Applications of p-median techniques to facilities design problems
Ashayeri, J.; Heuts, R.M.J.; Tammel, B.

Publication date:
1997

Link to publication

Citation for published version (APA):
Discussion paper

Tilburg University
APPLICATIONS OF P-MEDIAN TECHNIQUES TO FACILITIES DESIGN PROBLEMS: AN IMPROVED HEURISTIC

By J. Ashayeri, R. Heuts and B. Tammel

March 1997

ISSN 0924-7815
APPLICATIONS OF P-MEDIAN TECHNIQUES TO FACILITIES DESIGN PROBLEMS: AN IMPROVED HEURISTIC

J. Ashayeri, R. Heuts and B. Tammel
Department of Econometrics, Tilburg University, The Netherlands

Facilities design is closely related to efficient use of available resources. This paper presents a heuristic approach to solve two core problems of a good facilities design: facility location and facility layout. The latter group of problems will be solved for warehouse and production systems in particular. All these problems can be formulated as so-called p-median clustering problems. Teitz & Bart developed the vertex substitution method to solve those problems. This paper introduces effective improvements on this heuristic. The approach is tested on a large number of randomly generated cases and on problems taken from the literature. The results demonstrate the effectiveness and superiority of our method.

Key words: facilities design, clustering, heuristic approach

INTRODUCTION

Facilities design
In recent years it has become increasingly important to utilize the available resources as efficiently as possible. A term which is closely related with this aspect of competitive behaviour is facilities design. It is the design of the physical environment of an activity. In other words, facilities design attempts to organise the tangible fixed assets (e.g. buildings, machines, trucks) of an activity (e.g. warehousing, production) in such a way that the utilisation of resources is improved in order to attain the objectives of an activity.

To achieve a good facilities design, two aspects in particular are of great importance (see also Figure 1). First, the facility location needs to be carefully chosen. The determination of an appropriate facility location is in many cases essential for conducting a satisfactory process. Choosing where to build the facility depends on the distance between several demand locations and the potential facility locations, among other criteria. The costs involved with traversing these distances is an important factor.

Figure 1 shows that along with the facility location also the design of a good layout of the facility is important. This aspect of facilities design is relevant in several processes. In this paper we look at the design of the layout in warehouse and production systems in particular. The increasing interest for these layout aspects is a consequence of the following developments. In recent years many companies are redesigning their distribution systems from so-called indirect distribution to more centralised direct distribution. However, this forces companies to improve
the efficiency of location assignment planning in the warehouse. Otherwise, the order picking process may be too slow, because of wrong item locations. Another development is the strive for shorter throughput times by means of cell production. In practice a useful tool to solve the facilities design problem, is cluster analysis. The increasing possibilities and the availability of powerful computers has contributed enormously to this development. The goal of cluster analysis is to cluster elements which show a large amount of similarity for certain criteria. The different criteria that are used to attain a proper grouping, depend on the area where the cluster analysis is applied to.

![Classification of major decisions](image)

Figure 1: Classification of major decisions

Facility location problems and clustering problems with respect to warehouse and production systems will be taken into consideration. In fact they all can be formulated as so-called p-median clustering problems. Before describing the p-median clustering problem and the solution techniques developed for this purpose, a few fundamental concepts are explained first.

**The distance matrix**

Many methods for solving clustering problems start with a distance matrix $D$. This matrix provides an indication of the equality between every two elements with respect to a certain criterion. In the case of facility location problems the objective is to select $p$ locations where a facility might eventually be located, and to allocate several demand locations to these facility locations in such way that the total distance involved will be minimised. In this type of problems the distance matrix can more or less directly be obtained from practice; for example; an element $d_{ij}$ can represent the distance between demand location $i$ and location $j$, where the facility might be set up.

If a clustering problem appears in a warehouse or a production environment, the distance matrix will be based on a binary matrix, which has elements either 0 or 1. The objective with respect to warehouse systems is to cluster those products that show a large amount of similarity in the order pattern. The order data can be found in a product-order matrix $P$. This matrix shows whether a product is demanded in a certain order or not. This product-order matrix is formed as
follows. We consider \( m \) products and \( n \) orders and we form an \( m \times n \) matrix:

\[
p_{ij} = \begin{cases} 1, & \text{product } i \text{ is demanded in order } j \ (i=1,...,m \text{ and } j=1,...,n), \\ 0, & \text{otherwise}. \end{cases}
\]

Remark: The purpose with regard to production systems is to group machines into cells that show a large amount of similarity in their production processes. A similar matrix with only 0 and 1 elements is now called a machine-product matrix. The elements have now the following interpretation:

\[
p_{ij} = \begin{cases} 1, & \text{product } j \text{ needs to be processed on machine } i \ (i=1,...,m \text{ and } j=1,...,n), \\ 0, & \text{otherwise}. \end{cases}
\]

To transform the binary product-order matrix into a distance matrix, we use a distance function. Consider two products \( i \) and \( j \) with vectors \( P_i \) and \( P_j \) respectively, in which the order data can be found (rows \( i \) and \( j \) of the product-order matrix). The distance between both products will be small (large) when there is large (small) similarity in their order patterns.

Späth defines a distance function as follows: A function \( d: \mathbb{R}^m \times \mathbb{R}^n \rightarrow \mathbb{R} \) is a distance function if for given \( P_i, P_j, \text{ and } P_z \in \mathbb{R}^n \) the following propositions hold:

1. \( d_{ij} = d(P_i, P_j) = 0 \) if \( P_i = P_j \);
2. \( d_{ij} = d(P_i, P_j) \geq 0 \ \forall \ P_i, P_j \in \mathbb{R}^n \);
3. \( d_{ij} = d(P_i, P_j) = d(P_j, P_i) \);
4. \( d_{ij} = d(P_i, P_j) \leq d(P_i, P_z) + d(P_z, P_j) \ \forall P_i, P_j, \text{ and } P_z \in \mathbb{R}^n \).

There are different distance functions that satisfy these propositions. If \( p_{ik} \) is the \( k^{th} \) element of vector \( P_i \), then \( P_i^T := (p_{i1}, ..., p_{im}) \). The Minkowski function between products \( i \) and \( j \) is then defined as

\[
d_x(P_i, P_j) = \left[ \sum_{k=1}^n |p_{ik} - p_{jk}|^x \right]^{1/x}, \quad x \geq 1.
\]

By taking different values for \( x \) we create different distance functions:

* \( L_1 \)-metric: \( x = 1 \): \( d_1(P_i, P_j) = \sum_{k=1}^n |p_{ik} - p_{jk}| \)
* \( L_2 \)-metric: \( x = 2 \): \( d_2(P_i, P_j) = \left[ \sum_{k=1}^n (p_{ik} - p_{jk})^2 \right]^{1/2} \)
* Chebychev-metric: \( x \) increases without bound; sometimes it is called the \( L_\infty \)-metric: \( d_\infty(P_i, P_j) = \max_k |p_{ik} - p_{jk}| \).
In every situation where two order patterns do not completely match each other, the Cheby-
chev-metric results in the value one. Therefore, much information is lost. Each order gives a
contribution to the distance between two products that equals zero or one (and consequently
also its squared value). Therefore, if the $L_1$-metric is used, each difference in the order pattern
will have a diminishing effect on the distance as a consequence of the square root. This does not
seem a logical distance function for the situation that is considered here. So we choose the $L_1$-
metric, which increases the distance between two products by one if a difference in the order
pattern appears. This means: $d_{ij} \leq n$. The distance between product $i$ and $j$ ($d_{ij}$) and use of the $L_1$-
metric can also be described as follows:

$$
\begin{align*}
    d_{ij} &= \sum_{k=1}^{n} \Gamma(p_{ik}, p_{jk}), \\
    \Gamma(p_{ik}, p_{jk}) &= 0 \text{ if } p_{ik} = p_{jk} \\
    &= 1 \text{ if } p_{ik} \neq p_{jk}.
\end{align*}
$$

It can be shown that an $m \times n$ product-order matrix leads to an $m \times m$ distance matrix. The
number of orders is irrelevant for the size of the distance matrix.

Remark: The transformation of the machine-product matrix in production systems is based on a
different distance function. The reason will be explained later on.

The clustering problem can now be formulated as a 0-1 integer programming model.

**The $p$-median model**

The number of clusters ($p$) in a $p$-median cluster problem is determined on *a priori* information
(for example, the number of aisles or tote-boxes in a warehouse). On the basis of the distance
matrix $P$, different elements (medians) will be selected to which the rest of the elements will be
allocated, so that $p$ clusters are created. Afterwards the value of $p$ may be changed, and the
procedure may be repeated.

A cluster median is defined as the element $j$ that is representative for all elements in the cluster.
Because we want to create $p$ clusters there are also $p$ medians, with $1 \leq p \leq m$. We now define
the following variables:

- $x_{ij} = 1$, element $i$ is allocated to median $j$;
- $0$, otherwise.
- $y_j = 1$, element $j$ is a median;
- $0$, otherwise.

