Branch-and-Cut for Separable Piecewise Linear Optimization: Computation

I.R. de Farias JR.,* R. Gupta, E. Kozyreff
Department of Industrial Engineering, Texas Tech University
{ismael.de-farias, rajat.gupta, ernee.kozyreff}@ttu.edu

M. Zhao
SAS, ming.zhao@sas.com

March 13, 2011

Abstract

We report and analyze the results of our extensive computational testing of branch-and-cut for piecewise linear optimization using the cutting planes given recently by Zhao and de Farias. Besides analysis of the performance of the cuts, we also analyze the effect of formulation on the performance of branch-and-cut. Finally, we report and analyze initial results on piecewise linear optimization problems with semi-continuous constraints.

Keywords: piecewise linear optimization, mixed-integer programming, knapsack problem, special ordered set, semi-continuous variable, polyhedral method, branch-and-cut

*Corresponding author

1 Introduction

Let n be a positive integer and $N = \{1, \ldots, n\}$. The *separable piecewise linear optimization problem* (SPLO) is:

$$\begin{aligned} \textit{maximize} \quad & \textstyle\sum_{j \in N} g_j(x_j) \\ \textit{s.t.} \quad & Ax \leq b \\ & x_j \in [0, u_j], \quad j \in N, \end{aligned}$$

where g_j is a continuous piecewise linear function $\forall j \in N$. In a companion paper, Zhao and de Farias [12] studied the inequalities valid for the feasible set of a knapsack relaxation of SPLO, modeled through *special ordered sets of type 2* (SOS2). Note that such inequalities are valid for the feasible set of SPLO, regardless of whether SPLO is tackled through the SOS2 approach [1], the "usual" MIP approach [2], or the more recent "logarithmic" (LOG) approach [10, 11]. In this paper we report on our computational testing of the efficiency of the inequalities given in [12], when used as cuts in a branch-and-cut (B&C) scheme to solve SPLO to proven optimality. Additionally, we discuss efficiency issues related to the choice of formulation for SPLO. Finally, we report on our initial results on an extension of the cuts, given in [12], for SPLO with semi-continuous constraints [3, 4]. The notation, assumptions, and definitions used in this paper are the same as in [12].

Let T be the number of *linear segments* of g_j, which we assume WLOG to be the same $\forall j \in N$, and $K = \{1, \ldots, T\}$. Our polyhedron of interest is the *piecewise linear optimization (PLO) knapsack polytope* $P = conv(S)$, where $S = \{\lambda \in \Re^{nT} : \lambda$ satisfies (1) – (4) $\forall k \in K, j \in N\}$,

$$\sum_{k \in K} \lambda_j^k \leq 1, \tag{1}$$

$$\lambda_j^k \geq 0, \tag{2}$$

$$\sum_{j \in N^+} \sum_{k \in K} a_j^k \lambda_j^k - \sum_{j \in N^-} \sum_{k \in K} a_j^k \lambda_j^k \leq b, \tag{3}$$

$$\left\{\lambda_j^1, \ldots, \lambda_j^T\right\} \text{ is SOS2}' \tag{4}$$

(SOS2$'$ was defined in [12]), $N^+ \cap N^- = \emptyset$ and $N^+ \cup N^- = N$, and $a_j^0, \ldots, a_j^T$ are the *endpoints* or *breakpoints* of g_j $\forall j \in N$ (we assume WLOG they are translated so that $0 = a_j^0 < \cdots < a_j^T = u_j$). Of particular importance to us are Theorems 1, 2, 9, 10, and 11 of [12], and the inequalities they establish, (13), (19), (50), (51), and (58), which we henceforth refer to as (PLO1) – (PLO5).

As shown in Theorems 12 – 15 of [12], when the variables x_j, $j \in N$, are *semi-continuous* (see [3, 4]), it is possible to strengthen (PLO1) – (PLO4). We refer to (PLO1) – (PLO4) strengthened to take into account semi-continuous constraints as (SC-PLO1) – (SC-PLO4).

Here we show how to use Inequalities (PLO1) – (PLO5), as well as (SC-PLO1) – (SC-PLO4), as cuts (called PLO and SC-PLO, respectively) in a B&C scheme to solve SPLO, and we evaluate their impact on difficult and large instances of the transshipment and transportation problems with continuous separable concave piecewise linear costs. Also, we investigate the performance of the SOS2 approach and the LOG and "usual" MIP models for these problems.

