Solving the Parallel Replacement Problem Under Economies of Scale and Non-Decreasing Demand with Branch-and-Cut

Joseph C. Hartman
Shangyuan Luo
Lehigh University

Report No. 04T-024

Solving the Parallel Replacement Problem under Economies of Scale and Non-Decreasing Demand with Branch-and-Cut

Joseph C. Hartman[†]**and Shangyuan Luo**

Department of Industrial and Systems Engineering

Lehigh University

Bethlehem, PA 18015-1582

Abstract

The parallel replacement problem considers the individual replacement schedule of each asset in a group of assets that operate in parallel and are economically interdependent. Economies of scale, including a fixed charge in any period in which an asset is purchased, are a common cause of economic interdependence. Valid inequalities are derived for the parallel replacement problem with both fixed and variable costs for integer programming formulations considering constant and non-decreasing demand. The cuts are motivated by the "no-splitting rule" in the literature which states that an optimal solution exists such that assets of the same age in the same time period are kept or replaced as a group. The constraints are shown to be valid for cases of non-decreasing demand, which include traditional replacement and capacity expansion problems. Experimental results illustrate the effectiveness of the cuts with respect to solving the integer programs in a branch-and-cut framework.

Keywords: Parallel Replacement, Economies of Scale, Fixed Charge Network, Minimum Cost Flow.

1 Introduction

The parallel replacement problem is concerned with determining minimum cost replacement schedules for each individual asset in a group of assets that operate in parallel and are economically interdependent. The replacement of assets is generally driven by economics, including increased operating and maintenance costs (O&M) of deteriorating assets and the availability of newer, more efficient assets in the marketplace. Unlike serial (single asset) replacement problems (see Fraser and Posey [5], for example), parallel replacement problems are combinatorial as groups of assets must be analyzed simultaneously.

A variety of parallel replacement problems have been studied in the literature. This paper is concerned with the parallel replacement problem under economies of scale, as first examined by Jones et al. [10], which assumes that a fixed charge is incurred in each period that an asset is purchased. Jones et al. [10] examine the problem with the use of dynamic programming and show that in the optimal solution, groups of same aged assets in the same time period are either kept or replaced in their entirety assuming constant demand

[†]This research was sponsored in part by NSF grants DMD-9713690 and DMI-9984891

and no capital budgeting constraints. This "no-splitting" rule greatly reduces the complexity of the problem. Chen [4] reformulates the problem as a 0-1 integer program and solves it using Benders' decomposition while McClurg and Chand [11] provide a forward algorithm and forecast horizon result.

Further generalizations of this problem (with fixed charges) have also been studied. Rajagopalan [13], with integer programming and a dual-based solution procedure, and Chand et al. [3], with enumerative and heuristic algorithms, examine the problem under the assumption of non-decreasing demand, also termed replacement and expansion. Hartman [6] further generalizes the problem to include demand and capital budgeting constraints with the use of integer programming. The "no-splitting rule" is generalized to the case of non-decreasing demand in all three of these papers.

This paper presents valid inequalities and a branch-and-cut solution procedure to an integer programming formulation for the deterministic, parallel replacement problem under economies of scale (PRES) in which a number of assets are required for operations in each period over a finite horizon of length T. Under the assumption of constant demand, all assets are utilized each period while they may be stored in the case of non-decreasing demand to take advantage of economies of scale in purchases. At the end of each period, an asset may be salvaged or retained, assuming it has not reached its maximum physical life N, at which time it must be retired. Assets may be replaced through the purchase of new assets. The purchase of assets is subject to a fixed charge, regardless of the order size. At the end of the finite horizon, all assets are salvaged. The solution consists of purchase and salvage decisions for each asset over the finite horizon with the objective to minimize discounted purchase and O&M costs less salvage values.

This paper is motivated by the integer programming results of Hartman [6] and the successful use of cutting planes in solving lot sizing problems (Barany et al. [1]). Conceptually, the parallel replacement problem and the lot sizing problem are similar. In the lot sizing problem, inventory purchases are made by trading off a fixed charge (setup cost) against inventory carry charges. In the parallel replacement problem under economies of scale, additional fixed charges are incurred if assets are not replaced simultaneously.

Hartman [6] illustrated that the linear programming relaxation of a restricted subproblem of PRES has integer extreme points. Specifically, if the binary variables (required for imposing a fixed charge with asset purchases) are fixed, then the optimal solution to the linear programming relaxation of the resulting formulation is integer-valued, if a feasible solution exists. Thus, branch and bound procedures must only focus on the T binary variables. In this paper, we provide valid inequalities which focus on these T variables to further reduce the difficulty of solving PRES under constant or non-decreasing demand constraints. (The results in Hartman [6] are valid for any demand.)

The valid inequalities are useful from a computational standpoint but they are also interesting as they are derived from the "no-splitting rule" which has been used to reduce the computation in earlier dynamic programming approaches to the problem. The rule states that an optimal solution to PRES exists such that all assets of the same age in the same time period are either kept or replaced as a group. This fact is used to tighten constraints which enforce the fixed charge in each period of an asset purchase.

This paper makes three contributions to the replacement analysis literature. First, we prove that PRES is NP-Hard, as it has not formally been stated in the literature. Second, a set of valid inequalities is defined for PRES. It is shown that these are valid for both constant and non-decreasing demand cases. Their relationship to the "no-splitting rule" is made clear in their development. Third, computational results show that the incorporation of these cuts into a branch-and-cut procedure drastically improves the solution time of both constant and non-decreasing demand cases of PRES. This is especially true for large problem instances, such as those from the railroad industry analyzed in Hartman and Lohmann [7]. As it is commonly assumed that the horizon should be at least twice the maximum age of an asset (Bean et al. [2]) to ensure optimal time zero decisions (for an infinite horizon problem), problems with maximum ages of 50 periods and horizons of 100 periods are not uncommon, leading to the need to solve large-scale problems.

2 PRES under Constant Demand

We now formulate PRES as a finite horizon integer program. The program is modified from PRP in Hartman [6] with no budgeting constraints and constant demand. It is further assumed that the number of assets in inventory at time zero is equal to demand and no asset in the initial inventory is equal to its maximum service life. This eliminates the automatic decision of having to replace an asset at time zero. Assuming constant demand and no capital budgeting constraints eliminates the need to store assets, as in Hartman [6].

An asset is defined by its age, $i = 0, 1, \ldots, N$ at the end of time period $j = 0, 1, \ldots, T$. An asset may be retained or salvaged after each period unless it reaches age N at which time it must be salvaged. The problem is solved over T periods, with purchases allowed at the end of periods $0, 1, \ldots, T-1$. All assets are sold at the end of time period T. The decision variables are summarized as follows:

X_{ij} = the number of i-period old assets in use from the end of period j to $j+1$, where X_{0j} also denotes asset purchases;
S_{ij} = the number of i-period old assets salvaged at the end of period j;
Z_j = 1 if a purchase is made in period j, else $Z_j = 0$.

The deterministic costs associated with each of these decisions are defined as follows:

p_j = discounted per-unit cost for a new asset purchased at the end of period j;
k_j = discounted fixed cost for any asset purchase at the end of period j;
c_{ij} = discounted O&M cost for an i-period old asset in use from the end of period j to $j+1$;
r_{ij} = discounted revenue for an i-period old asset salvaged at the end of period j.

Other relevant parameters include:

n_i = the number of i-period old assets available (in inventory) at time zero;
d = number of assets demanded in each period such that $d = \sum_i n_i$.

To aid in the discussion, part of the formulation may be visualized as a network, as illustrated in Figure 1. The network is drawn on two axes, with the y-axis representing the age of an asset ($i = 0, 1, \ldots, N$) and the x-axis representing the end of the time period ($j = 0, 1, \ldots, T$). Figure 1 represents a problem with $N = 3$ and $T = 5$. The nodes are labeled according to the age of an asset and the end of the time period, (i, j), although the labels have been removed from the figure for clarity.