$I$ = set of elements that have to be clustered.
$J$ = set of elements that can act as median.
The p-median clustering problem can now be formulated as follows:

\[
\begin{align*}
\min \sum_{i \in I} \sum_{j \in J} d_{ij} x_{ij} \\
\text{s.t.} \\
\sum_{j \in J} x_{ij} = 1, \quad \forall i \in I \\
\sum_{i \in I} y_j = p \\
x_{ij} \leq y_j, \quad \forall i \in I \text{ and } j \in J \\
x_{ij}, y_j \in \{0, 1\}, \quad \forall i \in I \text{ and } j \in J.
\end{align*}
\]

For facility location problems and clustering problems with respect to warehouse and production layout we need to note that the matrix \( D \) in the latter case is based on a binary matrix, whereas in the former case it is based on a non-binary one. Furthermore, \( D \) in a warehouse or production layout problem is square, symmetric, and the elements vary in the range of \( \{0,\ldots,n\} \). This concept will be discussed in the following section. More literature about the p-median cluster model can be found in Rosenwein and Captivo.

**P-MEDIAN SOLUTION TECHNIQUES**

There are several methods for solving the p-median clustering problem. One such method is Lagrange-relaxation with respect to one of the restrictions of the p-median problem. A technique that uses this relaxation is described in Boffey & Karkazis. Another method is the so-called dual ascent-method of Galvão. A third method is the branch & bound method, which is used frequently in practice. It can be an effective solution technique for the determination of the "optimal" solution of (mixed or pure) integer problems. For an extensive description of branch & bound methods see Williams.

These exact solution methods cause serious problems in execution time when the size of the clustering problem is large. Therefore, heuristics are developed that try to approximate the optimal solution as good as possible, in a less complicated way. One of these heuristics is the vertex substitution method developed by Teitz & Bart. As we shall report on the improvements of this heuristic method, the original algorithm will first be described by estimating the generalised vertex median of a graph (with corrections of errors in the original paper of Teitz & Bart).

**The Original Vertex Substitution (OVS) method**

Let us consider a network of \( m \) nodes. For each pair of nodes \( i \) and \( j \) on the network, the minimum distance \( d_{ij} \) is found. The network can be conveniently represented by a graph \( G \). We number all points from 1 to \( m \). The vertex median of the graph \( G \) is defined in the following...
way. Let the distance matrix $D$ of $G$ be the $m \times m$ symmetric matrix of $d_{ij}$ between all pairs of $i$ and $j$. The vertex median $k$ will be that vertex for which the sum of the elements in the corresponding column of $D$ is minimised. Let $V$ be some subset containing exactly $p$ vertices of $G$. We define $D_p^p(V)$ as the $m \times p$ matrix formed by the columns of the distance matrix $D$ of which the column number match the vertex numbers in $V$. Finally, we define $I = \{1, \ldots, m\}$. The task is now to find a set $(V)$ that is in some sense "closest" to all vertices in the system.

The algorithm:

1. Select the initial subset $V_1$ that contains vertices $1, 2, \ldots, p$.

2. Initialise $t=1$.
   Create a set of destination vertices $A_j$ such that
   \[
   A_j = \{ i \mid d_{ij} \leq d_{k,i} \quad \forall k \in V_t, \}, \text{ for } \forall i \in I \text{ and } j \in V_t
   \]
   The associated total distance is given by
   \[
   d_i = \sum_{j \in V_t} \sum_{i \in A_j} d_{ij}
   \]

3. Select some vertex $b$, not in $V_t$.

4. For each vertex $j$ in $V_t$, substitute $b$.

* If $d_{ij}$ is not the $i^{th}$ row minimum of submatrix $D_p^p(V_t)$, then no change will result in the $i^{th}$ row contribution to the total distance.

* If $d_{ij}$ is the $i^{th}$ row minimum of $D_p^p(V_t)$, then its replacement by $d_{ib}$ can have several outcomes depending upon whether
   \[
   d_{ib} \leq d_{ij} \quad \text{(a)}
   \]
   or
   \[
   d_{ij} \leq d_{ib} < d_{is} \quad \text{(b)}
   \]
   or
   \[
   d_{ij} \leq d_{is} \leq d_{ib} \quad \text{(c)}
   \]

where $d_{is}$ is the distance from $i$ to $s$ ($i,s \in V_t$), for which
\[
   d_{ij} \leq d_{is} \leq d_{s',s}, \text{ for } s' \in V_t \text{ and } s' \neq j, s.
\]
In other words, $d_{is}$ is the second smallest $i^{th}$ row element in $D_p^p(V_t)$.

The contribution to $d_i$ is now:

In case (a): $\Delta_{ib} = d_{ib} - d_{ij} \leq 0$;
In case (b): $\Delta_{ib} = d_{ib} - d_{ij} \geq 0$;
In case (c): $\Delta_{ib} = d_{is} - d_{ij} \geq 0$.
Replacing \( b \) for \( j \) leads to a reduction in total distance only if \( \Delta_{bj} = \sum_{i=1}^{m} \Delta_{bij} < 0 \).

5. Find the vertex \( k \) in \( V_t \) such that
\[ \Delta_{bk} < 0 \quad \text{and} \quad \Delta_{bk} = \min_{j=1,2,\ldots,p} \Delta_{bj}. \] (3)

6. * If a vertex \( k \) satisfies conditions (3), replace \( b \) for \( k \) in the subset, increase \( t \) by one, and label the new subset \( V_t \). Calculate \( A_t \) and the corresponding \( d_t \).
* If no vertex meets conditions (3), proceed with the old subset \( V_t \).

7. Select another vertex contained in the complement of subset \( V_t \) through \( V_t \) and not previously tried, and repeat steps 4 through 6.

8. When all vertices in the complement of \( V_t \) have been tried, define the resulting subset \( V_t \) as \( V_t \). Repeat steps 2 through 7. This step completes one iteration or cycle. Initialise a new cycle by defining \( d_1 = d_t \) and \( t = 1 \).

9. When one complete cycle results in no reduction of \( d_1 \), terminate the procedure. \( V_t \) is now an estimate of the vertex \( p \)-median of \( G \).

Vertex weights require an adjustment in the definition of the distance matrix in the following way:

\[
R := \text{weighted distance matrix}, \quad R = \text{HD} = [h_i d_{ij}], \quad \text{with} \quad H := \text{m} \times \text{m diagonal matrix}, \quad \text{with the vertex weights on the diagonal.}
\]

In the algorithm the distance matrix \( D \) will be replaced by the weighted distance matrix \( R \). The algorithm remains unmodified. For example, vertex weights can be very useful if in a warehouse system several products are ordered by different amounts.

To reach a solution for the three facilities design clustering problems, the same algorithm can be applied. Only the interpretation has to be adjusted slightly. This will be illustrated by an example concerning a clustering problem in a warehouse system where several products have to be grouped.

**Example 1.** The Original Vertex Substitution method.

Consider the product-order matrix \( P \) and the corresponding distance matrix \( D \):
Suppose we want to cluster five products into three groups \((p=3)\). In this case the original vertex substitution method is executed as follows.

1. Initialise subset \(V_1\) by \{1,2,3\}.

2. Determine the \(A_j\) corresponding to each median:

\[
A_j = \{i \mid d_{ij} \leq d_{ik} \forall k \in V_1\} \text{ and } j \in \{1,2,3\}.
\]

Now the following holds:

\[
A_1 = \{1\}, \quad A_2 = \{2\}, \quad A_3 = \{3,4,5\}.
\]

The total distance is

\[
d_1 = \sum_{i \in V_1} \sum_{k \in A_j} d_{ij} = d_{11} + d_{22} + d_{33} + d_{43} + d_{53} = 0 + 0 + 0 + 2 + 2 = 4.
\]

Now select product 4 \((b=4)\), and calculate the contributions to the distance as follows.

Replace product 1 \((j=1)\):

- \(i=1: d_{11}\) is the 1\(^{st}\) row minimum of \(D(V_1)\) (in this case the 5 \* 3 matrix consists of the first three columns of \(D\)). Hence: \(d_{11} \leq d_{12} \leq d_{14}\) (case (c)) \(\Rightarrow \Delta_{41} = 1\).

- \(i=2: d_{22}\) is not the 2\(^{nd}\) row minimum of \(D(V_1)\) \(\Rightarrow \Delta_{24} = 0\).

- \(i=3: d_{31}\) is not the 3\(^{rd}\) row minimum of \(D(V_1)\) \(\Rightarrow \Delta_{34} = 0\).

- \(i=4: d_{41}\) is not the 4\(^{th}\) row minimum of \(D(V_1)\) \(\Rightarrow \Delta_{44} = 0\).

