Testing the efficiency of adjacency algorithms for solving unit restricted models

Jan Kašpar, Róbert Marušák

Czech University of Life Sciences Prague
Faculty of Forestry and Wood Sciences

SSAFR 2015, August 19 – 21th, Uppsala, Sweden

INTRODUCTION

Adjacency constraints – important part of spatial harvest scheduling models since GIS tools were developed

⇨ legal constraints, environmental constraints etc.

Early solvers ⇨ **limited number of variables and constraints** !

this can lead to a loss of efficiency in solving some problems (as discussed by Torres-Rojo and Brodie 1990).

the reduction of constraints less relevant (Crowe et al. 2003)

There are many possibilities how to add adjacency constraints into the model

Pairwise constraints	✓	Very simple to create
	✗	High number of constraints
Constraints based on analytical algorithms (Yoshimoto, Brodie 1994)	✓	Low number of constraints
	✗	Mathematical modification is needed

Computer memory [**random-access memory**] („CM“) usage

Pairwise constraints

Database with spatial information

Model

No „CM“ is used

Constraints based on analytical algorithms

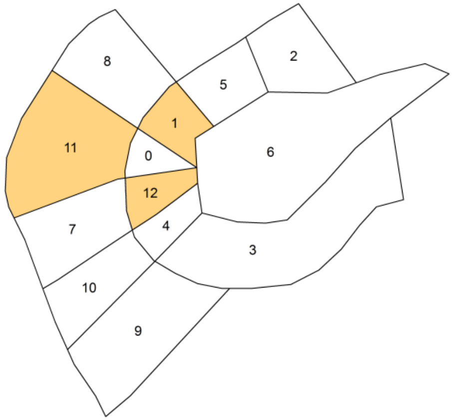
Database with spatial information

Mathematical modification

Model

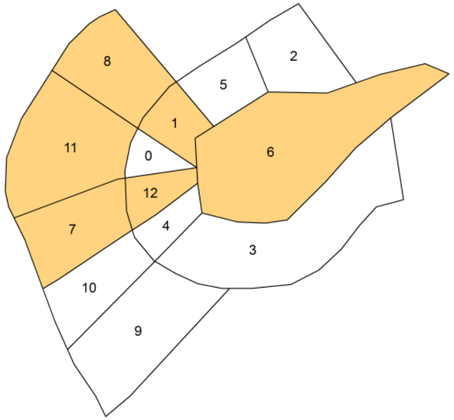
„CM“ is used

The effect of neighbourhood type



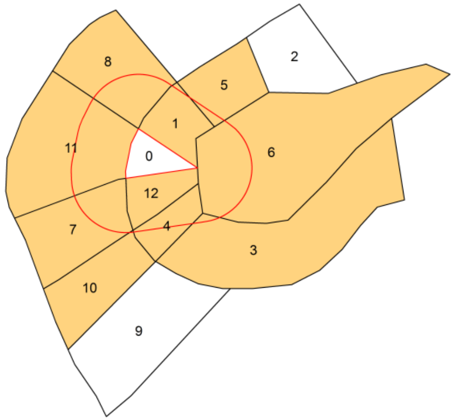
Neumann

3



Moore

6



Buffer 30 meters

10

The goal of the work

⇨ To compare time needed for creating and solving models with adjacency constraints:

- based on 3 different analytical algortihms + pairwise constraints
- for 3 different types of neighbourhood

⇨ Is it necessary to consume computer memory and time by using the analytical algorithms when the sophisticated softwares and powerful computers are available?

?

Material and methods

- **738** randomly generated spatial structures (**1-738 polygons**)

- **2-3 optimization** were made for each spatial structure –**2,000 optimization** in total

- Totally **12** variants =
  4 types of adjacency constraints × 3 types of neighbourhood