Referring to Figure 1, the number of i-period old initial inventory assets (n_i) is represented by the supply at each node $(i, 0)$, $i > 0$ and $i < N$. (Allowing $n_N > 0$ forces a replacement at time zero, which can be incorporated easily.) Flow between nodes (i, j) and $(i+1, j+1)$ represents assets in use (X_{ij}) from the

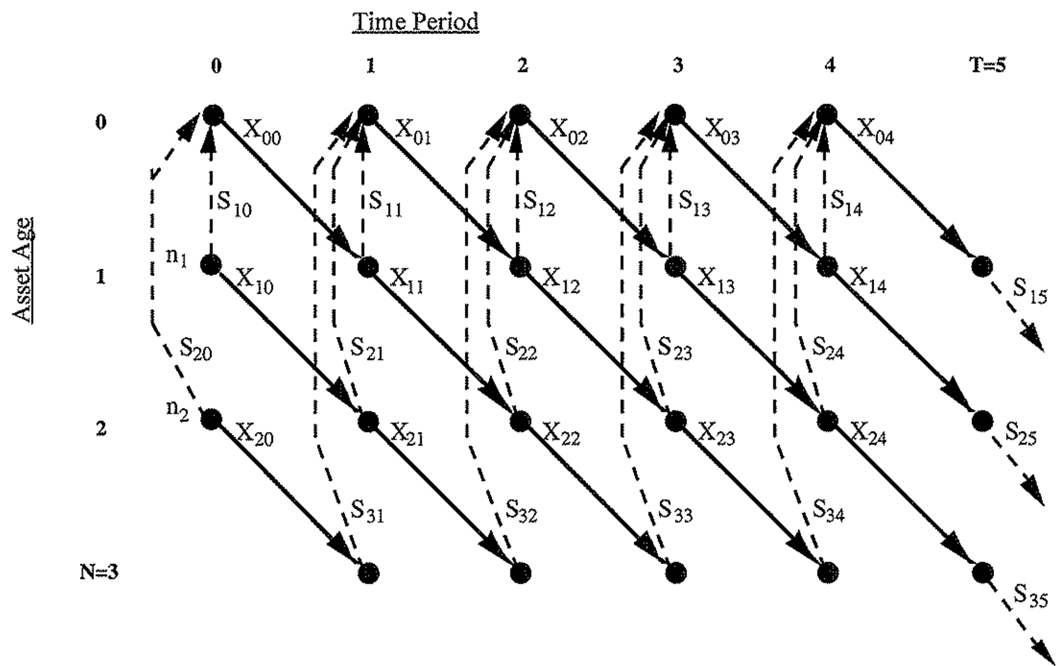

Figure 1: Network representation of PRES under constant demand with flow representing purchase (X_{0j}), utilization (X) and salvage (S) variables and initial inventory supply (n).

end of period j to the end of period $j+1$, at which time the assets are $i+1$ periods old. The variable X_{0j} denotes the purchase and one period use of a new asset. For this model it is assumed that a purchased asset is new and it must be retained for a minimum of one period before being eligible for salvage. Note that under the assumption of constant demand, any asset that is salvaged must be replaced, and thus a purchase must occur. The sale or salvage of an asset is represented as flow, S_{ij}, from any node (i,j), $i > 0$ and $j \geq 0$ to node $(0,j)$, denoting the subsequent purchase. All salvage arcs are represented with dashed lines to distinguish from utilization arcs in Figure 1.

With these variables and parameters, the integer programming formulation for PRES follows.

$$\min_{X,S,Z} \sum_{j=0}^{T-1} (p_j X_{0j} + k_j Z_j) + \sum_{j=0}^{T-1} \sum_{i=0}^{N-1} c_{ij} X_{ij} - \sum_{j=0}^{T} \sum_{i=1}^{N} r_{ij} S_{ij} \tag{1}$$

subject to:

$$X_{i0} + S_{i0} = n_i \quad \forall i \in \{1, 2, \ldots, N-1\} \tag{2}$$

$$\sum_{i=1}^{N} S_{ij} - X_{0j} = 0 \quad \forall j \in \{0, 1, \ldots, T-1\} \tag{3}$$

$$X_{(i-1)(j-1)} - X_{ij} - S_{ij} = 0 \quad \forall i \in \{1, 2, \ldots, N-1\},\ j \in \{1, 2, \ldots, T-1\} \tag{4}$$

$$X_{(i-1)(j-1)} - S_{ij} = 0 \quad \forall i = N,\ j \in \{1, 2, \ldots, T\} \text{ and } i \in \{1, 2, \ldots, N\},\ j = T \tag{5}$$

$$X_{0j} \leq dZ_j \quad \forall j \in \{0, 1, \ldots, T-1\} \tag{6}$$

$$X_{ij}, S_{ij} \in \{0, 1, 2, \ldots\} \quad (7)$$
$$Z_j \in \{0, 1\} \quad (8)$$

With constant demand, PRES is a fixed charge minimum cost flow problem. The objective function (1) minimizes discounted purchase and O&M costs less salvage values. Constraints (2) through (5) ensure flow throughout the network. Constraint (6) includes the fixed charge variable Z_j such that if any assets are purchased, the fixed charge is imposed. As demand d is constant and the number of initial assets in the system is d, the maximum number of assets that can be purchased in any period is d. All decision variables are general integers with the Z_j being restricted as binary.

For this formulation, there are T binary variables representing the periods of purchase, $N \times T$ utilization variables and $(N \times (T+1))$ - 1 salvage variables, totaling $(T + 2NT + N - 1)$ variables. There are T constraints to connect the purchase decisions with the binary variables and $((N+1) \times (T+1)) - 2$ flow balance constraints, totaling $(2T + NT + N - 1)$ constraints.

3 Difficulty of PRES

In this section, we illustrate that a restricted version of the PRES problem under constant demand is NP-Hard. While this may have been assumed, it has not been shown previously in the literature. To prove this, we transform an instance of the minimum directed Steiner network problem (Hwang et al. [9]), known to be NP-Hard (Hsu et al. [8]), to PRES. The minimum directed Steiner network problem is defined as follows: given a set of starting vertices S and a set of terminating vertices V' on a directed, acyclic graph $G(V, E)$, we want to find a subgraph G^* with the minimum total edge weight such that G^* is $S, T-$connected. We consider the case where there is one terminating node, v_0, and no nodes in S are connected. Nodes in S and v_0 are referred to as terminal nodes while the remaining nodes are referred to as non-terminal nodes. Finally, define w_{ij} as the weight (length) of arc (i, j) in G.

Theorem 1 *PRES with $p_j = c_{ij} = s_{ij} = 0 \; \forall ij, \; j \geq i$ is NP-Hard.*

Proof. Given an instance of the minimum directed Steiner network problem, define the following sets: starting vertices S and non-terminal nodes $V_1 \cup V_2 \cup V_3$. These are shown schematically in Figure 2 such that nodes in V_1 are connected to S, nodes in V_3 are connected with v_0, and nodes in V_2 are connected with both S and v_0.

Before constructing an instance of PRES, we provide a different graphical representation than previously defined, as given in Figure 3. The nodes in the left column, labeled i, represent the possible ages of initial clusters of assets. As in our previous formulation, flow from these nodes is defined as n_i, or the number of i-period old assets at time zero. Flow is from a node i to a node j, representing the time period that the initial cluster is replaced. An arc (i, j) exists if $i + j \leq N$. The nodes in column j represent the time periods that a cluster is replaced. Note that not all arcs are shown for clarity. An arc $(j, j+k)$ exists if $k \leq N$. Finally, arcs (j, T) exist if assets are retained from the end of period j to the end of the horizon T, when they are replaced. These arcs exist for $j \geq T - N$ and $j \leq T - 1$.

In our previous formulation, costs were defined according to periods. In this formulation, costs are defined accordingly by the length of time an asset is held. For example, arcs (i, j) carry the cost of retaining an

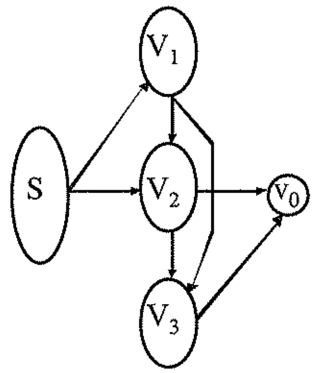

Figure 2: Schematic of sets of nodes and arcs in directed, acyclic network.

i-period old asset for j periods before salvaging the asset. Arcs $(j, j+k)$ represent the cost to purchase an asset at time period j, utilize it for k periods, and then salvage the asset. Finally, there is a fixed charge associated with each $(j, j+k)$ arc. We will use the term c_{ij} for costs between arcs and k_j for the fixed charge associated with flow from node j.

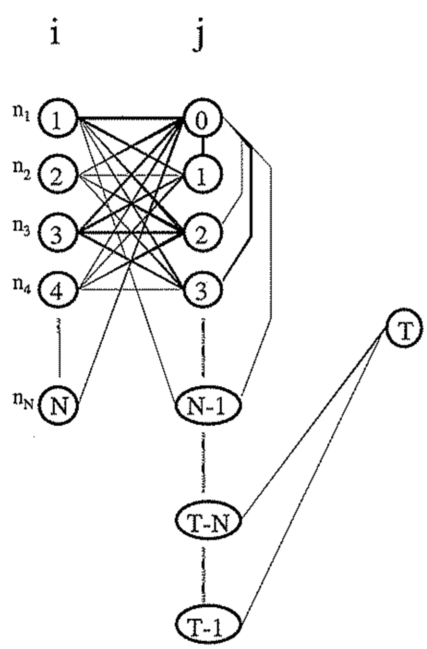

Figure 3: Different graphical representation of PRES. Note that all arcs are directed (from left to right) and arcs are omitted from nodes $j > 0$ to $j + k$, $j + k \leq N$ for clarity.

Note the similarities in Figures 2 and 3. This provides the motivation for the new graphical representation. Intuitively, our approach to constructing an instance of PRES from the minimum directed Steiner network problem is as follows: the nodes in S in Figure 2 correspond to those defined as i in Figure 3. The nodes in V_1 and V_2 correspond to those defined as j in Figure 3 where $j \leq N - 1$. The nodes in V_3 correspond to those j where $j \geq T-N$. Finally, v_0 corresponds to T. The difficulty in performing the transformation from the Steiner network to PRES is due to the systematic number of arcs emanating from each node in PRES. We accomplish this by adding nodes and arcs to the Steiner network.

