OPTIMIZATION APPROACH TO THE PROBLEM OF SPECTRAL RECONSTRUCTION OF GRAPHS

Tatjana Davidović

Mathematical Institute
Serbian Academy od Sciences and Arts
http://www.mi.sanu.ac.rs/~tanjad
(tanjad@mi.sanu.ac.rs)

Joint work with Robert Doža, Predrag Djordjević,
Marko Milenković, Milica Andjelić

SMSCG2025, Oct. 02-05, 2025

Presentation outline

Searching for hypothesis in graph theory

- Extremal problems on graphs - a very popular research field in GT
- The goal is to find a graph that maximizes (minimizes) some parameter
- Theoretical results are straightforward only for some special classes of graphs
- In the most general cases, exhaustive search over all graphs should be performed
- It is usually done by the computer enumeration

Specialized softwares

- GRAPH, 1984 (https://www.mi.sanu.ac.rs/novi_sajt/research/projects/GRAPH.zip)
- AutoGraphiX, 1997, 2009, 2015 (https://www.autographix.ca)
- newGRAPH, 2004 (https://www.mi.sanu.ac.rs/newgraph/)
- PHOEG, 2008 (https://phoeg.umons.ac.be/phoeg)

Metaheuristic approach

- Algorithms are developed for each particular problem
- *A priori* knowledge about the problem is included
- The resulting graph characteristics are used as a hypothesis
- Researchers try to prove it theoretically

- Spectral reconstruction of graphs (SRG)

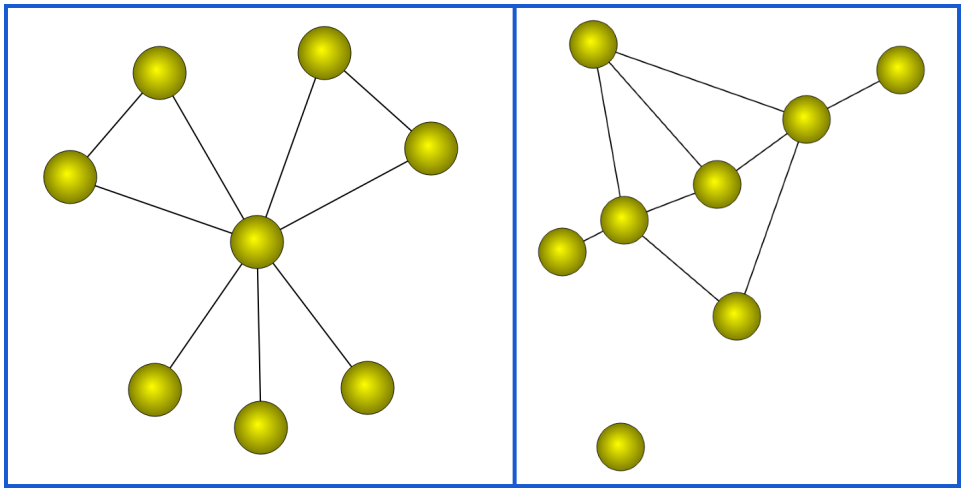
Spectral distance

- **Spectrum of graph:** $S = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ - eigenvalues of graph G, i.e., the roots of its characteristic polynomial $P_G(x) = det(xI - A)$
- Only some special classes of graphs, e.g., complete graphs, paths, cycles, are determined (to the isomorphism) by the spectrum with respect to A
- **Spectral distance of G_1 and G_2**

$$D = \sqrt{(\mu_1 - \lambda_1)^2 + (\mu_2 - \lambda_2)^2 + \cdots + (\mu_n - \lambda_n)^2}$$

where $S_1 = (\mu_1, \mu_2, \ldots, \mu_n)$ and $S_2 = (\lambda_1, \lambda_2, \ldots, \lambda_n)$ are spectra for G_1 and G_2, respectively.

