Metaheuristic Approaches to Optimization Problems on 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 Dragan Stevanović, Luka Radanović,
Predrag Djordjević, Robert Doža, Milica Andjelić
Abdelkadir Fellague, Dragutin Ostojić

Jan. 26, 2026

Acknowledgements

- Financial support:
  1. Science Fund of the Republic of Serbia, Grant #6767, Lazy walk counts and spectral radius of graphs—LZWK
  2. Serbian Ministry of Science, Technological Development, and Innovations, Agreement No. 451-03-66/2024-03/200029
- Research framework: Summer Research Internship for Students, Mathematical Institute, Serbian Academy of Sciences and Arts

Presentation outline

1. Definition of problems
   - Spectral reconstruction of graphs (SRG)
   - Maximization of spectral radius (MSR)
2. Metaheuristic methods
   - General Variable Neighborhood Search (GVNS)
   - Improvement-based Bee Colony Optimization (BCOi)
3. Implementation of the proposed methods
   - GVNS for SRG
   - GVNS for MSR
4. Experimental evaluation
5. Concluding remarks

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
- Examples: tightness maximization, minimization of the least eigenvalue, etc.

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

1. Spectral reconstruction of graphs (SRG)
2. Maximization of spectral radius (MSR)

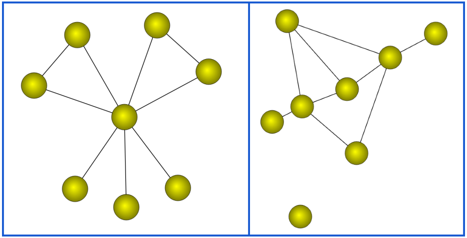
Spectral distance

- **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.

Ccspectral 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 $C = (c_1, c_2, \dots, c_n)$ and $S = (\lambda_1, \lambda_2, \dots, \lambda_n)$, 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.

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

Finding connected graphs with maximum index

- We are searching for a connected undirected graph $G = (V, E)$
- With specified $n = |V|$, $m = |E|$
- Described by the Adjacency matrix $A = [a_{ij}]_{n \times n}$
- Such that its maximum eigenvalue (λ_1) is as large as possible
- It is known that we need to examine only threshold graphs[1]

[1] Simić, S. K., Li Marzi, E. M., Belardo, F., Connected graphs of fixed order and size with maximal index: structural considerations, Le Matematiche, 59(1,2):349–365, 2004.

The application of threshold graphs

Threshold graphs are attractive for investigation due to their numerous applications:[2]

- medicine (genetics, neonatology)
- psychology (evaluation and grading of candidates)
- computer science (aggregation of LP formulation, scheduling and timetabling, synchronization of parallel processes)
- etc.

[2] Mahadev, N. V. R., Peled, U. N., Threshold graphs and related topics, Elsevier, 1995.

Threshold graphs

Iterative construction: We start with a single vertex and add new vertices that are either isolated or adjacent to all already included vertices.
More formally,

$$G_{p_1} = K_{p_1},$$

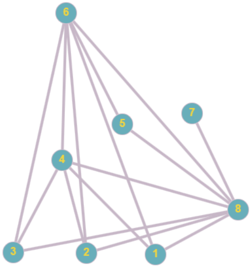
$$G_{p_1,p_2,\ldots,p_k} = \overline{G_{p_1,p_2,\ldots,p_{k-1}}} \vee K_{p_k}$$

where $p_1, p_2, \ldots, p_k$, are positive integers,
$\overline{G}$ denotes the complement of G (and changes type of vertices) and
$\vee$ denotes the join of two graphs (connects new p_k vertices to all already existing).

The type of vertices alternates (between isolated or adjacent to all previous).

An example of threshold graph

$$A = \begin{bmatrix} 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$



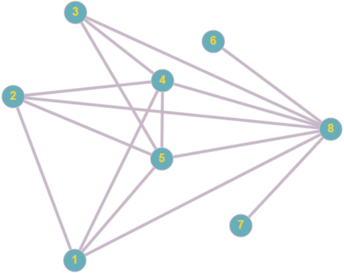
Binary representation:

$$\mathcal{R} = \{1, 0, 0, 1, 0, 1, 0, 1\}$$

$n = 8,\ m = 15,\ \lambda_1 = 4.37$

Threshold graph maximizing spectral radius

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$



Binary representation:

$$\mathcal{R} = \{1, 1, 0, 1, 1, 0, 0, 1\}$$

$n = 8,\ m = 15,\ \lambda_1 = 4.52$

Problem statement

$$\max_G \max_i S = \{\lambda_i, i = 1, 2, \ldots, n\}$$

- $G = (V, E)$ is undirected graph with $|V| = n$ vertices and $E = m$ edges
- S is spectrum of graph G, i.e., a set of n eigenvalues λ_i (calculated as the roots of polynomial $A - \lambda I$, usual complexity $O(n^3)$)
- Adjacency matrix

$$A = [a_{ij}]_{n \times n} = \begin{cases} 1, & (i, j) \in E, \\ 0, & \text{otherwise.} \end{cases}$$