Given the Steiner network, we construct a network for PRES as in Figure 3 as follows. First consider the nodes we defined as j in the PRES network. Given the $V_1 + V_2$ nodes in the Steiner network, then there are at most $|V_1 + V_2|$ arcs emanating from any node in S. Define the j nodes in PRES as $j' + V_3$ where $j' = V_1 + V_2$ *plus* an additional $|V_1 + V_2| - 1$ nodes. We label these nodes from 0 to j'. The remaining nodes, from V_3, are labeled from j' to $T-1$.

The nodes in S in the Steiner network correspond directly with the nodes defined as i in the PRES network. For the node in S with highest degree (break ties arbitrarily), connect an arc in the PRES network from i to all nodes in j'. For the node with next highest degree, connect to all nodes 0 to $j'-1$. Continue in this manner noting that each node in S will contain its original arcs (from the Steiner network) and additional arcs. For each original arc from the Steiner network, define the cost $c_{ij} = w_{ij}$ and for each newly created arc, define c_{ij} arbitrarily high. In PRES, the nodes i now have degree $2|V_1 + V_2| - 1$ to $|V_1 + V_2|$ and correspond to initial asset ages of $N - j'$ to $N - j' - |V_1 + V_2| - 1$, respectively. We assume one asset in each initial asset cluster in PRES as the Steiner network is only concerned with fixed (not per unit) charges.

We continue construction of the network for PRES by defining arcs connecting V_1, V_2, V_3, and v_0 from the Steiner network in exactly the same manner in PRES. Note that in PRES, we define a single fixed charge for an asset purchase in a given time period. However, in the Steiner network, it may be possible that there are multiple arcs emanating from a node in V_1, V_2 or V_3 with different costs. As all paths from V_1, V_2 and V_3 are directed and acyclic towards node v_0, we can determine which of the multiple arcs emanating from a node is on the shortest path to v_0. Thus, for a given node, we can eliminate the inferior arcs at that node as it cannot be in the optimal solution (due to acyclic network). After this reduction, we define the per unit cost as $c_{ij} = 0$ and the fixed charge $k_j = w_{ij}$ for the remaining arcs.

Finally, we complete the network by assuming the maximum asset age N is arbitrarily large such that there is an arc from each node $j = 0, 1, \ldots, T-1$ to each node k, where $k \leq T$. (Note that some of these arcs are already present from the previous transformation steps. This step creates arcs not present.) The costs on these new arcs are arbitrarily large, as they are not present in the original Steiner network.

Thus, we have transformed our instance of the minimum directed Steiner network problem to an instance of PRES. This is done in polynomial time as we have added $|V_1 + V_2 - 1|$ nodes and a polynomial number of arcs based on the total number of nodes.

Solving this instance of PRES provides the optimal solution to the minimum directed Steiner network problem. This is because the Steiner network is embedded in the PRES network and additional arc costs are arbitrarily high such that they will not be included in the solution. As we assume that the initial asset clusters are of size one, the solution of PRES produces a cost which correlates directly with the Steiner (per arc) cost. Thus, PRES is NP-Hard. □

This provides further motivation for our research. We now return to our original formulation of PRES and present our solution approach.

4 Valid Inequalities for PRES

As shown in Hartman [6], if the Z_j variables are either 0 or 1, the remaining X and S variables are integer as the extreme points of the linear programming relaxation are integer. This is clear in the constant demand formulation as it is a minimum cost flow problem with fixed charges associated with the X_{0j} arcs which are enforced in Constraint (6). The valid inequalities in this paper are designed to tighten these constraints, forcing the Z_j variables to be 0-1.

Although we derive one class of valid inequalities, we present one subset separately for clarity as follows.

Theorem 2 *The following inequality:*

$$S_{ij} \leq n_{i-j} Z_j \quad \forall j < N,\ \forall i > j \tag{9}$$

is valid for PRES.

Proof. First note that with $j < N$ and $i > j$, this refers to the sale of an initial inventory cluster. As the number of assets in an initial inventory cluster of age i is n_i, then:

$$S_{ij} \leq n_{i-j} \quad \forall j < N,\ \forall i > j.$$

If $S_{ij} > 0$, then $Z_j = 1$ as the sale of an asset requires the purchase of a new asset. Thus, Constraints (9) are valid for PRES. □

We refer to these inequalities as "Lower Triangle Cuts," or LTC, because if one were to draw a line through nodes $(0,0)$, $(1,1)$, ..., (N,N) in Figure 1, the nodes for which one can write Constraints (9) form a triangle below the line. They are illustrated in the following example.

Example 1:

Consider an example with $T = 20$, $N = 8$ and $d = 31$ with an initial inventory of assets such that $n_1 = 9$, $n_2 = 8$, $n_3 = 2$, $n_5 = 8$ and $n_6 = 4$. The linear programming relaxation of this problem is such that each asset cluster is kept until its maximum age $N = 8$. This results in the following:

$$Z_j = \{0, 0, 0.13, 0.26, 0, 0.06, 0.26, 0.29, 0, 0, 0.13, 0.26, 0, 0.06, 0.26, 0.29, 0, 0, 0.13, 0.26\}.$$

The six-year old initial asset cluster is kept such that $X_{60} = X_{71} = S_{82} = X_{02} = \ldots = 4$. This results in $Z_2 = 4/31 = 0.13$. The pattern is similar for all clusters as they are all kept to their maximum age. The optimal solution to the linear programming relaxation is 45,470.

We introduce the lower triangle inequalities (9) now. For example, for the six-year old cluster, we include:

$$\begin{aligned} S_{6,0} &\leq 4Z_0 \\ S_{7,1} &\leq 4Z_1 \\ S_{8,2} &\leq 4Z_2 \end{aligned}$$

Introducing all LTC (35 in total) results in a linear programming relaxation solution of 48,298 with:

$$Z_j = \{0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0.39, 0, 0, 0.61, 0, 0, 0, 0, 0.39, 0\}.$$

This objective function represents a 6.22% improvement on the lower bound produced by the linear programming relaxation without LTC. In the new solution, the initial clusters of age five and six are retained for two years while the remaining clusters are retained for five years. This results in two clusters of 12 and 19 assets that are retained for 8 years in each replacement cycle through the end of the horizon. The first 13 periods of the solution with the LTC is given in Figure 4. □

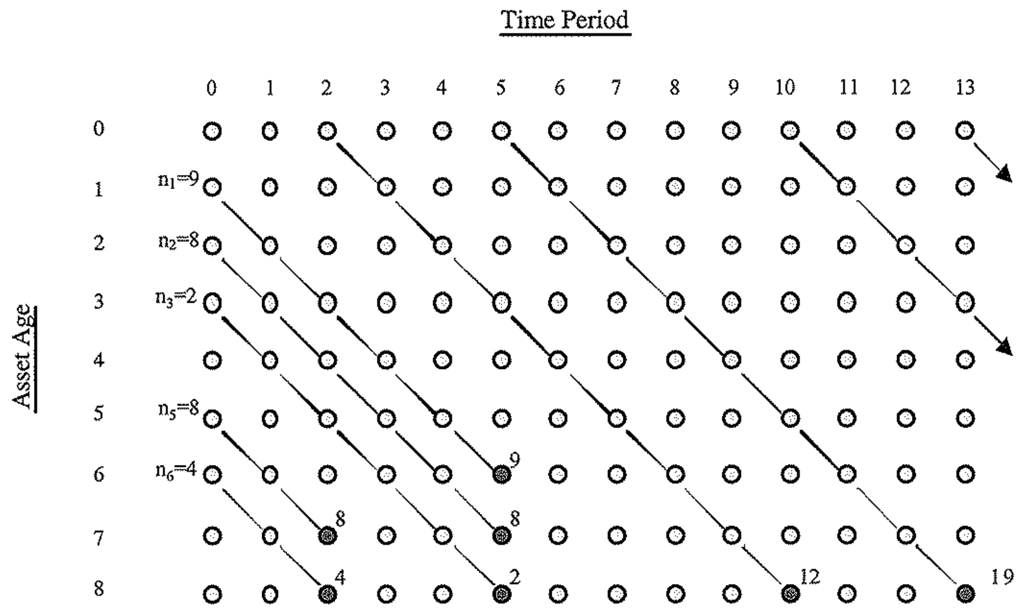

Figure 4: Solution network for first 13 periods of the example with LTC.

Our second class of valid inequalities are similar to the traditional flow cover inequalities derived for fixed charge networks ([12]) in that they are derived by isolating some flow through a node (or set of notes). However, the cuts are tailored to the parallel replacement problem in that they are a direct result of the no-splitting rule (NSR). NSR, first proved by Jones et al. [10] for constant demand and homogeneous assets, states that an optimal solution to PRES exists such that any cluster of same aged assets in the same time period are either kept or replaced in their entirety.