Gurobi optimization software was used for analysis

Personal computer Intel® Core™ i7-2600 CPU @ 3.40 GHz was used

The models were written in Java programming language

Mathematical model

$$\max Z = \sum_{i=1}^{n} \sum_{p=1}^{5} v_{ip} \cdot x_{ip}$$

$$\sum_{p=1}^{5} x_{ip} \leq 1 \quad \forall i = 1,2,\ldots,n$$

!+ **Adjacency constraints**

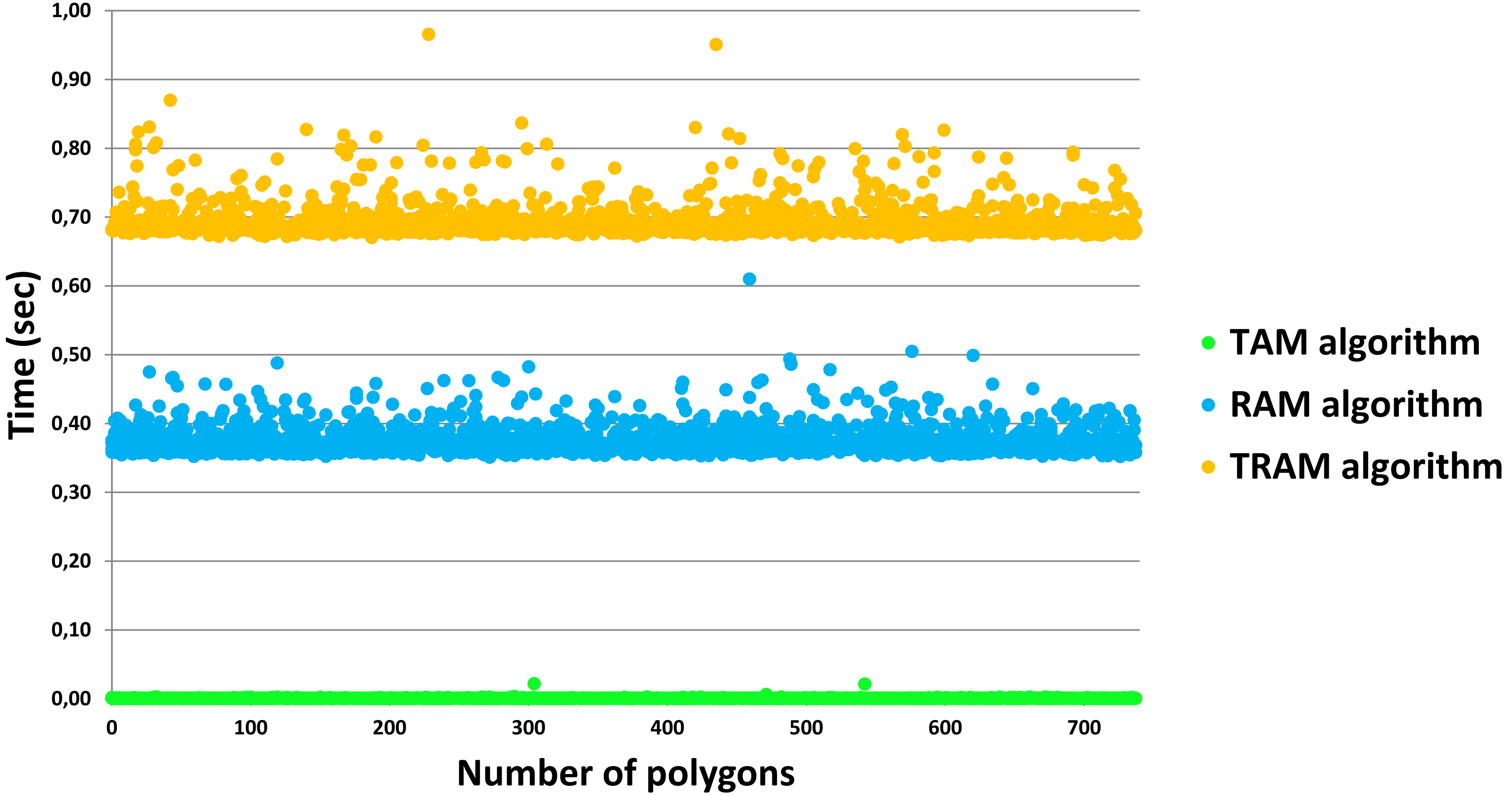
		Evaluated time		
	Type of adjacency constraints	Algorithm processing time	Constraints creating time	Solving time
$\mathbf{M} \cdot x \leq \mathbf{A} \cdot 1$ $\mathbf{M} = \mathbf{A} + \mathbf{B}$	Triangular adjacency matrix (TAM)	✓	✓	✓
	Row adjacency matrix (RAM)	✓	✓	✓
	Triangular row adjacency matrix (TRAM)	✓	✓	✓
$x_i + x_j \leq 1$	Pairwise constraints	✗	✓	✓

