# ALC 2022/2023
# $2^{nd}$ Project – Harvest Scheduling using SMT

19/Oct/2022, version 1.0

## Overview

The 2nd ALC project is to develop a software tool for solving the Harvest Scheduling problem. In order to solve this problem, students must use solvers for Satisfiability Modulo Theories (SMT).

## Problem Specification

Consider an agricultural area that is split into several plots of land. Each plot of land is also known as a *unit* where some agricultural product will be planted. The product to be planted in each unit depends on the type of soil, topography, among other features.

However, the agricultural area contains a specie of land animal that needs protection. Hence, part of the overall agricultural area is going to be defined as a natural reserve where farming will not be allowed. The natural reserve to be defined must have a given minimum size area. Another key feature of the natural reserve is that it needs to be contiguous such that the animals to be protected can move around the natural reserve without leaving it.

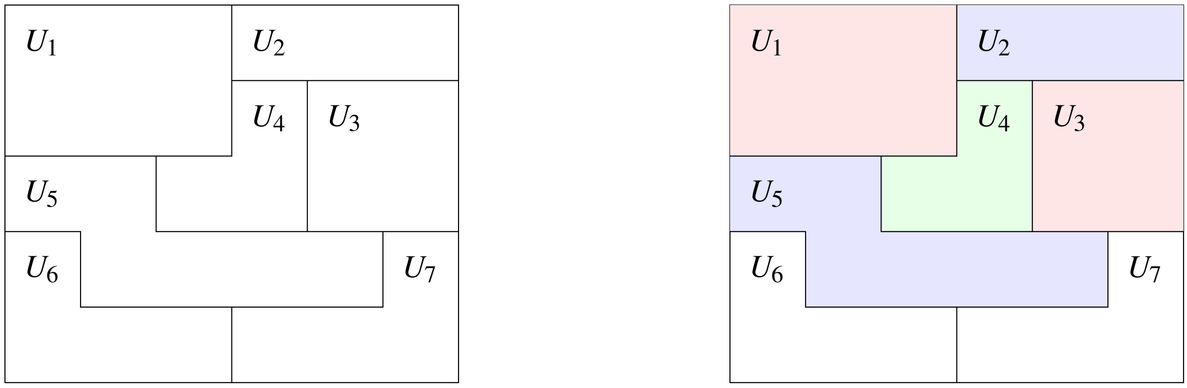
The harvest problem with adjacency constraints is not new [5]. There has also been several works on how to design connected natural reserves [3, 2]. Note that each work has its specific goals and properties to be satisfied. Moreover, contiguity constraints are common to many problems such as the electoral districting problem [4, 6, 1].

Let $U$ define the set of $n$ units of land. All units in $U$ are numbered from 1 to $n$, i.e. $U = \{1, 2, \ldots, n\}$. Let $A_i$ define the area of unit $i \in U$.

For each unit of land the farmer defines the type of agricultural product that is the most appropriate of that particular unit such that it provides a given profit. However, the profit depends on when the harvesting can be made. Let $T = \{1, 2, \ldots, k\}$ define the set of $k$ time periods for harvesting. Moreover, let $P_{ij}$ denote the profit of harvesting unit $i$ ($i \in U$) in time period $j$ ($j \in T$).

Your goal is to help building the harvest's schedule of the agricultural area such that the profit is maximized. However, there are several restrictions that need to be satisfied:

- Each unit of land cannot be harvested more than once in the $T$ time periods.
- Two adjacent units (i.e. units that share a common frontier) cannot be harvested in the same time period.




(a) Units of land

(b) Solution

Figure 1: Split of the agricultural land in units (left) and solution (right)

| | $U_1$ | $U_2$ | $U_3$ | $U_4$ | $U_5$ | $U_6$ | $U_7$ |
|---|---|---|---|---|---|---|---|
| $A_i$ | 6 | 3 | 4 | 3 | 6 | 4 | 4 |

Table 1: Areas for each unit of land in Figure 1.

- The natural reserve must be a contiguous area with a given minimum area size of $A_{min}$ such that $A_{min} \geq 0$.

Consider a problem instance where the land is divided into 7 units as in Figure 1 and the areas of the units are presented in Table 1. Suppose that there are 3 harvest periods and Table 2 shows the profits for each unit when harvested in each of the 3 periods. Moreover, consider also that the minimum size of the natural reserve area must be at least 7.

Figure 1 (b) shows an optimal solution to the problem with a total profit of 18. Additionally, the units in blue, red and green are the ones harvested in periods 1, 2 and 3, respectively. Moreover, the units in white define the natural reserve.

Observe that the units selected for the natural reserve define a contiguous area. Moreover, the size of the natural reserve is 8, which is larger than the defined minimum size.

In this example, all units are either harvested or part of the natural reserve. However, there could be optimal solutions where one or more units are neither harvested or included in the natural reserve. Finally, in a given period, the number of harvested units might be 0.