The remainder of the paper is organized as follows. In Section 2 we describe the computer platform used in the tests, the problems tested, how the instances were generated, which data was collected, and the separation heuristics for the cuts. In Section 3 we give the results for the transshipment problem. In Section 4 we give the results for the transportation problem. In Section 5 we analyze the performance of the PLO cuts for large values of T. In Section 6 we give results on B&C with SC-PLO cuts for SPLO with semi-continuous constraints. Finally, in Section 7 we present conclusions and we give directions for further research.

2 Platform, Instances, Separation Heuristics, and Tests Conducted

We performed our computational tests in the Texas Tech High Performance Computing Center's Intel Xeon E5450 3.0GHz CPU with 16GB RAM nodes (two CPUs on a single board for each node) [8]. We used the callable libraries of CPLEX 12 and GUROBI 4. We ran both on a single thread, as multiple thread computational times on either software appeared random, and therefore are not reproducible. The computational times and number of branch-and-bound (B&B) nodes for CPLEX and GUROBI were different in all instances, but in almost all cases they led to the same conclusions. For this reason, we report the results of GUROBI only. In the small number of cases in which CPLEX and GUROBI disagree we report the results of both.

We tested B&C with (PLO1), (PLO2), and (PLO5) on difficult and large instances of the *transshipment problem*:

$$\begin{aligned} \textit{minimize} \quad & \textstyle\sum_{i\in M}\sum_{j\in M} f_{ij}(x_{ij}) \\ \textit{s.t.} \quad & \textstyle\sum_{j\in M}(x_{ij}-x_{ji}) = b_i, \quad i\in M \\ & x_{ij}\ge 0, \qquad i,j\in M, \end{aligned} \tag{5}$$

where M is the set of nodes of the network and f_{ij} is a continuous concave piecewise linear function $\forall i, j \in M$. We assume WLOG that $\sum_{i\in M} b_i = 0$. We also tested (PLO1) – (PLO4) on large instances of the *transportation problem*:

$$\textit{minimize} \quad \textstyle\sum_{i\in I}\sum_{j\in J} f_{ij}(x_{ij})$$

$$\textit{s.t.} \quad \textstyle\sum_{i\in M} x_{ij} = s_i, \qquad i\in I \tag{6}$$

$$\textstyle\sum_{j\in M} x_{ij} = d_j, \qquad j\in J \tag{7}$$

$$x_{ij}\ge 0, \qquad i,j\in M,$$

where I is the set of supply nodes, J the set of demand nodes, s_i the supply of node i, and d_j the demand of node j. We assume WLOG that $\sum_{i\in I} s_i = \sum_{j\in J} d_j$. Additionally, we tested the largest instances of Vielma and Nemhauser [11], which are transportation instances, but much smaller than ours. Finally, we tested the SC-PLO cuts on instances of the transportation problem with semi-continuous constraints.

Our instances were generated in the same way as in [9], with the following exceptions. First, we tested a much wider range of values of T. Specifically, for transshipment T varied between 3 and 20; for transportation, between 5 and 200; and for transportation with semi-continuous constraints, $T \in \{5, 10\}$. Second, to improve the chances that $\sum_{i \in I} s_i = \sum_{j \in J} d_j$ (thus facilitating the generation of instances), the values of s_i ranged between $T+1$ and $2(T+10)|J|/|I| - T - 1$, rather than $T+1$ and $T+20|J|/|I|$ as in [9]. As for transportation with semi-continuous constraints, we imposed $x_{ij} \in \{0\} \cup \left[a_{ij}^1, a_{ij}^T\right]$, or equivalently $\lambda_{ij}^1 \in \{0, 1\}$, $\forall i \in I, j \in J$ ($a_{ij}^0, \ldots, a_{ij}^T$ are the breakpoints of f_{ij} $\forall i \in I, j \in J$).

The separation of both PLO and SC-PLO cuts seems to be difficult, so we adopted simple heuristics for them. Because the separation of SC-PLO cuts is done similarly to that of PLO cuts, we comment only on the second. For cut generation, we considered all constraints (5) for the transshipment instances, and (6) and (7) for the transportation instances.