Results

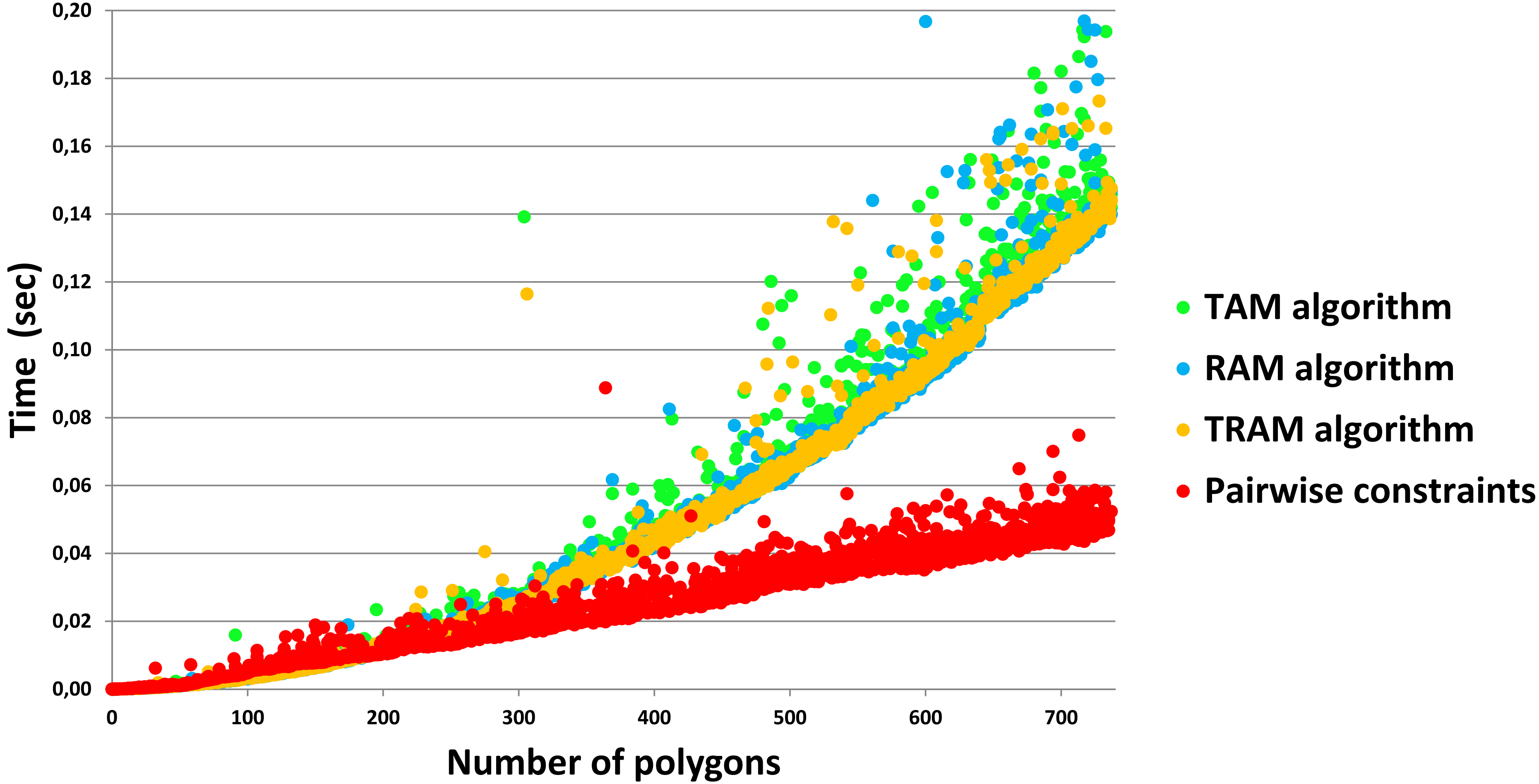
Neighbourhood	Neighbours statistics		
	Mean	Standard deviation	Max / Min
Neumann	3.9	2.9	28 / 0
Moore	4.2	3.1	28 / 0
Buffer	7.0	5.3	44 / 0

	Number of resulted constraints			
	TAM	RAM	TRAM	Pairwise constraints
Neumann	0 - 522	0 – 423	0 - 423	0 – 1,472
Moore	0 - 530	0 – 441	0 - 441	0 – 1,628
Buffer	0 - 620	0 - 527	0 - 527	0 – 2,778