Given NSR and constant demand, it should be clear that every time there is a sale of assets (cluster or clusters), an ensuing purchase must occur. Thus, examining Figure 4, we can trace the sale of any cluster throughout the horizon T back to time period 0, when it was merely an initial cluster in inventory. This is critical, as we know the number of assets in each cluster at time zero, and thus we have some information to bound flow through the network. It is these bounds that can be used to tighten Constraints (6).

Flow cover inequalities ([12]) are derived when the capacity of inflow arcs exceed the capacity (demand) at a node. These are used to write cover inequalities in order to improve the lower bounds of linear programming relaxations. In our application, there is no situation in which the capacity of inbound arcs exceed the demand at a node (or equivalently, the capacity of outbound arcs). However, we take a similar tactic by isolating the flow of clusters (on arcs) and identifying their original source, inventory at time zero. We motivate the approach by continuing the previous example.

Example 2:

Given the solution posed in the previous example, we note that the 12 assets sold at time period 10 are comprised of the 4 six-year old assets and 8 five-year old assets at time zero which are both replaced at the end of time period two, as illustrated in Figure 4. These correspond to parameters n_5 and n_6, respectively. Consider nodes, labeled (i, j), $(0, 2), (1, 3), \ldots, (8, 10)$ as one "super" node. The flow into this node is $\sum_{i=1}^{N} S_{i,2}$ while the flow out is $S_{1,3}, S_{2,4}, \ldots, S_{8,10}$, such that by conservation of flow:

$$\sum_{i=1}^{N} S_{i,2} = \sum_{i=1}^{N} S_{i,i+2}$$

We can write this out to be clear in our discussion, such that:

$$S_{1,2} + S_{2,2} + \ldots + S_{7,2} + S_{8,2} = S_{1,3} + S_{2,4} + S_{3,5} + S_{4,6} + S_{5,7} + S_{6,8} + S_{7,9} + S_{8,10}$$

Now if we only consider a subset of the inflow to this super node, such as $S_{7,2}$ and $S_{8,2}$, then:

$$S_{7,2} + S_{8,2} \leq S_{1,3} + S_{2,4} + S_{3,5} + S_{4,6} + S_{5,7} + S_{6,8} + S_{7,9} + S_{8,10}$$

Further, we know that $S_{7,2}$ and $S_{8,2}$ are bounded by the initial assets at time zero, such that:

$$S_{7,2} + S_{8,2} \leq n_5 + n_6,$$

Our fractional value of Z_{10} resulted from the purchase of assets in time period 10, which was precipitated by asset sales in the same period, or $S_{8,10}$. We isolate this value of $S_{8,10}$, which is clearly bounded by the demand d, such that:

$$S_{7,2} + S_{8,2} - S_{1,3} - S_{2,4} - S_{3,5} - S_{4,6} - S_{5,7} - S_{6,8} - S_{7,9} \leq S_{8,10} \leq dZ_{10}$$

However, we know that $S_{7,2}$ and $S_{8,2}$ are bounded by n_5 and n_6, respectively, such that:

$$S_{7,2} + S_{8,2} - S_{1,3} - S_{2,4} - S_{3,5} - S_{4,6} - S_{5,7} - S_{6,8} - S_{7,9} \leq (n_5 + n_6)\, Z_{10} = 12Z_{10}$$

Including this constraint cuts off the fractional value $Z_{10} = 0.39$ in the linear programming relaxation. To cut off the fractional Z_{13} value, the following constraint can also be added:

$$S_{6,5} + S_{7,5} + S_{8,5} - S_{1,6} - S_{2,7} - S_{3,8} - S_{4,9} - S_{5,10} - S_{6,11} - S_{7,12} \leq 19Z_{13}$$

Including these two constraints increases the lower bound to 48,555, a 0.532% improvement over the previous lower bound. The network solution is given in Figure 5. □

As illustrated in the example, each inequality requires three components:

1. A super node which is a set P of nodes $(0, j)$ with all of their associated "diagonal nodes" $(1, j + 1), (2, j + 2), \ldots, (N, j + N)$ for each j defined by all $(0, j)$ nodes.
2. A set I of inflow nodes with at least one (i, j), $i > j$ for each $(0, j) \in P$.
3. A set O of outflow nodes (i, j), $j \geq i$ and $O \subseteq P$.

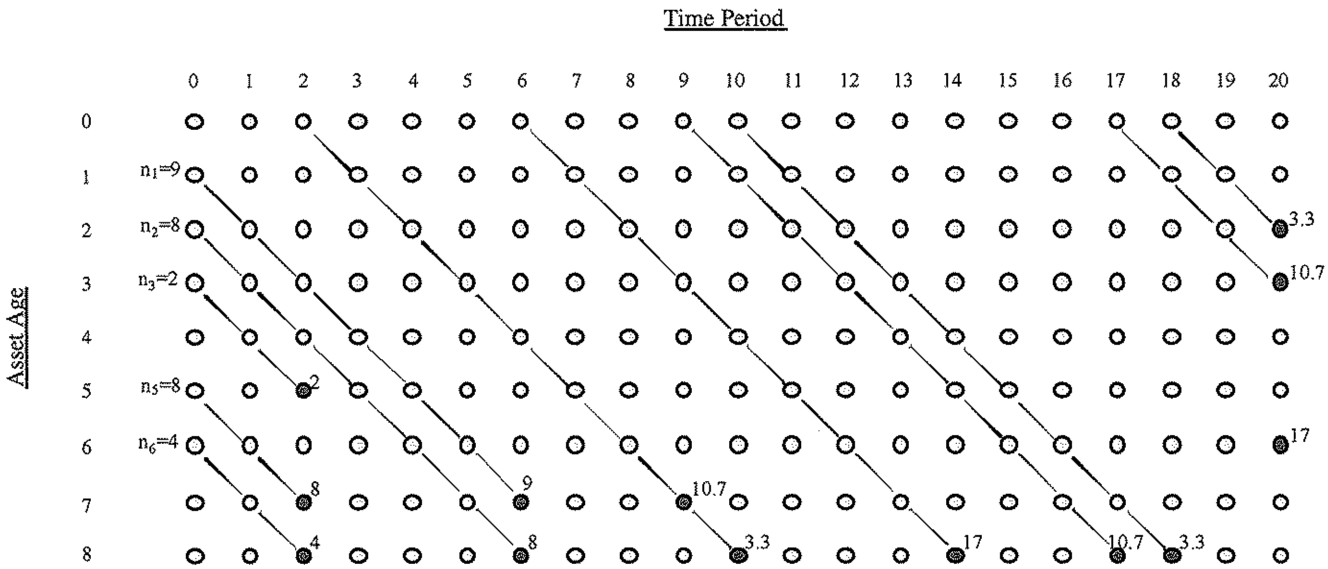

Figure 5: Solution network for example with LTC two NSRC.

Given these definitions, we can write the following.

Theorem 3 *The following inequality:*

$$\sum_{(i,j)\in I} S_{ij} - \sum_{(i,j)\in P\setminus O} S_{ij} \leq \left(\sum_{(i,j)\in I} n_{i-j}\right)\left(\sum_{(i,j)\in O} Z_j\right) \tag{10}$$

is valid for PRES.

Proof. To prove that this inequality is valid for PRES, one must note that, by definition:

$$\sum_{(i,j)\in I} S_{ij} = \sum_{(i,j)\in I} n_{i-j},$$

and thus:

$$\sum_{(i,j)\in I} S_{ij} - \sum_{(i,j)\in P\setminus O} S_{ij} \leq \sum_{(i,j)\in I} n_{i-j},$$

as all variables are non-negative.

Further note that the sales of assets in O result in a purchase, such that:

$$\sum_{(i,j)\in O} S_{ij} \leq d \sum_{(i,j)\in O} Z_j.$$

As:

$$\sum_{(i,j)\in I} S_{ij} - \sum_{(i,j)\in P\setminus O} S_{ij} \leq \sum_{(i,j)\in O} S_{ij},$$

and:

$$\sum_{(i,j)\in I} n_{i-j} \leq d,$$

we can write our valid inequality:

$$\sum_{(i,j)\in I} S_{ij} - \sum_{(i,j)\in P\setminus O} S_{ij} \leq \left(\sum_{(i,j)\in I} n_{i-j}\right)\left(\sum_{(i,j)\in O} Z_j\right)$$

□

We define these valid inequalities as NSRC, or "no-splitting rule cuts." We continue our previous example to illustrate the diversity of this class of cuts.

Example 3:
As shown in Figure 5, we can see that the three oldest clusters are sold at time period 2. This leads to the following inequality:

$$S_{5,2} + S_{7,2} + S_{8,2} - S_{1,3} - S_{2,4} - S_{3,5} - S_{4,6} - S_{5,7} - S_{6,8} \leq (n_3 + n_5 + n_6)(Z_9 + Z_{10}) = 14(Z_9 + Z_{10})$$

Notice that the above constraint merely combines two constraints that could have been written on Z_9 and Z_{10} individually. This can also be extended over two replacement cycles:

$$S_{5,2} + S_{7,2} + S_{8,2} - S_{1,3} - S_{2,4} - S_{3,5} - S_{4,6} - S_{5,7} - S_{6,8} -$$

$$S_{1,10} - S_{2,11} - S_{3,12} - S_{4,13} - S_{5,14} - S_{6,15} - S_{7,16} -$$

$$S_{1,11} - S_{2,12} - S_{3,13} - S_{4,14} - S_{5,15} - S_{6,16} \leq 14(Z_{17} + Z_{18})$$

The following constraint can also be added:

$$S_{7,6} + S_{8,6} - S_{1,7} - S_{2,8} - S_{3,9} - S_{4,10} - S_{5,11} - S_{6,12} - S_{7,13} \leq 17Z_{14}$$

Including these constraints increases the lower bound to 48,766 (0.435% improvement) and results in the linear programming relaxation solution partially depicted in Figure 6.