Let $\bar{\lambda}$ be the optimal LP relaxation solution just obtained. The heuristics we now give are not specific for transportation or transshipment, but for SPLO in general. Let $j \in N$ and suppose that $\bar{\lambda}_j^k > 0$ for some $k \in K$. We let

$$u_j = min\left\{k \in K : \bar{\lambda}_j^k > 0\right\}$$

and

$$v_j = max\left\{k \in K : \bar{\lambda}_j^k > 0\right\}.$$

Regarding inequality (PLO1), when $N^- \neq \emptyset$, we take $N_1^- = \emptyset$. So the inequality is determined when j and s are chosen. For each $j \in N$ with $\bar{\lambda}_j^k > 0$ for some $k \in K$, if $v_j - u_j \geq 2$ we test whether (PLO1) with $s = u_j$ is violated by $\bar{\lambda}$, and if yes we add (PLO1) to the cutpool. If $u_j > 1$, we also test for $s = 1$, and if $\bar{\lambda}$ violates (PLO1) we add (PLO1) (with $s = 1$) to the cutpool. If $u_j > 1$ and $\sum_{k \in K} \bar{\lambda}_j^k < 1$, we test whether (PLO1) with $s = u_j$ is violated by $\bar{\lambda}$, and if yes we add (PLO1) to the cutpool.

Regarding inequality (PLO2), we also take $N_1^- = \emptyset$ when $N^- \neq \emptyset$. For each $j \in N$ with $\bar{\lambda}_j^k > 0$ for some $k \in K$, if $v_j - u_j \geq 2$ and $u_j \geq 2$, we test whether (PLO2) with $s = u_j$ is violated by $\bar{\lambda}$, and if yes we add (PLO2) to the cutpool.

Regarding inequalities (PLO3), (PLO4), and (PLO5), we use the same simple approach that is used in [9] for separating generalized cover inequalities. Specifically, all l_j's are chosen as in [9]; C in (PLO3) and (PLO4) is chosen as C^+ is in [9]; C^+ and C^- in (PLO5) are chosen as in [9]. Finally, in (PLO3) and (PLO5),

$$C_1 = \left\{ j \in C : l_j \geq 3 \text{ and } a_j^{l_j - 1} > b - \sum_{i \in C - \{j\}} a_i^l \right\}.$$

We evaluate the performance of GUROBI on the piecewise linear transshipment and transportation instances with and without the addition of PLO cuts. In either case, we examine how much preprocessing, primal heuristics, and cutting planes influence the number of enumeration nodes and computational time. In other words, we evaluate GUROBI with and without PLO cuts, and:

- preprocessing, primal heuristics, and GUROBI cuts off, which we call B&B setting

- precisely one of preprocessing, primal heuristics, or GUROBI cuts on (and the other two off)
- default setting.

Depending on the results, we performed additional tests. For example, when GUROBI cuts appeared to be relevant, we tried to identify which ones are the most relevant. Also, we minded the GUROBI parameters. The most important example is the addition of user cuts to the cutpool. GUROBI filters which user cuts are added to the cutpool, and as a result sometimes only a small fraction of them are actually used. We tuned parameters to see when using more cuts than allowed by default would lead to a better performance. All parameter tuning will be explicitly mentioned.

To identify which formulation performs best, we conducted all tests above with the LOG, MIP, and SOS2 formulations. Finally, we note that the difference between the optimal value of all instances and the optimal value of their LP relaxations is extremely small. This is typical of the problems we tested, see for example [7, 9]. For this reason, we do not report the optimal values or the integrality gaps of the instances.

3 Computational Results for Transshipment Instances

We tested 85 piecewise linear transshipment instances. The instances can be grouped into 17 sets with same number of nodes, which we denote as η, and same value of T. Because preliminary tests indicated these instances to be extremely difficult, we allowed for a maximum computational time of 7,200 seconds of CPU. Here, in most cases, adopting the "Aggressive User Cuts" option when separating PLO cuts led to a better performance than default or the "Very Aggressive User Cuts" option. Therefore, we report on this option. (For the remainder of this section, default refers to GUROBI's default setting, except for "Aggressive User Cuts" on).

In Tables 1, 2, and 3 we compare the average solution time and the number of instances solved to proven optimality, and in Tables 4, 5, and 6 the average number of enumeration nodes, for GUROBI in: default setting; B&B; default with PLO cuts (PLO1), (PLO2), and (PLO5) added; and B&B with (PLO1), (PLO2), and (PLO5) added. In Tables 7, 8, and 9 we give the average number of cuts (PLO1), (PLO2), and (PLO5) separated and added. In Tables 10 and 11 we give the average number of MIP cuts used by GUROBI in the LOG and MIP formulations, respectively.