- \(i=5: d_{51}\) is not the 5\(^{th}\) row minimum of \(D(V_1)\) \(\Rightarrow \Delta_{54} = 0\).

The above leads to...
Now replace product 2 \((j = 2)\) in a similar manner:
\[
\Delta_{42} = 0 + 1 + 0 + 0 + 0 = 1;
\]
Replacement of product 3 \((j = 3)\) gives:
\[
\Delta_{43} = 0 + 0 + 1 - 2 + 1 = 0.
\]
5. None of the replacements imply a reduction of the total distance (all contributions are positive or zero).

7. Now select product 5. Replacement leads to the following effects on the distance:
\[
\Delta_{51} = 1; \quad \Delta_{52} = 1; \quad \Delta_{53} = 0.
\]

Again no replacement reduces the distance. A complete cycle is finished, and the subset still contains products 1, 2 and 3. We conclude that the initial solution equals the final solution.

Group 1 := \{ 1 \}, median: product 1;
Group 2 := \{ 2 \}, median: product 2;
Group 3 := \{ 3, 4, 5 \}, median: product 3.

The corresponding distance equals four \((d_5 = 4)\).

**IMPROVEMENTS ON THE ORIGINAL VERTEX SUBSTITUTION METHOD**

**The Adjusted Vertex Substitution (AVS) method**

In this section we will show that in many cases the solution of the OVS method can be improved. To achieve this improvement, the heuristic needs some adjustments. Steps 1 through 3 remain unmodified. In step 4 the replacement effect of \( j \) for \( b \) on \( d_i \) is calculated. In the previous section we mentioned that only the \( i \)-th row contribution on the distance can be changed if \( d_{ij} \) is the \( i \)-th row minimum of \( D^p_i(V_r) \). Now we look again at the replacement effect of \( j \) (in \( V_r \)) for \( b \) (not in \( V_r \)) on the distance. Suppose \( d_{ij} \) is the \( i \)-th row minimum of \( D^p_i(V_r) \) and not \( d_{ik} \). If \( d_{ib} \) is smaller than \( d_{ik} \), replacing \( j \) for \( b \) does not change the \( i \)-th row contribution to the distance resulting from the original method. However, the distance does really decline with \( d_{ib} - d_{ik} \). To implement the above adjustment in the algorithm, we change step 4 as follows:

4. For each vertex \( j \) (in \( V_r \)) substitute \( b \) (not in \( V_r \)).

Consider \( d_{ir} \) to be the \( i \)-th row minimum of \( D^p_r(V_r) \), or \( d_{ir} \leq d_{ik} \) with \( k \in V_r \).
* If $d_i$ is not the $i^{th}$ row minimum of $D_p^n(V_i)$, we distinguish two possibilities:
  a) if $d_{ik} > d_i$ then no change appears in the $i^{th}$ row contribution.
  b) if $d_{ik} < d_i$ then $\Delta_{bj} = d_{bj} - d_i$.
* In case $d_i$ is the $i^{th}$ row minimum of $D_p^n(V_i)$, step 4 remains unchanged.

Calculation $\Delta_{bj} = \sum_{i=1}^n \Delta_{bj}$.

The rest of the algorithm remains the same. This means that a difference with the OVS method arises if for a certain $i$ the following holds:

$$d_i > d_k \land d_{ij} < d_{ji}, \text{ with } k \in V_i$$

(5)

To illustrate the above adjustment, we apply the AVS method to the same example solved earlier. Remember that the total distance of the original heuristic was four.

**Example 2. The Adjusted Vertex Substitution method.**

The distance matrix is:

$$D := \begin{bmatrix}
0 & 1 & 1 & 3 & 3 \\
1 & 0 & 2 & 4 & 4 \\
1 & 2 & 0 & 2 & 2 \\
3 & 4 & 2 & 0 & 4 \\
3 & 4 & 2 & 4 & 0
\end{bmatrix}$$

Once again we form $V_i$ given by $\{1, 2, 3\}$; the $A_i$'s remain unchanged; and $d_i = 4$.

3 & 4. Select product 4 ($b = 4$).

Replace product 1 ($j = 1$):

The minima of row 1 through 5 of $D_4^3(V_i)$ are: $d_{11}$, $d_{22}$, $d_{33}$, $d_{44}$ and $d_{55}$.

i=1: (5) does not hold: $d_{11}$ is the 1st row minimum of $D_4^3(V_i)$, $\rightarrow_1 \Delta_{41}$ remains 1.

i=2: (5) does not hold: $d_{21}$ is not the 2nd row minimum of $D_4^3(V_i)$, but $d_{24} \geq d_{22}$ $\rightarrow_2 \Delta_{41}$ remains 0.

i=3: (5) does not hold: $d_{31}$ is not the 3rd row minimum of $D_4^3(V_i)$, but $d_{34} \geq d_{33}$ $\rightarrow_3 \Delta_{41}$ remains 0.

i=4: (5) holds: $d_{41}$ is not the 4th row minimum of $D_4^3(V_i)$ and $d_{44} < d_{43}$ $\rightarrow_4 \Delta_{41} = 0 - 2 = -2$.

i=5: (5) does not hold: $d_{51}$ is not the 5th row minimum of $D_4^3(V_i)$, but $d_{54} \geq d_{53}$ $\rightarrow_5 \Delta_{41}$ remains 0.
The above leads to \( \Delta_{4i} = \sum_{j=1}^{5} \Delta_{4j} = -1 \) (instead of 1).

Then replace product 2 \((j=2)\) in a similar manner. This gives
\[ \Delta_{42} = 0 + 1 + 0 - 2 + 0 = -1. \]
Replacement of product 3 \((j = 3)\) gives:
\[ \Delta_{43} = 0 + 0 + 1 - 2 + 1 = 0. \]

5. Replacement of 1 by 4 in \( V_1 \) reduces the distance by 1.

6. Form \( V_2 = \{ 2, 3, 4 \} \) and \( d_2 = 3 \).

7. Select product 5.
Replacement of products 2, 3 and 4 gives respectively: \( \Delta_{52} = 0 \), \( \Delta_{53} = 0 \) and \( \Delta_{54} = 0 \). These figures are zero, so no changes are made.

A complete cycle is finished and \( V_1 \) has changed, so we start a new cycle with
\( V_1 = V_2 \) and \( d_1 = d_2 \). This new cycle produces no further improvement, thus the final result after 2 cycles is: \( V_1 = \{ 2, 3, 4 \} \) with \( A_2 = \{ 1, 2 \} \), \( A_3 = \{ 3, 5 \} \) and \( A_4 = \{ 4 \} \). The corresponding total distance is \( d_1 = 3 \).

Given the above example, we can see that the AVS method can find an improvement during the procedure, in contrast with the OVS method. This leads to actual replacement of a median. The OVS heuristic proceeds with the old subset. Therefore, not in all cases is the AVS method at least as good as the OVS heuristic. A negative difference may occur in exceptional situations, because the OVS method analyses other combinations of medians than the AVS method, so it may produce a (slightly) better solution. In all test cases (see last section) this situation occurred only once, with a small difference in distance.

**The Adjusted Vertex Substitution method with Multiple starting points (AVS-M)**

Example 2 showed that the adjustment in the algorithm indeed lead to the intended effect: the solution was improved. However, studying this example closer we see after some calculations that the solution is still a local optimum. Consider \( V_1 \) to be \( \{ 1, 4, 5 \} \), then the total distance equals two, which is the global optimum. Using multiple different initial subsets can be an adequate way to resolve this problem in many cases.

Obviously one of the major problems is the arbitrary choice of the initial subset. The basis is once again the AVS method: initialise \( V_1 = \{ 1, \ldots, p \} \) and apply the improvement heuristic. We consider the solution to be the best solution so far (BEST), and we will see if this solution can be improved by using another initial subset. In case \( p < m \), obviously the first \( p \) vertices will not
span \( \mathbb{R}^m \). Therefore, we use the following initial subset: \( V_1 = \{2, \ldots, p + 1\} \) in the second iteration of the algorithm. The AVS method is applied to this initial subset again. If the solution is better than BEST, then BEST will be modified. If not, BEST remains unchanged. This procedure is repeated \( k \) times, with \( p + k = m + 1 \) (\( n + 1 \) in case of facility location problems). This way \( \mathbb{R}^m \) is spanned by the initial subsets. This procedure consequently produces a solution, which is as good as or better than the solution of the AVS method.

**Example 3.** The Adjusted Vertex Substitution method with Multiple starting points.

Consider once more distance matrix \( D \) in equation (4). Example 2 shows that the AVS method with initial subset \( V_1 = \{1, 2, 3\} \) gives a solution: \( \text{BEST} = 3 \).

