A class of strong valid inequalities for the discrete lot-sizing and scheduling problem

Citation for published version (APA):

Document status and date:
Published: 01/01/1993

Publisher’s PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:
• A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
• The final author version and the galley proof are versions of the publication after peer review.
• The final published version features the final layout of the paper including the volume, issue and page numbers.

Link to publication

General rights
Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.
• Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
• You may not further distribute the material or use it for any profit-making activity or commercial gain
• You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the “Taverne” license above, please follow below link for the End User Agreement:
www.tue.nl/taverne

Take down policy
If you believe that this document breaches copyright please contact us at:
openaccess@tue.nl
providing details and we will investigate your claim.

Download date: 06. Apr. 2021
A class of strong valid inequalities for the discrete lot-sizing and scheduling problem

Stan van Hoesel
Antoon Kolen

Eindhoven, March 1993
The Netherlands
A class of strong valid inequalities for the discrete lot-sizing and scheduling problem

Stan van Hoesel* Antoon Kolen†

March 3, 1993

Abstract

A class of strong valid inequalities for the discrete lot-sizing and scheduling problem is presented. Necessary and sufficient conditions are derived for valid inequalities in this class to be facet-defining. The separation problem for these inequalities is shown to be solvable in polynomial time by dynamic programming.

Key words: Discrete Lot-sizing and Scheduling, Strong Valid Inequalities, Dynamic Programming, Separation.

AMS Subject classification: 90B.

*Address: Eindhoven University of Technology, P.O. Box 513, 5600 MB Eindhoven, The Netherlands. E-mail: HOESEL@BS.WIN.TUE.NL
†Address: University of Limburg, P.O. Box 616, 6200 MD Maastricht, The Netherlands. E-mail: A.KOLEN@KE.RULIMBURG.NL
1 Introduction

Production planning decisions in industry are made on two distinct levels: a strategic (long-term) level and an operational (short-term) level. On the strategic level planning systems are used to develop a rough production plan for the coming years, whereas on the operational level detailed production decisions are specified, typically for some months. On both levels a planner is faced with the same types of costs, the inventory holding costs and the production costs. The production costs consist of a fixed component, typical for lot-sizing, and a production-size dependent component. The most suitable class of models for strategic planning are economic lot-sizing models. These are capable of handling problems with relatively long periods, in which a large production capacity must be divided among several goods. For operational planning one usually takes a discrete lot-sizing and scheduling model, where periods are so short that only a single item can be produced in a fixed amount. Such production processes arise naturally in bulk processing industries like chemical plants.

The history of the Economic Lot-Sizing Problem (ELSP) goes back to the late fifties, when the two seminal papers of Wagner and Whitin [19] and Manne [11] were published. In [19] a dynamic programming algorithm was developed to solve the single-item uncapacitated version of the economic lot-sizing problem. In [11] Manne suggests a linear programming approach for the multi-item capacitated version of the problem. Dynamic programming and linear programming are the two basic techniques that have been used for solving these and other versions of the economic lot-sizing problem. The polyhedral structure of the single-item economic lot-sizing problem has been investigated by Barany, Van Roy and Wolsey [1], [2]. Their valid inequalities have been implemented successfully in a cutting-plane algorithm for the multi-item problem. Krarup and Bilde [9] provide a polynomial-size complete linear description for the single-item economic lot-sizing problem by splitting the production variables. See also Eppen and Martin [3] for a detailed description of the technique of variable splitting. Pochet [12] and Pochet and Wolsey [13], [14] describe valid inequalities for ELSP with backlogging and with production capacities in each period. Van Hoesel et al. [8] describe the convex hull of solutions for the problem with two types of fixed costs (set-up costs and start-up costs).

Research on the Discrete Lot-sizing and Scheduling Problem (DLSP) has started only recently by Schrage [18]. Schrage typified production processes with a so-called all-or-nothing policy as discrete lot-sizing and scheduling problems, and he distinguished between two different kinds of fixed production costs: set-up costs (typical for economic lot-sizing) and start-up or change-over costs (typical for discrete lot-sizing). A straightforward dynamic programming recursion solves the single-item DLSP. Fleischmann [4] proposes a branch and bound algorithm by use of Lagrangian relaxation of the capacity constraints of the problem for the multi-item version. In [5] he reformulates this problem as a travelling salesman problem with time windows, and he proposes a lower bounding procedure based on Lagrangian relaxation of the travelling salesman model. The complexity of DLSP and some variants is discussed by Salomon [15]. A linear reformulation by use of variable splitting is described in van Hoesel and Kolen [7]. Valid inequalities for the single-item discrete lot-sizing and scheduling problem will be developed in this manuscript. Magnanti and Vachani [10] and Sastry [17] describe facet-defining inequalities for the more general problem, in which set-up costs are included.

Section 2 contains the formulation for the single-item discrete lot-sizing problem and some basic facts on the convex hull of solutions for this problem. In section 3 we describe a class of
strong valid inequalities, the so-called hole-bucket inequalities. We will derive necessary and sufficient conditions under which these are facet-defining. In section 4 we describe a dynamic programming based separation algorithm for the hole-bucket inequalities. Finally, in section 5, other classes of valid inequalities are mentioned. Moreover, the relation with polyhedral results for related discrete lot-sizing problems is discussed.

2 Preliminaries

2.1 Integer programming formulation for DLSP

Consider the single-item version of DLSP, i.e., we have one item that must be produced. The planning horizon consists of \( T \) periods, and in each period \( t \in \{1, \ldots, T\} \) a demand of \( d_t \) units of the item occurs. This demand must be satisfied by production in one or more of the periods up to \( t \). Since an all-or-nothing production policy is assumed in each period, i.e., production is either at capacity (constant over all periods) or zero, the production speed can be normalized to one unit per period. This implies that the demands can be restricted to be binary, also (see Fleischmann [4]). A maximal set of consecutive periods in which production takes place is called a production batch. Such a batch must begin with a period in which a start-up takes place. In the generic formulation of the problem as suggested by Fleischmann [4] the inventory at the end of each period is denoted by a third type of variables. However, this type of variables can be eliminated to create the formulation below.

The following parameters and variables are used to describe the single-item DLSP. They are defined for each period \( t \in \{1, \ldots, T\} \).