In all tables each entry refers to 5 instances. Also, we computed both arithmetic and geometric averages, but because they led to the same conclusions, we report only the arithmetic averages. Tables 1, 4, 7, and 10 give results for the LOG formulation, Tables 2, 5, 8, and 11 for the MIP formulation, and Tables 3, 6, and 9 for the SOS2 approach. In Tables 1 – 6, "% reduction" is relative to the default setting. In Tables 7 – 9, the column "Added" gives the average number of PLO cuts that passed GUROBI's user cuts filter. In Tables 10 and 11, IB is implied bounds, Cv is cover, FC is flow cover, and G is Gomory. These and MIR are the cuts that GUROBI separated for the LOG and MIP models.

Note that, even on "Aggressive User Cuts" mode, GUROBI added to the cutpool only 13% of the PLO cuts separated for LOG in default setting and 6% in B&B; 8% for MIP in default setting and 7% in B&B; 9% for SOS2 in default setting and 7% in B&B. As CPLEX does not filter user cuts, it allowed all PLO cuts separated. As expected, the percentage reduction in the number of enumeration nodes due to PLO cuts was greater with CPLEX than GUROBI. However, the percentage reduction in computational time was greater with GUROBI. Obviously several factors may be involved here. But because the percentage of PLO cuts allowed by GUROBI was so small, we conjecture that its filter was efficient and we suggest researching further how to filter PLO cuts efficiently.

The impact of PLO cuts both on computational time and number of enumeration nodes was enormous. Out of the 85 instances tested, GUROBI in default setting was able to solve to proven optimality only 44 instances (51%), as opposed to 74 (87%) with PLO cuts, using the LOG model; 29 (34%), as opposed to 69 (81%), using the MIP model; and 35 (41%), as opposed to 72 (84%), using the SOS2 approach. The average reduction from default in computational time by adding PLO cuts was 62%, with the LOG model; 60%, with the MIP model; and 59%, with the SOS2 approach. The average reduction from default in the number of enumeration nodes by adding PLO cuts was 98%, with the LOG and MIP models, and 99% with the SOS2 approach. In summary, the use of PLO cuts improved our ability to solve the instances tremendously, not only by reducing their computational time and number of enumeration nodes, but also by solving many more instances that GUROBI could not solve without them.

We now turn our attention to the impact of the branch-and-cut features present in GUROBI default (preprocessing, primal heuristics, and MIP cutting planes) on the number of enumeration nodes and computational time for the instances. First we note that of these features, the only ones present in the SOS2 approach of GUROBI are preprocessing and primal heuristics.

The number of nodes and computational time for LOG, MIP, and SOS2 when only preprocessing is on or when only primal heuristics is on is almost the same as B&B. So, of these alternatives, we only report the results for B&B. On the other hand, the results for when only the MIP cuts are on (for the LOG and MIP formulations) are almost the same as default. So of these two, we only report the results for default. Thus, we compare the performance of default with B&B, first without the addition of PLO cuts. We note that in the case of the SOS2 approach, the performance of default and B&B are not significantly different, which is not surprising in the light of the previous discussion on the efficiency of GUROBI's pre-processing and primal heuristics for SPLO.

The default setting reduced the number of enumeration nodes of B&B by 64% for LOG and 63% for MIP. As mentioned above, this reduction is mostly due to MIP cutting planes. For the LOG model, the vast majority of cuts were Gomory, specifically 90% of the MIP cuts. For the MIP model, flow cover was the most used, 80% total, while Gomory was a distant second place, 16%. This is to be expected, since the MIP formulation exposes the flow cover structure explicitly in the model, which is not the case with the LOG formulation.

Despite the significant reduction in number of nodes, the use of MIP cuts in GUROBI default increased the computational time. At first, the average increase seems modest, 0.2% for LOG and 3% for MIP. However, if we average the percentage reduction in time for B&B over default for the instance sizes where either one of default or B&B was solved to proven optimality (first 13 entries of Table 1 and first 10 entries of Table 2) we obtain different results, 16% for LOG and 6% for MIP. This shows that the increase is significant for LOG and higher than MIP's. We also note that for LOG, in a few cases the increase was almost 50% (entries 15.25, 20.20, 25.15, and 30.6 of Table 1), while for MIP the individual increases were smaller. A possible explanation for this is the large presence of Gomory cuts. Because they are dense, when they are used in large quantities, typically a reduction of 60% in the number of enumeration nodes does not decrease computational time. Note that many more Gomory cuts were added for LOG than for MIP.

This computational time result is in agreement with tests conducted previously, see for example [6, 9], on platforms that are different. There, as well as here, the use of MIP tools for SPLO (and other combinatorial problems of similar type), did not improve the computational time required to solve the problem, and in some cases made it worse.