An iteration of the algorithm with \( V_1 = \{2, 3, 4\} \) does not lead to an improvement. Consequently, the initial subset is the final solution with \( d_1 = 3 \geq \text{BEST} \). This means that BEST will not be changed.

A final repetition of the algorithm with \( V_1 = \{3, 4, 5\} \) does lead to an improvement by replacing product 3 by product 1: \( \alpha_{31} = -1 \). The total distance reduces to two and BEST is modified.

Because the AVS method is repeated \( k \) times with \( p + k = m + 1 \) (or \( k = 3 \)), the procedure is now terminated and BEST represents the final distance. At last we have reached the global optimum of this example with \( V_1 = \{1, 4, 5\} \), with:

- \( A_1 = \{1, 2, 3\} \)
- \( A_4 = \{4\} \)
- \( A_5 = \{5\} \).

**Comparison of vertex substitution methods**

The examples showed that each adjustment of the algorithm lead to a better solution. Of course the question is: is the improvement structural or is distance matrix (4) an exception? Another important question is: what are the processing times and what level of improvement do they introduce in general? This section will answer these questions by comparing the three heuristics when running a large number of test cases generated by a Turbo-Pascal 7.0 program. The computer used for this purpose is an Olivetti P 82 modulo M4.

The generated distance matrices are based on a binary matrix for warehouse and production layout problems, and are based on non-binary matrices for facility location problems. These two different groups of matrices give different results. Two factors are of great importance for the performance of the heuristics. First, the number of vertices that must be clustered is relevant for the execution time: if this number increases, the execution time will also increase. Second, the number of clusters (\( p \)) is important for the level of the costs: if \( p \) increases, the corresponding costs will decrease.
When the distance matrices are based on binary matrices, the size of the binary matrix is varied between 5x5 and 70x70. For each test case we take three different values for p. Each size of the binary matrix will be generated five times, and the vertex substitution methods will be applied (120 cases are tested). After computing the means, the results are given in Table 1. The comparison is mainly based on the size of the matrix and on the value of p. The results are shown in the 3rd, 5th and 7th columns. First, the results of the OVS method for each situation (mean distance, mean execution time (in seconds)) are given, then the results are given for the AVS heuristic and the AVS-M method respectively.

The table shows that the AVS method yields a reduction between 2 and 4% for almost all cases with respect to the OVS heuristic. The AVS-M method improves the results of the AVS heuristic, with respect to this group of distance matrices, by about 1%

The execution time does seem to increase gradually when the value of p increases. The number of actions to complete the procedure reaches its maximum at p = m/2. This can be illustrated by an example in which 15 products have to be clustered. If p = 5 then in each cycle 10*5 = 50 substitutions will be checked. If p = 6 this number is 6*9 = 54, if p = 7 then it is 7*8 = 56 and if p = 8 then 8*7 = 56 replacements are examined. In general this means that the execution time will increase if the value of p increases up to m/2. If this value exceeds m/2, the execution time of the heuristics will decrease.

Table 1 shows that the required execution time for the AVS-M method in comparison with the other two heuristics does not increase significantly to make the method impractical. The largest execution time among all test cases is 28.3 seconds. The magnitude of the execution time obviously depends on the number of cycles that the algorithm needs to complete. More often the AVS method will give better results than the OVS method. Therefore, the termination of the process occurs in most cases after more cycles. This explains the (small) difference in the required execution times. For all test cases the OVS method reached its solution after at most four cycles, and the AVS method needed at most five cycles.
Table 1: *Comparison of the three vertex substitution methods using binary distance matrices*

<table>
<thead>
<tr>
<th>Matrix size</th>
<th>p</th>
<th>Results; Ex time</th>
<th>p</th>
<th>Results; Ex time</th>
<th>p</th>
<th>Results; Ex time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>OVS</td>
<td></td>
<td>OVS</td>
<td></td>
<td>OVS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AVS</td>
<td></td>
<td>AVS</td>
<td></td>
<td>AVS</td>
</tr>
<tr>
<td></td>
<td></td>
<td>AVS-M</td>
<td></td>
<td>AVS-M</td>
<td></td>
<td>AVS-M</td>
</tr>
<tr>
<td>5 * 5</td>
<td>2</td>
<td>4.6 ; 0.00</td>
<td>3</td>
<td>2.4 ; 0.00</td>
<td>4</td>
<td>1 ; 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.6 ; 0.00</td>
<td></td>
<td>2.2 ; 0.00</td>
<td></td>
<td>1 ; 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>4.6 ; 0.00</td>
<td></td>
<td>2.2 ; 0.00</td>
<td></td>
<td>1 ; 0.00</td>
</tr>
<tr>
<td>10 * 10</td>
<td>3</td>
<td>23.2 ; 0.00</td>
<td>5</td>
<td>14.2 ; 0.00</td>
<td>7</td>
<td>8.2 ; 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>22.0 ; 0.00</td>
<td></td>
<td>14.0 ; 0.00</td>
<td></td>
<td>7.4 ; 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>22.0 ; 0.03</td>
<td></td>
<td>13.8 ; 0.01</td>
<td></td>
<td>7.4 ; 0.01</td>
</tr>
<tr>
<td>20 * 20</td>
<td>5</td>
<td>98.0 ; 0.00</td>
<td>10</td>
<td>56.2 ; 0.00</td>
<td>15</td>
<td>27.2 ; 0.00</td>
</tr>
<tr>
<td></td>
<td></td>
<td>94.4 ; 0.00</td>
<td></td>
<td>54.4 ; 0.02</td>
<td></td>
<td>23.2 ; 0.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>94.0 ; 0.19</td>
<td></td>
<td>53.8 ; 0.14</td>
<td></td>
<td>23.2 ; 0.05</td>
</tr>
<tr>
<td>30 * 30</td>
<td>5</td>
<td>272.0 ; 0.01</td>
<td>10</td>
<td>197.0 ; 0.03</td>
<td>15</td>
<td>141.0 ; 0.03</td>
</tr>
<tr>
<td></td>
<td></td>
<td>262.4 ; 0.02</td>
<td></td>
<td>191.6 ; 0.05</td>
<td></td>
<td>134.8 ; 0.04</td>
</tr>
<tr>
<td></td>
<td></td>
<td>262.0 ; 0.76</td>
<td></td>
<td>190.2 ; 0.76</td>
<td></td>
<td>134.4 ; 0.64</td>
</tr>
<tr>
<td>40 * 40</td>
<td>5</td>
<td>540.6 ; 0.03</td>
<td>10</td>
<td>430.8 ; 0.05</td>
<td>15</td>
<td>341.0 ; 0.06</td>
</tr>
<tr>
<td></td>
<td></td>
<td>522.2 ; 0.04</td>
<td></td>
<td>411.8 ; 0.06</td>
<td></td>
<td>333.6 ; 0.09</td>
</tr>
<tr>
<td></td>
<td></td>
<td>520.4 ; 1.90</td>
<td></td>
<td>411.6 ; 2.58</td>
<td></td>
<td>331.6 ; 2.45</td>
</tr>
<tr>
<td>50 * 50</td>
<td>5</td>
<td>887.0 ; 0.06</td>
<td>10</td>
<td>743.8 ; 0.09</td>
<td>15</td>
<td>628.8 ; 0.12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>864.8 ; 0.08</td>
<td></td>
<td>720.8 ; 0.11</td>
<td></td>
<td>610.8 ; 0.13</td>
</tr>
<tr>
<td></td>
<td></td>
<td>864.6 ; 4.97</td>
<td></td>
<td>719.2 ; 5.40</td>
<td></td>
<td>609.4 ; 6.33</td>
</tr>
<tr>
<td>60 * 60</td>
<td>5</td>
<td>1377.8 ; 0.06</td>
<td>10</td>
<td>1165.6 ; 0.13</td>
<td>15</td>
<td>1016.6 ; 0.15</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1327.8 ; 0.14</td>
<td></td>
<td>1129.6 ; 0.22</td>
<td></td>
<td>983.0 ; 0.24</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1324.6 ; 7.43</td>
<td></td>
<td>1127.8 ; 11.11</td>
<td></td>
<td>981.0 ; 12.67</td>
</tr>
<tr>
<td>70 * 70</td>
<td>5</td>
<td>1904.6 ; 0.12</td>
<td>10</td>
<td>1655.2 ; 0.20</td>
<td>15</td>
<td>1472.2 ; 0.28</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1853.4 ; 0.15</td>
<td></td>
<td>1610.0 ; 0.29</td>
<td></td>
<td>1424.6 ; 0.43</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1848.2 ; 11.34</td>
<td></td>
<td>1606.2 ; 20.43</td>
<td></td>
<td>1422.8 ; 24.80</td>
</tr>
</tbody>
</table>