This leads to another valid inequality from this class, as the initial clusters are splitting. This only complicates the input flow to the super node. The following can be written:

$$S_{6,1} + S_{7,1} + S_{7,2} + S_{8,2} - S_{1,2} - S_{2,3} - S_{3,4} - S_{4,5} - S_{5,6} - S_{6,7} - S_{7,8} -$$

$$S_{1,3} - S_{2,4} - S_{3,5} - S_{4,6} - S_{5,7} - S_{6,8} \leq 12(Z_9 + Z_{10})$$

□

Similar constraints can be written for the other set of clusters and for extending these to two replacement cycles. We now illustrate our implementation in a branch-and-cut framework.

5 Branch and Cut Implementation

We have implemented a branch-and-cut framework to solve PRES using our two sets of valid inequalities, LTC and NSRC. For problems that are not excessively large in parameters N and T, the LTC can be included in the root node formulation. However, the NSRC cuts must be dynamically generated.

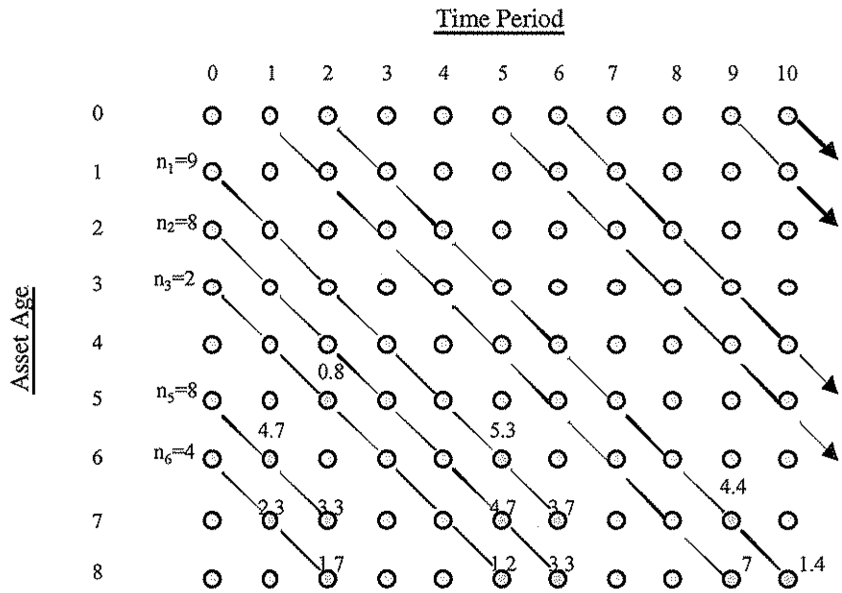

Figure 6: Solution network for first 10 periods of the continued example with LTC and NSRC.

We have performed numerous experiments, as there is clearly a tradeoff between the number of cuts generated at each node (in the branch-and-bound tree) and the entire solution time of the algorithm. The solutions are highly dynamic as changes in the solution in the first few periods of the horizon have a drastic effect on solutions in the latter portion of the horizon. This is due to the cyclic nature of the solution (as evident in the network). This fact, along with the fact that costs are discounted, motivate us to generate valid inequalities early in the horizon before generating cuts later in the horizon. Thus, in our implementation, we generate a single valid inequality, re-formulate the linear program, and solve it instead of generating multiple cuts in a single iteration. If no cuts can be generated after solution of the re-formulated linear program, we branch on the first non-integer valued Z variable in the horizon and repeat the cut generation procedure.

6 Computational Results

We follow the computational design of Chen [4]. Specifically, we generate data as follows: We use five different sizes for the horizon $T = 20, 40, 60, 80, 100$ and four sizes for the number of initial clusters, $g = 5, 10, 20, 30$. The physical life of the asset is given as

$$N = g + [0, \lambda(T - g)]^+$$

where parameter $\lambda \in 0.2, 0.6$. For problems with smaller λ, the physical life N is relatively small, compared with the solution horizon T, while for problems with larger λ, the physical life N is relatively large. Other parameters used to generate the problem data are given in Table 1. For each of the 20 possible combinations of g and T, 10 problems, five with $\lambda = 0.2$ and five with $\lambda = 0.6$, are randomly generated.

The branch-and-cut procedure was implemented on a laptop computer with a Pentium II and 768 MB of memory running Windows XP. The algorithm was implemented in C++ using Visual C++ 6.0. All linear

Table 1: Data generation for computational experiments.

Parameters	Data
Age of assets in cluster α_i	$U[0, N]^1$
Size of cluster n_i	$U[2, 10]$
Fixed cost K	$U[1000, 5000]$
Unit purchase cost p_j	$U[100, 500]$
Unit O&M cost c_{ij}	$\alpha e^{\beta j}$, where $\alpha = U[10, 50]$, $\beta = U[0.01, 01]$
Unit salvage value s_{ij}	$p_j e^{-\gamma j/\alpha}$, where $\gamma = U[1, 10]$

programs were solved with CPLEX 8.1 through the callable library.

Tables 2 and 3 summarize our computational results. Table 2 provides the average solution time, nodes evaluated in the branch-and-bound tree, and cuts generated, from the five randomly generated instances for each set of problem parameters when solving PRES using default CPLEX and when incorporating the LTC. On average, including the LTC reduces the computation time by 47% while reducing the number of nodes and cuts generated by over 76%. (The improvement in computation time was calculated as the difference between the solution times with and without LTC divided by the solution time without LTC.) For problems in which solving the MIP directly resulted in large run times (such as $k = 18$ and $k = 39$), the results are quite impressive as solution times were reduced by nearly 98%. Note that optimal solutions were found in all cases.

Table 3 provides the average results when solving PRES with both the LTC and NSRC. These results are compared with solving PRES directly and with just the LTC. These results illustrate the effectiveness of the LTC, as the solution times improved an average of only 4.57% over solving PRES with the LTC. However, combining LTC and NSRC results in solution time improvements by an average of nearly 50% when compared to solving PRES directly.

7 PRES under Non-Decreasing Demand

We have separated the general case of non-decreasing demand from constant demand because it can no longer be modeled as a fixed charge minimum cost flow problem. We introduce the following variable to allow for assets to be stored (As noted in Hartman [6] and Chand et al. [3], early purchases may be made in order to take advantage of economies of scale. See Hartman for more discussion.):

Y_{ij} = i-period old assets in storage from the end of period j to $j + 1$;

All costs and parameters remain as before with the exception that demand is periodic, or:

d_j = number of assets demanded from the end of period j to $j + 1$.

The inventory carrying cost for stored assets is defined as c'. Further, it is assumed that $\sum_i n_i \leq d_0$. The revised formulation follows:

Table 2: Comparisons with solving MIP with default CPLEX and with LTC.

