VŠB TECHNICKÁ

VSB TECHNICAL



www.vsb.cz

Nice Application of Complete Graph Decomposition

Petr Kovář

joint work with

Dalibor Lukáš, Tereza Kovářová, Michal Kravčenko, Michal Merta

VSB - Technical University of Ostrava, Czech Republic

Seminár z teórie grafov, 11.3. 2021, Bratislava



- 2 Graph theory formulation
- 3 Equivalent formulations
- 4 Known results
- 5 Main result
- 6 A couple of related results and approximations

կլլ

When solving real life problems

- **a** large $n \times n$ matrix, $n \dots$ millions
- numerical method

When solving real life problems

- **a** large $n \times n$ matrix, $n \ldots$ millions
- numerical method
- parallelize the computation
- N processes, $N \times N$ blocks (submatrices) B_{ij}

$$\begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N1} & B_{N2} & \cdots & B_{NN} \end{pmatrix}$$

When solving real life problems

- **a** large $n \times n$ matrix, $n \ldots$ millions
- numerical method
- parallelize the computation
- N processes, $N \times N$ blocks (submatrices) B_{ij}

$$\begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N1} & B_{N2} & \cdots & B_{NN} \end{pmatrix}$$

We have a dense matrix!

When solving real life problems

- **a** large $n \times n$ matrix, $n \ldots$ millions
- numerical method
- parallelize the computation
- N processes, $N \times N$ blocks (submatrices) B_{ij}

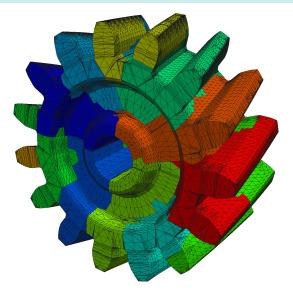
$$\begin{pmatrix} B_{11} & B_{12} & \cdots & B_{1N} \\ \vdots & \vdots & \ddots & \vdots \\ B_{N1} & B_{N2} & \cdots & B_{NN} \end{pmatrix}$$

We have a dense matrix!

- \blacksquare distribute N blocks to each processor
- (geometrically) closely related blocks to the same processor

FEM & BEM





Many points, split into N machines.

IμI

Parallel machine without shared memory.

We prefer parallelization

- load balanced
- memory balanced

կլլ

Parallel machine without shared memory.

We prefer parallelization

- load balanced
- memory balanced

Computation of block B_{ij} , requires

- ∎ *i*-th and
- *j*-th parts of the geometry

կլլ

Parallel machine without shared memory.

We prefer parallelization

- load balanced
- memory balanced

Computation of block B_{ij} , requires

- ∎ *i*-th and
- *j*-th parts of the geometry

To each CPU as few different indices as possible. (e.g. not all blocks from one row)

կլլ

Parallel machine without shared memory.

We prefer parallelization

- load balanced
- memory balanced

Computation of block B_{ij} , requires

- ∎ *i*-th and
- *j*-th parts of the geometry

To each CPU as few different indices as possible. (e.g. not all blocks from one row)

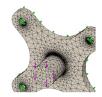
Not all N^2 blocks fit into the memory of one CPU! (nor all n^2 elements of the matrix)