Now we compare the performance of default and B&B with PLO cuts added to both. As in the previous case, their performances are not too different for the SOS2 approach. On the other hand, the default setting reduced the number of enumeration nodes of B&B by 54% for LOG and 36% for MIP, and it reduced the computational time by 13% for LOG and 11% for MIP. So even though B&B was better than default when the PLO cuts were not added, in our tests the best performance overall was obtained when the PLO cuts were added on the top of the default setting.

We now discuss the effect of formulation on the computational results, first without the addition of the PLO cuts. Of the three approaches, LOG was the most efficient, followed by SOS2, followed by MIP. The LOG formulation solved to proven optimality 44 instances in default and 47 in B&B, as opposed to 35 in default and 36 in B&B for SOS2, and 29 in default and 31 in B&B for MIP. In number of nodes, LOG reduced the number of SOS2 nodes by 93% in default and by 81% in B&B, and the number of MIP nodes by 46% in default and 45% in B&B. In computational time, LOG reduced the computational time of SOS2 by 15% in both default and B&B, and the computational time of MIP by 24% in default and 22% in B&B.

The superiority of SOS2 over MIP (even when equipped with cutting planes, primal heuristic, and pre-processing) had already been noted and analyzed in the literature, see for example [5, 6, 9]. The superiority of LOG over SOS2 was noted in [10, 11] when T (the number of partitions) is large, but it was not analyzed. Here again LOG is superior to SOS2. It remains to determine why. The first explanation is that LOG allows for the use of all MIP features present in GUROBI, especially MIP cutting planes. But in the present tests, B&B was superior to default for LOG, so that cannot be the reason. Another possibility is GUROBI's implementation of SOS2 branching being inferior to its 0-1 variable dichotomy branching implementation. As we will see in Section 5, different implementations of SOS2 branching can lead to much difference in performance. One other possibility is that the LOG formulation may be breaking some of the symmetry inherent in transshipment with SOS2. We tried to explore further this possibility by repeating all tests with GUROBI in a mode where symmetry is explored aggressively. However, no improvement was detected (same with CPLEX). We note that in case LOG is exploring symmetry, a different branching scheme for SOS2 may eliminate the advantage of LOG and lead to an improved SOS2 approach. We will leave this issue as a topic for further research.

With PLO cuts added, again LOG was superior to SOS2, which was superior to MIP. The LOG formulation solved to proven optimality 74 instances in default and B&B, as opposed to 68 in default and B&B for SOS2, and 69 in default and 70 in B&B for MIP. In number of nodes, LOG reduced the number of SOS2 nodes by 71% in default and by 37% in B&B, and the number of MIP nodes by 45% in default and 25% in B&B. In computational time, LOG reduced the computational time of SOS2 by 22% in default and 4% in B&B, and the computational time of MIP by 29% in default and 9% in B&B. Here again, we tested GUROBI in a mode that exlores symmetry aggressively, but did not obtain different results (same with CPLEX). Overall, the best setting was LOG in default with PLO cuts.

4 Computational Results for Transportation Instances

We tested 180 piecewise linear transportation instances. The instances can be grouped into 3 sets with same value of T, each containing 12 sets with same number of supply (denoted as s) and

Table 1: Average solution time and # of instances solved for transshipment with the LOG model

	Default		**B&B**			**Def. + PLO cuts**			**B&B + PLO cuts**		
$\eta.T$	Time	#Sol.	Time	%Red.	#Sol.	Time	%Red.	#Sol.	Time	%Red.	#Sol.
15.20	19	5	16	15.78	5	22	-15.78	5	34	-78.9	5
15.25	49	5	26	46.93	5	23	53.06	5	32	34.69	5
15.30	26	5	17	34.61	5	46	-76.92	5	43	-65.38	5
20.20	208	5	117	43.75	5	44	78.84	5	39	81.25	5
25.15	1,617	4	853	47.24	5	134	91.71	5	142	91.21	5
25.20	2,377	4	2,426	-2.06	4	194	91.83	5	207	91.29	5
30.6	1,854	5	1,074	42.07	5	81	95.63	5	59	96.81	5
30.10	3,287	3	2,844	13.47	4	119	96.37	5	134	95.92	5
30.15	3,325	3	3,142	5.5	3	299	91.01	5	358	89.23	5
40.10	5,089	2	5,384	-5.79	2	525	89.68	5	675	86.7	5
50.6	7,200	0	7,078	1.69	4	872	87.88	5	1,733	75.93	4
60.4	6,685	1	7,200	-7.7	0	707	89.42	5	1,134	83.03	5
70.3	5,772	2	7,200	-24.74	0	289	94.99	5	291	94.9	5
70.5	7,200	0	7,200	0.0	0	3,132	56.5	4	3,419	52.51	3
80.5	7,200	0	7,200	0.0	0	4,949	31.26	4	6,078	15.83	1
90.5	7,200	0	7,200	0.0	0	6,160	14.44	1	6,766	6.02	1
100.5	7,200	0	7,200	0.0	0	7,200	0.0	0	7,200	0.0	0