Parameters:
\( d_t \): the demand of the item in period \( t \);
\( f_t \): the start-up cost in period \( t \);
\( c_t \): the unit production cost of the item in period \( t \);

Variables:
\( x_t \): the production of the item in period \( t \);
\( y_t \): \( \begin{cases} 1 & \text{if a start-up is incurred in period } t \\ 0 & \text{otherwise.} \end{cases} \)

\[
\text{(DLSP)} \quad \min \sum_{t=1}^{T} (f_t y_t + c_t x_t) \quad (1)
\]

s.t.
\[
\sum_{\tau=1}^{t} x_{\tau} \geq d_{1,t} \quad (1 \leq t \leq T) \quad (2)
\]
\[
x_1 \leq y_1 \quad (3)
\]
\[
x_t \leq x_{t-1} + y_t \quad (2 \leq t \leq T) \quad (4)
\]
where \( d_{1,t} \) is the cumulative demand of the periods \( 1, \ldots, t \). We assume that \( z_0 \) is equal to zero.

The constraints 2 ensure that the ending inventory of all periods is nonnegative. Note that a positive ending inventory for period \( T \) is allowed. The starting inventory of the first period is assumed to be zero. The constraints 3 and 4 force a start-up when production of the item takes place in period \( t \) but not in the preceding period \( t - 1 \). Constraints 5 force the variables to be binary. These constraints are relaxed in the linear programming relaxation of DLSP. The problem above is, in the notation introduced by Salomon et al. [16], denoted by 1/1/SI/G/A, i.e., there is one machine, one item, Sequence Independent start-up costs (set-up costs in their terminology), time-dependent (General) production and inventory costs, and start-up times are Absent.

### 2.2 Dimension of the convex hull of DLSP

For a given instance of DLSP, it is not straightforward to determine the dimension of the convex hull, since it depends on the demand function \( d \). However, it will be shown that a few simplifying assumptions suffice to get rid of the demand dependency. Moreover, these assumptions ensure a full-dimensional polyhedron.

Suppose that we are given a \( T \)-period instance with demand function \( d \), i.e., \( d_t \ (1 \leq t \leq T) \) denotes the demand in period \( t \). There are \( 2T \) variables and there is no explicit equation. Thus, the dimension is at most \( 2T \). The actual dimension depends on the demands as is shown by the following example. If \( d_1 = 1 \), then \( y_1 = x_1 = 1 \) must hold. In general, if \( d_1 = d_2 = \ldots = d_t = 1 \), then \( y_1 = x_1 = x_2 = \ldots = x_t = 1 \), which yields \( t + 1 \) equations.

In the remainder of this section, we eliminate this demand dependency by assuming that \( d_1 = 0 \). For any particular instance, this can be achieved by ignoring the periods before the first period without demand, and putting the start-up costs in the first non-demand period to zero. A useful assumption already in the model is that overproduction is allowed, i.e., ending inventory of period \( T \) can be positive. One advantage of this assumption is that the demand dependency that arises when \( d_T = 0 \) is eliminated (in case the total production is forced to equal the total demand). Under the two given assumptions the dimension of the convex hull of DLSP is not dependent on the demands anymore, as follows from the following theorem.

**Theorem 1** The convex hull of a \( T \)-period instance of DLSP has dimension \( 2T \), under the assumption that \( d_1 = 0 \).

Before we prove this theorem, the idea behind the proof is clarified. The dimension \( d(P) \) of a polyhedron \( P \) is the dimension of the smallest affine subspace in which it can be contained. Usually this dimension is determined by exhibiting \( d(P) + 1 \) affinely independent feasible solutions in the affine subspace. However, one can also exhibit \( d(P) \) linearly independent vectors which are the component-wise difference of two feasible solutions in the affine subspace. These difference-vectors are called directions. This is the way we will prove the theorem. Thus, we exhibit \( 2T \) linearly independent directions. Moreover, this technique will
also be applied to the facet proofs, where we will exhibit \(2T - 1\) directions, each consisting of the difference of two feasible solutions which satisfy the valid inequality as equality. This technique makes the facet proofs more transparent in general. This is true especially in case of the constructive descriptions of the inequalities in section 3.

**Proof.** The proof falls apart in two cases. One case where the direction \(y_t = 1\) (all variables not mentioned have value zero) is established, and one where the direction \(x_t = 1\) is established. In both cases feasible solutions are used which have a large part in common. This part consists of the values of the variables of periods other than period \(t\). For each \(\tau \neq t\), we fix \(y_\tau = x_\tau = 1\).

**Case 1:** direction \(y_t = 1\) \((1 \leq t \leq T)\). In the first solution we take \(y_t = 1; x_t = 0\), and in the second solution we take \(y_t = 0; x_t = 0\).

Note that this results in feasible solutions. Since \(d_1 = 0\), and period \(t\) is the only period without production, all demands can be fulfilled.

**Case 2:** direction \(x_t = 1\) \((1 \leq t \leq T)\). In the first solution we take \(y_t = 1; x_t = 1\), and in the second solution we take \(y_t = 1; x_t = 0\).

Since we have exhibited \(2T\) linear independent directions, this ends the proof.

\[\square\]

### 2.3 Facet-defining inequalities in the model

In this subsection the facet-defining inequalities that are part of the model are evaluated, under the assumptions made in the preceding subsection. Only for the first type of inequalities we will prove this. In the other cases similar reasoning can be applied. In the facet-proofs we will repeatedly make use of feasible solutions that satisfy the inequality as equality. Such solutions will be called *optimal solutions* with respect to that inequality.

#### Production inequalities

These are the inequalities that ensure that there is enough production, i.e., the inequalities \(\sum_{\tau=1}^{t} x_\tau \geq d_{1,t} (1 \leq t \leq T)\).

**Lemma 2** *The inequalities \(\sum_{\tau=1}^{t} x_\tau \geq d_{1,t} (1 \leq t \leq T)\) are facet-defining if and only if either \(d_t = 1\) and \(d_{t+1} = 0\) \((t < T)\) or \(d_t = 1\) \((t = T)\).*

**Proof.** If \(d_t = 0\), then \(\sum_{\tau=1}^{t-1} x_\tau \geq d_{1,t-1}\) and \(x_t \geq 0\) imply \(\sum_{\tau=1}^{t} x_\tau \geq d_{1,t} (1 \leq t \leq T)\). If \(t < T\) and \(d_{t+1} = 1\), then a similar implication holds. Thus, the conditions are certainly necessary.

