Exact ILP Algorithms for the MWSCP

Ali Forouz. Hafshejani

Electrical and Computers Engineering
Isfahan University of Technology

February 2024

Table of Contents

Abstract

Problem

The Maximum Weighted Submatrix Coverage Problem (also known as MWSCP) is a recently introduced problem with applications in data mining. It is concerned with selecting K submatrices of a given numerical matrix such that the sum of the matrix-entries, which occur in at least one of the selected submatrices, is maximized.

Preliminary

Concerning mathematics, assume further $\mathbb{N}_K = \{1, \dots, K\}$.

Mathematical definition

Definition (The Maximum Weighted Submatrix Coverage Problem)

Let $\mathcal{M} \in \mathbb{R}^{m\times n}$ be an $m \times n$ matrix, and $\mathcal{M}_{ij}$ indicate the value of the cell in row i and column j. Let $\mathcal{R} = \{1, \dots, m\}$ and $\mathcal{C} = \{1, \dots, n\}$ indicate the index-set of the rows and columns of $\mathcal{M}$, respectively. Let K be a given integer and $(\mathcal{R}_k, \mathcal{C}_k) \subset (R, C), k \in \mathbb{N}_K$, i.e., each $(\mathcal{R}_k, \mathcal{C}_k)$ is a (possibly empty) submatrix of $\mathcal{M}$. For a set of K submatrices $(\mathcal{R}_k, \mathcal{C}_k)$, let $COV(\cup_{k\in\mathbb{N}_K}(\mathcal{R}_k, \mathcal{C}_k), i, j) = 1$, if and only if cell (i, j) is occurring in a least one of the submatrices $(\mathcal{R}_k, \mathcal{C}_k), k \in \mathbb{N}_K$, and zero otherwise. The goal of the MWSCP is to find K submatrices $(\mathcal{R}_1^*, \mathcal{C}_1^*), \dots, (\mathcal{R}_K^*, \mathcal{C}_K^*)$, which maximize $\sum_{i\in R, j\in C} \mathcal{M}_{ij} \cdot COV(\cup_{k\in\mathbb{N}_K}(\mathcal{R}_k, \mathcal{C}_k), i, j)$.

Example

	1	2	3	4	5	6	7
1	−4.4	−2.2	1.0	2.8	−0.3	−1.8	−1.9
2	2.3	2.4	−1.1	5.3	1.0	1.8	0.8
3	−4.1	1.0	0.4	−4.2	3.0	2.1	−2.2
4	−5.6	1.8	4.1	3.5	−2.2	−4.2	−4.3
5	0.3	2.2	−3.9	3.2	4.1	−1.1	−2.1
6	−4.3	−2.0	−3.1	4.1	2.0	5.1	1.1

(a) Exemplary instance

	1	2	3	4	5	6	7
1	−4.4	−2.2	1.0	2.8	−0.3	−1.8	−1.9
2	2.3	2.4	−1.1	5.3	1.0	1.8	0.8
3	−4.1	1.0	0.4	−4.2	3.0	2.1	−2.2
4	−5.6	1.8	4.1	3.5	−2.2	−4.2	−4.3
5	0.3	2.2	−3.9	3.2	4.1	−1.1	−2.1
6	−4.3	−2.0	−3.1	4.1	2.0	5.1	1.1

(b) Optimal solution for $K = 2$

Figure: Exemplary instance of the MWSCP and optimal solution for K = 2. The selected submatrices are given in red ($\mathcal{R} = \{1, 3, 4\}$, $\mathcal{C} = \{3, 4\}$) and blue ($\mathcal{R} = \{2, 3, 5, 6\}$, $\mathcal{C} = \{2, 4, 5, 6\}$), note that they overlap in cell $(3, 4)$. The solution value is 41.8.