- As A is symmetric, S consists of real values

Problem complexity

- Finding a connected threshold graph that maximizes spectral radius in the general case is an *open problem*
- It can be formulated as a *combinatorial optimization problem*
- Exhaustive search over all threshold graphs of the same size is *NP-hard problem*, and therefore, computationally intractable for large values of n and m
- The largest search space corresponds to the medium number of edges, i.e., for m close to $n(n-1)/4$

GVNS - pseudocode

```
procedure GVNS(Problem input data, k_max, STOP)
    x_best ← InitSolution()
    repeat
        k ← 1
        repeat
            x' ← RandomSolution(x_best, N_k)                ▷ Shaking
            x'' ← VND(x')                                   ▷ Local Improvement
            if (f(x'') < f(x_best)) then                    ▷ Neighborhood Change
                x_best ← x''
                k ← 1
            else
                k ← k + 1
            end if
            Terminate ← StoppingCriterion(STOP)
        until (k > k_max ∨ Terminate)
    until (Terminate)
    Return(x_best, f(x_best))
end procedure
```

BCOi - pseudocode

```
procedure BCOi(Problem input data, B, NC, STOP)
    Terminate ← 0
    while !Terminate do
        for b ← 1, B do                                   ▷ // Determine initial population
            Solution(b) ← GenerateSolution()              ▷ // Initial Forward pass
        end for
        Update(x_best)
        for u ← 1, NC do
            Normalization()                               ▷ // Backward pass
            U ← Loyalty()
            for b ← 1, U do
                Recruitment(Solution(b))
            end for
            for b ← 1, B do                               ▷ // Forward pass
                Transform(Solution(b))
            end for
            Update(x_best)
        end for
        Terminate ← StoppingCriterion(STOP)
    end while
    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 $\mathcal{N}_1$
  2. Moving a pair of edges $\mathcal{N}_2$
  3. Moving three edges simultaneously $\mathcal{N}_3$
- **Shaking:** in neighborhood k move up to k edges to preserve feasibility

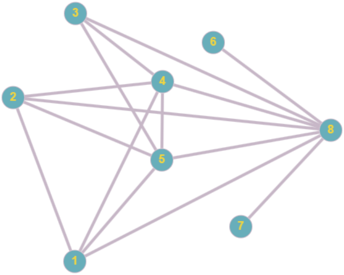
Solution representation

- **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

Solution representation: Binary sequence

$$\mathcal{R} = \{r_1, r_2, \ldots, r_n\}, \ \sum_{i=1}^{n} (i-1) \cdot r_i = m$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \end{bmatrix}$$



$$\mathcal{R} = \{1, 1, 0, 1, 1, 0, 0, 1\},$$

$$n = 8, \ m = 15, \ \lambda_1 = 4.52$$

Generation of initial solution for GVNS

- In GVNS, greedy procedure that fills sequence elements from the tail is used
- The first element is equal to one, by convention
- The last element equals one to ensure connectivity
- The resulting solution accumulates ones at the end of sequence