Block matrix with numbers related to difficulty

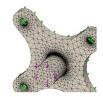
	14	100	18	8 11	7 17										
	25	17 8	13 20	14	<u> </u>	16	13	14	14	10	9	13	10	13	10
		16	1 H H 1 H H	18		15 12	19	14	14 11	13 9	12	11 8 FT7 18	13	8 10 14 10	12
15 6			20	17		1918	10	· · ·	14 141 17 13	194111	15	141118	10	11410 114	12
21 11	15 15		20	13	11	11	11	18		13	14	8	11	13	10
17 00 5		21		17	15 18 17 18 X	13 10 #11611	12	19 14		16 13	16	11 10	11	9 7 14 14 13 14	13
23 5 12 -	* * *	~~! H			17 福台	10 I	21 12			14 24 18		13#1			
12 20	1212	10	11		14	17	- 22	9	12	11	9	14	9	14	12
18 1	1000	14	13 14	19	26	10.00	15 17	17	7 12	11 8	14	14 10	12	10 13	15
10 20			1/17	H ''	last in		40.00		9 田	12		16 H 16		16	
19	7 14	12	11 14	15 18 20			19 20	11	10 12	10 ¹¹ 15 10	12	18 4 13	14		13 18
14	13	11	10	19	17	22		11	12	9	11	13	15		14 8
		18		13 10		16			. –	-	17 8			16	16
17	10		13 14	10	12	12	9		18	23	13	15	16	13	9
19	11 15	15													
	1.01111			10	9 14	14 10	12			4	18		17	12 9	10
	19 10 10 18 10			10	18 8	# 9 10 3 # 10	12	27 17			19 14	17		1 12	10
14			18 <mark>-11</mark> 18-11 13-14	10 8	18 8	H9 .	12 13	27 17 18 15 15 11	The states		19 14 18	17 211 17	17 18 18	11 11 11 11 11 11 11 12 11 12 11 12	10 15
14 11	11 18		16 16	10 8 9	18 18 18 18 18 1 9 15 1	11 9 10 3 11 13 12 13		18 15 15	16		19 14	117 211 17 18 18 18	16 18	11 11 11 11 11 11 11 12 11 12 11 12	
11	9 9 9	∞ 16 15	18 14 13 14	8 9	18 18 1 9 15 1 8 12 12 12	10 3 11 13 11 11 10 13 11 10 13 13 11 10	13 9	18 15 15 11 22 11 11 16	16 18 11 10		19 14 18 -	日 7 2 日 17 17	18 18 20 14 9 21	10 11 11 11 11 11 17 11 17 11	15 15
· · ·	11 11 9 8	∞ 16	18 <mark>18</mark> 18 <mark>18</mark> 13 14	8 9	18 111 18 12 9 15 1 8 12	10 3 11 13 11 11 10 13 11 10 13 13 11 10	13 9 13	18 15 15	18 11	20	19 14 18	18 17 18 18 18	18 18 220 14 9 21 21	10 11 11 11 11 11 17 11 17 11 11	15 15 18 19
11	11 16 9 6 9 16 16	∞ 16 15	10 <mark>16 1314 1016 1314 10 3 11</mark>	8 9	18 12 9 15 12 8 12 12 13 13 15 13	19 10 11 13 11 13 11 10 13 11 10 13 11 10 13	13 9 13	18 15 15 11 22 11 11 16	18 11 18 11 18 11 18 17	20 - 20 - 20 - 20 - 20 - 20 - 20 - 20 -	19 14 18 1 18 2 18 21 19 1 14 2		18 18 220 14 9 21 21	11 12 3 13 7 17 11 17 11 17 11 17 11	15 15 18 19 20 9 20
11 12 9	11 9 8 9 16 16 16 16 16 16 16 16 16 16 16 16 16	20 16 15 13 9	1011 1011 1011 3114 3114 3114 3114 1011 1011	8 9 9 10	18 11 9 15 8 12 12 13 11 13 15 13 15 13 15 13 15 17 10	1 1 1 1 1 1 1 1	13 9 13 14	18 15 15 12 22 1 11 16 15 15	16 16 16 11 18 17 12 16 15		19 14 18 1 18 2 18 21 19 7 14 2 7 47	18 17 18 18 18	18 18 220 14 9 21 21	167 12 3 1 3 17 11 17 11 17 11 18 17 18 17 18 17	15 15 18 19 20 11 9 20
11 12	11 11 9 6 9 16 16 16 16 16 16 16 16 16 16	∞ 16 15 13	1011 1016 1016 1011 1011	8 9 9 10	18 18 19 14 9 15 14 8 12 12 13 15 14 13 15 14 6 17	1 1 1 1 1 1 1 1	13 9 13 14	18 15 15 11 22 11 11 16 15	16 16 16 11 18 17 12 16 15	20 - 20 - 20 - 20 - 20 - 20 - 20 - 20 -	19 14 18 1 18 2 18 21 19 7 14 2 7 47		18 18 20 14 9 21 21 19	167 110 117 117 117 117 117 117 11	15 15 18 19 20 11 9 20
11 12 9	9 11 11 9 9 9 10 10 11 9 8 12 14 14 11 11 10 11 10 11 10 11 10 11 10 11 10 11 10 11 10 11 10 10	20 16 15 13 9	10 10 10 16 13 14 10 1 10 1 10 10 11 12 13 11 12 13 11 12 13 11 12 13 14	8 9 9 10	18 11 9 15 8 12 12 13 11 13 15 13 15 13 15 13 15 17 10	19 11 12 11 13 11 13 11 13 11 13 11 13 11 15 16 11 15 16 11 15 16 11 15 16 17 18 18 19 10 10 10 10 10 10 10 10 10 10	13 9 13 14	18 15 15 12 22 1 11 16 15 15	18 18 18 18 17 12 16 13 10 13 10 10 10 10 10 10 10 10 10 10	20 20 20 20 14 16 5	19 14 18 1 18 2 18 21 19 7 14 2 7 47	17 21 17 18 18 17 18 18 18 18 18 18 18 18 18 18 18 18 18	18 18 20 14 9 21 19 19 19 19 19 19 19 19 19 19 19 19 19	167 12 3 1 13 7 1 13 7 1 13 7 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1	15 15 18 19 20 11 9 20

