Katedra aplikované matematiky

IT4Innovations Centrum excelence

A nice application of decompositions of graphs

Petr Kovář

joint work with

Dalibor Lukáš, Tereza Kovářová, Michal Kravčenko

VŠB – Technical University of Ostrava

DiMaS, 27.11. 2012, VŠB–TU Ostrava

evropský sociální fond v ČR
EVROPSKÁ UNIE
MINISTERSTVO ŠKOLSTVÍ, MLÁDEŽE A TĚLOVÝCHOVY
OP Vzdělávání pro konkurenceschopnost

INVESTICE DO ROZVOJE VZDĚLÁVÁNÍ
Svět vědy CZ 1.07/2.3.00/35.0018

Outline of the talk

Motivation

Graph theory formulation

Known results

Main result

A couple of small results and approximations

End

Motivation

When solving real life problems

- large $n \times n$ matrix, $n \ldots$ millions
- numerical method

Motivation

When solving real life problems

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

Motivation

When solving real life problems

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

We have a dense matrix!

Motivation

When solving real life problems

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

We have a dense matrix!

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

Parallelization

Parallel machine without shared memory.

We prefer parallelization

- memory balanced
- load balanced

Parallelization

Parallel machine without shared memory.

We prefer parallelization

- memory balanced
- load balanced

Computation of block B_{ij}, requires

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

Parallelization

Parallel machine without shared memory.

We prefer parallelization

- memory balanced
- load 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)

Parallelization

Parallel machine without shared memory.

We prefer parallelization

- memory balanced
- load 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)

Question: which N blocks to which CPU?

- one diagonal block (longest computation)
- $(n-1)$ non-diagonal blocks

Question: which N blocks to which CPU?

- 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}

Question: which N blocks to which CPU?

- 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

Question: which N blocks to which CPU?

- 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 nad a vertex labeling $f : V(G) \to \{0, 1, \ldots, m\}$. 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 $\{\ell(x, y) : xy \in E(G)\} = \{1, 2, \ldots, m\}$.

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 nad a vertex labeling $f : V(G) \to \{0, 1, \ldots, m\}$. 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 $\{\ell(x, y) : xy \in E(G)\} = \{1, 2, \ldots, m\}$.

- graceful not only trees
- K_k is graceful iff $k \leq 4$

Definition (ρ-labeling)

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

Definition (ρ-labeling)

Let G be a graph with m edges nad a vertex labeling $f : V(G) \to \{0, 1, \ldots, 2m\}$. We say f 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.

Definition (ρ-labeling)

Let G be a graph with m edges nad a vertex labeling $f : V(G) \to \{0, 1, \ldots, 2m\}$. We say f 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.

- not all complete graphs have a ρ-labeling
- infinitely many have

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 $a_i - a_j$.

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 $a_i - a_j$.

Example

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

Equivalent formulation

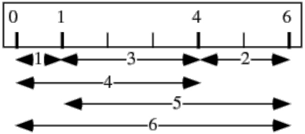
Definition (perfect difference sets)

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 $a_i - a_j$.

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



Example (Continued...)

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

Example (Continued...)

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

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

Example (Continued...)

Based on the perfect difference set $\{0, 1, 4, 6\}$ for $m = 6$ we cane 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) to each process

Example (Continued...)

Based on the perfect difference set $\{0, 1, 4, 6\}$ for $m = 6$ we cane 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) to each process
- the higher n the better ratio

Yet another formulation

Decomposing a complete graph into complete subgraphs – this seems as 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. BIBDs are also known as 2-designs and are denoted as $2-(v,k,\lambda)$ designs.

Yet another formulation

Decomposing a complete graph into complete subgraphs – this seems as 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. BIBDs are also known as 2-designs and are denoted as $2-(v, k, \lambda)$ designs.

In our case:

- $k = k$
- $v = n = k^2 - k + 1$
- $b = n$ this is unusual!
- $\lambda = 1$

Constructions

Sufficient condition: $k-1$ is a prime power.

Constructions

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.

Constructions

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.

Pavel Jahoda will tell more.

Constructions

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.

Pavel Jahoda will tell more.

Yet, K_7 does not decompose K_{43}.
Also, K_{11} does not decompose K_{111}.
...

Constructions

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.

Pavel Jahoda will tell more.

Yet, K_7 does not decompose K_{43}.
Also, K_{11} does not decompose K_{111}.
...

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

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

Constructions

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.

Pavel Jahoda will tell more.

Yet, K_7 does not decompose K_{43}.
Also, 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):
`http://www.ccrwest.org/diffsets/diff_sets/baumert.html`

Main result

Main result

. . . is the nice application!

Main result

…is the nice application!

Construction for certain (not all) values.

Main result

. . . 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.

Main result

…is the nice application!

Construction for certain (not all) values.

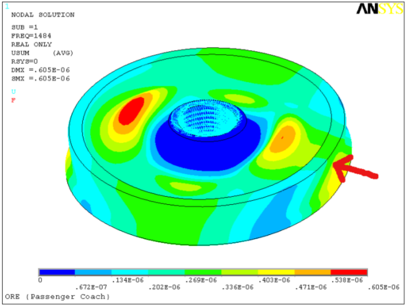
It has been implemented and successfully tested:
Fast BEM matrices of size n up to millions, distributed to hundreds of nodes N.

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

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

…details given by Dalibor Lukáš in the previous talk…

Motivation: railway wheel noise elimination by profiling



Courtesy of J. Szweda, Department of Mechanics, VŠB–TU Ostrava

Additional constructions

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

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

Additional constructions

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

Additional constructions

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

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

Additional constructions

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

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).

Graphs similar to complete graphs.

Graphs similar to complete graphs.

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

Graphs similar to complete 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

Graphs similar to complete 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 $k = 28, 29, \ldots$

Graphs similar to complete 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 $k = 28, 29, \ldots$

...difficult (almost) as for complete graphs

Graphs similar to complete 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 $k = 28, 29, \ldots$

. . . difficult (almost) as for complete graphs

Decomposition needs not to be cyclic

$K_7 - K_{3,3}$ decomposes K_{25}, yet no ρ-labeling.

Subgraphs need not to be isomorphic

- for odd N subgraphs can be isomorphic (exceptions!)
- for even N subgraphs cannot be isomorphic (parity)

Subgraphs need not to be isomorphic

- for odd N subgraphs can be isomorphic (exceptions!)
- for even N subgraphs cannot be isomorphic (parity)

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.

Subgraphs need not to be isomorphic

- for odd N subgraphs can be isomorphic (exceptions!)
- for even N subgraphs cannot be isomorphic (parity)

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 isomorphic subgraphs on pq vertices.

Example

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

Subgraphs need not to be isomorphic

- for odd N subgraphs can be isomorphic (exceptions!)
- for even N subgraphs cannot be isomorphic (parity)

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 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).

Further approximations

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

- constructing dense graphs using a greedy computer search

Further approximations

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

Further approximations

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

Further approximations

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.
E.g. decomposing K_{559} into 559 isomorphic subgraphs each on 28 vertices.
(Theoretical optimum: 25 vertices.)

Thank you for your attention.