With regard to the group of distance matrices based on non-binary matrices (facility location problems), we observed a similar behaviour of the heuristics as shown in Table 2. The structure of Table 2 is like Table 1. The elements are now randomly generated as follows: \( d_{ij} = \text{round}(\text{random}(2^m)) \). This enlarged range shows that the improved heuristic results in much better solution (larger reductions in the distance) compared with the original heuristic. The differences between the OVS and the AVS method vary between 0 and 30 %. The difference between the AVS and the AVS-M method can increase to 15 %.
Table 2: Comparison of the three vertex substitution methods using non-binary based distance matrices

<table>
<thead>
<tr>
<th>Matrix size</th>
<th>p</th>
<th>Results; Ex.time</th>
<th>p</th>
<th>Results; Ex.time</th>
<th>p</th>
<th>Results; Ex.time</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td><strong>OVS</strong></td>
<td></td>
<td><strong>AVS</strong></td>
<td></td>
<td><strong>AVS-M</strong></td>
</tr>
<tr>
<td>10 * 5</td>
<td></td>
<td>45.6; 0.00</td>
<td></td>
<td>45.2; 0.00</td>
<td></td>
<td>44.6; 0.00</td>
</tr>
<tr>
<td>20 * 15</td>
<td></td>
<td>126.6; 0.00</td>
<td></td>
<td>114.8; 0.00</td>
<td></td>
<td>111.6; 0.10</td>
</tr>
<tr>
<td>30 * 25</td>
<td></td>
<td>191.0; 0.03</td>
<td></td>
<td>153.4; 0.03</td>
<td></td>
<td>151.2; 0.46</td>
</tr>
<tr>
<td>40 * 35</td>
<td></td>
<td>371.8; 0.04</td>
<td></td>
<td>264.4; 0.04</td>
<td></td>
<td>263.2; 1.43</td>
</tr>
<tr>
<td>50 * 45</td>
<td></td>
<td>548.6; 0.04</td>
<td></td>
<td>436.6; 0.07</td>
<td></td>
<td>421.8; 2.89</td>
</tr>
<tr>
<td>60 * 55</td>
<td></td>
<td>931.8; 0.07</td>
<td></td>
<td>677.0; 0.10</td>
<td></td>
<td>648.4; 5.90</td>
</tr>
<tr>
<td>70 * 65</td>
<td></td>
<td>1217.0; 0.11</td>
<td></td>
<td>952.6; 0.19</td>
<td></td>
<td>902.4; 10.16</td>
</tr>
<tr>
<td>70 * 70</td>
<td></td>
<td>1411.0; 0.10</td>
<td></td>
<td>1007.4; 0.16</td>
<td></td>
<td>961.2; 12.70</td>
</tr>
</tbody>
</table>

Comparison of the heuristic and the optimal solution

In the previous section we noticed that each adjustment in the algorithm introduced a significant improvement in the solution. However, the remaining question is: what is the quality of the solution in comparison with the optimal solution? We try to find out whether the heuristics can provide a good alternative to the exact solution techniques, which are in many cases complex and expensive. In the comparison only the AVS-M method will be considered because it is the superior heuristic.

Following the previous section, several random distance matrices are generated. We considered 10 * 10 matrices and gradually enlarged the size. For each size we created five different distance
matrices. The optimal solution is found by using an optimisation program called OMP. OMP uses a branch & bound-technique to reach the exact solution. It is common knowledge that this method often encounters problems with (mixed) integer problems. The extent of these problems together with the performance of the heuristic in comparison with the optimal solution can be found in Table 3. The table shows that the quality of the heuristic is exceptionally good. In only one test case the solution was suboptimal. The difference was 0.3%. Furthermore, the required execution time of OMP increases enormously, in contrast with the required execution time of the heuristic. In case of the two last matrices of size 50 * 50 (indicated by ★ in the table), OMP did not reach the optimal solution, even after 17 hours! The best solution then found was not even close to the solution of the heuristic, which was reached in ±5.5 seconds. Enlarging the matrix size further gave even more insurmountable problems with the execution time.

Table 3: Comparison of the AVS-M method with the optimal solution

<table>
<thead>
<tr>
<th>Size</th>
<th>AVS-M</th>
<th>OMP</th>
</tr>
</thead>
<tbody>
<tr>
<td>10 * 10</td>
<td>20 ; 0.0 s.</td>
<td>11 ; 0.0 s.</td>
</tr>
<tr>
<td></td>
<td>20 ; 2 s.</td>
<td>11 ; 1 s.</td>
</tr>
<tr>
<td>20 * 20</td>
<td>65 ; 0.1 s.</td>
<td>73 ; 0.1 s.</td>
</tr>
<tr>
<td></td>
<td>65 ; 4 s.</td>
<td>73 ; 4 s.</td>
</tr>
<tr>
<td>30 * 30</td>
<td>130 ; 10 s.</td>
<td>146 ; 1.2 s.</td>
</tr>
<tr>
<td></td>
<td>130 ; 26 s.</td>
<td>146 ; 39 s.</td>
</tr>
<tr>
<td>40 * 40</td>
<td>226 ; 2.1 s.</td>
<td>303 ; 2.7 s.</td>
</tr>
<tr>
<td></td>
<td>226 ; 16 m.</td>
<td>303 ; 1.1 h.</td>
</tr>
<tr>
<td>50 * 50</td>
<td>427 ; 5.3 s.</td>
<td>457 ; 5.5 s.</td>
</tr>
<tr>
<td></td>
<td>427 ; 4.7 h.</td>
<td>455 ; 5.2 h.</td>
</tr>
</tbody>
</table>

It should be clear that the quality of the heuristic depends mainly on the size of the set of possible medians (J). We learned that comparison with OMP with respect to large matrices is impossible. Therefore, we reduced the value of m, and generated random 15 * 80 distance matrices, for which OMP can find a solution in short time. For 16 of such matrices the result of the heuristic is compared with the optimal solution. To check whether the required execution time depends on the range of the elements, we divided these matrices into four groups. In each group the elements have different ranges: group 1: 0-15, group 2: 0-160, group 3: 0-400 and group 4: 0-1000. The results are shown in Table 4.
Table 4: Comparison with regard to 15 * 80 matrices

<table>
<thead>
<tr>
<th>$d_i \in {0,15}$</th>
<th>AVS-M</th>
<th>OMP</th>
<th>$d_i \in {0,160}$</th>
<th>AVS-M</th>
<th>OMP</th>
<th>$d_i \in {0,400}$</th>
<th>AVS-M</th>
<th>OMP</th>
<th>$d_i \in {0,1000}$</th>
<th>AVS-M</th>
<th>OMP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2 ; 2.7 s.</td>
<td>4 ; 2.6 s.</td>
<td>3 ; 2.5 s.</td>
<td>6 ; 2.6 s.</td>
<td>112 ; 3.8 s.</td>
<td>97 ; 2.8 s.</td>
<td>58 ; 2.9 s.</td>
<td>100 ; 2.9 s.</td>
<td>244 ; 2.5 s.</td>
<td>275 ; 3.2 s.</td>
<td>238 ; 3.4 s.</td>
</tr>
<tr>
<td></td>
<td>2 ; 5 s.</td>
<td>4 ; 11 s.</td>
<td>3 ; 7 s.</td>
<td>6 ; 9 s.</td>
<td>112 ; 8 s.</td>
<td>97 ; 7 s.</td>
<td>58 ; 7 s.</td>
<td>100 ; 9 s.</td>
<td>244 ; 8 s.</td>
<td>267 ; 9 s.</td>
<td>238 ; 6 s.</td>
</tr>
</tbody>
</table>

The different ranges of the elements of the distance matrix do not seem to influence the required processing time significantly. In only one case the AVS-M heuristic produced a suboptimal solution. Table 4 shows that for set J, containing 80 elements, the heuristic also operates well.

APPLICATION OF THE AVS-M METHOD TO GROUP TECHNOLOGY (GT)

In the previous section the heuristics were tested on random matrices to answer the question: can the AVS-M method provide a good alternative to a complex exact solution technique? In this section we will examine whether the confirming answer will also apply to matrices found in the literature. These cases are mostly based on GT-problems in production systems. Table 5 shows a set of examples with their characteristics.