Table 2: Average solution time and # of instances solved for transshipment with the MIP model

	Default		**B&B**			**Def. + PLO cuts**			**B&B + PLO cuts**		
$\eta.T$	Time	#Sol.	Time	%Red.	#Sol.	Time	%Red.	#Sol.	Time	%Red.	#Sol.
15.20	126	5	139	-10.31	5	65	48.41	5	39	69.04	5
15.25	257	5	261	-1.55	5	106	58.75	5	59	106.0	5
15.30	1,435	5	864	39.79	5	147	89.75	5	114	92.05	5
20.20	720	5	814	-13.05	5	93	87.08	5	64	91.11	5
25.15	5,651	2	3,933	30.4	3	578	89.77	5	375	93.36	5
25.20	4,977	2	4,921	1.125	2	564	88.66	5	417	91.62	5
30.6	5,107	2	4,539	11.12	2	198	96.12	5	98	98.08	5
30.10	5,611	2	5,770	-2.83	2	335	94.02	5	282	94.97	5
30.15	5,874	1	5,889	-0.25	1	2,121	63.89	5	1,042	82.26	5
40.10	7,200	0	6,887	4.34	1	2,550	64.58	5	1,309	81.81	5
50.6	7,200	0	7,200	0.0	0	2,167	69.9	4	1,978	75.52	4
60.4	7,200	0	7,200	0.0	0	1,530	78.75	5	1,137	81.43	5
70.3	7,200	0	7,200	0.0	0	485	93.26	5	452	93.72	5
70.5	7,200	0	7,200	0.0	0	3,638	49.47	3	3,568	50.44	3
80.5	7,200	0	7,200	0.0	0	6,130	14.86	1	5,908	17.94	2
90.5	7,200	0	7,200	0.0	0	6,877	4.486	1	6,951	3.45	1
100.5	7,200	0	7,200	0.0	0	7,200	0.0	0	7,200	0.0	0

Table 3: Average solution time and # of instances solved for transshipment with the SOS2 approach

	Default		**B&B**			**Def. + PLO cuts**			**B&B + PLO cuts**		
$\eta.T$	Time	#Sol.	Time	%Red.	#Sol.	Time	%Red.	#Sol.	Time	%Red.	#Sol.
15.20	171	5	154	9.94	5	38	77.78	5	30	82.45	5
15.25	122	5	130	-6.55	5	44	63.93	5	41	66.39	5
15.30	198	5	174	12.12	5	112	43.43	5	65	67.17	5
20.20	333	5	323	3.0	5	73	78.07	5	43	87.08	5
25.15	2,185	4	2,367	-8.32	4	267	87.78	5	186	91.48	5
25.20	4,529	2	4,510	0.41	2	323	92.86	5	287	93.66	5
30.6	3,220	4	3,201	0.59	5	65	97.98	5	47	98.54	5
30.10	5,162	2	5,226	-1.23	2	131	97.46	5	104	97.98	5
30.15	5,368	2	5,577	3.89	2	652	87.85	5	562	89.53	5
40.10	6,007	1	5,993	0.23	1	813	86.46	5	765	87.26	5
50.6	7,200	0	7,200	0.0	0	1,855	74.23	4	1,779	75.29	4
60.4	7,200	0	7,200	0.0	0	2,269	68.48	5	1,627	77.4	5
70.3	7,200	0	7,200	0.0	0	623	91.34	5	494	93.14	5
70.5	7,200	0	7,200	0.0	0	3,962	44.97	3	3,652	49.28	3
80.5	7,200	0	7,200	0.0	0	6,072	15.67	1	6,091	15.4	1
90.5	7,200	0	7,200	0.0	0	7,200	0.0	0	7,200	0.0	0
100.5	7,200	0	7,200	0.0	0	7,200	0.0	0	7,200	0.0	0

Table 4: Average number of enumeration nodes for transshipment with the LOG model