Cospectral graphs



Graph $\Omega_{8,5}$ and its non-isomorphic co-spectral mate

- **Isomorphic graphs** - same structure up to the vertex labels
- **Co-spectral graphs** - spectral distance D equals zero

If a non-isomorphic co-spectral graph is found, graph is not determined by the spectrum

Spectral reconstruction of graphs

Find a graph G whose spectrum is equal to the given vector C

Optimization problem: minimize spectral distance between given vector $C = (c_1, c_2, \ldots, c_n)$ and $S = (\lambda_1, \lambda_2, \ldots, \lambda_n)$, the spectrum of a graph that we construct

$$D = \sqrt{(c_1 - \lambda_1)^2 + (c_2 - \lambda_2)^2 + \cdots + (c_n - \lambda_n)^2} \; (= 0)$$

NP-hard problem: all graphs with given number of vertices (n), edges (m), and triangles (t) should be examined, i.e., $\binom{n(n-1)/2}{m}$ graphs, s.t.

$$m = \frac{1}{2}\sum_{i=1}^{n} c_i^2 \text{ and } t = \frac{1}{6}\sum_{i=1}^{n} c_i^3$$

Definition of metaheuristics

- Optimization environment, general methods, recipes
- Incomplete search techniques
- Iterative, stochastic algorithms
- Classified as:
  - Mathematically-founded or nature-inspired
  - Explore a single solution or population of solutions
  - Construct new solutions or improve the existing ones

Basic steps of MH methods

- Generation of initial solution (population)
- Solution construction/transformation
- Improvement of the current best solution (intensification)
- Avoiding local optimum trap (diversification)
- Checking of stopping criterion fulfilment