```
procedure INIT_SOL(n, m, R)
    R[1] ← 1
    R[n] ← 1
    for i ← 2, n − 1 do
        R[i] ← 0
    end for
    e = m − n + 1
    i ← n
    while (i > 1) ∧ (e > 0) do
        if i − 1 <= e then
            R[i] ← 1
            e ← e − i + 1
        end if
        i ← i − 1
    end while
end procedure
```

Generation of initial solution for BCOi

- This initial solution is not suitable for BCOi
- It allows small number of transformations
- For BCOi we modify initial solution generation by using 2 as the counter increment
- Initial population in BCOi is obtained by random perturbations of the modified initial solution

Definition of neighborhoods/transformations

Four transformations yielding to a feasible solution can be defined:

1. Each combination $\{1...01...10...1\}$ can be replaced with $\{1...10...01...1\}$
2. Analogously, $\{1...10...01...1\}$ could be replaced with $\{1...01...10...1\}$
3. If $r_i = 1$, $r_j = r_k = 0$, and $i = j + k$, then it is possible to modify this solution in such a way that $r_i = 0$, $r_j = r_k = 1$
4. Analogously, if $r_i = 0$, $r_j = r_k = 1$, and $i = j + k$, then it is possible to modify this solution in such a way that $r_i = 1$, $r_j = r_k = 0$

The first two transformations preserve the number of zeros/ones, while the last two enable to modify (increase or decrease) these numbers, however, without violating their sum, i.e., the value of m.

Searching over neighborhoods/transformations

1. The first two transformations define one neighborhood $\mathcal{N}_1$, while the last two are joined into $\mathcal{N}_2$
2. VND systematically searches $\mathcal{N}_1$ and $\mathcal{N}_2$ in the First-Improvement manner
3. In the forward pass of BCOi, for each solution we perform a random number of transformations (o)
4. For each transformation we randomly choose between $\mathcal{N}_1$ and $\mathcal{N}_2$
5. The corresponding transformation is performed on randomly selected (feasible) positions

Experimental Setup

GVNS and BCOi are implemented in C++ and executed on Intel Xeon E5-2620 v3, 2.40GHz, 32 GB RAM Under Linux 4.19.12 and compiled with GCC 4.8.3.

Test Examples:
SRG: $K_{9,1}$, S_{10}^{+}, R_4 with 12 vertices, Cubic graphs on 12 vertices (C_1, C_2, C_3)
MSR: Medium instances with 30 and 50 vertices. Number of edges was selected randomly from the middle of valid interval.

Repetition: 30 times, seed equal to $n * i + m$.

Stopping criterion: 10000(2000) function evaluations (to ensure fair comparison).
Parameter values are determined intuitively.
GVNS: $k_{max} = n/2$,
BCOi: $BEES = 5$, $NC = 10$, $o = rnd(n/3, 2n/3)$

SRG Results

GVNS without initial VND, using $\mathcal{N}_1$

Graph	#succ	av. eval.	av. obj.	t_{best}	t_{tot}
$G10-1$	30/30	**2105.77**	0.00	0.072	0.072
$G12-1$	0/30	100000.00	**0.11**	2.411	5.362
$G12-2$	0/30	100000.00	**0.11**	2.680	5.407
$G12-3$	0/30	100000.00	**0.12**	3.348	5.378
Cu_1	**30/30**	**9191.90**	0.00	0.475	0.475
Cu_2	**30/30**	**5790.00**	0.00	0.301	0.301
Cu_8	**30/30**	**39296.73**	0.00	2.036	2.036
$H16$	**1/30**	**97575.20**	0.60	6.545	11.888
$R12-4$	**10/30**	**83110.00**	0.39	2.063	4.395

MSR Results

Graph	Init.sol.	GVNS			BCOi		
	Obj.val.	#bests	best obj.	av. obj.	#bests	best obj.	av. obj.
$G_{30,100}$	10.96	30	12.34	12.34	13	12.34	12.10
$G_{30,220}$	18.12	30	20.03	20.03	30	20.03	20.03
$G_{30,300}$	22.16	**30**	23.65	**23.65**	26	23.65	23.64
$G_{30,400}$	27.04	30	27.58	27.58	30	27.58	27.58
$G_{50,100}$	10.38	30	**10.87**	**10.87**	30	10.38	10.38
$G_{50,300}$	19.58	**30**	**22.89**	**22.89**	25	22.50	22.19
$G_{50,500}$	26.80	**30**	**30.33**	**30.33**	1	30.18	30.08
$G_{50,1000}$	41.87	30	44.02	44.02	30	44.02	44.02

Results (cont.)

The binary sequences that correspond to the best obtained solutions are:

$R(G_{30,100}) = \{111111101111100000000000000001\}$
$R(G_{30,220}) = \{111111111111111111101000000001\}$
$R(G_{30,300}) = \{111110111111111111111111000001\}$
$R(G_{30,400}) = \{111111101111111111111111111101\}$
$R(G_{50,100}) = \{11110111111000000000000000000000000000000000000001\}$
$R(G_{50,300}) = \{11011111111111111111111000000000000000000000000001\}$
$R(G_{50,500}) = \{11111111111111011111111111111110000000000000000001\}$
$R(G_{50,1000}) = \{11111111111111111111111111111111111111101111100001\}$

The hypothesis: Maximum spectral radius value corresponds to graphs whose binary sequences contain 1s concentrated at the beginning.

Summary and conclusion

- We implemented GVNS as the incomplete search for SRG and MSR
- We implemented GVNS and BCOi as the incomplete search for graphs with maximum radius
- Promising results for further development 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

- MISANU Summer Internship since 2020
- Call: May-June
- Realization July-August
- Various topics in mathematics, mechanic, and computer science

Thank you for the attention!

Questions?

Tatjana Davidović
tanjad@mi.sanu.ac.rs