NP-Hard

Search Space

For $K > 1$, once all the columns set variables are fixed ($\mathcal{C}_k; \forall k \in \mathbb{N}_K$) it remains to decide for each row i and each submatrix k whether i should be part of $\mathcal{R}_k$ or not. Those K decisions per row does not enjoy the monotonicity or the anti-monotonicity properties.

Bottleneck

Actually, those K decisions per row cannot be optimally taken in polynomial time anymore.

Theorem

For fixed variables $\mathcal{C}_k$ ($\forall k \in \mathbb{N}_K$), fixing optimally $\mathcal{R}_k$ ($\forall k \in \mathbb{N}_K$) is NP-Hard.

Approches

In this presentaion, different approaches were introduced.

Basics of Mathematical Model

Suppose binary variables are as follows.

$r_i^k = 1 \iff$ row $i \in R$ is selected in submatrix k.

$c_j^k = 1 \iff$ column $j \in C$ is selected in submatrix k.

$s_{ij} = 1 \iff$ cell (i, j) is chosen in at least one of the selected submatrices.

IPCQ

The first mathematical modeling of the problem with the boolean quadric polytope is as follows.

$$\max \sum_{i \in R, j \in C} \mathcal{M}_{ij} s_{ij}$$

$$K s_{ij} \geq \sum_{k=1}^{K} r_i^k c_j^k \quad \forall i \in R, j \in C \tag{SRCQ-LINK1}$$

$$s_{ij} \leq \sum_{k=1}^{K} r_i^k c_j^k \quad \forall i \in R, j \in C \tag{SRCQ-LINK2}$$

$$s_{ij} \in \{0,1\} \quad \forall i \in R, j \in C$$

$$r_i^k \in \{0,1\} \quad \forall i \in R; k \in \mathbb{N}_K$$

$$c_j^k \in \{0,1\} \quad \forall j \in C; k \in \mathbb{N}_K$$

First linear formulation

All variables are binary.

$r_i^k = 1 \iff$ row $i \in R$ is selected in submatrix k.

$c_j^k = 1 \iff$ column $j \in C$ is selected in submatrix k.

$e_{ij}^k = 1 \iff$ cell (i, j) is chosen in submatrix k.

$s_{ij} = 1 \iff$ cell (i, j) is chosen in at least one of the selected submatrices.

$\text{ILP}_{\text{Derval}}$

$$\max \sum_{i \in R, j \in C} \mathcal{M}_{ij} s_{ij}$$

$$s_{ij} \geq e_{ij}^k \quad \forall i \in R, j \in C; k \in \mathbb{N}_K \tag{SE-LINK1}$$

$$s_{ij} \leq \sum_{k=1}^{K} e_{ij}^k \quad \forall i \in R, j \in C \tag{SE-LINK2}$$

$$e_{ij}^k + 1 \geq r_i^k + c_j^k \quad \forall i \in R, j \in C; k \in \mathbb{N}_K \tag{RCE-LINK1}$$

$$2e_{ij}^k \leq r_i^k + c_j^k \quad \forall i \in R, j \in C; k \in \mathbb{N}_K \tag{RCE-LINK2}$$

$$s_{ij} \in \{0,1\} \quad \forall i \in R, j \in C$$

$$e_{ij}^k \in \{0,1\} \quad \forall (i,j) \in R \times C; k \in \mathbb{N}_K$$

$$r_i^k \in \{0,1\} \quad \forall i \in R; k \in \mathbb{N}_K$$

$$c_j^k \in \{0,1\} \quad \forall j \in C; k \in \mathbb{N}_K$$

Simplify formulation

(RCE-LINK2) can be disaggregated to the following two sets of constraints (RE-LINK) and (CE-LINK).