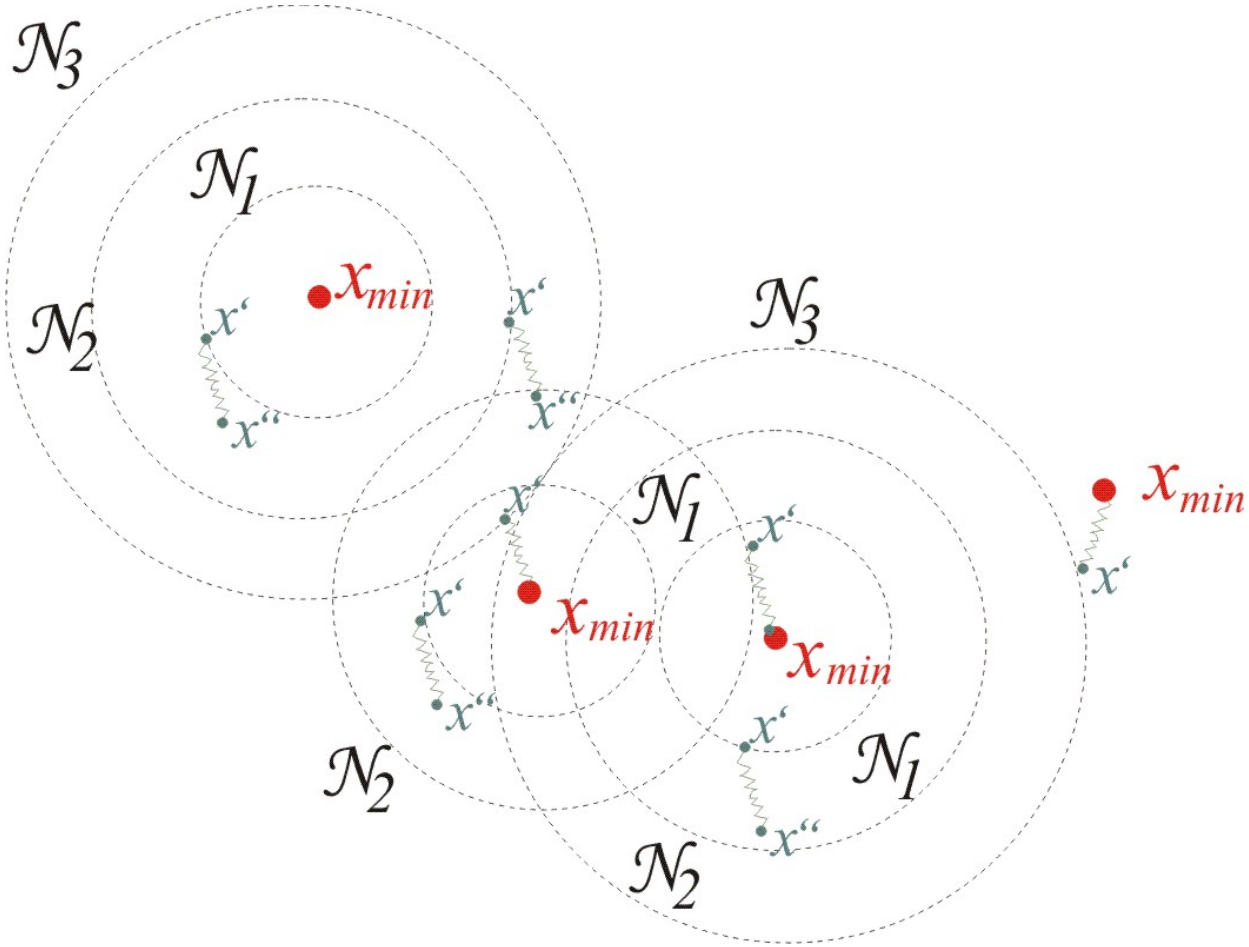
Examples of metaheuristics

- Simulated Annealing (SA)
- Tabu Search (TS)
- Greedy Randomized Adaptive Search Procedure (GRASP)

- Genetic Algorithm (GA)
- Ant Colony Optimization (ACO)
- Particle Swarm Optimization (PSO)

- Variable Neighborhood Search (VNS): Mladenović and Hansen, 1995 (1997)
- Bee Colony Optimization (BCO): Lučić and Teodorović, 2001 (2006)

Illustration of VNS at work



VNS - pseudocode

procedure GVNS(Problem input data, k_{max}, STOP)
    $x_{best} \leftarrow$ InitSolution()
    repeat
        $k \leftarrow 1$
        repeat
            $x' \leftarrow$ RandomSolution$(x_{best}, \mathcal{N}_k)$ ▷ Shaking
            $x'' \leftarrow$ VND(x') ▷ Local Improvement
            if $(f(x'') < f(x_{best}))$ **then** ▷ Neighborhood Change
                $x_{best} \leftarrow x''$
                $k \leftarrow 1$
            else
                $k \leftarrow k + 1$
            end if
            $Terminate \leftarrow$ StoppingCriterion$(STOP)$
        until $(k > k_{max} \vee Terminate)$
    until (Terminate)
    Return$(x_{best}, f(x_{best}))$
end procedure

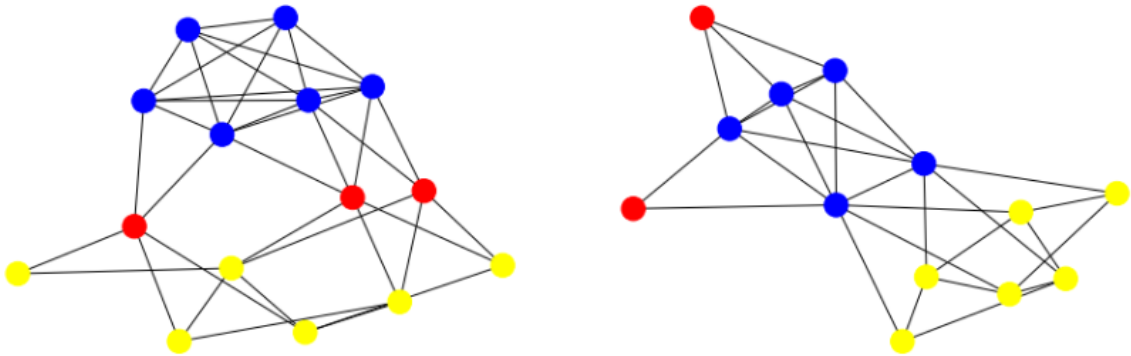
Solutions, neighborhoods, and shaking

- **Feasible solutions:** graphs with n vertices, m edges and t triangles
- **Transformations:** move edges and preserve the number of triangles
- **Neighborhoods:**
  1. Moving a single edge (N_1)
  2. Moving a pair of edges (N_2)
  3. Moving three edges simultaneously (N_3)
- **Shaking:** in neighborhood k move up to k edges to preserve feasibility, three variants
  1. Drop-and-add a single edge k times (SH_1)
  2. Add-and-drop a single edge k times (SH_2)
  3. Add k edges and drop k edges (SH_3)