ηп

- one diagonal block (longest computation)
- (N-1) non-diagonal blocks



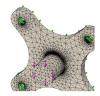
- one diagonal block (longest computation)
- $\bullet \ (N-1) \text{ non-diagonal blocks}$
- if B_{ij} then also B_{ji} (same CPU)
- if B_{ij} , B_{ik} , and B_{lj} then also B_{lk}



- one diagonal block (longest computation)
- (N-1) non-diagonal blocks
- if B_{ij} then also B_{ji} (same CPU)
- if B_{ij} , B_{ik} , and B_{lj} then also B_{lk}

This translates into

- decomposing K_N into N subgraphs G_1, G_2, \ldots, G_N
- each with (N-1)/2 edges
- each with as few vertices as possible



- one diagonal block (longest computation)
- (N-1) non-diagonal blocks
- if B_{ij} then also B_{ji} (same CPU)
- if B_{ij} , B_{ik} , and B_{lj} then also B_{lk}

This translates into

- decomposing K_N into N subgraphs G_1, G_2, \ldots, G_N
- each with (N-1)/2 edges
- each with as few vertices as possible

First suppose G_1, G_2, \ldots, G_N isomorphic to complete graph.



Graph theory formulation



Decompose K_N into N copies of K_k – dense subgraphs. Necessary condition: $N = k^2 - k + 1$.

Graph theory formulation



Decompose K_N into N copies of K_k – dense subgraphs. Necessary condition: $N = k^2 - k + 1$.

Particularly easy if the decomposition is cyclic:

Definition (Graceful labeling)

Let G be a graph with m edges and a vertex labeling $\lambda: V(G) \rightarrow \{0, 1, \dots, m\}$. The *length* of an edge xy is

$$\ell(x,y) = \min\{|\lambda(x) - \lambda(y)|, \ 2m + 1 - |\lambda(x) - \lambda(y)|\}.$$

We call f a graceful labeling if the set of edge lengths $\{\ell(x,y) : xy \in E(G)\} = \{1, 2, \dots, m\}.$



Graceful and ρ -labeling

The famous Graceful Tree Conjecture: "All trees graceful."

Graceful and ρ -labeling

կլլ

The famous Graceful Tree Conjecture: "All trees graceful."

Other graphs are interesting as well

• K_k is graceful iff $k \leq 4$

Graceful and $\rho\text{-labeling}$

կլլ

The famous Graceful Tree Conjecture: "All trees graceful."

Other graphs are interesting as well

- K_k is graceful iff $k \leq 4$
- unicyclic graphs

Graceful and ρ -labeling

կլլ

The famous Graceful Tree Conjecture: "All trees graceful."

Other graphs are interesting as well

- K_k is graceful iff $k \leq 4$
- unicyclic graphs

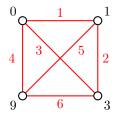
Definition (ρ -labeling)

Let G be a graph with m edges nad a vertex labeling $f: V(G) \rightarrow \{0, 1, \ldots, 2m\}$. The *length* of xy is $\ell(x, y) = \min\{|\lambda(x) - \lambda(y)|, 2m + 1 - |\lambda(x) - \lambda(y)|\}$. We say λ is a ρ -labeling labeling if the set of edge lengths $\{\ell(x, y) : xy \in E(G)\} = \{1, 2, \ldots, m\}$.



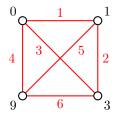
Theorem (Rosa 1967)

A graph G with m edges allows a cyclic decomposition of K_{2m+1} iff G has a ρ -labeling.



Theorem (Rosa 1967)

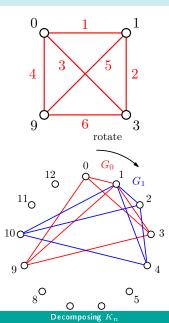
A graph G with m edges allows a cyclic decomposition of K_{2m+1} iff G has a ρ -labeling.