Table 5: The set of problems

<table>
<thead>
<tr>
<th>Number</th>
<th>Problem</th>
<th>Machines (m)</th>
<th>Parts (n)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.</td>
<td>Chan &amp; Milner 9</td>
<td>15</td>
<td>10</td>
</tr>
<tr>
<td>2.</td>
<td>De Witte 10</td>
<td>12</td>
<td>19</td>
</tr>
<tr>
<td>3.</td>
<td>Chandrasekharan &amp; Rajagopalan 11</td>
<td>8</td>
<td>20</td>
</tr>
<tr>
<td>4.</td>
<td>Burbidge 12</td>
<td>16</td>
<td>43</td>
</tr>
<tr>
<td>5.</td>
<td>Carrie 13</td>
<td>20</td>
<td>35</td>
</tr>
<tr>
<td>6.</td>
<td>Chandrasekharan &amp; Rajagopalan 14</td>
<td>8</td>
<td>20</td>
</tr>
<tr>
<td>7.</td>
<td>Kumar &amp; Vanelli 15</td>
<td>30</td>
<td>45</td>
</tr>
<tr>
<td>8.</td>
<td>Askin 16</td>
<td>14</td>
<td>24</td>
</tr>
<tr>
<td>9.</td>
<td>Viswanathan 17</td>
<td>10</td>
<td>12</td>
</tr>
</tbody>
</table>

In these cases the matrices are binary machine-product matrices. Later on we shall see that distance function (1) can be improved in case of GT-problems. Therefore, at first we consider
these problems to be product-order matrices. For each problem we vary the value of \( p \) five times. This implies that 40 comparisons are made. For all cases the AVS-M method found the optimal solution.

Originally the problems in Table 5 belong to the Group Technology problems. In practice this important group of clustering problems has often to be solved, to assign products to machine cells (yet to be formed) in such a way that within a machine cell as few as possible machines are idle (voids) and as few as possible products need to be processed outside a machine cell (exceptions). This can be visualised as follows: rearrange the rows of the machine-product matrix in such a manner that the rows corresponding to machines within one cell succeed each other. Then rearrange the columns in such a way that the columns corresponding to those products that are allocated to one cell, succeed each other too. In this way so-called diagonal blocks are developed. The number of voids equals the number of zeros within the blocks, and the number of exceptions equals the number of ones outside the blocks.

On the basis of this diagonal assignment the quality of different heuristics can be compared with each other. In general this is done with the help of grouping efficiency. An increasing coefficient implies a better quality of the clustering. Kerr & Balakrishnan applied their Spreadsheet Cell Formation Algorithm (SCFA) to the first eight problems in Table 5. They used the following grouping efficiency measure:

\[
e_1 = \frac{n_1}{\sum_{i=1}^{p} P_i Q_i} - 1 + \frac{n_1}{n_1 + n_0}, \quad -1 \leq e_1 \leq 1
\]

where \( n_1 \) denotes the number of ones within the diagonal blocks, and \( n_0 \) is the number of ones outside the diagonal blocks (exceptions). \( P_i \) and \( Q_i \) represent the number of products and the number of machines in cell \( i \) respectively. This grouping efficiency measure clearly emphasises maximising the number of ones within the blocks. Minimising the number of voids is of secondary importance. The term \( \sum_{i=1}^{p} P_i Q_i \) implies that in most cases an equal partitioning of machines and products in cells leads to a higher grouping efficiency.

If minimising the number of voids also has priority, then other grouping efficiency measures are available. Chandrasekharan & Rajagopalan use grouping efficiency measure:

\[
e_2 = q \left[ \frac{n_1}{\sum_{i=1}^{p} P_i Q_i} \right] + (1-q) \left[ 1 - \frac{n_0}{mn - \sum_{i=1}^{p} P_i Q_i} \right], \quad -1 \leq e_2 \leq 1
\]
where \( q \) is the weighting factor that enables the analyst to alter the emphasis between intercell movement. Even if \( q = 0.5 \), the number of voids is more important than if measure (6) is used.

To reach a higher grouping efficiency, distance function (1) is no longer used. We now want to minimise the number of voids and exceptions. Therefore, each element of the distance matrix is set to \( 2^n \) and like Viswanathan\(^7\) the following distance function is used:

\[
d_{ij} = \sum_{k=1}^{n} \Gamma (p_{ik}, p_{jk}), \quad \text{with} \quad \Gamma (p_{ik}, p_{jk}) = \begin{cases} -2 & \text{if } p_{ik} = p_{jk} = 1; \\ +1 & \text{if } p_{ik} \neq p_{jk}; \\ 0 & \text{otherwise.} \end{cases} \tag{8}
\]

If \( p_{ik} = p_{jk} = 1 \) then product \( k \) is processed on machines \( i \) and \( j \). If they are placed in one machine cell, this means that two ones appear in a diagonal block. If \( p_{ik} \neq p_{jk} \) and machine \( i \) and \( j \) are placed together in one cell, then a void or an exception occurs.

On the basis of such a distance matrix the AVS-M method will cluster the machines. Comparing this heuristic with the SCFA procedure requires the use of grouping efficiency measure (6). Therefore we assign the products to the machine cells as follows. Given the number of operations in each cell for each product, the product will be assigned to the cell in which this number is the largest. If a choice between two cells occurs, assign the product to the cell with the smallest number of machines in it. If grouping efficiency measure (7) is used, we proceed with checking for each product whether the sum of voids and exceptions can be reduced by assigning it to another cell. This situation can occur if the reassignment leads to a larger reduction of the number of voids than the increase in the number of exceptions.

The difference of both assignment methods can be illustrated well by the example of Viswanathan\(^7\). In this situation we consider ten machines and twelve products. Of course the heuristic gives the same machine cells in both cases (which is the same cell partition as in Viswanathan\(^7\)). If the products are assigned to the machine cells on the basis of (6), we want to minimize \( n_0 \). The procedure described above gives \( \sum_{i=1}^{3} P_i Q_i = 51, n_1 = 34 \) and \( n_0 = 7 \). This leads to \( e_1 = 0.496 \) and \( e_2 = 0.783 \). If products are allocated to the machine cells on the basis of grouping efficiency measure (7), we are also interested in the number of voids. The process described above gives \( \sum_{i=1}^{3} P_i Q_i = 49, n_1 = 33 \) and \( n_0 = 8 \). This leads to \( e_1 = 0.492 \) and \( e_2 = 0.788 \). This shows that the assignment of products to the machine cells depends on the characteristics of the grouping efficiency measure.

The whole process is applied to the problems in Table 5. The results are shown in appendix 1, with the grouping efficiencies of the SCFA displayed in brackets. From this table we quickly see that in all cases the AVS-M method gives a grouping efficiency measure that is at least as big as
CONCLUSIONS

For facilities design we distinguished three different categories of clustering problems: (1) facility location problems, (2) clustering problems concerning warehouse, and (3) concerning production layout. These problems can be formulated as p-median clustering problems. Exact solution techniques are available, but in many cases the required execution time of these methods may increase dramatically. Therefore, the use of heuristics is necessary in these cases. One of those heuristics is the vertex substitution method developed by Teitz & Bart. We proposed an adjustment that improves the quality of the heuristic substantially. If this whole procedure is repeated with another initial subset each time, the quality may increase once again.

From the comparison with the optimal solution we conclude that in particular the adjusted vertex substitution method with multiple starting points gives exceptionally good results. In addition the processing time is polynomial. Another advantage of vertex substitution methods is that they are applicable to different problems, such as storage assignment in warehousing, assignment of locations in a distribution network or in constructing production cells. On the contrary, many specific heuristics for GT can only be used in specific situations.

Our investigation into the quality of the AVS method and the AVS-M method has proved that the latter heuristic is exceptionally good; in a large number of situations it turned out to be optimal.
REFERENCES


### Appendix 1: Output with Respect to 9 GT Problems

<table>
<thead>
<tr>
<th>Problem</th>
<th>Mach.</th>
<th>Number of clusters</th>
<th>Gr. Eff. AVS-M (SCFA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Parts</td>
<td>1,7,10, 2,7,10,11,12</td>
<td>1.7,10, 2,5,8, 3,5,8,13,15</td>
<td>e_i = 0.92 (e_c=0.92)</td>
</tr>
<tr>
<td>Parts</td>
<td>2,3,4,5, 18,19</td>
<td>2,3,4,5,6, 3,4,5,6,8,9</td>
<td>e_i = 0.43 (e_c=0.36)</td>
</tr>
<tr>
<td>Parts</td>
<td>12,13,14,15, 16,17,18,19</td>
<td>12,13,14, 9,10,11, 10,11,12</td>
<td>e_i = 0.46 (e_c=0.45)</td>
</tr>
<tr>
<td>Parts</td>
<td>1,5,10,12,15</td>
<td>1,5,6,7, 18,20, 2,4,7,8</td>
<td>e_i = 0.85 (e_c=0.85)</td>
</tr>
<tr>
<td>Parts</td>
<td>1,2,3,4,5, 25, 2,4,7,10,18,26, 25,29</td>
<td>1,2,3,4,5, 3,20,22,24, 2,8,9,11,13, 14,16,17,19</td>
<td>e_i = 0.75 (e_c=0.75)</td>
</tr>
<tr>
<td>Parts</td>
<td>1,2,3,4,9,10, 19,20, 1,3,7,21, 4,6,18,19, 5,16,17,27, 2,10,11,12, 20,23,31, 32,33,39, 40,41</td>
<td>1,2,3,4,9,10, 12,13,16,17, 19,20, 8,14,16,19, 14,15,16, 19,21,23, 29,33,41,43</td>
<td>e_i = 0.48 (e_c=0.48)</td>
</tr>
<tr>
<td>Parts</td>
<td>7,8,9,14,15, 24,25,28,29,35</td>
<td>7,8,9,14,15, 2,48, 1,3,13,21, 22,30</td>
<td>e_i = 0.37 (*)</td>
</tr>
<tr>
<td>Mach.</td>
<td>4,8,27,28</td>
<td>4,8,27,28, 9,19,20,29, 19,20,29, 30</td>
<td>1,3,13,21, 22,30, 4,6,18,19, 2,6,37,38</td>
</tr>
<tr>
<td>Problem</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
<tr>
<td>---------</td>
<td>-----</td>
<td>-----</td>
<td>-----</td>
</tr>
<tr>
<td>Parts 8</td>
<td>7,8,9,18</td>
<td>4,5,21,24</td>
<td>1,6,10,11,12,13,14,15,16,22</td>
</tr>
<tr>
<td>Mach. 8</td>
<td>1,12,13</td>
<td>2,3,10,11</td>
<td>6,8,9,14</td>
</tr>
<tr>
<td>Parts 9</td>
<td>1,6,10,11,12,13,14,15,16,22</td>
<td>2,3,4,5,7,8,9,17,18,19,20,21,23,24</td>
<td>1,2,3,4,5,7,10,11,12,13</td>
</tr>
<tr>
<td>Mach. 9</td>
<td>6,8,9,14</td>
<td>1,2,3,4,5,7,10,11,12,13</td>
<td>4,5,9</td>
</tr>
</tbody>
</table>