Problem				MIP			MIP with LTC			Reduction with LTC		
k	λ	g	T	Nodes	Cuts	Time	Nodes	Cuts	Time	Nodes	Cuts	Time
1	0.2	5	20	9.60	18.60	0.24	8.20	5.20	0.22	14.6%	72.0%	5.9%
2	0.2	5	40	280.00	30.00	0.74	93.00	10.60	0.54	66.8%	64.7%	27.5%
3	0.2	5	60	267.60	30.00	1.42	111.20	12.20	0.98	58.4%	59.3%	30.7%
4	0.2	5	80	1,385.20	34.80	5.54	732.40	11.20	3.49	47.1%	67.8%	37.1%
5	0.2	5	100	353.60	25.20	4.37	74.40	18.00	2.90	79.0%	28.6%	33.6%
6	0.2	10	20	55.40	23.40	0.38	10.80	5.80	0.28	80.5%	75.2%	26.0%
7	0.2	10	40	373.20	37.80	1.02	85.80	8.60	0.60	77.0%	77.2%	41.1%
8	0.2	10	60	515.60	39.80	2.06	282.00	12.20	1.54	45.3%	69.3%	25.5%
9	0.2	10	80	1,895.60	40.60	8.60	891.20	14.40	5.43	53.0%	64.5%	36.8%
10	0.2	10	100	2,625.80	53.60	18.40	895.40	16.80	10.90	65.9%	68.7%	40.7%
11	0.2	20	20	64.60	16.60	1.10	3.80	3.80	0.28	94.1%	77.1%	74.1%
12	0.2	20	40	4,495.80	44.20	7.25	66.60	3.20	0.98	98.5%	92.8%	86.5%
13	0.2	20	60	11,572.00	51.80	32.17	273.80	9.20	2.33	97.6%	82.2%	92.8%
14	0.2	20	80	1,087.20	58.00	7.53	532.00	9.20	5.38	51.1%	84.1%	28.5%
15	0.2	20	100	9,343.80	56.60	51.34	2,082.80	12.60	23.04	77.7%	77.7%	55.1%
16	0.2	30	20	64.40	16.20	0.57	1.40	1.80	0.34	97.8%	88.9%	40.9%
17	0.2	30	40	20,367.40	44.88	47.83	26.60	2.80	0.84	99.9%	93.8%	98.2%
18	0.2	30	60	49,503.40	49.80	157.43	215.00	6.00	3.52	99.6%	88.0%	97.8%
19	0.2	30	80	216.60	45.80	3.88	13.40	2.80	2.03	93.8%	93.9%	47.8%
20	0.2	30	100	14,577.20	62.60	89.75	1,151.20	13.00	18.29	92.1%	79.2%	79.6%
21	0.6	5	20	18.80	14.40	0.37	5.60	3.80	0.32	70.2%	73.6%	12.0%
22	0.6	5	40	165.80	17.60	1.02	41.60	5.40	0.83	74.9%	69.3%	18.6%
23	0.6	5	60	467.00	19.40	2.86	135.60	6.20	1.60	71.0%	68.0%	44.1%
24	0.6	5	80	2,113.40	21.00	13.94	821.00	8.00	7.30	61.2%	61.9%	47.6%
25	0.6	5	100	2,690.80	25.40	31.06	699.00	6.20	13.94	74.0%	75.6%	55.1%
26	0.6	10	20	77.00	19.40	0.58	6.20	4.60	0.30	91.9%	76.3%	47.2%
27	0.6	10	40	206.60	22.20	0.94	28.60	4.80	0.67	86.2%	78.4%	28.8%
28	0.6	10	60	308.53	24.80	8.48	184.80	5.40	1.85	40.1%	78.2%	78.2%
29	0.6	10	80	1,268.80	24.60	11.58	562.40	8.60	8.06	55.7%	65.0%	30.4%
30	0.6	10	100	1,192.80	33.00	20.28	196.80	13.00	11.78	83.5%	60.6%	41.9%
31	0.6	20	20	63.80	17.00	0.70	5.00	3.80	0.58	92.2%	77.6%	18.0%
32	0.6	20	40	947.00	26.00	2.84	85.20	4.00	1.11	91.0%	84.6%	60.9%
33	0.6	20	60	1,076.80	31.40	5.76	198.20	5.00	2.84	81.6%	84.1%	50.7%
34	0.6	20	80	650.40	35.40	6.28	74.40	3.80	2.95	88.6%	89.3%	53.0%
35	0.6	20	100	791.40	37.40	21.50	281.80	9.40	15.67	64.4%	74.9%	27.1%
36	0.6	30	20	74.20	16.00	0.53	0.80	2.80	0.43	98.9%	82.5%	19.5%
37	0.6	30	40	389.20	25.20	1.79	7.80	1.00	0.72	98.0%	96.0%	59.6%
38	0.6	30	60	2,332.00	40.20	9.68	170.20	5.00	3.83	92.7%	87.6%	60.4%
39	0.6	30	80	3,889.40	40.40	302.19	477.20	7.20	8.90	87.7%	82.2%	97.1%
40	0.6	30	100	459.80	49.20	13.58	73.40	6.20	11.98	84.0%	87.4%	11.8%
			Avg:	3,455.94	33.01	22.44	290.17	7.34	4.49	76.9%	76.5%	46.7%

Table 3: Comparisons with solving MIP with all cuts versus default CPLEX with and without LTC.

Problem				MIP with LTC and NSRC			Reduction from MIP with LTC			Reduction from MIP		
k	λ	g	T	Nodes	Cuts	Time	Nodes	Cuts	Time	Nodes	Cuts	Time
1	0.2	5	20	2.60	5.60	0.25	68.3%	-7.7%	-13.5%	72.9%	69.9%	-6.8%
2	0.2	5	40	22.60	38.00	0.51	75.7%	-258.5%	5.2%	91.9%	-26.7%	31.3%
3	0.2	5	60	20.00	37.80	0.92	82.0%	-209.8%	6.9%	92.5%	-26.0%	35.5%
4	0.2	5	80	124.60	104.40	2.15	83.0%	-832.1%	38.2%	91.0%	-200.0%	61.1%
5	0.2	5	100	0.00	7.20	1.36	100%	60.0%	53.2%	100%	71.4%	68.9%
6	0.2	10	20	1.20	8.20	0.30	88.9%	-41.4%	-4.9%	97.8%	65.0%	22.4%
7	0.2	10	40	12.60	31.80	0.81	85.3%	-269.8%	-34.8%	96.6%	15.9%	20.6%
8	0.2	10	60	27.20	49.80	1.07	90.4%	-308.2%	30.2%	94.7%	-25.1%	48.0%
9	0.2	10	80	146.20	248.40	4.43	83.6%	-1625.0%	18.5%	92.3%	-511.8%	48.5%
10	0.2	10	100	118.41	222.80	6.91	86.8%	-1226.2%	36.6%	95.5%	-315.7%	62.4%
11	0.2	20	20	1.80	6.60	0.32	52.6%	-73.7%	-11.3%	97.2%	60.2%	71.2%
12	0.2	20	40	7.40	38.40	0.98	88.9%	-1100%	0.0%	99.8%	13.1%	86.5%
13	0.2	20	60	25.00	76.00	1.81	90.9%	-726.1%	22.4%	99.8%	-46.7%	94.4%
14	0.2	20	80	67.20	137.00	3.75	87.4%	-1389.1%	30.4%	93.8%	-136.2%	50.2%
15	0.2	20	100	143.20	407.80	14.88	93.1%	-3136.5%	35.4%	98.5%	-620.5%	71.0%
16	0.2	30	20	1.60	1.60	0.52	-14.3%	11.1%	-54.5%	97.5%	90.1%	8.7%
17	0.2	30	40	4.60	17.60	1.08	82.7%	-528.6%	-28.8%	100%	60.8%	97.7%
18	0.2	30	60	47.80	102.00	3.30	77.8%	-1600.0%	6.3%	99.9%	-104.8%	97.9%
19	0.2	30	80	13.40	19.40	2.33	0.0%	-592.9%	-14.6%	93.8%	57.6%	40.1%
20	0.2	30	100	202.00	244.40	16.93	82.5%	-1780.0%	7.5%	98.6%	-290.4%	81.1%
21	0.6	5	20	2.20	8.60	0.34	60.7%	-126.3%	-5.6%	88.3%	40.3%	7.1%
22	0.6	5	40	7.80	13.20	0.76	81.3%	-144.4%	8.2%	95.3%	25.0%	25.3%
23	0.6	5	60	28.80	34.40	1.42	78.8%	-454.8%	11.1%	93.8%	-77.3%	50.3%
24	0.6	5	80	229.80	129.00	5.01	72.0%	-1512.5%	31.4%	89.1%	-514.3%	64.0%
25	0.6	5	100	299.80	170.60	14.29	57.1%	-2651.6%	-2.5%	88.9%	-571.7%	54.0%
26	0.6	10	20	1.00	4.80	0.32	83.9%	-4.3%	-3.9%	98.7%	75.3%	45.2%
27	0.6	10	40	8.20	17.80	0.69	71.3%	-270.8%	-3.3%	96.0%	19.8%	26.5%
28	0.6	10	60	33.20	34.40	1.88	82.0%	-537.0%	-1.8%	89.2%	-38.7%	77.8%
29	0.6	10	80	100.60	106.40	4.44	82.1%	-1137.2%	44.9%	92.1%	-332.5%	61.6%
30	0.6	10	100	42.20	64.60	10.25	78.6%	-396.9%	13.0%	96.5%	-95.8%	49.5%
31	0.6	20	20	1.60	4.80	0.65	68.0%	-26.3%	-13.2%	97.5%	71.8%	7.1%
32	0.6	20	40	23.00	58.40	1.60	73.0%	-1360.0%	-43.6%	97.6%	-124.6%	43.8%
33	0.6	20	60	69.40	72.00	2.67	65.0%	-1340.0%	6.1%	93.6%	-129.3%	53.7%
34	0.6	20	80	17.80	31.40	2.85	76.1%	-726.3%	3.5%	97.3%	11.3%	54.7%
35	0.6	20	100	53.00	97.60	11.38	81.2%	-938.3%	27.4%	93.3%	-161.0%	47.1%
36	0.6	30	20	0.40	1.60	0.57	50.0%	42.9%	-33.7%	99.5%	90.0%	-7.6%
37	0.6	30	40	2.00	7.80	0.85	74.4%	-680.0%	-17.8%	99.5%	69.0%	52.4%
38	0.6	30	60	46.60	52.60	3.80	72.6%	-952.0%	0.9%	98.0%	-30.8%	60.8%
39	0.6	30	80	115.00	149.60	7.03	75.9%	-1977.8%	21.0%	97.0%	-270.3%	97.7%
40	0.6	30	100	15.80	34.00	10.48	78.5%	-448.4%	12.5%	96.6%	30.9%	22.8%
			Averages	52.19	72.46	3.65	73.7%	-781.9%	4.6%	95.0%	-92.8%	49.6%

