APGen - The assignment problem generator

Lars Relund Nielsen[*]
Department of Accounting, Finance and Logistics
Aarhus School of Business
Fuglesangs Allé 4
DK-8210 Aarhus V
Denmark

Christian Roed Pedersen[†]
Department of Operations Research
University of Aarhus
Ny Munkegade, Building 1530
8000 Aarhus C
Denmark

Version 2.0 - Dec. 2005

Abstract

This manual provide documentation for the *Assignment Problem Generator* (*APGen*) which generate problem instances for the bicriterion assignment problem. Also instances for the bicriterion multi modal assignment problem can be generated. The instance generated will be output as an xml file which may be converted to a desired format using an xslt stylesheet. Instances for similar single criterion problems can also be generated by e.g. only considering the first criterion.

Keywords: Linear assignment problem, testing, generator.

Contents

1 Introduction

This manual provide documentation for the *Assignment Problem Generator* (*APGen*) which generate problem instances for the bicriterion assignment problem (BiAP). Also instances for the bicriterion multi modal

[*] Corresponding author, e-mail: lars@relund.dk.

[†] e-mail: roed@imf.au.dk.

row \ column	0	1
0	$\begin{pmatrix}547\\532\end{pmatrix}$	$\begin{pmatrix}138\\777\end{pmatrix}$ $\begin{pmatrix}357\\721\end{pmatrix}$ $\begin{pmatrix}594\\462\end{pmatrix}$ $\begin{pmatrix}840\\272\end{pmatrix}$
1	$\begin{pmatrix}113\\828\end{pmatrix}$ $\begin{pmatrix}178\\732\end{pmatrix}$ $\begin{pmatrix}493\\408\end{pmatrix}$ $\begin{pmatrix}752\\219\end{pmatrix}$	$\begin{pmatrix}85\\320\end{pmatrix}$

Figure 1: An example of the assignment cost matrix $(n = 2)$.

assignment problem can be generated (BiMMAP). For a short introduction to BiAP and BiMMAP see Appendix A. An instance generated will be output as an xml file which may be converted to a desired format using an xslt stylesheet. For an introduction to xml and xslt see Møller and Schwartzbach [1].

The APGen generator has been used to generate instances for testing in e.g. Pedersen, Nielsen, and Andersen [2, 3].

The underlying data structure used is an *assignment cost matrix* as shown in Figure 1. Given the *dimension* n of the assignment problem, we consider a matrix with n rows and columns numbered 0,…,$n-1$. For a fixed row i and column j we consider a given *cell* containing a set of L_{ij} *entries*, namely, the set of bicriterion costs $(c^1_{ij1}, c^2_{ij1}), \dots, (c^1_{ijL_{ij}}, c^2_{ijL_{ij}})$.

The documentation is ordered as follows. The program parameters are given in Section 2 and output is described in Section 3. The remaining section describe the public members for the generator class in the class library, which are available on request. Finally, Appendix A gives a short introduction to BiAP and BiMMAP.

2 Program parameters

The program uses command line passing for catching a number of run directives. The following flags/options can be used:

`-dim` Dimension n of problem instance to be generated (compulsory).

`-maxEnt` Upper bound for entry range (compulsory).

`-minEnt` Lower bound for entry range (optional - default is one). This entry must of course not be larger than `maxEnt`.

`-maxCost` Upper bound for cost range (compulsory), i.e. costs are generated in $\{0, \dots, \texttt{maxCost}\}$. Observe that `maxCost` must be chosen sufficiently large ($\geq$ `maxEnt` $-\,1$).

`-method` The method used for generating costs (see Section 2.1).

`-shape` The shape parameter. Possible values depend on which method is used (see Section 2.1).

`-out` Output file name without xml extension (compulsory).

`-seed` The seed (if not specified generate one randomly).