* Due to the lack of data the grouping efficiency of the SCFA could not be calculated.
<table>
<thead>
<tr>
<th>No.</th>
<th>Author(s)</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>9661</td>
<td>U. Gneezy and J. Potters</td>
<td>An Experiment on Risk Taking and Evaluation Periods</td>
</tr>
<tr>
<td>9662</td>
<td>H.J. Bierens</td>
<td>Nonparametric Nonlinear Co-Trending Analysis, with an Application to Interest and Inflation in the U.S.</td>
</tr>
<tr>
<td>9663</td>
<td>J.P.C. Blanc</td>
<td>Optimization of Periodic Polling Systems with Non-Preemptive, Time-Limited Service</td>
</tr>
<tr>
<td>9664</td>
<td>M.J. Lee</td>
<td>A Root-N Consistent Semiparametric Estimator for Fixed Effect Binary Response Panel Data</td>
</tr>
<tr>
<td>9665</td>
<td>C. Fernández, J. Osiewalski and M.F.J. Steel</td>
<td>Robust Bayesian Inference on Scale Parameters</td>
</tr>
<tr>
<td>9666</td>
<td>X. Han and H. Webers</td>
<td>A Comment on Shaked and Sutton’s Model of Vertical Product Differentiation</td>
</tr>
<tr>
<td>9667</td>
<td>R. Kollmann</td>
<td>The Exchange Rate in a Dynamic-Optimizing Current Account Model with Nominal Rigidities: A Quantitative Investigation</td>
</tr>
<tr>
<td>9668</td>
<td>R.C.H. Cheng and J.P.C. Kleijnen</td>
<td>Improved Design of Queueing Simulation Experiments with Highly Heteroscedastic Responses</td>
</tr>
<tr>
<td>9669</td>
<td>E. van Heck and P.M.A. Ribbers</td>
<td>Economic Effects of Electronic Markets</td>
</tr>
<tr>
<td>9670</td>
<td>F.Y. Kumah</td>
<td>The Effect of Monetary Policy on Exchange Rates: How to Solve the Puzzles</td>
</tr>
<tr>
<td>9671</td>
<td>J. Jansen</td>
<td>On the First Entrance Time Distribution of the M/D/∞ Queue: a Combinatorial Approach</td>
</tr>
<tr>
<td>9672</td>
<td>Y.H. Farzin, K.J.M. Huisman and P.M. Kort</td>
<td>Optimal Timing of Technology Adoption</td>
</tr>
<tr>
<td>9673</td>
<td>J.R. Magnus and F.J.G.M. Klaassen</td>
<td>Testing Some Common Tennis Hypotheses: Four Years at Wimbledon</td>
</tr>
<tr>
<td>9674</td>
<td>J. Fidrmuc</td>
<td>Political Sustainability of Economic Reforms: Dynamics and Analysis of Regional Economic Factors</td>
</tr>
<tr>
<td>9675</td>
<td>M. Das and A. van Soest</td>
<td>A Panel Data Model for Subjective Information on Household Income Growth</td>
</tr>
<tr>
<td>9677</td>
<td>B. van Aarle and S.-E. Hougaard Jensen</td>
<td>Output Stabilization in EMU: Is There a Case for an EFTS?</td>
</tr>
<tr>
<td>9678</td>
<td>Th.E. Nijman, F.A. de Roon and C. Veld</td>
<td>Pricing Term Structure Risk in Futures Markets</td>
</tr>
<tr>
<td>No.</td>
<td>Author(s)</td>
<td>Title</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>9679</td>
<td>M. Dufwenberg and U. Gneezy</td>
<td>Efficiency, Reciprocity, and Expectations in an Experimental Game</td>
</tr>
<tr>
<td>9680</td>
<td>P. Bolton and E.-L. von Thadden</td>
<td>Blocks, Liquidity, and Corporate Control</td>
</tr>
<tr>
<td>9681</td>
<td>T. ten Raa and P. Mohnen</td>
<td>The Location of Comparative Advantages on the Basis of Fundamentals only</td>
</tr>
<tr>
<td>9682</td>
<td>S. Hochguertel and van Soest</td>
<td>The Relation between Financial and Housing Wealth of Dutch A. Households</td>
</tr>
<tr>
<td>9684</td>
<td>F.Y. Kumah</td>
<td>Common Stochastic Trends in the Current Account</td>
</tr>
<tr>
<td>9685</td>
<td>U. Gneezy and M. Das</td>
<td>Experimental Investigation of Perceived Risk in Finite Random Walk Processes</td>
</tr>
<tr>
<td>9686</td>
<td>B. von Stengel, A. van den Elzen and D. Talman</td>
<td>Tracing Equilibria in Extensive Games by Complementary Pivoting</td>
</tr>
<tr>
<td>9687</td>
<td>S. Tijs and M. Koster</td>
<td>General Aggregation of Demand and Cost Sharing Methods</td>
</tr>
<tr>
<td>9688</td>
<td>S.C.W. Eijffinger, H.P. Huizinga and J.J.G. Lemmen</td>
<td>Short-Term and Long-Term Government Debt and Nonresident Interest Withholding Taxes</td>
</tr>
<tr>
<td>9689</td>
<td>T. ten Raa and E.N. Wolff</td>
<td>Outsourcing of Services and the Productivity Recovery in U.S. Manufacturing in the 1980s</td>
</tr>
<tr>
<td>9690</td>
<td>J. Suijs</td>
<td>A Nucleolus for Stochastic Cooperative Games</td>
</tr>
<tr>
<td>9691</td>
<td>C. Seidl and S. Traub</td>
<td>Rational Choice and the Relevance of Irrelevant Alternatives</td>
</tr>
<tr>
<td>9692</td>
<td>C. Seidl and S. Traub</td>
<td>Testing Decision Rules for Multiattribute Decision Making</td>
</tr>
<tr>
<td>9693</td>
<td>R.M.W.J. Beetsma and H. Jensen</td>
<td>Inflation Targets and Contracts with Uncertain Central Banker Preferences</td>
</tr>
<tr>
<td>9694</td>
<td>M. Voorneveld</td>
<td>Equilibria and Approximate Equilibria in Infinite Potential Games</td>
</tr>
<tr>
<td>9695</td>
<td>F.B.S.L.P. Janssen and A.G. de Kok</td>
<td>A Two-Supplier Inventory Model</td>
</tr>
<tr>
<td>9696</td>
<td>L. Ljungqvist and H. Uhlig</td>
<td>Catching up with the Keynesians</td>
</tr>
<tr>
<td>9697</td>
<td>A. Rustichini</td>
<td>Dynamic Programming Solution of Incentive Constrained Problems</td>
</tr>
<tr>
<td>No.</td>
<td>Author(s)</td>
<td>Title</td>
</tr>
<tr>
<td>------</td>
<td>----------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>9698</td>
<td>G. Gürkan and A. Y. Özge</td>
<td>Sample-Path Optimization of Buffer Allocations in a Tandem Queue - Part I: Theoretical Issues</td>
</tr>
<tr>
<td>9699</td>
<td>H. Huizinga</td>
<td>The Dual Role of Money and Optimal Financial Taxes</td>
</tr>
<tr>
<td>96100</td>
<td>H. Huizinga</td>
<td>The Taxation Implicit in Two-Tiered Exchange Rate Systems</td>
</tr>
<tr>
<td>96101</td>
<td>H. Norde, F. Patrone and S. Tijs</td>
<td>Characterizing Properties of Approximate Solutions for Optimization Problems</td>
</tr>
<tr>
<td>96102</td>
<td>M. Berg, A. De Waegenaere and J. Wielhouwer</td>
<td>Optimal Tax Reduction by Depreciation: A Stochastic Model</td>
</tr>
<tr>
<td>96103</td>
<td>G. van der Laan, D. Talman and Z. Yang</td>
<td>Existence and Approximation of Robust Stationary Points on Polytopes</td>
</tr>
<tr>
<td>96104</td>
<td>H. Huizinga and S.B. Nielsen</td>