$$\min_{X,Y,S,Z} \sum_{j=0}^{T-1} (p_j X_{0j} + k_j Z_j) + \sum_{j=0}^{T-1} \sum_{i=0}^{N-1} (c_{ij} X_{ij} + c'_{ij} Y_{ij}) - \sum_{j=0}^{T} \sum_{i=1}^{N} r_{ij} S_{ij} \tag{11}$$

subject to:

$$\sum_{i=0}^{N-1} X_{ij} \geq d_j \quad \forall j \in \{0, 1, \ldots, T = 1\} \tag{12}$$

$$X_{i0} + Y_{i0} + S_{i0} = n_i \quad \forall i \in \{1, 2, \ldots, N-1\} \tag{13}$$

$$\sum_{i=1}^{N} S_{ij} - X_{0j} - Y_{0j} \leq 0 \quad \forall j \in \{0, 1, \ldots, T-1\} \tag{14}$$

$$X_{(i-1)(j-1)} + Y_{(i-1)(j-1)} - X_{ij} - Y_{ij} - S_{ij} = 0 \quad \forall i \in \{1, 2, \ldots, N-1\},\ j \in \{1, 2, \ldots, T-1\} \tag{15}$$

$$X_{(i-1)(j-1)} + Y_{(i-1)(j-1)} - S_{ij} = 0\ \forall i = N,\ j \in \{1, 2, \ldots, T\}; i \in \{1, 2, \ldots, N\},\ j = T \tag{16}$$

$$X_{0j} + Y_{0j} \leq m_j Z_j \quad \forall j \in \{0, 1, \ldots, T-1\} \tag{17}$$

$$X_{ij}, Y_{ij}, S_{ij} \in \{0, 1, 2, \ldots\} \tag{18}$$

$$Z_j \in \{0, 1\} \tag{19}$$

This formulation differs from PRES through the inclusion of a demand constraint (12), as this is no longer a minimum cost flow formulation. This is because an increase in demand may lead to the purchase of more assets than salvaged in a given period, as noted in constraint (13). The storage variables Y_{ij} are added as a decision whenever a usage decision, X_{ij}, is made. Note that stored assets do not contribute to demand.

The inclusion of the storage variables adds $N \times T$ variables to the formulation when compared to the constant demand case. Additionally, there are T constraints for demand.

The remaining constraints and conditions are as before (with stored assets), although the maximum number of assets purchased is no longer d. The following theorem tightens the value of m_j such that it is equal to the greatest demand in any period from the current period to the end of the horizon. Assuming the solution is bounded, there is no advantage to purchasing more assets than can possibly be used in a period over the remaining horizon.

Theorem 4 *The inequality*

$$X_{0j} \leq m_j Z_j \quad \forall j \in \{0, 1, \ldots, T-1\}$$

is valid for PRES with:

$$m_j = \max_{k=j,j+1,\ldots,T-1} d_k \quad \forall j \in \{0, 1, \ldots, T-1\}, \tag{20}$$

if the problem is bounded.

Proof. As the problem is bounded, no asset can be purchased for profit. (Note that if this were possible, then the optimal solution would be to buy an infinite number of assets. See Hartman [6] for cost conditions to assure a bounded solution.) This is a proof by contradiction. Assume an optimal solution exists such that X_{0j} assets are purchased in some period j and

$$X_{0j} > d_k \quad \forall k \in \{j, j+1, \ldots, T-1\}.$$

Now, from this solution, construct a new solution such that:

$$X'_{0j} = \max_{k=j,j+1,\ldots,T-1} d_k \quad \forall k \in \{j, j+1, \ldots, T-1\}.$$

Note that this solution is still feasible as there is no period in which there are fewer than the necessary d_j assets. However, the cost of the new solution is lowered by the purchase price of an asset times the reduction in assets purchased. Thus, X_{0j} cannot be optimal. Furthermore, the maximum number of assets that would ever be purchased in a given period is defined by m_j in (20) and Constraint (17) is valid under this definition of m_j. □

8 Valid Inequalities

The previously defined inequalities LTC and NSRC, shown valid for PRES, are also valid for the case of non-decreasing demand. This stems directly from the fact that the no-splitting rule is also valid under non-decreasing demand (see Hartman [6]) and that the valid inequality (10) is defined as a less-than-or-equal-to constraint. As the flow into a supernode is defined by the initial clusters, the flow out of the supernode can only be greater than or equal to the inflow under non-decreasing demand. Thus, constraint (10) is valid. A similar argument holds for LTC. The proofs that LTC and NSRC are valid follow directly and are thus omitted.

9 Computational Results

We repeat the computational experiments for the non-decreasing demand case to illustrate their use. To generate the test problems, we follow the previous method except that demand in each period is generated as follows:

θ: probability of increasing demand over the previous period j;

ϕ_j : increase in demand in period j over period $j-1$.

The value of ϕ_j is given as follows:

$$\begin{aligned} \phi_j &= U\left[1, \rho d_0 + 1\right], \text{ if an increase in demand occurs in period } j, \\ &\quad\; 0, \text{ otherwise,} \end{aligned}$$

where ρ is a factor that decides the relative size of the increased demand. And the demands over the whole horizon are calculated as in the following formula:

$$d_j = d_{j-1} + \phi_j, \quad j = 1, 2, \ldots, T-1$$

We generate the testing problems as follows: We use two different sizes for probability of increasing demand over the previous period, $\theta = (0.2, 0.6)$ and factor $\rho = (0.01, 0.1)$, which indicate the distribution of increased demands over the whole horizon. We test on two problem sizes, $(g, T) = (10, 40), (20, 80)$ and fix the factor $\lambda = 0.2$. The other parameters were defined as previously.

For each data combination, we randomly generate five problems. The results are shown in Tables 4 and 5. The figures in the table represent the average of the results of the five generated data sets for each problem instance. The solution statistics (branch-and-bound nodes, cuts generated and solution time) are the same as in the constant demand case.

Table 4 shows the results of solving PRES under non-decreasing demand directly with and without use of the LTC. Inclusion of the LTC decreases the solution time an average of 7.8%. However, it requires a drastic increase in the number of nodes and cuts generated. While the overall statistics are not as impressive as the constant demand case (which is expected as the initial cluster values provide tighter bounds in the constant demand case), we are encouraged by the drastic decrease in solution time for the harder instances. For example, the four problem instances with $g = 30$ and $T = 100$ require, on average, 760.2, 331.9, 64.7, and 202.5 seconds, for solution of PRES directly. These respective times are decreased, on average, to 17.3, 16.0, 15.9, and 16.2 seconds with inclusion of the LTC.

Table 5 illustrates that the NSRC are much more effective (with respect to LTC) in the case of non-decreasing demand when compared to the constant demand case. Here, the solution time is reduced an average of 35.4% over solving PRES directly and 18.8% over solving PRES with LTC. In the four large instances noted earlier, solution with both NSRC and LTC results in average solution time reductions of 97.9%, 94.7%, 78.8%, and 93.0% when compared to solving PRES directly. The results, for both constant and non-decreasing demand, illustrate that the LTC and NSRC merit inclusion when solving the integer programming formulation of PRES.

10 Conclusions and Directions for Future Research

We have defined valid inequalities for an integer programming formulation for the parallel replacement problem with fixed and variable costs. The parallel replacement problem is concerned with the replacement schedule (periodic keep and replace decisions) for each individual asset in a group of assets that operate in parallel and are economically interdependent. Specifically, we examined the case where a fixed charge is incurred in each period when an asset is purchased assuming both constant and non-decreasing demand. The valid inequalities are motivated from the "no-splitting rule" from earlier research which states than an optimal solution exists such that assets of the same age in the same time period are either kept or replaced as a group. This motivated the development of the valid inequalities which utilize this fact when tightening the constraints used to enforce the fixed charge associated with asset purchases. The work was further motivated by illustrating the problem was NP-Hard which had not been shown previously in the literature.

Computational results show that the valid inequalities drastically improve the solution time, especially for large-scale instances. For the case of constant demand, the lower triangle cuts, or LTC, were shown to be highly effective while the no-splitting rule cuts, or NSRC, were shown to be more critical for the case of non-decreasing demand.

This paper has focused on the homogeneous asset case in that all assets are similar over time and there is only one type of asset available for purchase in each time period. It should be clear that the problem becomes more complicated in the heterogeneous asset case where multiple types of assets are available in each period for replacement over time. This is clearly a more realistic instance as manufacturers and service providers generally have a number of suppliers from which to choose their equipment. Current research is

Table 4: Comparisons with solving MIP with default CPLEX and with LTC for non-decreasing demand.

