomial functions in constraints.

Tammes problem (“Kissing problem” generalization) (1)

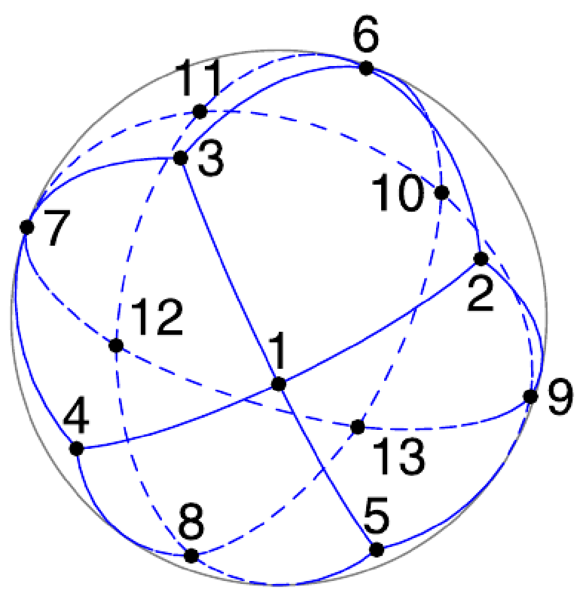
Tammes problem (optimal packing of circles on a sphere): arrange N points on a unit sphere to maximize minimal pairwise distance

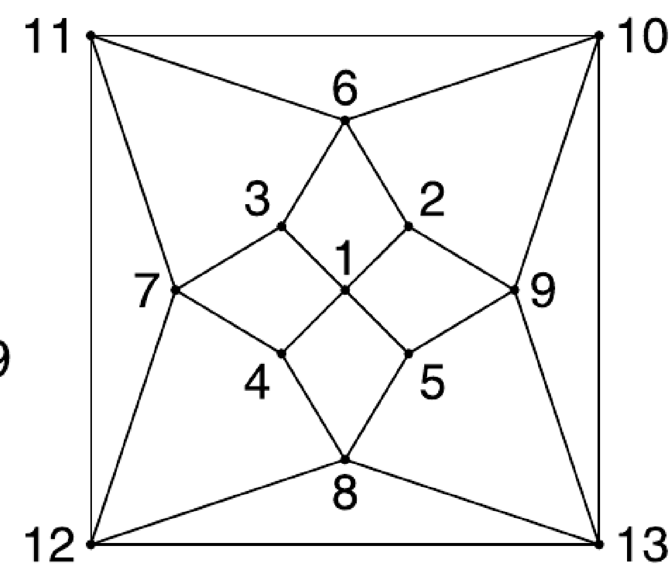
$$\min_{1\leqslant i<j\leqslant N}\{\|x_i - x_j\|\} \to \max_{x_i\in\mathbb{R}^3,\ i=1:N} \quad s.t.:\ \|x_i\|=1\ (i=1:N).$$

Oleg Musin, Alexey Tarasov: (**2012**) *The strong* ***thirteen spheres problem***. Discrete & Computational Geometry, 48(1):128–141; (**2015**) *The Tammes Problem for* ***N=14****, Experimental Mathematics*, 24:4, 460-468

It is not difficult to make a conjecture about optimal arrangement, but it is very difficult to prove that is global optimum!

The optimal configuration of 14 points was conjectured more than 60 years ago, but **computer-assisted proof** has been done on 2015 **by enumeration of the irreducible contact graphs.**





Super

$$f_o(x) \to \max_x,$$
$$x = (x_B, x_C) \in Q, x_B \in \{0,1\}^{n_B}, x_C \in \mathbb{R}^{n_C}$$

(P)

$$Q = \left\{ f_i(x_B, x_C) \leqslant 0 (i \in \mathrm{I}), g_j(x_B, x_C) = 0 (j \in \mathrm{J}) \right\}$$
$$= \text{may be something else} \ldots$$

One of the algorithms is Branch-and-Bound (B&B) known from ~1960 (Land Al. H., Doig A. G. and etc.)