Sufficiency can be concluded from the following. As a basic solution, we take \(y_\tau = 1\) \((1 \leq \tau \leq T)\), and \(x_\tau = d_\tau\) \((1 \leq \tau \leq t)\), \(x_\tau = 1\) \((t + 1 \leq \tau \leq T)\). For each period \(\tau\) after \(t\), the directions \(y_\tau = 1\) and \(x_\tau = 1\) are easily created. This leaves the directions for the periods \(\{1, \ldots, t\}\). In the following table only the variables that have values different from the basic solution are mentioned.
\[
\begin{array}{|c|c|c|}
\hline
\tau \in \{1, \ldots, t\} & \text{Direction} & \text{First solution} & \text{Second solution} \\
\hline
d_\tau = 0 (\tau \neq 1): & y_\tau = 1 & x_\tau = 1, x_t = 0 & y_\tau = 0 \\
d_\tau = 0: & x_\tau = 1, x_t = -1 & x_\tau = 1, x_t = 0 \\
d_\tau = 1: & y_\tau = 1 & x_\tau = 0, x_1 = 1 & y_\tau = x_\tau = 0, x_1 = 1 \\
d_\tau = 1: & x_\tau = 1, x_1 = -1 & x_\tau = 0, x_1 = 1 & \\
\hline
\end{array}
\]

Thus we have \(2(T-t) + 2t - 1 = 2T - 1\) linear independent directions. This suffices to show that the inequality is a facet-defining one.

\[\square\]

**Start-up inequalities**

These are the inequalities \(x_1 \leq y_1\) and \(x_t \leq x_{t-1} + y_t\) \((2 \leq t \leq T)\).

(a) The inequality \(y_1 \geq x_1\) is facet defining.

(b) The inequalities \(x_{t-1} + y_t \geq x_t\) \((t \geq 2)\) define facets, unless \(d_2 = \ldots = d_t = 1\).

**Variable bounding inequalities**

These are the inequalities that bound the variables in the model:

\[
x_t \geq 0; \quad y_t \geq 0; \quad x_t \leq 1; \quad y_t \leq 1. \quad (1 \leq t \leq T)
\]

(a) The inequalities \(x_t \geq 0\) \((1 \leq t \leq T)\) define facets, except if \((t = 1\) and \(d_2 = 1\)), or if \((t > 1\) and \(d_2 = \ldots = d_t = 1\)).

(b) The inequalities \(x_t \leq 1\) \((1 \leq t \leq T)\) define facets, except if \(t = 1\).

(c) The inequalities \(y_t \geq 0\) \((1 \leq t \leq T)\) define facets, except if \(t = 1\), or if \((t > 1\) and \(d_2 = \ldots = d_t = 1\)).

(d) The inequalities \(y_t \leq 1\) \((1 \leq t \leq T)\) define facets for all \(t\).

The facet-proofs for the start-up inequalities and the variable bounding inequalities are similar to the proof for the production inequalities.

### 3 The hole-bucket inequalities

#### 3.1 Inequalities with 0/1-coefficients

We will consider inequalities with coefficients 0 and 1 for the production and start-up variables, i.e., these inequalities have the following structure.

\[
\sum_{t \in X} x_t + \sum_{t \in Y} y_t \geq \alpha(X,Y)
\]  
(6)
where $X$ and $Y$ are subsets of $\{1, \ldots, T\}$ and $\alpha(X,Y)$ is a positive integer dependent on $X$ and $Y$. The only exceptions known with respect to this structure are the variable upper bounding inequalities, and the start-up inequalities. We start with some properties of these inequalities that simplify the exposition.

(P1) $1 \notin Y$. 
If $1 \in Y$, then $y_1 = x_1$ for all feasible solutions satisfying $6$ as equality.

(P2) $Y \neq \emptyset$. 
If $Y = \emptyset$, then the inequality defines a facet if and only if it is a production inequality.

(P3) $T \in X \cup Y$. 
If $T \notin X \cup Y$, then one can restrict the question whether the inequality is facet-defining to a smaller planning horizon. See the following two lemmas.

Lemma 3 Consider an inequality of type 6, satisfying P1 and P2, such that $t < T$ is the last period in $X \cup Y$.

(i) This inequality is valid if and only if it is valid with respect to the problem restricted to the planning horizon $\{1, \ldots, t\}$.

(ii) This inequality is facet-defining if and only if it is facet-defining with respect to the problem restricted to the planning horizon $\{1, \ldots, t\}$.

Proof. If the inequality is valid with respect to $\{1, \ldots, t\}$, then it is certainly valid with respect to $\{1, \ldots, T\}$: any feasible solution of the problem on the full planning horizon cut to the restricted planning horizon must be feasible for the restricted problem. Thus, certainly, such a solution satisfies the valid inequality. If it is not valid with respect to $\{1, \ldots, t\}$, then take a feasible solution which violates the inequality. Extending this solution with $x_r = y_r = 1$ for $T > t$, creates a feasible solution with respect to $\{1, \ldots, T\}$ that also violates the inequality. This ends the proof of the first part of the lemma.

Suppose that the inequality is facet-defining with respect to $\{1, \ldots, t\}$. Then there are $2t - 1$ linear independent directions using the variables $x_r, y_r$ ($\tau \leq t$) only. Moreover, there is a feasible solution satisfying the inequality as equality with $\sum_{\tau=1}^{t} x_\tau > d_{1,t}$. Otherwise, all optimal feasible solutions would satisfy $\sum_{\tau=1}^{t} x_\tau = d_{1,t}$, implying that the inequality is not a facet, since $Y \neq \emptyset$. Take this solution and extend it with $x_\tau = y_\tau = 1$ ($t < \tau \leq T$). Using this solution as a basis the directions $x_\tau = 1$ and $y_\tau = 1$ ($t < \tau \leq T$) can be created. For the direction $x_\tau = 1$, the basis is taken as a first solution, and the second solution is like the first but with $x_\tau = 0$. The latter solution is taken as a first solution to create the direction $y_\tau = 1$. The second solution has also $y_\tau = 0$. Thus, $2(T - t)$ new linear independent directions have been created, and therefore the inequality is also facet-defining with respect to $\{1, \ldots, T\}$. Finally, suppose that the inequality is not facet-defining with respect to $\{1, \ldots, t\}$. Then there is a set of valid inequalities that dominate the given one. However, these inequalities are also valid with respect to $\{1, \ldots, T\}$, which has been proved in the first part of this lemma. Thus, the original inequality is also dominated with respect to the planning horizon $\{1, \ldots, T\}$ by these inequalities which implies that it is not a facet with respect to this planning horizon.
The following lemma shows that the choices for restricted planning horizons can be limited to periods \( \{1, \ldots , t\} \), where \( t \) is a demand period.

