

ARMLIB 0.1: Simple Example and Formulations

Juan Pablo Vielma and Marcos Goycoolea

October 24, 2007

1 Introduction

In this file we introduce an simple example for the ARMLIB 0.1 instances format. We also describe two basic integer programming formulations for the ARM problem (see [3, 5, 6]) and the file formats associated to them. These two formulations were chosen mainly for their simplicity; we note that there are many other alternative formulations for the ARM problem (see [1, 2, 4, 7]) The objective of this file is not to define a standard format for ARM integer programming formulations, but to provide examples that may be of help when using the ARMLIB 0.1 DataSet and Instance formats.

2 Example

In this section we define a forest example used to illustrate the DataSet and Instance formats. The forest map of the example is depicted in figure 1.

1	2	3
4	5	6

Figure 1: Example forest map

The stand set of this forest is $V = \{1, 2, 3, 4, 5, 6\}$ and tables 1–5 define stand attributes for these stands and for three periods.

v	1	2	3	4	5	6
a_v	1	1	1	2	2	2

Table 1: Area for example: in file `area.area`

v	1	2	3	4	5	6
$p_{v,1}$	10	10	10	20	20	20
$p_{v,2}$	11	11	11	22	22	22
$p_{v,3}$	12	12	12	24	24	24

Table 2: Profit for example: in file `profit.profit`

v	1	2	3	4	5	6
$\alpha_{v,1}$	10	10	10	20	20	20
$\alpha_{v,2}$	11	11	11	22	22	22
$\alpha_{v,3}$	12	12	12	24	24	24

Table 3: First set of volumes for example: in file `first_volume.volume`

v	1	2	3	4	5	6
$\alpha_{v,1}$	30	30	30	10	10	10
$\alpha_{v,2}$	33	33	33	11	11	11
$\alpha_{v,3}$	36	36	36	12	12	12

Table 4: Second set of volumes for example: in file `second_volume.volume`

v	0	1	2	3	4	5
g_v	1	1	1	2	2	2

Table 5: Initial age for example: in file `age.age`

We also define two types of adjacencies for this example. Table 2 defines the neighbours when two stands are considered adjacent if they touch and table 2 does the same when two stands are considered adjacent if they touch at least in a line segment.

v	$N(v)$
1	$\{2, 4, 5\}$
2	$\{1, 3, 4, 5, 6\}$
3	$\{2, 5, 6\}$
4	$\{1, 2, 5\}$
5	$\{1, 2, 3, 4, 6\}$
6	$\{2, 3, 5\}$

Table 6: First type of adjacency: in file `point.adjacency`

v	$N(v)$
1	$\{2, 4\}$
2	$\{1, 3, 5\}$
3	$\{2, 6\}$
4	$\{1, 5\}$
5	$\{2, 4, 6\}$
6	$\{3, 5\}$

Table 7: Second type of adjacency: in file `line.adjacency`

3 Integer Programming formulations

In what follows, we denote a set of nodes $C \subset V$ as a *cluster* for an adjacency type if it is contiguous under that adjacency.

Sections 3.1 and 3.2 describe the basics of two types of formulations and section 3.4 describes the formats of some files associated to the formulations.

3.1 Cell Formulation

This formulation uses one binary variable for each node $v \in V$ and each period t . The formulation is based in the concept of *minimally infeasible cluster*. A *cluster* C is an *infeasible cluster* for maximum area A if $\sum_{v \in C} a_v > A$. An *infeasible cluster* C is a *minimally infeasible cluster* if for all $v \in C$ either $C \setminus \{v\}$ is not contiguous for corresponding adjacency type or $\sum_{v \in C \setminus \{v\}} a_v \leq A$. We denote the set of all *minimally infeasible cluster* as Λ^+ .

3.1.1 Basic Formulation

The basic cell formulation is:

$$\max \sum_{t \in \mathcal{T}} \sum_{v \in V} p_{v,t} y_{v,t} \quad (1)$$

s.t.

$$\sum_{v \in C} y_{v,t} \leq |C| - 1 \quad \forall C \in \Lambda^+, \forall t \in \mathcal{T} \quad (2)$$

$$\sum_{t \in \mathcal{T}} y_{v,t} \leq 1 \quad \forall v \in V \quad (3)$$

$$y_{v,t} \in \{0, 1\} \quad \forall v \in V, \forall t \in \mathcal{T} \quad (4)$$

where

$$\bullet \ y_{v,t} = \begin{cases} 1 & \text{if the stand associated to } v \text{ is harvested in period } t. \\ 0 & \text{if not.} \end{cases}$$