All parameters are given as integers except `out`. Example:

```
apgen -dim 8 -maxEnt 15 -minEnt 10 -maxCost 1000 -out apmat -seed 23413 -method
1 -shape 10
```

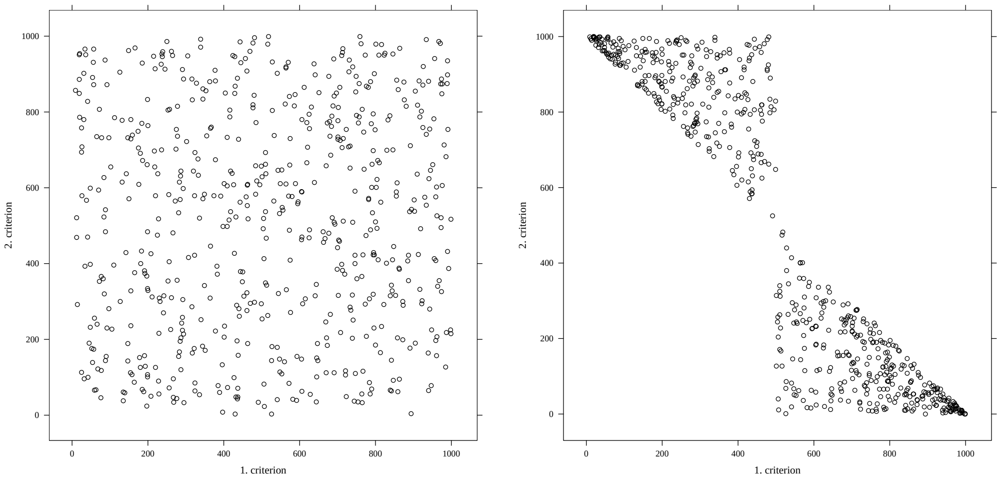

Figure 2: BiAP instance - Costs for all cells using $\mathsf{shape} = 0$ (left) and $\mathsf{shape} = 1$ (right).

```
<?xml version="1.0" encoding="ISO-8859-1"?> <!--Cost matrix for the
bicriterion (multi modal)
 assignment problem generated using APGen.
 Lars Relund Nielsen and Christian Roed Pedersen (c)-->
<apmat dimension="2" version="2.0" costrange="0-1000"
   entryrange="1-4" method="1" shape="0" seed="234213">
    <cell row="0" column="0" entries="1">
        <entry c1="547" c2="532"/>
    </cell>
    <cell row="0" column="1" entries="4">
        <entry c1="138" c2="777"/>
        <entry c1="357" c2="721"/>
        <entry c1="594" c2="462"/>
        <entry c1="840" c2="272"/>
    </cell>
    <cell row="1" column="0" entries="4">
        <entry c1="113" c2="828"/>
        <entry c1="178" c2="732"/>
        <entry c1="493" c2="408"/>
        <entry c1="752" c2="219"/>
    </cell>
    <cell row="1" column="1" entries="1">
        <entry c1="85" c2="320"/>
    </cell>
</apmat>
```

Figure 3: The xml output file.

2.1 Possible methods and shapes

Cost are generated differently depending on whether we generate instances for BiAP or BiMMAP. If BiAP is considered, then $\texttt{maxEnt} = 1$ and `shape` define the correlation between the costs. If $\texttt{shape} = 0$, costs are generated randomly in $\{0, \dots, \texttt{maxCost}\}$, and if $\texttt{shape} = 1$, costs (c^1, c^2) are generated negatively correlated as follows

$$
\begin{aligned}
c^1 < \frac{\texttt{maxCost}}{2} &\Rightarrow c^2 \in \left\{\texttt{maxCost} - c^1, \dots, \texttt{maxCost}\right\} \\
c^1 \geq \frac{\texttt{maxCost}}{2} &\Rightarrow c^2 \in \left\{0, \dots, \texttt{maxCost} - c^1\right\}
\end{aligned}
$$

Plots of the costs in all cells of a BiAP instance of size $n = 50$ using shape 0 and 1 are given in Figure 2. The `method` parameter have no effect if we consider BiAP instances.