**Lemma 4** Consider a valid inequality of type 6, such that \( t < T \) is the last period in \( X \cup Y \). If this inequality is facet-defining, then \( d_t = 1 \).

**Proof.** Suppose that \( d_t = 0 \). If \( t \in X \), then any feasible solution with \( x_t = 1 \) can be improved by setting \( x_t = 0 \) and \( y_t = 1 \) for \( t = t + 1, \ldots , T \). This does not affect the feasibility of the solution, since \( \sum_{r=1}^{t} x_r \geq \sum_{r=1}^{t-1} x_r \geq d_{t-1} = d_t \). Therefore, all feasible solutions that satisfy 6 as equality satisfy \( x_t = 0 \) also. If \( t \in Y \), then the proof proceeds analogously.

From the previous two lemmas it follows that facet-defining inequalities satisfy the property P3 for some restricted planning horizon \( \{1, \ldots , t\} (t \leq T) \). Therefore, P3 is assumed to hold in the following.

### 3.2 Hole-bucket inequalities

In the following we consider inequalities of type 6 with the additional requirement that \( X \) and \( Y \) have no periods in common. The main goal is to derive necessary and sufficient conditions under which these inequalities define facets. The left-hand side of the facet-defining inequalities can be partitioned into so-called *holes* and *buckets* due to the following lemma.

**Lemma 5** A facet-defining inequality of type 6 with \( X \cap Y = \emptyset \) satisfies the following condition. If \( t \in Y \), then \( t - 1 \in X \) or \( t - 1 \in Y \).

**Proof.** Suppose that there is a period \( t \) with \( t \in Y \) and \( t - 1 \notin X \cup Y \). Each feasible solution \((x, y)\) with \( y_t = 1 \) can be improved by putting \( y_t = 0 \) and \( y_{t-1} = x_{t-1} = 1 \), which implies that each optimal solution satisfies \( y_t = 0 \), and thus the inequality is not a facet-defining one.

By the previous lemma the planning horizon can be partitioned in batches of periods \( B_k = \{t_k, \ldots , u_k\} (1 \leq k \leq K) \), such that \( t_k \in X \) and \( B_k \{t_k\} \subseteq Y \) and a set of periods \( H = \{h_l|1 \leq l \leq L\} = \{1, \ldots , T\} \setminus (X \cup Y) \). Thus, the variables corresponding to \( B_k \) in the left-hand side of the hole-bucket inequality are \( x_{t_k} + y_{t_k+1} + \ldots + y_{u_k} \). Such batches are called buckets, since the use of one or more of the periods in a bucket \( B_k \) for production contributes at least one unit to the value of the left-hand side of 6. On the other hand, in a period of \( H \), a so-called hole, production can take place without any contribution to the left-hand side of 6.

**EXAMPLES**

<table>
<thead>
<tr>
<th>( t )</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
<th>12</th>
</tr>
</thead>
<tbody>
<tr>
<td>( d_t )</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>
Clearly, the right-hand side is the minimum number of buckets that must be used for production of at least one demand.

3.3 Greedy algorithm

Given a set of holes and buckets, covering the demands of the periods \( \{1, \ldots, T\} \) by production in the buckets, i.e., the determination of the right-hand side \( \alpha(X, Y) \) in order to make the hole-bucket inequality a valid one, is done by use of the following greedy algorithm. It makes use of the observation that production in holes does not contribute to the value of the left-hand side of the hole-bucket inequality, and production in a bucket contributes at least one unit to the value of the left-hand side. The size of a bucket \( B \) is its number of periods. It is denoted by \( |B| \).

**Greedy algorithm**

The periods with demand are considered in an increasing order one by one. The production period for the demand from period \( t \) is chosen in a greedy way in the following order of preference.

(a) If there is an empty hole in \( \{1, \ldots, t\} \), then it is chosen to produce \( d_t \).

(b) If there is a bucket in \( \{1, \ldots, t\} \) that is only partially used for production, then the first free period in this bucket is chosen to produce \( d_t \).

(c) If all buckets in \( \{1, \ldots, t\} \) are either empty or full, then a largest empty available bucket is chosen to produce \( d_t \) in its first period. In this case the value of the left-hand side is incremented by one (a bucket is called available if it is empty and if its first period is in \( \{1, \ldots, t\} \); a bucket is called completely available if it is empty and if its last period is in \( \{1, \ldots, t\} \)).

**Theorem 6** The greedy algorithm determines the minimum number of buckets that must be used for production.

**Proof.** The proof is by induction on the number of periods. We use the following induction hypothesis. There is an optimal solution (the production periods of this solution are denoted by \( S_{\text{opt}}(t) \)) that produces in all periods chosen by the greedy algorithm to produce
the demands of the periods \( \{1, \ldots, t\} \). The set of production periods of the latter solution is denoted by \( S_g(t) \). Thus, the induction hypothesis is \( S_g(t) \subseteq S_{opt}(t) \). Clearly, the hypothesis holds for period 1. Suppose it holds for period \( t \).

If \( d_{t+1} = 0 \), then the hypothesis also holds for \( S_{opt}(t+1) = S_{opt}(t) \), since \( S_g(t+1) = S_g(t) \). Therefore, we suppose that \( d_{t+1} = 1 \).

If there is a hole in \( \{1, \ldots, t+1\} \) not in \( S_g(t) \), then this hole is chosen as a production period in \( S_g(t+1) \) by the greedy algorithm. If this hole is not in \( S_{opt}(t) \), then we can take any period \( \tau \) in \( S_{opt}(t) \) not used by \( S_g(t+1) \), and transfer its production to the hole. This ensures that \( S_g(t+1) \subseteq S_{opt}(t+1) \) and thus that the solution \( S_{opt}(t+1) \) is feasible with respect to the restricted planning horizon \( \{1, \ldots, t+1\} \).

If there is no hole available, but if there is a bucket partially used for production, then the greedy algorithm takes a period in this bucket for production. Like in the previous case, we can take any period in \( S_{opt}(t) \setminus S_g(t+1) \) and transfer its production to the period chosen by the greedy algorithm.