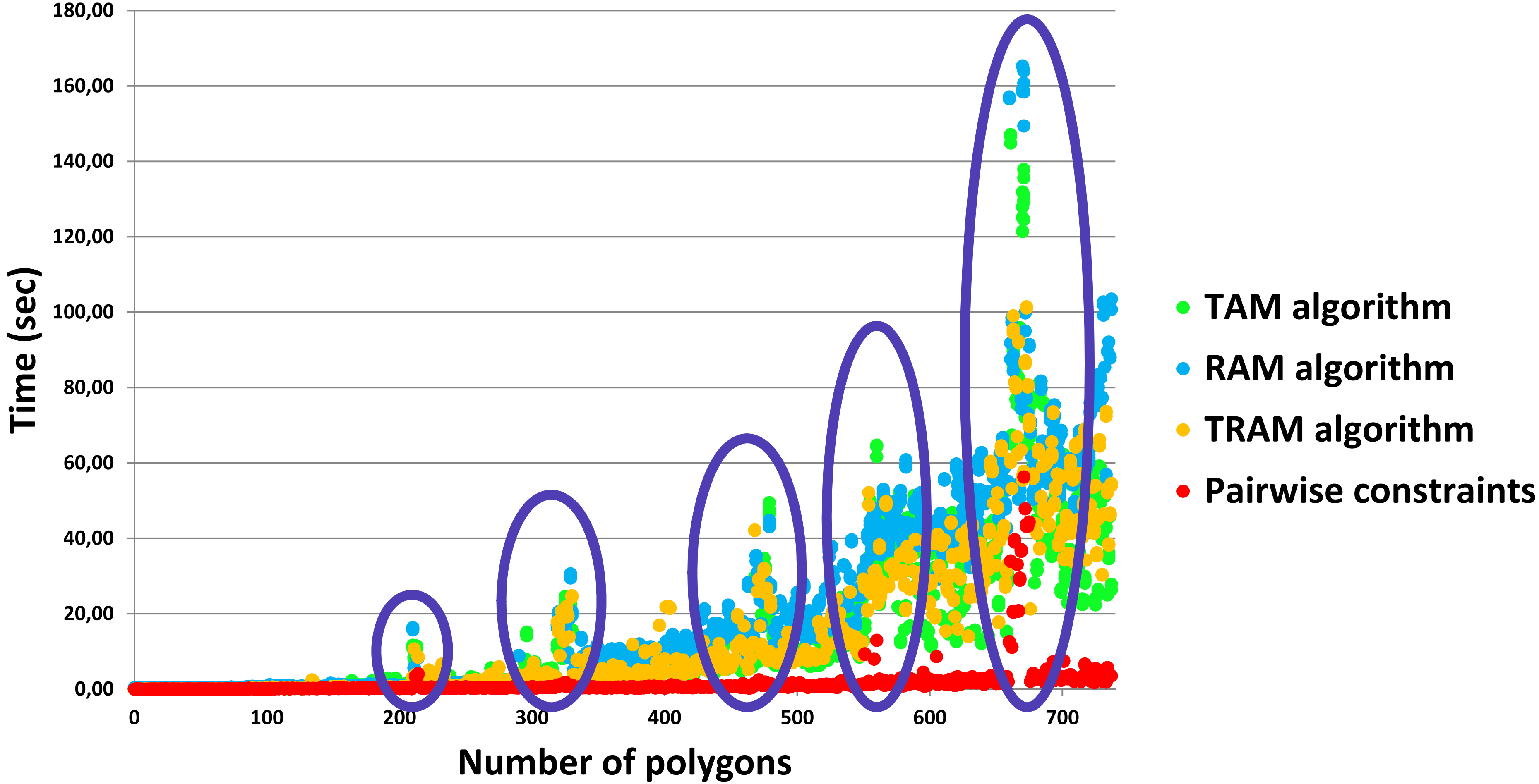
PROCESSING TIME FOR DIFFERENT ALGORITHMS – Buffer neighbourhoods



CONSTRAINTS CREATING TIME FOR DIFFERENT ALGORTIHMS – Buffer neighbourhoods



MODEL SOLVING TIME OF DIFFERENT ALGORITHMS – Buffer neighbourhoods



DISCUSSION AND CONCLUSION

- Time needed for creating constraints in the model and solving is strongly related to the number of neighbours

- Solving time depends not only on the number of polygons but also on the spatial structure

- **Analytical algorithms reduce the total number of constraints, however they can lead to the loss of eficiency of branch-and-cut algorithm for solving the model**

Thank you for your attention!

This research was supported by the project of the National Agency for Agriculture Research (No. QJ1320230) and the Internal Grant Agency of Faculty of Forestry and Wood Sciences Czech University of Life Sciences in Prague (No. B07/15).