Solution representation

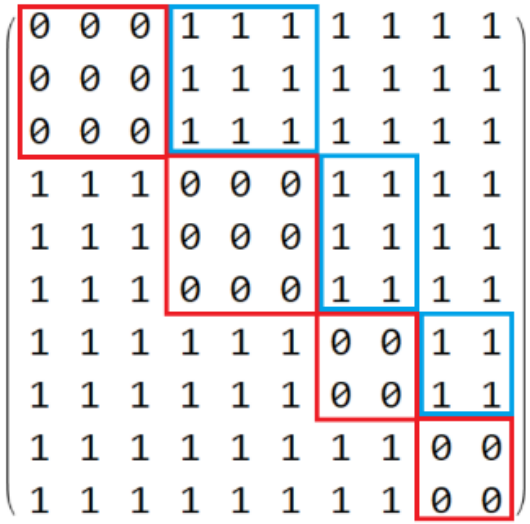
- **Threefold representation:**
  1. Adjacency matrix (for computing the spectrum)
  2. Lists of active and inactive edges
  3. For each edge - number of triangles it creates (or would create if active)
- Data structures (2) and (3) are updated only upon improvements
- **Initial solution:** Maximum clique, maximum bipartite graphs, and combinatoins
  1. Constructive approach
  2. Destructive approach

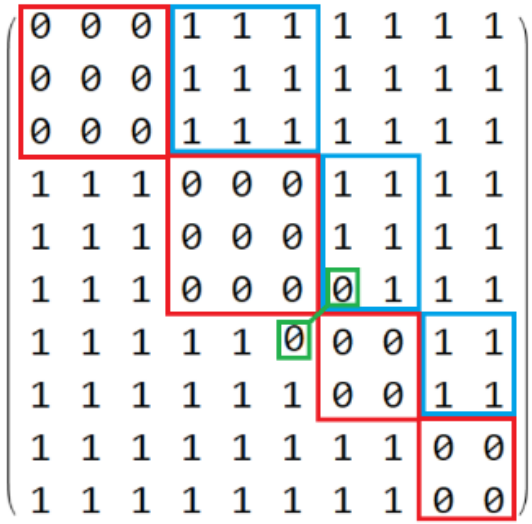
Constructive approach



- Generate largest possible complete graph K, which does not exceed the designated number of triangles (blue vertices).
- Add a vertex and start connecting it to the vertices of K until the next connection exceeds the designated number of triangles (red vertices).
- Repeat the previous step until we meet the triangle requirement, and label the constructed graph as G;
- If there are any remaining vertices or edges, create a maximum (yellow) bipartite graph B disconnected to G.

Destructive approach





- Construct an adjacency matrix with K blocks of zeroes on the main diagonal (red blocks), differentiating in dimension, where K is largest possible clique.
- Remove edges in adjacent blocks (blue blocks) carefully, so that for each vertex, we remove edges in only one adjacent block.

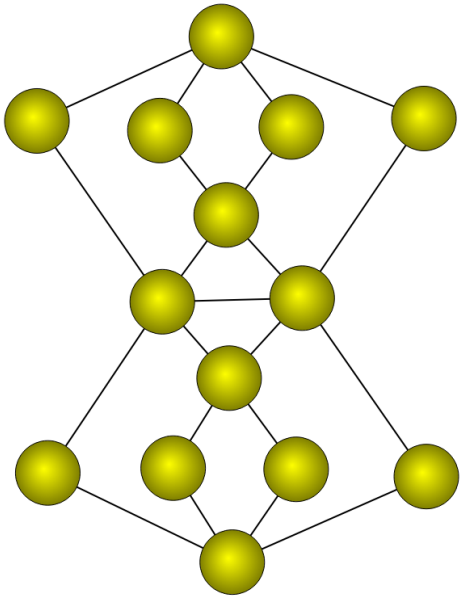
Experimental Setup

The implementation of GVNS is performed in C++ and executed on Intel Xeon E5-2620 v3, 2.40GHz, 32 GB RAM Under Linux Operating System and compiled with GCC 13.2.0.