	Default	B&B		Def. + PLO cuts		B&B + PLO cuts	
$\eta.T$	Nodes	Nodes	%Red.	Nodes	%Red.	Nodes	%Red.
15.20	9,972	11,705	-17.37	546	94.52	952	90.45
15.25	12,267	16,834	-37.22	342	97.21	660	94.61
15.30	6,310	8,040	-27.41	652	89.66	741	88.25
20.20	29,023	42,652	-46.95	438	98.49	606	97.91
25.15	168,839	249,014	-47.48	1,204	99.28	1,621	99.03
25.20	144,815	449,428	-210.34	1,222	99.15	1,689	98.83
30.6	241,763	451,226	-86.63	1,592	99.34	2,038	99.15
30.10	222,788	642,184	-188.24	1,032	99.53	1,775	99.20
30.15	230,829	517,038	-123.99	2,171	99.05	3,269	98.58
40.10	197,821	613,385	-210.07	2,790	98.58	4,542	97.70
50.6	268,889	740,425	-175.36	4,360	98.37	12,436	95.37
60.4	321,817	1,253,942	-289.60	5,322	98.34	13,643	95.76
70.3	235,155	753,397	-220.38	1,839	99.21	4,384	98.13
70.5	126,790	407,905	-221.71	6,705	94.71	13,159	89.62
80.5	83,305	296,901	-256.40	6,601	92.07	16,559	80.12
90.5	62,540	223,295	-257.04	5,532	91.15	12,472	80.05
100.5	42,946	161,051	-275.00	2,885	93.28	6,763	84.25

Table 5: Average number of enumeration nodes for transshipment with the MIP model

	Default	B&B		Def. + PLO cuts		B&B + PLO cuts	
$\eta.T$	Nodes	Nodes	%Red.	Nodes	%Red.	Nodes	%Red.
15.20	70,452	89,675	-27.28	1,510	97.85	1,302	98.15
15.25	109,275	127,857	-17.00	1,701	98.44	1,324	98.78
15.30	470,910	360,488	23.44	2,822	99.4	2,385	99.49
20.20	194,841	237,535	-21.91	1,100	99.43	1,135	99.41
25.15	664,683	923,245	-38.9	5,107	99.23	5,346	99.19
25.20	485,813	822,335	-69.26	4,350	99.1	3,652	99.24
30.6	621,309	2,052,059	-230.27	3,920	99.36	3,346	99.46
30.10	715,605	1,312,370	-83.39	3,486	99.51	5,093	99.28
30.15	287,183	859,853	-199.4	17,696	93.83	11,628	95.95
40.10	291,834	754,546	-158.55	14,219	95.12	10,528	96.39
50.6	165,811	1,144,273	-590.1	6,608	96.01	14,357	91.34
60.4	128,332	1,086,020	-746.25	5,219	95.93	14,049	89.05
70.3	106,304	928,584	-773.51	1,657	98.44	6,255	94.11
70.5	64,967	642,132	-888.39	4,536	93.01	14,695	77.38
80.5	38,637	465,171	-1,103.90	4,137	89.29	16,002	58.58
90.5	28,965	341,841	-1,080.18	3,363	88.38	12,302	57.52
100.5	19,351	226,762	-1,071.83	1,292	93.32	6,491	66.45

Table 6: Average number of enumeration nodes for transshipment with the SOS2 approach

	Default	**B&B**		**Def. + PLO cuts**		**B&B + PLO cuts**	
$\eta.T$	Nodes	Nodes	%Red.	Nodes	%Red.	Nodes	%Red.
15.20	352,673	325,453	7.71	1,122	99.95	1,194	99.66
15.25	197,901	216,402	-9.34	925	99.53	1,107	99.44
15.30	280,909	254,896	9.26	2,182	99.22	1,622	99.42
20.20	396,332	394,383	0.49	955	99.75	847	99.78
25.15	2,112,977	2,354,152	-11.41	3,461	99.83	3,009	99.85
25.20	3,248,977	3,346,952	-3.01	2,515	99.92	2,965	99.90
30.6	4,538,896	4,698,361	-3.51	2,854	99.93	2,387	99.94
30.10	4,847,685	5,065,709	-4.49	2,015	99.95	2,228	99.95
30.15	3,324,306	3,618,237	-8.84	6,412	99.80	6,399	99.80
40.10	2,734,485	2,821,902	-3.19	6,327	99.76	6,587	99.75
50.6	3,156,942	3,307,116	-4.75	15,344	99.51	17,387	99.44
60.4	2,837,364	2,914,406	-2.71	31,410	98.89	26,271	99.99
70.3	2,502,456	2,574,267	-2.86	13,892	99.44	11,756	99.53
70.5	1,410,147	1,443,881	-2.39	21,143	98.50	20,721	98.53
80.5	939,498	971,422	-3.39	21,715	97.68	22,887	97.56
90.5	702,864	719,886	-2.42	16,957	97.58	17,726	97.47
100.5	522,391	535,181	-2.44	8,608	98.35	9,125	98.25