If there is no hole or partially used bucket available, then the greedy algorithm takes a largest empty available bucket, say \( B_k = \{t_k, \ldots, u_k\} \). Thus, \( t_k \leq t \). If \( S_{opt}(t) \) uses this bucket also, then the hypothesis holds for \( S_g(t+1) \). So, suppose that \( S_{opt}(t) \) does not contain any periods of \( B_k \). Since \( S_{opt}(t) \) uses all the buckets and holes of \( S_g(t) \), there must be some period \( \tau \) in \( \{1, \ldots, t+1\} \) that is in \( S_{opt}(t) \) but not in \( S_g(t) \). Since there are no holes without production in \( \{1, \ldots, t+1\} \setminus S_g(t) \), period \( \tau \) is in a bucket \( B_l \). The size of this bucket is at most \( |B_k| \), since the greedy algorithm takes the largest bucket. We create a new solution \( S_{opt}(t+1) \) from \( S_{opt}(t) \) by transferring the production from \( B_l \) to \( B_k \). This solution uses the same number of buckets as \( S_{opt}(t) \), and \( S_g(t+1) \subseteq S_{opt}(t+1) \). So, it remains to be shown that \( S_{opt}(t+1) \) is feasible, i.e., that \( |S_{opt}(t+1) \cap \{1, \ldots, u\}| \geq d_{1,u} \) for \( u = 1, \ldots, T \). By construction of \( S_g(t+1) \subseteq S_{opt}(t+1) \) this holds for \( u \leq t + 1 \). The production in \( B_k \) can be arranged such that the earliest periods in \( B_k \) produce. Suppose that \( s \) is the last production period in \( B_k \). Then \( |S_{opt}(t+1) \cap \{1, \ldots, u\}| = |S_{opt}(t) \cap \{1, \ldots, u\}| \geq d_{1,u} \) for all \( u > s \). Now, if \( |S_{opt}(t+1) \cap \{1, \ldots, u\}| < d_{1,u} \) for some \( u \in \{t+2, \ldots, s\} \), then this also holds for \( u - 1 \), since \( d_u \leq 1 \) and \( x_u = 1 \). Thus, it holds for all earlier periods in \( B_k \), including \( t+1 \), a contradiction. This proves that \( S_{opt}(t+1) \) is feasible, thereby ending the correctness proof of the greedy algorithm.

\( \Box \)

### 3.4 Necessary conditions

The greedy algorithm is an important tool in establishing necessary and sufficient conditions for the determination of facet-defining hole-bucket inequalities. The following conditions are necessary for a hole-bucket inequality to be facet-defining.

(N1) Suppose that the first hole \( h \) precedes the first bucket \( B = \{t, \ldots, u\} \) of size bigger than one. For each \( \tau \) with \( h \leq \tau < t \), the number of holes in \( \{h, \ldots, \tau\} \) exceeds the total demand \( d_{h,\tau} \).

(N2) Suppose that the greedy algorithm must determine a production period for \( d_t = 1 \).
(a) If there is a partially filled bucket \( B \) and no empty hole, then an empty bucket of size at least \(|B|\) must be available in \( \{1, \ldots, t\} \).

(b) If there are no partially filled buckets and no empty holes available and the last filled bucket is \( B \), then an empty bucket of size at least \(|B|\) must be completely available in \( \{1, \ldots, t\} \) besides \( B \).

Recall that a bucket is available if it is empty and if its first period is in \( \{1, \ldots, t\} \); a bucket is completely available if it is empty and if its last period is in \( \{1, \ldots, t\} \).

(N3) For \( d_t \) the greedy algorithm chooses a new empty (maximal) bucket.

(N4) If there are at least two buckets, then there should be enough holes before \( B_1 \) to cover the demands of the periods up till \( u_1 \), i.e., the demand pattern is such that it is possible to leave \( B_1 \) without production.

In the proof on the necessity of these conditions, variations in the left-hand side of a hole-bucket inequality are used to determine whether a hole-bucket inequality is dominated by another inequality. A solution is called optimal with respect to a valid inequality if it satisfies the inequality as equality. We will use the fact that if a set of optimal solutions to a certain inequality is a subset of the set of optimal solutions of another inequality, then the latter inequality dominates the first, which indicates that the first can not be facet-defining.

In the following cases a hole-bucket inequality is not facet-defining. If the variable \( x_t \) of a hole \( t \) is added to the left-hand side of a hole-bucket inequality \( ax + \beta y \geq \gamma \), and \( \gamma \) increases, then all optimal solutions to \( ax + \beta y \geq \gamma \) have production in \( t \), i.e., they satisfy \( x_t = 1 \). If the variable \( y_u \), where \( u \) is the last period of a bucket is deleted from \( ax + \beta y \geq \gamma \), and \( \gamma \) remains equal, then all optimal solutions to \( ax + \beta y \geq \gamma \) satisfy \( y_u = 0 \).

Proof. The necessity of each of the conditions N1 - N4 is proved separately.

Suppose that for some period N1 does not hold. Then take the minimal period \( \tau \), i.e., \( d_{h,\tau} \) is the number of holes in \( \{h, \ldots, \tau\} \). By the minimality of \( \tau \), there are holes available for all demand periods in \( \{h, \ldots, \tau\} \). Filling the holes is cheapest, since there are only buckets of size one besides the holes. However, this implies that all the holes are filled by any optimal solution, and thus the inequality is not a facet.

Suppose that for \( d_t \) there is a partially filled bucket \( B \) available, but there are no empty buckets with size at least \(|B|\) and no empty hole available in \( \{1, \ldots, t\} \), i.e., N2a is violated. Let \( s \) be the first period for which production is chosen to take place in \( B \). Suppose that the last period of \( B \) is turned into a hole. If \( s \) precedes the hole, then \( B \) may be chosen, since it is still maximal. Otherwise, \( d_s \) is produced in that hole. However, for the succeeding demand period (which is period \( t \) or an earlier period) \( B \) is still a maximal available bucket and may therefore be chosen by the greedy algorithm. Thus, in all cases the original greedy solution remains optimal, implying that the inequality is not a facet and that N2a must be satisfied.

Suppose that for \( d_t \) a new bucket must be chosen by the greedy algorithm and no bucket with size at least \(|B|\) is completely available in \( \{1, \ldots, t\} \). If no maximal bucket is available at all, then N2a is violated for the periods that have their demand produced in the last filled maximal bucket. Thus, we may assume that there is only one empty maximal bucket \( B \), which is partially in \( \{1, \ldots, t\} \). Thus, turning the last period of \( B \) into a hole leaves \( B \)
the largest empty available bucket for \( t \), and therefore it can still be chosen by the greedy algorithm. Thus, the original greedy solution remains optimal, implying that the inequality is not a facet and that N2b must be satisfied.