3.1.2 Volume Flow Constraints

Volume flow constraints are implemented in the cell formulation as:

$$\sum_{v \in V} \alpha_{v, \text{next}(t)} y_{v, \text{next}(t)} \leq (1 - \frac{L}{100}) \sum_{v \in V} \alpha_{v,t} y_{v,t} \quad \forall t \in \mathcal{T} \setminus \{\text{last}(\mathcal{T})\} \quad (5)$$

$$\sum_{v \in V} \alpha_{v, \text{next}(t)} y_{v, \text{next}(t)} \geq (1 + \frac{U}{100}) \sum_{v \in V} \alpha_{v,t} y_{v,t} \quad \forall t \in \mathcal{T} \setminus \{\text{last}(\mathcal{T})\} \quad (6)$$

3.1.3 Average Ending Age Constraints

Average ending age constraints are implemented in the cell formulation as:

$$\begin{aligned} & \sum_{v \in \bar{V}} a_v (g_v + \bar{N} Y - (\bar{N} Y + g_v) y_{v, \text{last}(t)}) \\ & - \sum_{v \in \bar{V}} a_v \left(\sum_{t \in \mathcal{T} \setminus \{\text{last}(\mathcal{T})\}} (\text{next}(t) Y + g_v) y_{v,t} \right) \geq \bar{G} \sum_{v \in \bar{V}} a_v \quad (7) \end{aligned}$$

3.2 Cluster Formulation

We say that a cluster C is feasible for a given maximum area A if $\sum_{v \in C} a_v \leq A$. We denote the set of all feasible clusters by Ω . This formulation uses a binary variable for each feasible cluster C . For each feasible cluster C we define:

$$\begin{aligned} \bullet \ g_C &= \sum_{v \in C} g_v. \\ \bullet \ a_C &= \sum_{v \in C} a_v. \end{aligned}$$

- $p_{C,t} = \sum_{v \in C} p_{C,t}$.
- $\alpha_{C,t} = \sum_{v \in C} \alpha_{v,t}$.

3.2.1 Basic Formulation

The basic cluster formulation is:

$$\max \sum_{t \in \mathcal{T}} \sum_{C \in \Omega} p_{C,t} x_{C,t} \quad (8)$$

s.t.

$$\sum_{c \in \Omega(K)} x_{C,t} \leq 1 \quad \forall K \in \Pi, \forall t \in \mathcal{T} \quad (9)$$

$$\sum_{t=1}^T \sum_{C \in \Omega(\{v\})} x_{C,t} \leq 1 \quad \forall v \in V \quad (10)$$

$$x_{C,t} \in \{0, 1\} \quad \forall C \in \Omega, \forall t \in \mathcal{T} \quad (11)$$

where

- $x_{C,t} = \begin{cases} 1 & \text{if cluster } C \text{ is harvested in period } t \\ 0 & \text{if not.} \end{cases}$
- Π is the set of all *maximal cliques* in the forest graph $G(V, E)$.
- $\Omega(K)$ is the set of all clusters that intersect *maximal clique* K .
- $\Omega(\{v\})$ is the set of all clusters that contain *basic cell* v .

3.2.2 Volume Flow Constraints

Volume flow constraints are implemented in the cluster formulation as:

$$\sum_{C \in \Omega} \alpha_{C, \text{next}(t)} x_{C, \text{next}(t)} \leq (1 - \frac{L}{100}) \sum_{C \in \Omega} \alpha_{C,t} x_{C,t} \quad \forall t \in \mathcal{T} \setminus \{\text{last}(\mathcal{T})\} \quad (12)$$

$$\sum_{C \in \Omega} \alpha_{C, \text{next}(t)} x_{C, \text{next}(t)} \geq (1 + \frac{U}{100}) \sum_{C \in \Omega} \alpha_{C,t} x_{C,t} \quad \forall t \in \mathcal{T} \setminus \{\text{last}(\mathcal{T})\} \quad (13)$$