- Not all complete graphs have a ρ-labeling,
- infinitely many complete graphs have.

Example





Equivalent formulation



Definition (perfect difference sets)

A set of integers $\{a_1, a_2, \ldots, a_k\} \subseteq [0, N]$ such that every nonzero residue modulo N can be uniquely expressed in the form $a_i - a_j$.

Definition (perfect difference sets)

A set of integers $\{a_1, a_2, \ldots, a_k\} \subseteq [0, N]$ such that every nonzero residue modulo N can be uniquely expressed in the form $a_i - a_j$.

Example

 $\{0, 1, 4, 6\}$ is a perfect difference set for m = 6: 1 = 1 - 0, 2 = 6 - 4, 3 = 3 - 1, 4 = 4 - 0, 5 = 6 - 1, 6 = 6 - 0.

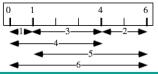
Definition (perfect difference sets)

A set of integers $\{a_1, a_2, \ldots, a_k\} \subseteq [0, N]$ such that every nonzero residue modulo N can be uniquely expressed in the form $a_i - a_j$.

Example

 $\{0, 1, 4, 6\}$ is a perfect difference set for m = 6: 1 = 1 - 0, 2 = 6 - 4, 3 = 3 - 1, 4 = 4 - 0, 5 = 6 - 1, 6 = 6 - 0.

Perfect ruler (Guy 1994) has k distinct marks s.t. any distance $1, 2, 3, 4, \ldots, N$ can be measured. E.g. 0, 1, 4, 6



Based on the perfect difference set $\{0, 1, 4, 6\}$ for m = 6 we can decompose $K_{2m+1} = K_{13}$ cyclically.

Based on the perfect difference set $\{0, 1, 4, 6\}$ for m = 6 we can decompose $K_{2m+1} = K_{13}$ cyclically.

- label vertices of K_4 by 0, 1, 4, 6
- find a cyclic decomposition of K_{13}

Based on the perfect difference set $\{0, 1, 4, 6\}$ for m = 6 we can decompose $K_{2m+1} = K_{13}$ cyclically.

- label vertices of K_4 by 0, 1, 4, 6
- find a cyclic decomposition of K_{13}

A 13 by 13 block matrix to 13 machines, so that:

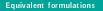
- 1 computation of diagonal block + 12 non-diagonal blocks to each process
- 4 rows (and 4 columns) of the geometry to each process

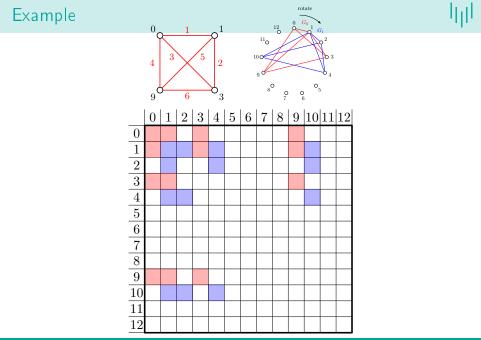
Based on the perfect difference set $\{0, 1, 4, 6\}$ for m = 6 we can decompose $K_{2m+1} = K_{13}$ cyclically.

- label vertices of K_4 by 0, 1, 4, 6
- find a cyclic decomposition of K_{13}

A 13 by 13 block matrix to 13 machines, so that:

- 1 computation of diagonal block + 12 non-diagonal blocks to each process
- 4 rows (and 4 columns) of the geometry to each process
- the higher N (N = 13) the better ratio





Decomposing a complete graph into complete subgraphs – this seems to be a design theory problem.

Definition

A block design (BIBD) is a collection B of b subsets (called blocks) of a finite set X of v elements such that any element of X is contained in the same number r of blocks, every block has the same number k of elements, and each pair of distinct elements appear together in the same number λ of blocks.

A symmetric BIBDs are (also known as 2-designs) are denoted as $2 - (v, k, \lambda)$ designs has b = v. ($b \ge v$ by Fisher's inequality.)

Decomposing a complete graph into complete subgraphs – this seems to be a design theory problem.

Definition

A block design (BIBD) is a collection B of b subsets (called blocks) of a finite set X of v elements such that any element of X is contained in the same number r of blocks, every block has the same number k of elements, and each pair of distinct elements appear together in the same number λ of blocks.