Suppose that there is a hole \( h \) available for \( d_T \). Let \( \tau \) be maximal such that in the set of periods \( \{1, \ldots, \tau\} \) no demand is placed in a hole. Then, for all periods \( \{\tau+1, \ldots, T\} \) there are holes available in \( \{\tau+1, \ldots, T\} \). Thus, any optimal solution will use these holes. Therefore, if \( T \in X \), then \( x_T = 0 \) for any optimal solution, and if \( T \in Y \), then \( y_T = 0 \) for any optimal solution. Next, suppose that there is a partially filled bucket available for \( d_T \). Then turning the last period of this bucket into a hole has no influence on the optimal value of the left-hand side of the inequality. We may conclude that an empty bucket must be chosen by the greedy algorithm for \( d_T \).

Finally, if N4 does not hold, then the first bucket \( B_1 \) must be filled. Thus, then all optimal solutions satisfy \( x_t + \ldots + y_u \geq 1 \) as equality, and since the inequality contains more than one bucket, it can not be a facet.

3.5 Sufficient conditions

Conditions N1 - N4 are also sufficient. This is shown by constructing, for a given hole-bucket inequality, optimal solutions with properties that allow the construction of \( 2T - 1 \) linearly independent directions. These optimal solutions are characterized in S1-S3. Consider a hole-bucket inequality satisfying N1 - N4.

(S1) For each bucket \( B = \{t, \ldots, u\} \) there is an optimal solution with period \( u \) being the only production period in \( B \), i.e., \( y_u = x_u = 1, y_t = \ldots = y_{u-1} = x_t = \ldots = x_{u-1} = 0 \).

(S2) There is a bucket \( A \) such that

(a) For any bucket \( B \) earlier than \( A \) there is an optimal solution such that \( B \) is empty and there is one production period in \( A \).

(b) For any bucket \( B \) later than \( A \), there is an earlier bucket \( C \), with respect to \( B \), such that \( C \) is empty and there is one production period in \( B \).

(S3) For each hole \( h \) there is an optimal solution that leaves \( h \) unused for production.

Proof. If the greedy algorithm fills the buckets \( \{C_1, \ldots, C_L\} \subseteq \{B_1, \ldots, B_K\} \) in this order, then this is the order of appearance of these buckets on the planning horizon. Note that from a set of empty maximal buckets the first is chosen.

It follows directly that S1 holds for all buckets left empty and for \( C_L \), by N3. Moreover, these solutions are such that \( C_1, \ldots, C_{L-1} \) are completely filled with production. Now suppose that S1 holds for \( C_{i+1}, \ldots, C_L \) and let \( S_{i+1} \) be the solution with production only in the last period of \( C_{i+1} \), such that only maximal buckets are used and such that \( C_1, \ldots, C_i \) are filled with production, i.e., the filling up to \( C_i \) is in accordance with the greedy algorithm. At the moment \( C_i \) is filled with its first unit of production it is completely available by N2b. Thus,
we take the last period of $C_i$ for this production. At the succeeding demand period another maximal bucket $C$ after $C_i$ is available by N2a. This may be $C_{i+1}$ or an earlier maximal bucket, but its size is at least $|C_i|$. Thus, this and all other demands can be transferred to $C$, leaving at least one period available for the production of the unit in the last period of $C_{i+1}$. The rest of the solution remains equal to $S_{i+1}$. By induction, $S_1$ has been proved now. Note that this solution, denoted by $S_i$, only uses maximal buckets, since $S_{i+1}$ does so. This is important in the proof of S2, which follows.

Suppose $B_k$ is the first bucket left empty by the greedy algorithm. Then we claim that $|B_1| = \ldots = |B_{k-1}| \geq |B_k|$. Clearly, $B_1, \ldots, B_{k-1}$ have non-decreasing size, since they are all filled by the greedy algorithm. Suppose that there is some $B_i = \{t_i, \ldots, u_i\} \ (i < k)$, that is bigger than its direct predecessor. By N2 all demands from $t_i$ and later periods are placed in buckets of size bigger than the size of $B_1$. Thus, the cumulative demand of $\{1, \ldots, t_i - 1\}$ is enough to force all the buckets $B_1, \ldots, B_{i-1}$ to be filled, in contradiction with N4 (N4 requires that $B_1$ can be left empty). Therefore, all buckets before $B_k$ are of the same size. By the same reasoning one can show that $B_k$ is not bigger than $B_{k-1}$.

If $|B_k| < |B_{k-1}|$, then we take $A = B_k$. In the solution, where $B_{k-1}$ is filled with one unit, $B_k$ is left empty, since it is not maximal. By N4 we know all production in $B_1$ can be transferred to later periods. If a bucket before $B_k$ is empty we use this one, otherwise one unit of the production in $B_1$ can be transferred to the first period of $B_k$ and the rest to $B_{k-1}$. This gives the desired solution for $B_1$. Since the production of each $B_i$ can be transferred to $B_1$, the solution for $B_i$ is also available. Finally, consider the solution where a bucket $B$ is filled with one unit. In this solution $B_k$ is empty, and therefore the solutions in S2 can be created.

If $|B_k| = |B_{k-1}|$, then we take $A = B_{k+1}$. Consider the solution, where $B_{k+1}$ is filled with one unit of production. Then $B_1, \ldots, B_{k-1}$ are filled and $B_k$ is empty, since they are filled according to the greedy algorithm. By N4, we may transfer the production of $B_1$ to $B_k$. Thus, we can generate the desired solutions for $B_1, \ldots, B_k$. For buckets after $B_{k+1}$ the filling with one unit leaves $B_k$ empty. This proves S2.

The proof of S3 is easier. Let $B$ be the first bucket of size bigger than one. Consider the optimal solution where $B$ is filled with one unit of production. If the first hole $h$ is later than $B$, then we can transfer production from $h$ to $B$. Otherwise, if $h$ is before $B$, then its production can be moved to $B$ by N1. It follows that the first hole can be left empty. Clearly, this implies that later holes can be left empty.

With the conditions S1-S3, we can now create the following directions.

**Case 1.** $y_\tau = 1$ if $\tau \notin Y$.

The direction $y_\tau = 1$ is created as follows. If $\tau$ is a hole, then we take the solution in which the hole is empty as a second solution. In the first we take $y_\tau = 1$. If $\tau$ is not a hole, then it must be the first period in a bucket $B = \{t, \ldots, u\}$. From S2 it follows that there exists a solution that does not use $B$ for production, if there are at least two buckets. With this solution the direction $y_\tau = 1$ can be created easily. If there is only one bucket then it has size at least two (see P2). Again from S2 it follows that there is a solution which produces in the last period of this bucket. Again the direction $y_\tau = 1$ can be created easily.
Case 2. $y_{\tau} = x_{\tau} = 1$, $x_t = -1$ if $\tau \in Y$, where $B = \{t, \ldots, u\}$ contains $\tau$.