If we consider BiMMAP, costs can be generated using 3 different methods. If $\texttt{method} = 1$, the $\texttt{shape} \in \{-100, \dots, 100\}$ parameter describes the curve of the function along which the entries are generated. A negative shape corresponds to generating the entries along a concave-like function, using shape 0 generates entries fluctuating along a straight line, and finally, a positive shape means generating entries along a convex-like function. Some plots of the cell cost vectors for instances generated using $\texttt{method} = 1$ and different shapes are given in Figures 4-6 ($n = 2$). Note, using negative shapes, in general, generate many unsupported nondominated cost vectors for each cell, while positive shapes generate many extreme supported nondominated cost vectors for each cell.

If $\texttt{method} = 2$, the cost interval is split into $2{\cdot}\texttt{shape}$ intervals and costs are generated in intervals $1, 3, 5, \dots, \texttt{shape} - 1$. This method can be used to generate costs in groups. Note, the number of groups is equal to `shape`, and we assume that $\texttt{shape} \in \{1, \dots, \lfloor \texttt{maxEnt}/2 \rfloor\}$. Some plots of the cell costs for instances generated using $\texttt{method} = 2$ and shapes 3 and 4 are given in Figures 7-8 ($n = 2$).

If $\texttt{method} = 3$, the two-dimensional cost interval is split into four equally sized regions. Assuming that the cells are numbered from 1 to n^2 (row-wise numbering) then the cost vectors in the cells $1, 3, \dots$ are generated using the upper/left cost region, and cost vectors in the cells $2, 4, \dots$ are generated using the lower/right cost region. That is, the costs differ from cell to cell as shown in Figure 9 ($n = 2$). The `shape` parameter is defined as for $\texttt{method} = 1$.

3 Output

Output is written to <filename>.xml where <filename> is the filename specified by `out`. The xml format is simple to understand and illustrated in Figure 3 which is the corresponding xml file to the assignment cost matrix in Figure 1. Note, in general, the xml format is very verbose resulting in large file sizes. However, the xml file may be converted to a desired format using an xslt stylesheet.

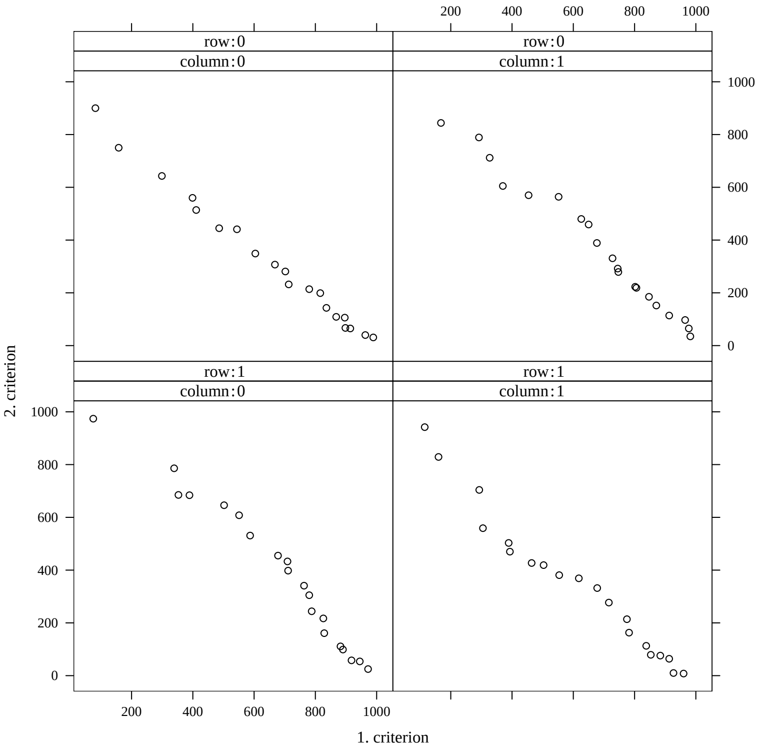

Figure 4: Costs in the different cells for an instance generated with dimension=2, `method` $= 1$ and `shape` $= 0$.