<td>The Coordination of Capital Income and Profit Taxation with Cross-Ownership of Firms</td>
</tr>
<tr>
<td>96105</td>
<td>H. Degryse</td>
<td>The Total Cost of Trading Belgian Shares: Brussels Versus London</td>
</tr>
<tr>
<td>96106</td>
<td>H. Huizinga and S.B. Nielsen</td>
<td>The Political Economy of Capital Income and Profit Taxation in a Small Open Economy</td>
</tr>
<tr>
<td>96107</td>
<td>T. Dieckmann</td>
<td>The Evolution of Conventions with Endogenous Interactions</td>
</tr>
<tr>
<td>96108</td>
<td>F. de Jong and M.W.M. Donders</td>
<td>Intraday Lead-Lag Relationships Between the Futures-, Options and Stock Market</td>
</tr>
<tr>
<td>96109</td>
<td>F. Verboven</td>
<td>Brand Rivalry, Market Segmentation, and the Pricing of Optional Engine Power on Automobiles</td>
</tr>
<tr>
<td>96110</td>
<td>D. Granot, H. Hamers and S. Tijs</td>
<td>Weakly Cyclic Graphs and Delivery Games</td>
</tr>
<tr>
<td>96112</td>
<td>A. De Waegenaere, R. Kast and A. Lapied</td>
<td>Non-linear Asset Valuation on Markets with Frictions</td>
</tr>
<tr>
<td>96113</td>
<td>R. van den Brink and P.H.M. Ruys</td>
<td>The Internal Organization of the Firm and its External Environment</td>
</tr>
<tr>
<td>96114</td>
<td>F. Palomino</td>
<td>Conflicting Trading Objectives and Market Efficiency</td>
</tr>
<tr>
<td>96115</td>
<td>E. van Damme and S. Hurkens</td>
<td>Endogenous Stackelberg Leadership</td>
</tr>
<tr>
<td>96116</td>
<td>E. Canton</td>
<td>Business Cycles in a Two-Sector Model of Endogenous Growth</td>
</tr>
<tr>
<td>9701</td>
<td>J.P.J.F. Scheepens</td>
<td>Collusion and Hierarchy in Banking</td>
</tr>
<tr>
<td>No.</td>
<td>Author(s)</td>
<td>Title</td>
</tr>
<tr>
<td>-----</td>
<td>-----------</td>
<td>-------</td>
</tr>
<tr>
<td>9702</td>
<td>H.G. Bloemen and E.G.F. Stancanelli</td>
<td>Individual Wealth, Reservation Wages and Transitions into Employment</td>
</tr>
<tr>
<td>9703</td>
<td>P.J.J. Herings and V.J. Vannetelbosch</td>
<td>Refinements of Rationalizability for Normal-Form Games</td>
</tr>
<tr>
<td>9705</td>
<td>C. Fernández and M.F.J. Steel</td>
<td>On the Dangers of Modelling Through Continuous Distributions: A Bayesian Perspective</td>
</tr>
<tr>
<td>9706</td>
<td>M.A. Odijk, P.J. Zwaneveld, J.S. Hooghiemstra, L.G. Kroon and M. Salomon</td>
<td>Decision Support Systems Help Railned to Search for 'Win-Win' Solutions in Railway Network Design</td>
</tr>
<tr>
<td>9707</td>
<td>G. Bekker, R.J. Hodrick and D.A. Marshall</td>
<td>The Implications of First-Order Risk Aversion for Asset Market Risk Premiums</td>
</tr>
<tr>
<td>9708</td>
<td>C. Fernández and M.F.J. Steel</td>
<td>Multivariate Student-(t) Regression Models: Pitfalls and Inference</td>
</tr>
<tr>
<td>9709</td>
<td>H. Huizinga and S.B. Nielsen</td>
<td>Privatization, Public Investment, and Capital Income Taxation</td>
</tr>
<tr>
<td>9710</td>
<td>S. Eijffinger, E. Schaling and M. Hoeberichts</td>
<td>Central Bank Independence: a Sensitivity Analysis</td>
</tr>
<tr>
<td>9711</td>
<td>H. Uhlig</td>
<td>Capital Income Taxation and the Sustainability of Permanent Primary Deficits</td>
</tr>
<tr>
<td>9712</td>
<td>M. Dufwenberg and W. Güth</td>
<td>Indirect Evolution Versus Strategic Delegation: A Comparison of Two Approaches to Explaining Economic Institutions</td>
</tr>
<tr>
<td>9713</td>
<td>H. Uhlig</td>
<td>Long Term Debt and the Political Support for a Monetary Union</td>
</tr>
<tr>
<td>9714</td>
<td>E. Charlier, B. Melenberg and A. van Soest</td>
<td>An Analysis of Housing Expenditure Using Semiparametric Models and Panel Data</td>
</tr>
<tr>
<td>9715</td>
<td>E. Charlier, B. Melenberg and A. van Soest</td>
<td>An Analysis of Housing Expenditure Using Semiparametric Cross-Section Models</td>
</tr>
<tr>
<td>9716</td>
<td>J.P. Choi and S.-S. Yi</td>
<td>Vertical Foreclosure with the Choice of Input Specifications</td>
</tr>
<tr>
<td>9717</td>
<td>J.P. Choi</td>
<td>Patent Litigation as an Information Transmission Mechanism</td>
</tr>
<tr>
<td>9718</td>
<td>H. Degryse and A. Irmen</td>
<td>Attribute Dependence and the Provision of Quality</td>
</tr>
<tr>
<td>9719</td>
<td>A. Possajennikov</td>
<td>An Analysis of a Simple Reinforcing Dynamics: Learning to Play an &quot;Egalitarian&quot; Equilibrium</td>
</tr>
<tr>
<td>9720</td>
<td>J. Jansen</td>
<td>Regulating Complementary Input Supply: Cost Correlation and Limited Liability</td>
</tr>
<tr>
<td>9721</td>
<td>J. ter Horst and M. Verbeek</td>
<td>Estimating Short-Run Persistence in Mutual Fund Performance</td>
</tr>
<tr>
<td>No.</td>
<td>Author(s)</td>
<td>Title</td>
</tr>
<tr>
<td>-----</td>
<td>-----------</td>
<td>-------</td>
</tr>
<tr>
<td>9722</td>
<td>G. Bekaert and S.F. Gray</td>
<td>Target Zones and Exchange Rates: An Empirical Investigation</td>
</tr>
<tr>
<td>9723</td>
<td>M. Slikker and A. van den Nouweland</td>
<td>A One-Stage Model of Link Formation and Payoff Division</td>
</tr>
<tr>
<td>9724</td>
<td>T. ten Raa</td>
<td>Club Efficiency and Lindahl Equilibrium</td>
</tr>
<tr>
<td>9725</td>
<td>R. Euwals, B. Melenberg and A. van Soest</td>
<td>Testing the Predictive Value of Subjective Labour Supply Data</td>
</tr>
<tr>
<td>9726</td>
<td>C. Fershtman and U. Gneezy</td>
<td>Strategic Delegation: An Experiment</td>
</tr>
<tr>
<td>9727</td>
<td>J. Potters, R. Sloof and F. van Winden</td>
<td>Campaign Expenditures, Contributions and Direct Endorsements: The Strategic Use of Information and Money to Influence Voter Behavior</td>
</tr>
<tr>
<td>9728</td>
<td>F.H. Page, Jr.</td>
<td>Existence of Optimal Auctions in General Environments</td>
</tr>
<tr>
<td>9731</td>
<td>A. Ridder, E. van der Laan and M. Salomon</td>
<td>How Larger Demand Variability may Lead to Lower Costs in the Newsvendor Problem</td>
</tr>
<tr>
<td>9732</td>
<td>K. Kultti</td>
<td>A Model of Random Matching and Price Formation</td>
</tr>
<tr>
<td>9733</td>
<td>J. Ashayeri, R. Heuts and B. Tammel</td>
<td>Applications of P-Median Techniques to Facilities Design Problems: an Improved Heuristic</td>
</tr>
</tbody>
</table>