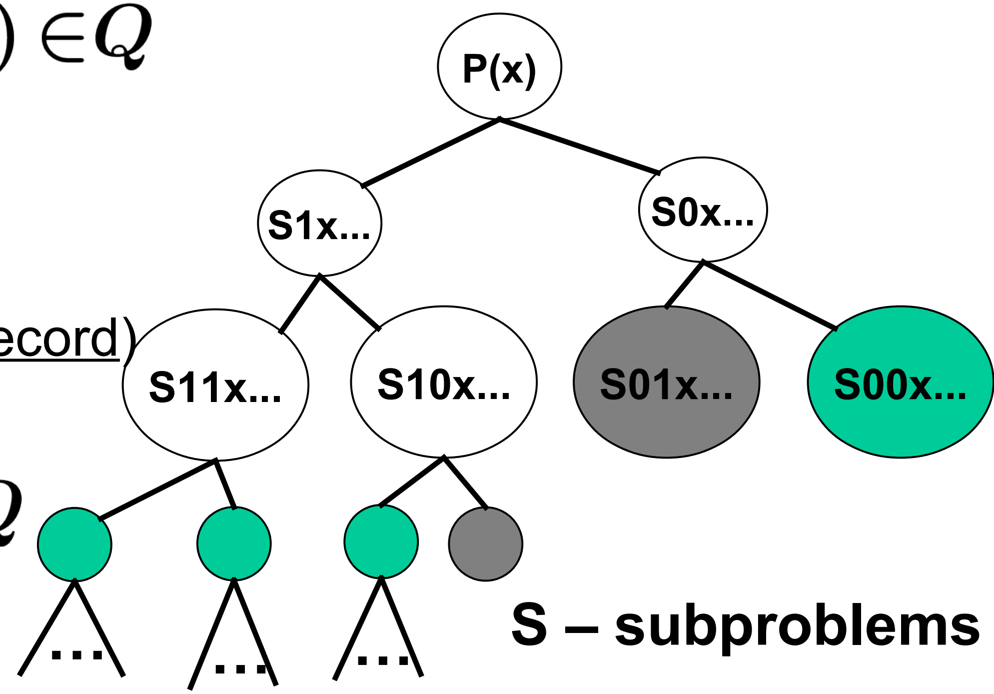
VERY briefly, B&B based on two interacting procedures:

Building the Search Tree	**Pruning Branch, Get Incumbents**
Recursive decomposition of feasible domain (Q), e.g. by fixing some x_B variables in accordance with some rules (branching)	Get lower bounds of obj. value for domain subsets; search feasible solutions $x' \in Q$ and keep the best ones, aka <u>incumbents</u> $f_o(x')$

Branch-and-bound for MI... problem (e.g. boolean)

General scheme of search tree traversing for problem (P)

$$f_o(x_B, x_C) \to \min_{(x_B, x_C)}, (x_B, x_C) \in Q$$



Current state of B&B (changed dynamically):

- list of nodes to be processed (green);
- upper-bound (**UB**) on MIN (aka <u>incumbent</u> | <u>record</u>)

$$\mathrm{UB} = f_o(x'_B, x'_C), (x'_B, x'_C) \in Q$$

S – subproblems

Node operation:

1) **calculate lower-bound of S, LB(S), by relaxation** of boolean and/or non-convex constraints to, e.g. LP or convex MINLP;

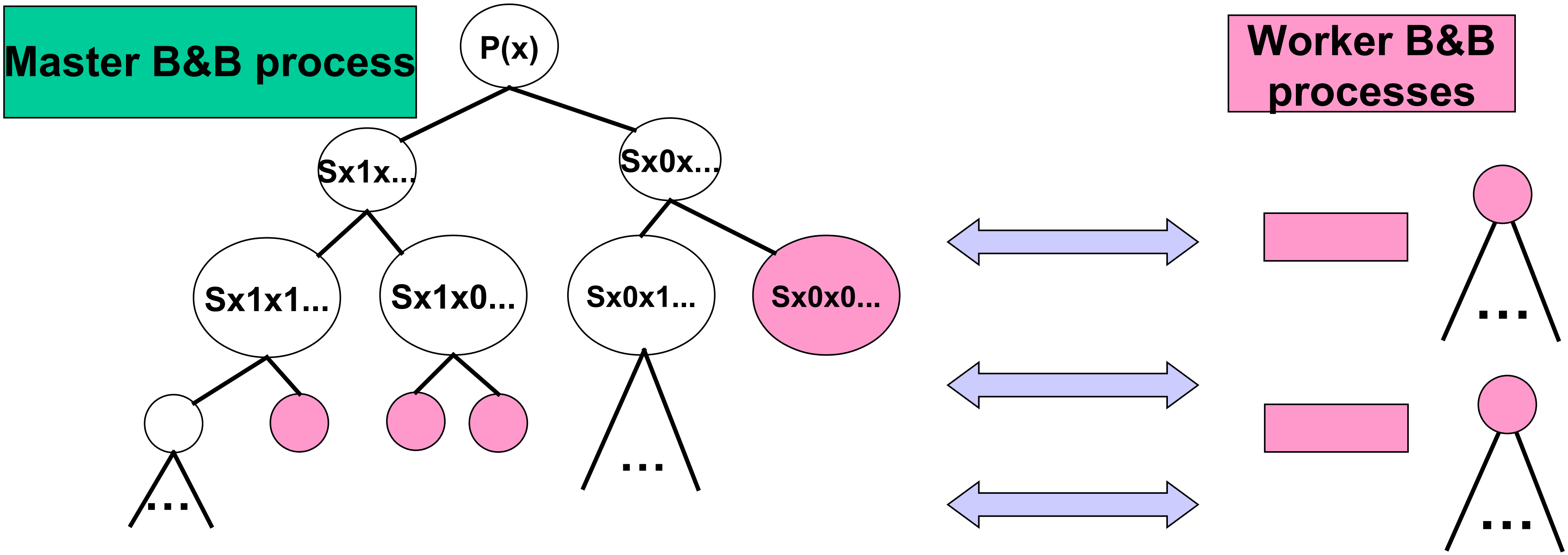
2) **if feasible vector was found** $\left(x''_B, x''_C\right) \in Q$ – update UB