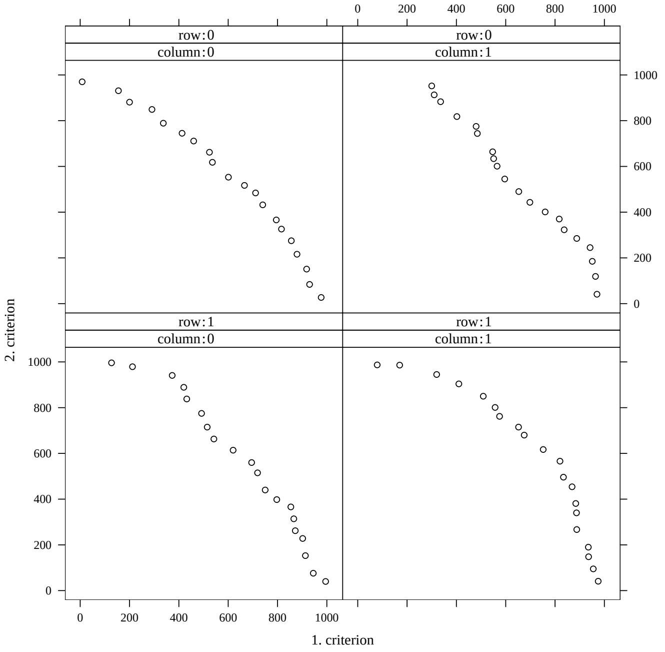

Figure 5: Costs in the different cells for an instance generated with dimension=2, `method` $= 1$ and `shape` $= -60$.

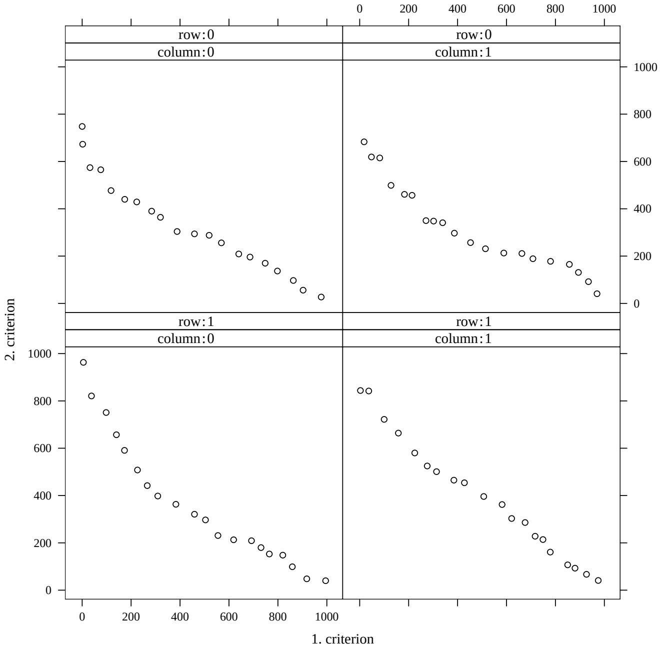

Figure 6: Costs in the different cells for an instance generated with dimension=2, `method` $= 1$ and `shape` $= 60$.