A symmetric BIBDs are (also known as 2-designs) are denoted as $2-(v,k,\lambda)$ designs has b=v. ($b\geq v$ by Fisher's inequality.)

In our case:

$$\bullet v = N = k^2 - k + 1$$

•
$$b = N$$
 (symmetric)
• $\lambda = 1$

(ALL)



Sufficient condition: k-1 is a prime power.



Sufficient condition: k-1 is a prime power.

Theorem (Singer 1934)

A perfect difference set with k elements exists if k-1 is a prime power.





Sufficient condition: k-1 is a prime power.

Theorem (Singer 1934)

A perfect difference set with k elements exists if k-1 is a prime power.

However, K_7 does not decompose K_{43} . Neither K_{11} does not decompose K_{111} .

. . .



Sufficient condition: k-1 is a prime power.

Theorem (Singer 1934)

A perfect difference set with k elements exists if k-1 is a prime power.

However, K_7 does not decompose K_{43} . Neither K_{11} does not decompose K_{111} .

Theorem (Hartke, Östergøard, Bryant, El-Zanati 2009)

There exists no $(K_6 - e)$ -decomposition of K_{29} .

. . .



Sufficient condition: k-1 is a prime power.

Theorem (Singer 1934)

A perfect difference set with k elements exists if k-1 is a prime power.

```
However, K_7 does not decompose K_{43}.
Neither K_{11} does not decompose K_{111}.
```

Theorem (Hartke, Östergøard, Bryant, El-Zanati 2009)

There exists no $(K_6 - e)$ -decomposition of K_{29} .

Overview for k<100 on web (Baumert): <code>http://www.ccrwest.org/diffsets/diff_sets/baumert.html</code>

. . .





... is the nice application!

Ш

... is the nice application!

Construction for certain (not all) values.

ЧH

կլ

... is the nice application!

Construction for certain (not all) values.

It has been implemented and successfully tested: Fast BEM matrices of size n up to millions, distributed to hundreds of

nodes N.

կլլ

... is the nice application!

Construction for certain (not all) values.

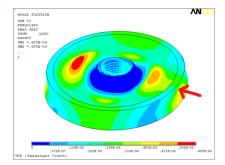
It has been implemented and successfully tested:

Fast BEM matrices of size $n\ {\rm up}$ to millions, distributed to hundreds of nodes N.

n	N=31, k=6	N=91, k=10	N=133,k=12
12 288	175 MB, 1 s	200 MB, 1 s	207 MB, 1 s
196 608	353 MB, 53 s	280 MB, 25 s	276 MB, 18 s
786 432	999 MB, 294 s	570 MB, 110 s	535 MB, 99 s
3 145 728			1911 MB, 596 s

Format: average memory [MB], CPU time per process [s]

Motivation: railway wheel noise elimination by profiling



Courtesy of J. Szweda, Department of Mechanics, VŠB – TUO

Distribution of processes to N CPUs needs not to be balanced (not isomorphic subgraphs).

Typical cluster has 2^t cores, e.g. 128.

ЧH

Distribution of processes to N CPUs needs not to be balanced (not isomorphic subgraphs).

Typical cluster has 2^t cores, e.g. 128. We know how to decompose K_{133} into 133 subgraphs K_{12} .

- Remove 5 rows and columns:
 - preferably from different K_{12} subgraphs
 - obtain some K_{12} , K_{11} , K_{10} , maybe a few smaller
 - or obtain many K_{12} , K_{11} , one K_7

Distribution of processes to N CPUs needs not to be balanced (not isomorphic subgraphs).

Typical cluster has 2^t cores, e.g. 128. We know how to decompose K_{133} into 133 subgraphs K_{12} .

- Remove 5 rows and columns:
 - preferably from different K_{12} subgraphs
 - obtain some K_{12} , K_{11} , K_{10} , maybe a few smaller
 - or obtain many K_{12} , K_{11} , one K_7

Or for 2014 cores decompose K_{993} into 933 copies of K_{31} .

Distribution of processes to N CPUs needs not to be balanced (not isomorphic subgraphs).

Typical cluster has 2^t cores, e.g. 128. We know how to decompose K_{133} into 133 subgraphs K_{12} .

Remove 5 rows and columns:

• preferably from different K_{12} subgraphs

- obtain some K_{12} , K_{11} , K_{10} , maybe a few smaller
- or obtain many K_{12} , K_{11} , one K_7

Or for 2014 cores decompose K_{993} into 933 copies of K_{31} .