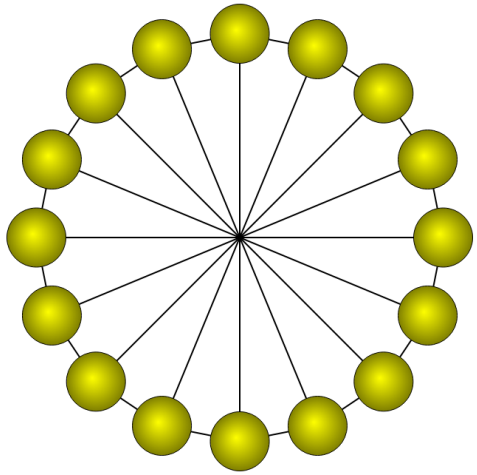
Test Examples:

- Cubic graph on 12 vertices (Cu_{12})
- Random graph on 12 vertices (G_{12})
- Symmetric graph with two triangles, tetragons, and pentagons on 14 vertices ($\Omega_{14,3}$)
- Star on 15 vertices ($K_{14,1} = S^0_{15,0}$)
- Ring with crossing edges on 16 vertices ($\Omega_{16,7}$)

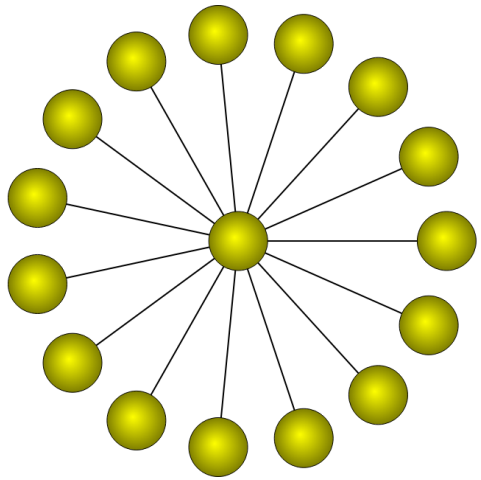
Graph examples



$\Omega_{14,3}$



$\Omega_{16,7}$



$S^0_{15,0}$

GVNS-SRG Results

Parameter values are determined intuitively:
GVNS: $k_{max} = m/2$, stopping criterion 1 000 000 evaluations.

Variants tried: N_1, $N_1 + N_2$, $N_1 + N_2 + N_3$, with and without initial LS

Best results: N_1 with initial LS and SH_1

Graph	#succ	av. eval.	av. obj.	av. t_{best}	av. t_{tot}
Cu_{12}	29/30	261712.80	0.012	1701.779	1962.849
G_{12}	2/30	957382.86	0.099	3159.891	6703.699
$\Omega_{14,3}$	27/30	393669.03	0.021	2180.229	2726.960
$S^0_{15,0}$	30/30	24728.26	0.000	144.398	144.398
$\Omega_{16,7}$	17/30	680950.63	0.147	2692.308	5126.023

Cospectral mate found for $S^0_{15,0}$.

Summary and conclusion

- We implemented GVNS as the incomplete search for SRG
- Promising preliminary results are obtained
- The main challenge is a large complexity of objective function computation (calculation of spectrum: $O(n^3)$)

- Optimization of the developed code is required
- The defined set of neighborhoods/transformations may be revisited
- The usage of memory and learning from previously visited solutions should be considered
- A careful parameter tuning need to be performed

Thank you for the attention!

Questions?

Tatjana Davidović
tanjad@mi.sanu.ac.rs