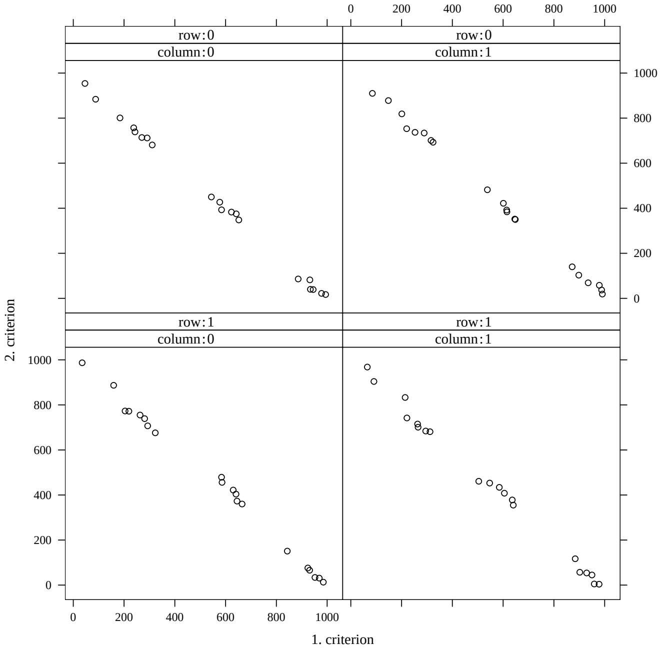

Figure 7: Costs in the different cells for an instance generated with dimension=2, `method` $= 2$ and `shape` $= 3$.

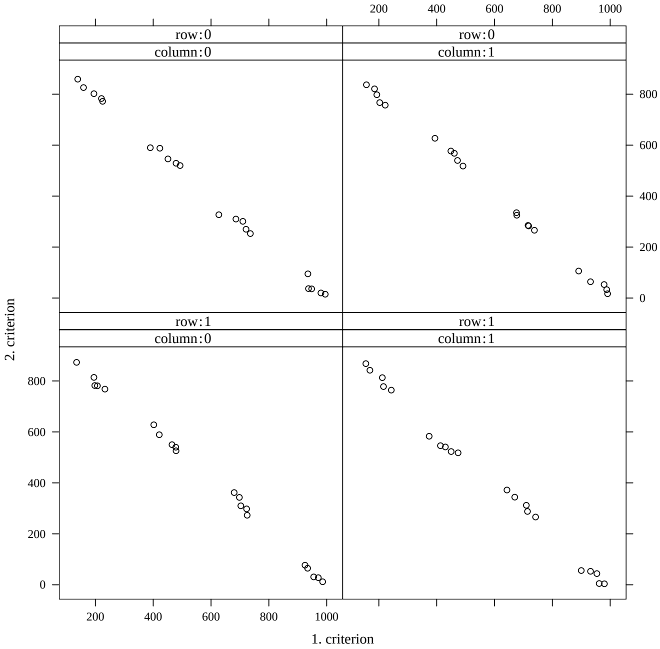

Figure 8: Costs in the different cells for an instance generated with dimension=2, `method` $= 2$ and `shape` $= 4$.

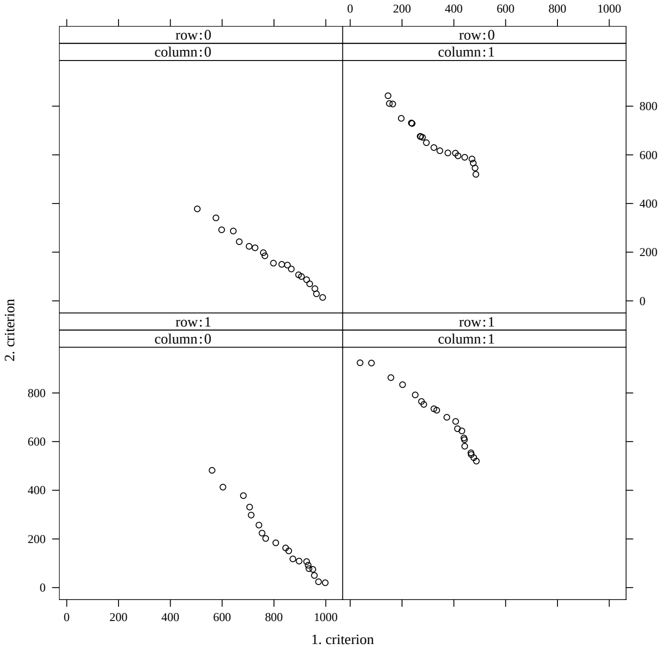

Figure 9: Costs in the different cells for an instance generated with dimension=2, `method` $= 3$ and `shape` $= 0$.