A graph theorists say: That's cheating! Yet, it works...

Distribution of processes to $N\ {\rm CPUs}$ needs not to be balanced (not isomorphic subgraphs).

Typical cluster has 2^t cores, e.g. 128. We know how to decompose K_{133} into 133 subgraphs K_{12} .

Remove 5 rows and columns:

• preferably from different K_{12} subgraphs

- obtain some K_{12} , K_{11} , K_{10} , maybe a few smaller
- or obtain many K_{12} , K_{11} , one K_7

Or for 2014 cores decompose K_{993} into 933 copies of K_{31} .

A graph theorists say: That's cheating! Yet, it works...

Computational time depends on the largest graph: preferably few small and a many large dense graphs (complete or "almost" complete graphs).

կլլ

If it has a ρ -labeling, then it can be used.

١пп

If it has a ρ -labeling, then it can be used.

- all graphs with at most 11 edges have a ρ -labeling
- many classes of sparse graphs

If it has a ρ -labeling, then it can be used.

- all graphs with at most 11 edges have a ρ -labeling
- many classes of sparse graphs

By hand up to n = 31:

- ρ -labeling of a graph with (n-1)/2 edges
- exceptions $n = 28, 29, \ldots$

If it has a ρ -labeling, then it can be used.

- all graphs with at most 11 edges have a ρ -labeling
- many classes of sparse graphs

By hand up to n = 31:

- ρ -labeling of a graph with (n-1)/2 edges
- exceptions $n=28,29,\ldots$
- ... difficult (almost) as for complete graphs

If it has a ρ -labeling, then it can be used.

- \blacksquare all graphs with at most 11 edges have a ho-labeling
- many classes of sparse graphs

By hand up to n = 31:

- ρ -labeling of a graph with (n-1)/2 edges
- exceptions $n = 28, 29, \ldots$

... difficult (almost) as for complete graphs

Decomposition needs not to be cyclic

 $K_7-K_{3,3}$ decomposes K_{25} , yet no ho-labeling. reference?

 for odd N subgraphs can be isomorphic (exceptions!)
 for even N subgraphs cannot be isomorphic (parity) (simple graph not, though isomorphic digraphs exist)

 for odd N subgraphs can be isomorphic (exceptions!)
 for even N subgraphs cannot be isomorphic (parity) (simple graph not, though isomorphic digraphs exist)

Lemma

Let r, s be odd. If G decomposes K_r into r copies and H decomposes $G[\overline{K_s}]$ into s copies, then a dense graph X on |H| vertices decomposes K_{rs} into rs copies of X.

 for odd N subgraphs can be isomorphic (exceptions!)
 for even N subgraphs cannot be isomorphic (parity) (simple graph not, though isomorphic digraphs exist)

Lemma

Let r, s be odd. If G decomposes K_r into r copies and H decomposes $G[\overline{K_s}]$ into s copies, then a dense graph X on |H| vertices decomposes K_{rs} into rs copies of X.

Theorem

If $r = p^2 - p + 1$ and $s = q^2 - q + 1$, then we can decompose K_{rs} into rs dense (for small s) isomorphic subgraphs on pq vertices.

Example

Decompose K_{147} (147 = 7 \cdot 21) into 147 isomorphic subgraphs on 15 vertices.

(ALL)

 for odd N subgraphs can be isomorphic (exceptions!)
 for even N subgraphs cannot be isomorphic (parity) (simple graph not, though isomorphic digraphs exist)

Lemma

Let r, s be odd. If G decomposes K_r into r copies and H decomposes $G[\overline{K_s}]$ into s copies, then a dense graph X on |H| vertices decomposes K_{rs} into rs copies of X.

Theorem

If $r = p^2 - p + 1$ and $s = q^2 - q + 1$, then we can decompose K_{rs} into rs dense (for small s) isomorphic subgraphs on pq vertices.

Example

Decompose K_{147} (147 = 7 \cdot 21) into 147 isomorphic subgraphs on 15 vertices. (Theoretical optimum: 13 vertices.)

If N is even...



If q is even, we decompose K_{pq} into pq subgraphs H_1, H_2, \ldots, H_{pq} (not isomorphic).

If N is even...



If q is even, we decompose K_{pq} into pq subgraphs H_1, H_2, \ldots, H_{pq} (not isomorphic).