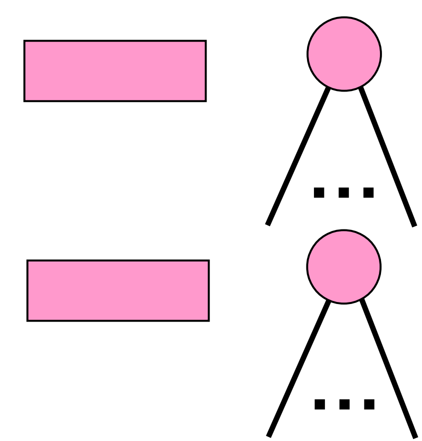
$$\mathrm{UB} := \min\{\mathrm{UB}, f_o(x''_B, x''_C)\}$$

3) **if LB(S) ⩾ UB – discard node** from the list (gray, “pruning” branch);

4) **select boolean variable (or add inequality with continuous variables) to decompose the node** and add new ones to the tree

Fine-grained parallelization of B&B (traditional approach)





Master & workers exchange with <u>subtrees</u> and <u>incumbents.</u>

Subtrees (subproblems) are generated dynamically

Usually, this approach is based on MPI and runs on high-performance clusters.

Rather intensive data flow.

Implementation requires low-level programming

Subtotal & Next topic DDBNB for Global Optimization

1) Смирнов С.А., Волошинов В.В. *Эффективное применение пакетов дискретной оптимизации в облачной инфраструктуре на основе эвристической декомпозиции исходной задачи в системе оптимизационного моделирования AMPL* // Программные системы: теория и приложения, No 28, 2016, с. 29–46
2) Волошинов В.В., Смирнов С.А.. *Оценка производительности крупноблочного алгоритма метода ветвей и границ в вычислительной среде Everest* // Программные системы: теория и приложения, т. 8, № 1, 2017, 105–119
3) Voloshinov V., Smirnov S. and Sukhoroslov, O., *Implementation and Use of Coarse-grained Parallel Branch-and-bound in Everest Distributed Environment* // Procedia Computer Science, 108, 2017, pp. 1532-1541
4) Smirnov S., Voloshinov V. *Implementation of Concurrent Parallelization of Branch-and-bound algorithm in Everest Distributed Environment* // Procedia Computer Science, 119, 2017, pp. 83–89

What else? Global optimization problems with examples of combinatorial geometry problems.

ParaSCIP, Flat Torus Packing N=7, 9

Our success story with ParaSCIP:

1. Running on HPC4/HPC5, NRC "Kurchatov Institute", ~22 000 cores, T-Platforms (458 in World Top500, 4 in Russia Top50)

KIAE has CentOS 6 with GCC 4.4 and doesn't support C++11 extensions required by SCIP.
Workaround: take another host with a similar CentOS version and devtoolset-7 (GCC 7.3); build solvers; copy them to the cluster.
It might give not optimal code due to difference in CPU architecture (unknown to compiler!)

2. Computer aided confirmation of one open problem in Combinatorial Geometry: Packing Flat Torus with N=9 congruent circles.