$\text{ILP}'_{\text{Derval}}$

$$\max \sum_{i\in R, j\in C} \mathcal{M}_{ij} s_{ij}$$

$$s_{ij} \geq e_{ij}^k \quad \forall i \in R, j \in C; k \in \mathbb{N}_K \tag{SE-LINK1}$$

$$s_{ij} \leq \sum_{k=1}^{K} e_{ij}^k \quad \forall i \in R, j \in C \tag{SE-LINK2}$$

$$e_{ij}^k + 1 \geq r_i^k + c_j^k \quad \forall i \in R, j \in C; k \in \mathbb{N}_K \tag{RCE-LINK1}$$

$$e_{ij}^k \leq r_i^k \quad \forall i \in R, j \in C; k \in \mathbb{N}_K \tag{RE-LINK}$$

$$e_{ij}^k \leq c_j^k \quad \forall i \in R, j \in C; k \in \mathbb{N}_K \tag{CE-LINK}$$

$$s_{ij} \in \{0,1\} \quad \forall i \in R, j \in C$$

$$e_{ij}^k \in \{0,1\} \quad \forall (i,j) \in R \times C; k \in \mathbb{N}_K$$

$$r_i^k \in \{0,1\} \quad \forall i \in R; k \in \mathbb{N}_K$$

$$c_j^k \in \{0,1\} \quad \forall j \in C; k \in \mathbb{N}_K$$

Reformulation

All variables are binary.

$f_{ij}^{k} = 1 \iff$ cell (i, j) by $\mathcal{M}_{ij} > 0$ is chosen in submatrix k.

$s_{ij} = 1 \iff$ cell (i, j) is chosen in at least one of the selected submatrices.

$\mathrm{ILP}_{\mathrm{Sinnl}}$

$$\max \sum_{i \in R, j \in C} \mathcal{M}_{ij} s_{ij}$$

$$s_{ij} + 1 \geq f_{ij}^k + f_{i'j'}^k \quad \forall (i,j),(i',j') \in R \times C : \mathcal{M}_{ij} > 0, \mathcal{M}_{i'j'} > 0, \mathcal{M}_{i'j} < 0; k \in \mathbb{N}_K \tag{SE-LINK1}$$

$$s_{ij} \leq \sum_{k=1}^{K} f_{ij}^k \quad \forall (i,j) \in R \times C : \mathcal{M}_{ij} > 0; k \in \mathbb{N}_K \tag{SF-LINK2}$$

$$s_{ij} \in \{0,1\} \quad \forall i \in R, j \in C$$

$$f_{ij}^k \in \{0,1\} \quad \forall (i,j) \in R \times C : \mathcal{M}_{ij} > 0; k \in \mathbb{N}_K$$

Formulation by decomposition

In this section, we obtain a formulation without e-variables from ($\mathrm{ILP}_{\mathrm{Derval}}$) using Benders feasibility cuts. By removing the e-variables and using propertices of dual, we would have a new formulation.

$\mathrm{ILP}_{\mathrm{Benders}}$

$$\max \sum_{i \in R, j \in C} \mathcal{M}_{ij} s_{ij}$$

$$s_{ij}^k + 1 \geq r_i^k + c_j^k \quad \forall (i,j) \in R \times C : \mathcal{M}_{ij} < 0; k \in \mathbb{N}_K \tag{SRC-LINK1}$$

$$s_{ij}^k \leq \sum_{k \in S} r_i^k + \sum_{k \in \mathbb{N}_K \setminus S} c_j^k \quad \forall (i,j) \in R \times C : \mathcal{M}_{ij} > 0; \forall S \subseteq \mathbb{N}_K \tag{SRC-LINK2}$$

$$r_i^k \in \{0,1\} \quad \forall i \in R; k \in \mathbb{N}_K$$

$$c_j^k \in \{0,1\} \quad \forall j \in C; k \in \mathbb{N}_K$$

$$s_{ij} \in \{0,1\} \quad \forall i \in R, j \in C$$

Result

Implementation

All of the algorithms were implemented in C++20 and IBM ILOG CPLEX Studio 22.1 with default settings. Also, the runs were made on parallel cores of an Intel Core i7-13650HX machine with 4.3 GHz and 16 GB of RAM.