Lemma

Let p be odd and q even. If G decomposes K_p into p copies and H decomposes $G[\overline{K_q}]$ into q copies, then K_{pq} can be decomposed into pq dense subgraphs X_1, X_2, \ldots, X_{pq} each on |H| vertices.

If N is even...



If q is even, we decompose K_{pq} into pq subgraphs H_1, H_2, \ldots, H_{pq} (not isomorphic).

Lemma

Let p be odd and q even. If G decomposes K_p into p copies and H decomposes $G[\overline{K_q}]$ into q copies, then K_{pq} can be decomposed into pq dense subgraphs X_1, X_2, \ldots, X_{pq} each on |H| vertices.

Leads to a recursive construction. Isolated values (case by case).



Up to N = 1000 ($k \simeq 40$):

constructing dense graphs using a greedy computer search

կլե

Up to $N = 1000 \ (k \simeq 40)$:

- constructing dense graphs using a greedy computer search
- corresponds to using more CPUs, not N but N' > N
- not balanced for CPU (nor memory) load

Roughly $N' \doteq \frac{7}{5}N$.

կլլ

Up to N = 1000 ($k \simeq 40$):

- constructing dense graphs using a greedy computer search
- corresponds to using more CPUs, not N but N' > N
- not balanced for CPU (nor memory) load

Roughly $N' \doteq \frac{7}{5}N$.

We beat this by theoretical constructions, though only for certain values – using more memory not more CPUs.

E.g. decomposing K_{559} into 559 isomorphic subgraphs each on 28 vertices.

կլե

Up to N = 1000 ($k \simeq 40$):

- constructing dense graphs using a greedy computer search
- corresponds to using more CPUs, not N but N' > N
- not balanced for CPU (nor memory) load

Roughly $N' \doteq \frac{7}{5}N$.

We beat this by theoretical constructions, though only for certain values – using more memory not more CPUs.

E.g. decomposing K_{559} into 559 isomorphic subgraphs each on 28 vertices.

Theoretical optimum: 25 vertices.

Optimization?



	14	22 17 8	13/20	6 11		16	13	14	14	10	9	13	10	13	10
		16		16		15 12	19	14	14 11 14 11	13 9	12	11 8 FT7 10	13	8 10	
10 0	22	18 17	20			19#1~		1.1	14 14 17 13	19#1		150		#114	
21 11	15 15	÷.		13	11	11	11	16		13	14	8	11	13	10
17 23 5		21 -		17	15 18 17 18 17 18	13 10 111 11	12	19 14 19 14		10 12		11 10 町14 13日 11	11	9 7 11 11 11 13	13
12		10	11			13 17	21 12	9	12	 11	9	14	9	14	12
12 20 18	MIN						22 18	-	7 12	11 8	-	14 10	-	10 13	
15 20		14	17 11 16	1 19			15 17	17	18 <mark>91</mark>	111 14 14		115 1611 12	12		15
19	10 20 P	12	12 18 14 12 18 14	lan lan			19 20	11	13 11 10 12	11/12/17 10/11/17 15/10	112	14 16 184 20 14 19	14		13 18
14	13	11	10	19	17	22		11	12	9	11	13	15	-	14 8
			19	13 10		16			23	_	17 8			I B	10
17	10	11 14	13 14	10	12	12	9		18	23	13	15	16	13	9
19	11 15 19 1014 18 1	15 20		10	10 18 1	14 10 19 10 3 11 10	12	27 17			19 18 19 14	17	17	12 9 11 12 11 12	10
14	11 18 9 8	16	10 16	8	9 11 15	11 12 11	13	18 15		1			18 18	3H 7	15
	9 6		13 14		8 12	11 10		15					20 H 20	17 11	
11	9	15	10	9	12	13	9	22 11 11 16	18 11 18 11	20	14	16 16 17 1	14 9 21	11	15
12	18 #11 18# 9 8		3 11 10 11	9	13 13 13 14 17	141 141 15 10	13	15	18 11 18 11 18 17 17		18 21 19 7				
9	12	9	12	10	10	11	14	11	15	20	14	13		16 17	
	14		13	40	145		15 17	F	14/01	14 18 184 18 18 18		Linese L	6 28		
3	9 14	14	11 13	18	15 11	194	14	5	8 13	17 8	19		13		
11	10	9	12	10	13	20	22 1 13 13	9	10	16	13		12 13 17	15 18	

... balance the sums among the subgraphs.

(ALL)

Thank you for your attention.