The direction $y_{\tau} = x_{\tau} = 1$, $x_t = -1$ is created by considering the optimal solution where $y_u = x_u = 1$. The production in $u$ is transferred to period $\tau$ in the first solution, necessary to create the direction, and $y_{\tau} = 1$ is added. The second direction is created from this optimal solution by transferring the production from $\tau$ to $t$. These solutions provide the suggested direction.

Case 3. $x_\tau = 1$, if $\tau \notin X$.

If $\tau$ is a hole, then this direction can be created directly from the solution in case 1. Otherwise, $\tau$ is in a bucket $B = \{t, \ldots, u\}$ ($\tau \neq t$). The direction $x_\tau = 1$ is created by considering the optimal solution where $y_{\tau-1} = x_{\tau-1} = 1$. This is the second solution. The first solution is created from the second one, by adding $x_\tau = 1$. These solutions provide the suggested direction.

Case 4. $x_\tau = 1$, $x_t = -1$, if $\tau \in X$ for some choice of $t \in X$.

The main problem here is to ensure that these directions are linearly independent. This is reached by making the following choices. Consider $A$ as defined in S2. Let $\tau \in B$, if $B$ precedes $A$, then $t$ is chosen as the first period of $A$. Otherwise, $t$ is chosen as the first period of an empty bucket preceding $B$, which exists according to S2. Note that for $A$ itself no direction is created.

The number of created directions is now: $(T - |Y|) + |Y| + (T - |X|) + (|X| - 1) = 2T - 1$. This is the desired number of directions.

### 4 Separation for the hole-bucket inequalities

In this section we define an acyclic network on which the separation problem for the hole-bucket inequalities can be solved as a shortest path problem. This network is defined such that each path corresponds to a facet-defining hole-bucket inequality and vice versa. In the following we consider a given fractional solution denoted by $(x, y)$.

We define the following set of nodes. Each node consists of a quadruple $(t, n, m, s)$ with the following definitions:

1. $t$ is the current period. It ranges from 1 to $T$;
2. $n$ is the number of freely usable periods up till $t$ (i.e., holes and free periods in partially filled buckets) for demand after period $t$;
3. $m$ is the number of empty buckets of the maximum size available up till period $t$;
4. $s$ is the maximum size of buckets available up till period $t$.

These are the four characteristics of a vertex in the network. $(0, 0, 0, 0)$ is the starting node, and $(T, n, m, s) (n, m, s \in \{1, \ldots, T\})$ are the ending nodes.

We define the arcs in the network with their corresponding costs as follows. An arc either represents a hole or a bucket:
An arc representing a hole for period \( t + 1 \):
This arc leaves vertex \( (t, n_1, m, s) \) and it enters vertex \( (t + 1, n_2, m, s) \), where the coefficient \( n_2 = n_1 + 1 - d_{t+1} \). Note that the maximum size of the buckets did not change and there are no new buckets of maximum size. There is one extra hole available, thus if period \( t + 1 \) is a not a demand period, then there is an extra freely usable period. The costs corresponding to this arc are zero: there are no variables added to the corresponding inequality, and there are no new buckets filled.

An arc representing a bucket of size \( l \) with periods \( t + 1, \ldots, t + l \)
This arc leaves vertex \( (t, n_1, m_1, s_1) \) and it enters vertex \( (t + l, n_2, m_2, s_2) \), where the coefficients \( n_i, m_i, s_i \) \( (i = 1, 2) \) have the following relations:

If \( l < s_1 \) and \( d_{t+1,t+l} \leq n_1 \), then \( n_2 = n_1 - d_{t+1,t+l}; m_2 = m_1; s_2 = s_1 \).
If \( l < s_1 \) and \( d_{t+1,t+l} > n_1 \), then \( n_2 = n_1 - d_{t+1,t+l} + s_1; m_2 = m_1 - 1; s_2 = s_1 \).
If \( l = s_1 \) and \( d_{t+1,t+l} \leq n_1 \), then \( n_2 = n_1 - d_{t+1,t+l} + s_1; m_2 = m_1; s_2 = s_1 \).
If \( l = s_1 \) and \( d_{t+1,t+l} > n_1 \), then \( n_2 = n_1 - d_{t+1,t+l}; m_2 = 1; s_2 = l \).
If \( l > s_1 \) and \( d_{t+1,t+l} \leq n_1 \), then \( n_2 = n_1 - d_{t+1,t+l} + l; m_2 = 0; s_2 = l \).

If \( l \leq s_1 \), then the maximum size of the buckets remains unchanged. Thus \( s_2 = s_1 \). If \( l < s_1 \), then there is no new bucket of maximum size, thus \( m_2 = m_1 \). If \( l = s_1 \), then there is one new bucket of maximum size, thus \( m_2 = m_1 + 1 \). If \( l > s_1 \), then the maximum size of the buckets did change. Thus \( s_2 = l \), and \( m_2 = 1 \). Now, the demands of the periods \( t + 1, \ldots, t + l \) are placed into the free periods first, and if there are not enough, then into a maximal bucket (note that only one is necessary, since \( l \leq s_2 \)). In the latter case the number of maximum buckets is decreased by one.

The costs of the arc are: the value of the variables added, i.e., \( x_{t+1} + y_{t+2} \ldots + y_{t+l} \) are counted positive; if a new maximal bucket is used, then one unit should be subtracted (the right-hand side increases by one).

Clearly, since we require that only maximal buckets are used, each path corresponds with a hole-bucket inequality. Moreover, each hole-bucket inequality that has its buckets used in order of non-decreasing size is represented by a path. Thus, all the hole-bucket inequalities (see condition N2) are represented. On the other hand, there are paths that do not represent a facet-defining hole-bucket inequality. To get rid of such paths, i.e., to make sure that each path represents a hole-bucket inequality that satisfies N1 - N4, we delete some arcs in the network:

(N1) No arc from \((t, 1, m, 1)\) to \((t + 1, 0, m + 1, 1)\): if the maximum bucket size is one, then the number of freely usable buckets (holes) should not drop from 1 to 0.
(N2a) No arc from \((t, s - 1, m_1, s)\) to \((t + l, n_2, 0, s)\) if \( d_{t+1,t+l} > 0 \), and no arc from \((t, s_1 - 1, m_1, s_1)\) to \((t + l, n_2, 0, s_2 > s_1)\) if \( d_{t+1,t+l} > s_1 \). In both cases the demands would be placed in a bucket with no maximum size bucket available.
(N2b) No arc from \((t, n_1, 0, s_1)\) to \((t + l, n_2, m_2, s_2)\) if \( d_{t+1,t+l-1} > n_1 \) if \( d_{t+1,t+l} > s_1 \). In this
case some demand would be placed in a new bucket with no maximum size bucket completely available.