Parameters of Runs

Runtimes of all algorithms report in seconds. ”ML” indicates memory limit reached, ”TL” indicates time limit of 300 seconds but solver doesn’t work, and ”GP” indicates gap limit less than 30 percent after 600 seconds.

Synthetic instances

The synthetic instances were constructed in such a way that at least a near-optimal solution is known by construction: To create an $m \times n$-instance, K submatrices (of a given dimension $r \times s$) were randomly created (Gaussian with variance $\frac{r \text{ or } s}{20}$) and then placed into an empty $m \times n$-matrix. For placing the submatrices in the instance, a parameter owas used which gives the minimum overlap (in % of cells) between submatrices the placement needs to have. After placing the submatrices in the $m \times n$-matrix, noise with given variance ρ and mean zero is added to every entry of the $m \times n$-matrix. With this procedure, the K submatrices give the optimal (or a near-optimal, due to the added noise) solution.

K Overlap

Instance					ILP		
K	m	o	ρ	r	Derval	Sinnl	Benders
2	100	0.0	0.0	25	0.2335	1.9330	0.0315
2	100	0.2	0.0	25	0.2544	4.9474	0.0477
2	100	0.4	0.0	25	0.1861	1.5346	0.0339
2	100	0.6	0.0	25	0.3250	0.6072	0.0313

K Noise

Instance					ILP		
K	m	o	ρ	r	Derval	Sinnl	Benders
2	50	0.0	0.0	16	0.1049	1.4020	0.0491
2	50	0.0	0.5	16	2.1939	28.9919	0.3506
2	50	0.0	1.0	16	81.5547	TL	50.3888
3	50	0.0	0.0	16	0.5499	127.2428	0.4972
3	50	0.0	0.5	16	11.2348	TL	2.9987
3	50	0.0	1.0	16	GP	TL	GP

Size Overlap

Instance					ILP		
K	m	o	ρ	r	Derval	Sinnl	Benders
2	50	0.0	0.0	16	0.2548	1.5489	0.0419
2	50	0.1	0.0	16	0.1683	1.4652	0.0389
2	50	0.2	0.0	16	0.1456	0.7397	0.0321
2	50	0.4	0.0	16	0.2808	0.2425	0.0206
2	100	0.0	0.0	33	0.6072	77.6524	0.0835
2	100	0.1	0.0	33	1.1576	28.3303	0.0885
2	100	0.2	0.0	33	0.7222	34.0175	0.0704
2	100	0.4	0.0	33	3.0368	8.9296	0.0586
2	200	0.0	0.0	50	1.3019	316.3238	0.1175
2	200	0.1	0.0	50	2.1365	ML	0.1807
2	200	0.2	0.0	50	1.6380	265.1710	0.1221
2	200	0.4	0.0	50	3.4386	123.5132	0.0749

Size Noise

Instance					ILP		
K	m	o	ρ	r	Derval	Sinnl	Benders
2	50	0.0	0.0	16	0.2680	1.5567	0.0430
2	50	0.0	0.5	16	2.7496	21.4312	0.3545
2	50	0.0	1.0	16	73.6836	TL	37.1137
2	100	0.0	0.0	33	0.6091	78.4539	0.0907
2	100	0.0	0.5	33	29.4875	GP	4.4827
2	100	0.0	1.0	33	TL	ML	TL
2	200	0.0	0.0	50	1.2690	TL	0.1292
2	400	0.0	0.0	50	2.0882	ML	0.4779

References

[1] Sinnl, M. (2021). Exact and heuristic algorithms for the maximum weighted submatrix coverage problem. European Journal of Operational Research, 298(3), 821-833.

[2] Derval, G., Branders, V., Dupont, P., & Schaus, P. (2019). The Maximum Weighted Submatrix Coverage Problem: A CP Approach. In: Rousseau, LM., Stergiou, K. (Eds.), Integration of Constraint Programming, Artificial Intelligence, and Operations Research (pp. 258–274). Springer International Publishing.