4 APGen Class Documentation

4.1 APMatGen Class Reference

4.1.1 Detailed Description

Class for generating problem instances for BiAP and BiMMAP.

The problem will be output as an xml file.

Author: Lars Relund Nielsen and Christian Roed Pedersen.

Version: 2.0

Public Member Functions

- APMatGen ()

  Initialize the object.

- void Generate ()

  Generate the problem instance and output it to an xml file (specified by SetValues).

- void SetValues (char $*$outFile, int dim, int minEntries, int maxEntries, int maxCost, int seedn, int genShape, int genMethod)

  Set input values.

References

[1] A. Møller and M.I. Schwartzbach. *An Introduction to XML and Web Technologies*. Addison-Wesley (in preparation), 2006.

[2] C.R. Pedersen, L.R. Nielsen, and K.A. Andersen. On the bicriterion multi modal assignment problem. Working Paper No. 2005/3, Department of Operations Research, University of Aarhus, 2005.

[3] C.R. Pedersen, L.R. Nielsen, and K.A. Andersen. A note on ranking assignments using reoptimization. Working Paper No. 2005/2, Department of Operations Research, University of Aarhus, 2005.

A Bicriterion assignment problems

The *linear assignment problem* (*AP*) is a well-known problem and may be considered as the problem of assigning n workers to n jobs. Each worker must be assigned to exactly one job and the objective is to minimize total cost. AP may be solved using the following IP-formulation

$$\begin{array}{lll} \text{min} & \sum_{i=1}^{n}\sum_{j=1}^{n} c_{ij}x_{ij} & \\ \text{st.} & \sum_{i=1}^{n} x_{ij} = 1, & j = 1, ..., n \\ & \sum_{j=1}^{n} x_{ij} = 1, & i = 1, ..., n \\ & x_{ij} \in \{0,1\} & i,j = 1, ..., n \end{array} \tag{1}$$

AP can be extended to consider two costs for each worker and job, named, the *bicriterion assignment problem* (*biAP*). Here we try to minimize both the first and second cost simultaneously. That is, we are interested in finding the set of *efficient assignments* or *nondominated costs* where one criterion can not be improved without worsening the other.

Another extension of AP is the *bicriterion multi modal assignment problem* (*BiMMAP*). Imagine a large global company with n specialists spread across the world and suppose that exactly n jobs have to be performed by these n specialist. Hence, each specialist must be assigned to exactly one job. For this problem, it seems relevant to consider two cost criteria to be minimized, namely travel time and travel cost. Since a specialist i possibly has multiple modes of transportation and routes to choose from in order to reach the destination of job j, several two-dimensional costs exists for each i and j. The BiMMAP is to find all efficient assignments of specialist to jobs. Note, for BiMMAP each cell in the assignment cost matrix may have more than one two-dimensional cost (see Figure 1). Moreover, all the costs in a cell will be nondominated.

Let x_{ijl} be a binary variable with value 1, if i is assigned to j using mode choice l, and 0 otherwise. Obviously, exactly one i must be assigned to each j using a specific mode choice which gives us the following mathematical formulation of BiMMAP.

$$\begin{aligned}
\min \ & \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{L_{ij}} c^1_{ijl}x_{ijl} \\
\min \ & \sum_{i=1}^{n}\sum_{j=1}^{n}\sum_{l=1}^{L_{ij}} c^2_{ijl}x_{ijl} \\
\text{st.} \ & \sum_{j=1}^{n}\sum_{l=1}^{L_{ij}} x_{ijl} = 1, \ \ i = 1,2,\ldots,n \\
& \sum_{i=1}^{n}\sum_{l=1}^{L_{ij}} x_{ijl} = 1, \ \ j = 1,2,\ldots,n \\
& x_{ijl} \in \{0,1\} \ \ \forall i,j,l
\end{aligned} \tag{2}$$

A two-phase procedure for BiMMAP applying an effective ranking scheme can be found in Pedersen et al. [3] and an algorithm for finding the K best assignments in Pedersen et al. [2].