(N3) If \( t + l = T \), then no arc from \( (t, n_1, m_1, s_1) \) to \( (t+l, n_2, m_2, s_2) \) if \( d_{t+1,t+l} \mod s_1 \neq 1 \), otherwise the demand of the last period does not take a new bucket.

(N4) No arc from \( (t, t, t, 0) \) to \( (t + l \neq T, n_2, m_2, l \geq 0) \), if \( d_{t+1,t+l} > n_2 \), otherwise the first bucket is necessarily used.

The costs of a path in the network, i.e., hole-bucket inequality, are exactly equal to the value of the variables of the solution \( (x, y) \) of the left-hand side of the inequality minus the value of the right-hand side of the inequality. Thus, we are looking for a path with negative costs. The network being polynomially sized in the number of periods, the separation problem will be proved to be solvable in polynomial time. The complexity of the algorithm of finding a negative length path from the starting vertex to one of the vertices, possibly intermediate, is determined by the number of arcs, since the network is acyclic. The number of vertices is \( O(T^4) \). From each vertex there are only \( O(T) \) possibilities to leave it. Thus, the complexity of the algorithm is \( O(T^5) \).

This ends the description of the separation algorithm.

5 Concluding remarks

A final remark concerning the hole-bucket inequalities concerns their number. There are exponentially many such facets, as follows from the following example. Suppose that on the planning horizon \( \{1, \ldots , T\} \) (\( T \geq 5 \)), we have the following demand pattern: \( d_1 = \ldots = d_{T-3} = 0, d_{T-2} = d_{T-1} = d_T = 1 \). Then any inequality with two or more buckets of size two and additional buckets of size one defines a facet.

We give two examples to show that many but not all facets for DLSP are also facet-defining in case overproduction is not allowed. The first example shows that facets of the full-dimensional polyhedron need not be facets of the original polyhedron. The second example shows that facets of the original polyhedron are not uniquely representable, i.e., there are representations which are not valid for the full polyhedron.

EXAMPLE 1: \( T = 4 \): \( d_1 = d_2 = d_3 = 0; d_4 = 1 \). The inequality \( x_1 + y_2 + x_3 + y_4 \geq 1 \) is a facet of the full polyhedron, but it is a lower dimensional face of the original one. The latter follows, since each feasible solution satisfying \( x_1 + y_2 + x_3 + y_4 = 1 \) also satisfies \( y_4 = x_4 \). Note that \( y_4 \geq x_4 \) is valid, if 7.2.2 holds as an equation, but not if it is relaxed. In general, the inequality \( y_t \geq x_t \) is valid under the condition \( d_{t-1,T-1} = 0 \), in the original polyhedron but not in the full polyhedron.

EXAMPLE 2: \( T = 6 \): \( d_1 = d_2 = d_3 = d_4 = 0; d_5 = d_6 = 1 \). The inequality \( y_3 + y_4 \geq x_4 \) is a facet in the original polyhedron, but it is not valid in the full polyhedron, because overproduction is allowed there. To make it a facet of the full polyhedron, the equation \( x_1 + x_2 + x_3 + x_4 + x_5 + x_6 = 2 \) must be added. Then we obtain the inequality \( x_1 + x_2 + x_3 + x_4 + x_5 + x_6 \geq 2 \). In general, if \( d_1 = 0 \) and \( d_{t+1,T} = t_2 - t_1 \), then \( y_t + 1 + \ldots + y_t \geq x_t \) is a facet in the original polyhedron, but it is not valid in the full polyhedron. The equation \( \sum_{t=1}^{T-1} x_t = d_{1,T} \) should be added to obtain the following facet for the full polyhedron.
Some final remarks with respect to related work.

First, the set of strong valid inequalities given here is not complete. There are many different types of valid inequalities known for DLSP (see van Hoesel [6]). These inequalities all have 0/1-coefficients, except for the start-up inequalities. For the more general problem considered in Magnanti and Vachani [10] and Sastry [17] the strong valid inequalities are more complicated in the sense that other coefficients are also possible there. Sastry gives the most complete description. He also considers holes, but he has no complete characterization. In a following paper we will discuss the generalization of the hole-bucket inequalities to the DLSP with set-up costs. Moreover, we will include computational experience.

References


<table>
<thead>
<tr>
<th>Number</th>
<th>Month</th>
<th>Author</th>
<th>Title</th>
</tr>
</thead>
<tbody>
<tr>
<td>93-01</td>
<td>January</td>
<td>P. v.d. Laan C. v. Eeden</td>
<td>Subset selection for the best of two populations: Tables of the expected subset size</td>
</tr>
<tr>
<td>93-03</td>
<td>February</td>
<td>Jan Beirlant John H.J. Einmahl</td>
<td>Asymptotic confidence intervals for the length of the shortt under random censoring.</td>
</tr>
<tr>
<td>93-04</td>
<td>February</td>
<td>E. Balas J. K. Lenstra A. Vazacopoulos</td>
<td>One machine scheduling with delayed precedence constraints</td>
</tr>
<tr>
<td>93-05</td>
<td>March</td>
<td>A.A. Stoorvogel J.H.A. Ludlage</td>
<td>The discrete time minimum entropy $H_\infty$ control problem</td>
</tr>
<tr>
<td>93-06</td>
<td>March</td>
<td>H.J.C. Huijberts C.H. Moog</td>
<td>Controlled invariance of nonlinear systems: nonexact forms speak louder than exact forms</td>
</tr>
<tr>
<td>93-07</td>
<td>March</td>
<td>Marinus Veldhorst</td>
<td>A linear time algorithm to schedule trees with communication delays optimally on two machines</td>
</tr>
<tr>
<td>93-08</td>
<td>March</td>
<td>Stan van Hoesel Antoon Kolen</td>
<td>A class of strong valid inequalities for the discrete lot-sizing and scheduling problem</td>
</tr>
</tbody>
</table>