3.3 Average Ending Age Constraints

Average ending age constraints are implemented in the cluster formulation as:

$$\begin{aligned}
& \sum_{v \in \bar{V}} a_v \left(g_v + \bar{N} Y - (\bar{N} Y + g_v) \sum_{C \in \Omega(\{v\})} x_{C, \text{last}(t)} \right) \\
& - \sum_{v \in \bar{V}} a_v \left(\sum_{t \in \mathcal{T} \setminus \{\text{last}(\mathcal{T})\}} (\text{next}(t) Y + g_v) \sum_{C \in \Omega(\{v\})} x_{C, t} \right) \geq \bar{G} \sum_{v \in \bar{V}} a_v \quad (14)
\end{aligned}$$

3.4 Formulation File Formats

3.4.1 Set Files

We use the set file format to save feasible clusters, minimally infeasible clusters and cliques. Clusters are stored in a text file named `clusters_A_adj.txt` where A is the maximum area and `adj` is the name of the adjacency file used to generate them. Minimally infeasible clusters are stored in a text file named `NDMIS_A_adj.txt` where A is the maximum area and `adj` is the name of the adjacency file used to generate them. Cliques are stored in a text file named `cliques_adj.txt` where `adj` is the name of the adjacency file used to generate them.

Each file contains a family $\mathcal{S}$ of subsets of N . We assume that $\mathcal{S} = \{S^1, S^2, \dots, S^{|\mathcal{S}|}\}$ where $S^i \subset N$ for all $i \in \{1, \dots, |\mathcal{S}|\}$. We further assume that each $S^i = \{S_{i,1}, \dots, S_{i,|S^i|}\}$. The format is:

```

|S|
1 ∪ |S1|
S1,1 ∪ S1,2 ∪ ... ∪ S1,|S1|
```

Format 1: Format of set files.

3.4.2 MPS and LP file formats

The cell and cluster formulation for an instance described by `instance.arm` are stored in files named `instance_cell.ext` and `instance_cluster.ext` respectively, where `ext` is either `lp` or `mps`.

The variables and constraints are named as follows.

- Clusters variables $x_{C, \mathbf{TP}}$ are named `clusterCNtTP`, where `CN` is the cluster number as in `clusters_A_adj.txt` and `TP` is the time period in $\mathcal{T}$.

- Cell variables $y_{v, \mathbf{TP}}$ are named **cellvtTP**, where **v** is the cell in V and **TP** is the time period in $\mathcal{T}$.
- Clique constraints (9) are named **cliqueCNtTP**, where **CN** is the clique number as in **cliques_adj.txt** and **TP** is the time period in $\{1, \dots, T\}$.
- Cell constraints (10) are named **cellvtTP**, where **v** is the cell in V and **TP** is the time period in $\mathcal{T}$.
- Minimally infeasible cluster constraints (2) are named **ndmisMICNtTP**, where **MICN** is the minimally infeasible cluster number as in **NDMIS_A_adj.txt** and **TP** is the time period in $\mathcal{T}$.
- Volume constraints (5) and (12) are named **vfNLtTP** and **vfNUtTP** respectively, where **N** is the order in which the constraint is defined in **instance.arm** and **TP** is the time period in $\mathcal{T}$.
- Average ending age constraints (7) and (14) are named **avgage**.

References

- [1] M. Constantino, I. Martins, and J. Borges. A new mixed integer programming model for harvest scheduling subject to maximum area restrictions. 2006. To appear in *Operations Research*.
- [2] K. Crowe, J. Nelson, and M. Boyland. Solving the area-restricted harvest-scheduling model using the branch and bound algorithm. *Canadian J. Forest Research-revue Canadienne De Recherche Forestiere*, 33(9):1804–1814, 2003.
- [3] M. Goycoolea, A. T. Murray, F. Barahona, R. Epstein, and A. Weintraub. Harvest scheduling subject to maximum area restrictions: Exploring exact approaches. *Operations Research*, 53(3):490–500, 2005.
- [4] E. A. Gunn and E. W. Richards. Solving the adjacency problem with stand-centred constraints. *Canadian J. Forest Research-revue Canadienne De Recherche Forestiere*, 35(4):832–842, 2005.
- [5] I. Martins, M. Constantino, and J. G. Borges. A column generation approach for solving a non-temporal forest harvest model with spatial structure constraints. *European J. Operational Research*, 161(2):478–498, 2005.
- [6] M. E. McDill, S. A. Rebain, and J. Braze. Harvest scheduling with area-based adjacency constraints. *Forest Science*, 48(4):631–642, 2002.
- [7] S. F. Tóth, M. E. McDill, and S. George. Strengthening cover inequalities for area-based adjacency formulations of harvest scheduling models. 2007. Submitted to *Operations Research*.