# Project Goals

You are to implement a tool, or optionally a set of tools, invoked with command `proj2`. This set of tools should use a SMT solver to compute the harvest scheduling problem.

| Period | $U_1$ | $U_2$ | $U_3$ | $U_4$ | $U_5$ | $U_6$ | $U_7$ |
|---|---|---|---|---|---|---|---|
| 1 | 5 | 1 | 4 | 5 | 5 | 1 | 1 |
| 2 | 5 | 1 | 3 | 3 | 3 | 2 | 1 |
| 3 | 4 | 1 | 3 | 4 | 3 | 2 | 1 |

Table 2: Harvest profit in each period for each unit of land in Figure 1.

Your tool does not take any command-line arguments. The problem instance is to be read from the standard input.

Consider an instance file named `instance.hsp`. The tool is expected to be executed as follows:

```
proj2 < instance.hsp > solution.txt
```

The tool must write the solution to the standard output, which can then be redirected to a file (e.g., `solution.txt`).

The programming languages to be used are only C/C++, Java or Python. The formats of the files used by the tool are described below.

# File Formats

You can assume that all input files follow the description provided in this document. There is no need to check if the input file is correct. Additionally, all lines (input or output) must terminate with the end-of-line character.

## Input Format

The input file representing a problem instance is a text file that follows the following format:

- One line with an integer $n$ $(n > 1)$ defining the number of units;
- One line with an integer $k$ $(k > 1)$ defining the number of periods;
- One line with $n$ integers defining the area size of each of the $n$ units;
- A sequence of $n$ lines where the $i^{th}$ line defines the adjacent units of unit $i$. Each of the $n$ lines contains one integer $m$ $(m < n)$ followed by $m$ integers corresponding to the adjacent units;
- A sequence of $k$ lines where each line contains $n$ integers. The $j^{th}$ line of this sequence defines the profit for each unit in period $j$;
- One line with an integer defining the minimum area size for the natural reserve;

Finally, observe that units are always numbered from 1 to $n$ and that integers in the same line are always separated by a white space.

### Output Format

The output representing an optimal solution to the problem instance must comply with the following format:

- One line with an integer defining the profit obtained in the optimal solution;
- A sequence of $k$ lines where the $j^{th}$ line contains the number of units harvested in period $j$ followed by the identifiers of the respective harvested units in that period. The identifiers do not have to be sorted;
- One line with the number of units that define the natural reserve followed by the respective unit identifiers. The unit identifiers do not have to be sorted;

**Important:** The final version to be submitted for evaluation must comply with the described output. Project submissions that do not comply will be severely penalized, since each incorrect output will be considered as a wrong answer. An application that verifies if the output complies with the description will be available on the course's website.

## Example 1

The file describing the problem shown as an example is as follows:

```
7
3
6 3 4 3 6 4 4
3 2 4 5
3 1 3 4
4 2 4 5 7
4 1 2 3 5
5 1 4 3 6 7
2 5 7
3 5 6 3
5 1 4 5 5 1 1
5 1 3 3 3 2 1
4 1 3 4 3 2 1
7
```

The optimal solution corresponding to Figure 1 would be:

```
18
2 2 5
2 1 3
1 4
2 6 7
```

## Additional Information

The project is to be implemented in groups of one or two students.

The project is to be submitted through the course website. Jointly with your code, you should submit a short report describing the main features of your project. The report to be submitted must describe the problem to be solved, how to install and run your project, as well as a description of the encoding and of the algorithm used to solve it.

The evaluation will be made taking into account correctness and efficiency given a reasonable amount of CPU time (80%) and the report (20%).

The input and output formats described in this document must be strictly followed.

## Project Dates

- Project published: 20/10/2022.
- Project due: 02/11/2022 at 23:59.

## Omissions & Errors

Any detected omissions or errors will be added to future versions of this document. Any required clarifications will be made available through the course's official website.

## Versions

19/Oct/2022, version 1.0: Original version.

## References


[1] T. Almeida and V. M. Manquinho. Constraint-based electoral districting using a new compactness measure: An application to portugal. *Comput. Oper. Res.*, 146:105892, 2022.

[2] H. L. Beyer, Y. Dujardin, M. E. Watts, and H. P. Possingham. Solving conservation planning problems with integer linear programming. *Ecological Modelling*, 328:14–22, 2016.

[3] A. Billionnet. Designing an optimal connected nature reserve. *Applied Mathematical Modelling*, 36(5):2213–2223, 2012.

[4] L. Kotthoff, B. O'Sullivan, S. S. Ravi, and I. Davidson. Complex Clustering Using Constraint Programming: Modelling Electoral Map Creation. In *14th International Workshop on Constraint Modelling and Reformulation*, Sept. 2015.

[5] M. E. McDill, S. A. Rebain, and J. Braze. Harvest scheduling with area-based adjacency constraints. *Forest Science*, 48(4):631–642, 2002.

[6] H. Validi, A. Buchanan, and E. Lykhovyd. Imposing contiguity constraints in political districting models. *Oper. Res.*, 70(2):867–892, 2022.