Problem					MIP			MIP with LTC			Reduction with LTC		
k	ρ	θ	g	T	Nodes	Cuts	Time	Nodes	Cuts	Time	Nodes	Cuts	Time
1	0.01	0.2	15	20	7.40	3.40	0.49	408.80	32.20	1.00	-5424%	-847.1%	-103.4%
2	0.01	0.2	15	60	241.20	49.60	5.01	8.20	101.80	2.49	96.6%	-105.2%	50.2%
3	0.01	0.2	15	100	507.60	123.40	27.02	8.60	169.40	12.60	98.3%	-37.3%	53.4%
4	0.01	0.2	30	20	-	0.80	0.57	124.80	21.20	1.01	–	-2550%	-76.1%
5	0.01	0.2	30	60	517.60	28.40	7.99	7.60	84.60	3.00	98.5%	-197.9%	62.5%
6	0.01	0.2	30	100	46,110.6	113.60	760.16	4.40	177.60	17.35	100%	-56.3%	97.7%
7	0.01	0.6	15	20	2.00	2.40	0.49	300.00	35.60	1.03	-14900%	-1383%	-110.0%
8	0.01	0.6	15	60	99.20	53.80	4.31	11.40	83.60	2.69	88.5%	-55.4%	37.6%
9	0.01	0.6	15	100	846.00	132.00	38.70	1.80	156.40	14.37	99.8%	-18.5%	62.9%
10	0.01	0.6	30	20	-	3.80	0.62	20.20	23.60	0.72	–	-521.1%	-16.1%
11	0.01	0.6	30	60	231.60	34.60	5.78	4.20	72.20	2.84	98.2%	-108.7%	50.8%
12	0.01	0.6	30	100	15,552.0	110.20	331.91	4.00	210.20	16.04	100%	-90.7%	95.2%
13	0.1	0.2	15	20	3.20	2.40	0.50	117.00	25.00	0.92	-3556%	-941.7%	-84.2%
14	0.1	0.2	15	60	181.60	38.20	4.28	442.00	77.60	6.41	-143.4%	-103.1%	-49.9%
15	0.1	0.2	15	100	457.60	116.80	31.84	6.60	188.80	16.73	98.6%	-61.6%	47.5%
16	0.1	0.2	30	20	19.80	23.20	0.86	-	-	0.58	100%	100%	32.6%
17	0.1	0.2	30	60	515.00	22.60	8.22	344.20	82.80	6.91	33.2%	-266.4%	16.0%
18	0.1	0.2	30	100	2,478.0	76.20	64.67	6.80	140.20	15.94	99.7%	-84.0%	75.3%
19	0.1	0.6	15	20	9.40	1.80	0.55	405.40	24.20	1.03	-4212%	-1244%	-86.4%
20	0.1	0.6	15	60	18.00	32.60	2.18	41.00	62.40	2.77	-127.8%	-91.4%	-26.9%
21	0.1	0.6	15	100	95.60	111.60	16.56	6.40	112.60	12.32	93.3%	-0.9%	25.6%
22	0.1	0.6	30	20	0.40	2.20	0.59	8.00	19.60	0.72	-1900%	-790.9%	-21.9%
23	0.1	0.6	30	60	16.00	14.00	2.93	15.20	53.40	4.04	5.0%	-281.4%	-37.8%
24	0.1	0.6	30	100	13,690.4	99.20	202.46	7.00	156.80	16.16	99.9%	-58.1%	92.0%
				Avg:	3,400.0	49.87	63.28	95.98	87.99	6.65	-1316%	-408.1%	7.8%

modifying the valid inequalities to this case. This is also expected to produce good results as the no-splitting rule is also valid in this situation.

References

[1] I. Barany, T.J. Van Roy, and L.A. Wolsey. Strong formulations for multi-item capacitated lot sizing. *Management Science*, 30(10):1255–1261, 1984.

[2] J.C. Bean, J.R. Lohmann, and R.L. Smith. A dynamic infinite horizon replacement economy decision model. *The Engineering Economist*, 30:99–120, 1985.

[3] S. Chand, T. McClurg, and J. Ward. A model for parallel machine replacment with capacity expansion. *European Journal of Operational Research*, 121:519–531, 2000.

[4] Z. Chen. Solution algorithms for the parallel replacement problem under economy of scale. *Naval Research Logistics*, 45:279–295, 1998.

[5] J.M. Fraser and J.W. Posey. A framework for replacement decisions. *European Journal of Operational Research*, 40:43–57, 1989.

[6] J.C. Hartman. The parallel replacement problem with demand and capital budgeting constraints. *Naval Research Logistics*, 47:40–56, 2000.

[7] J.C. Hartman and J.R. Lohmann. Multiple options in parallel replacement analysis: Buy, lease or rebuild. *The Engineering Economist*, 42(3):223–248, 1997.

[8] T. Hsu, K. Tsai, D. Wang, and D.T. Lee. Steiner problems on directed acyclic graphs. In *Lecture Notes in Computer Science, Volume 1090, Proceedings of the Second Annual International Conference on Computing and Combinatorics*, pages 21–30, London, 1996. Springer-Verlag.

[9] F.K. Hwang, D.S. Richards, and P. Winter. The steiner tree problem. Number 53 in Annals of Discrete Mathematics. North-Holland, New York, 1992.

[10] P.C. Jones, J.L. Zydiak, and W.J. Hopp. Parallel machine replacement. *Naval Research Logistics*, 38:351–365, 1991.

[11] T. McClurg and Chand. A parallel machine replacment model. *Naval Research Logistics*, 49(3):275–287, 2002.

[12] G.L. Nemhauser and J. Wolsey. *Integer and Combinatorial Optimization.* John Wiley and Sons, New York, 1988.

[13] S. Rajagopalan. Capacity expansion and equipment replacement: A unified approach. *Operations Research*, 46(6):846–857, 1998.

Table 5: Comparisons with solving MIP with default CPLEX and with NSRC and LTC for non-decreasing demand.

Problem					MIP with LTC and NSRC			Reduction from MIP with LTC			Reduction from MIP		
k	ρ	θ	g	T	Nodes	Cuts	Time	Nodes	Cuts	Time	Nodes	Cuts	Time
1	0.01	0.2	15	20	5.20	16.80	0.76	98.7%	47.8%	24.7%	29.7%	-394.1%	-53.1%
2	0.01	0.2	15	60	0.80	121.00	2.30	90.2%	-18.9%	7.7%	99.7%	-144.0%	54.0%
3	0.01	0.2	15	100	3.00	195.00	12.52	65.1%	-15.1%	0.6%	99.4%	-58.0%	53.7%
4	0.01	0.2	30	20	0.40	0.80	0.59	99.7%	96.2%	41.6%	–	0.0%	-2.8%
5	0.01	0.2	30	60	9.60	92.00	4.19	-26.3%	-8.7%	-39.7%	98.1%	-223.9%	47.6%
6	0.01	0.2	30	100	36.60	193.80	15.82	-731.8%	-9.1%	8.8%	99.9%	-70.6%	97.9%
7	0.01	0.6	15	20	1.40	6.00	0.50	99.5%	83.1%	51.2%	30.0%	-150.0%	-2.5%
8	0.01	0.6	15	60	-	119.60	1.89	100%	-43.1%	29.6%	100%	-122.3%	56.0%
9	0.01	0.6	15	100	-	207.80	12.06	100%	-32.9%	16.1%	100%	-57.4%	68.8%
10	0.01	0.6	30	20	1.40	7.60	0.62	93.1%	67.8%	14.4%	–	-100%	0.6%
11	0.01	0.6	30	60	2.60	84.40	2.43	38.1%	-16.9%	14.6%	98.9%	-143.9%	58.0%
12	0.01	0.6	30	100	6.80	228.60	17.39	-70.0%	-8.8%	-8.4%	100%	-107.4%	94.8%
13	0.1	0.2	15	20	0.80	7.60	0.48	99.3%	69.6%	47.9%	75.0%	-216.7%	4.0%
14	0.1	0.2	15	60	5.60	101.80	3.71	98.7%	-31.2%	42.2%	96.9%	-166.5%	13.3%
15	0.1	0.2	15	100	6.20	229.40	16.64	6.1%	-21.5%	0.5%	98.6%	-96.4%	47.7%
16	0.1	0.2	30	20	-	-	0.58	–	–	0.7%	100%	100%	33.1%
17	0.1	0.2	30	60	8.40	80.40	5.27	97.6%	2.9%	23.7%	98.4%	-255.8%	35.9%
18	0.1	0.2	30	100	3.60	170.20	13.71	47.1%	-21.4%	14.0%	99.9%	-123.4%	78.8%
19	0.1	0.6	15	20	1.80	10.80	0.50	99.6%	55.4%	51.2%	80.9%	-500.0%	9.0%
20	0.1	0.6	15	60	1.00	50.40	1.55	97.6%	19.2%	43.9%	94.4%	-54.6%	28.9%
21	0.1	0.6	15	100	1.00	149.60	11.30	84.4%	-32.9%	8.2%	99.0%	-34.1%	31.8%
22	0.1	0.6	30	20	-	3.60	0.66	100%	81.6%	7.8%	100%	-63.6%	-12.4%
23	0.1	0.6	30	60	1.80	31.40	2.56	88.2%	41.2%	36.6%	88.8%	-124.3%	12.6%
24	0.1	0.6	30	100	2.40	173.20	14.12	65.7%	-10.5%	12.6%	100%	-74.6%	93.0%
				Avg:	4.18	95.08	5.92	36.5%	12.8%	18.8%	90.3%	-132.6%	35.4%