demand (denoted as d) nodes. The maximum computational time allowed for each instance is of 3,600 seconds of CPU. Here, unlike transshipment, adopting the "Aggressive User Cuts" or the "Very Aggressive User Cuts" option led to inferior performance in most cases. So we use GUROBI's default filter for user cut generation.

Again unlike transshipment, default performed better than B&B for the LOG and MIP models, and almost the same for the SOS2 approach. For the LOG formulation, default solved to proven optimality 94 instances (52%), as opposed to 39 (22%) by B&B when no PLO cuts were added, and 115 (64%), as opposed to 95 (53%) when the PLO cuts were added; in average, default reduced the number of nodes of B&B by 74% and the computational time by 28% when no PLO cuts were added, and the number of nodes by 45% and the computational time by 20% when the PLO cuts were added. For the MIP formulation, default solved to proven optimality 91 instances (51%), as opposed to 30 (17%) by B&B when no PLO cuts were added, and 108 (60%), as opposed to 80 (44%) by B&B when the PLO cuts were added; in average, default reduced the number of nodes of B&B by 55% and the computational time by 32% when no PLO cuts were added, and the number of nodes by 14% and the computational time by 26% when the PLO cuts were added. So we only report results for default. The better performance of default over B&B is due almost entirely to the MIP cutting planes. In the aforementioned studies [6, 9], MIP cuts were not efficient for transportation. Possibly, improvements in the strategies and implementations for MIP cutting planes made them more efficient. In any case, it is an interesting question, which we leave for further research, when MIP cutting planes are efficient for SPLO (as in the present case) and when they are not (as in the case of transshipment).

Table 7: Average # of PLO cuts for transshipment with the LOG model

	Default + PLO cuts					**B&B + PLO cuts**				
$\eta.T$	(PLO1)	(PLO2)	(PLO5)	Total	Added	(PLO1)	(PLO2)	(PLO5)	Total	Added
15.20	341	1,460	1,856	3,656	529	751	3,314	4,150	8,215	526
15.25	391	1,967	1,755	4,113	384	785	4,588	3,517	8,890	469
15.30	641	2,848	5,025	8,514	513	918	5,288	6,501	12,708	615
20.20	402	2,662	1,772	4,836	380	533	4,747	2,818	8,097	570
25.15	561	4,348	4,018	8,927	662	855	8,672	7,515	17,042	652
25.20	831	6,629	4,327	11,786	568	1,745	12,522	8,329	22,595	549
30.6	230	1,312	937	2,479	531	366	2,448	1,407	4,220	587
30.10	328	2,820	1,767	4,916	580	540	5,263	3,069	8,872	672
30.15	2,154	7,425	7,074	16,653	739	3,254	10,933	9,663	23,850	957
40.10	621	6,699	3,396	10,715	829	1,768	12,259	6,838	20,865	1,024
50.6	363	4,681	2,425	7,468	834	599	7,720	3,263	11,582	1,162
60.4	288	3,122	1,633	5,043	784	454	5,732	2,255	8,440	998
70.3	208	1,308	1,205	2,721	549	316	2,416	1,670	4,432	694
70.5	388	6,758	2,964	10,110	10,021	678	12,465	4,026	17,169	1,456
80.5	524	8,985	4,539	14,049	1,084	897	15,063	5,985	21,945	1,913
90.5	602	12,161	5,508	18,271	1,341	1,047	21,617	6,757	29,421	2,151
100.5	728	18,294	7,640	26,662	1,565	1,273	29,975	9,192	40,440	2,381

Table 8: Average # of PLO cuts for transshipment with the MIP model

	Default + PLO cuts					**B&B + PLO cuts**				
$\eta.T$	(PLO1)	(PLO2)	(PLO5)	Total	Added	(PLO1)	(PLO2)	(PLO5)	Total	Added
15.20	513	1,554	2,918	4,986	751	560	2,243	3,224	6,227	761
15.25	848	2,964	4,276	8,088	583	997	3,899	3,954	8,850	712
15.30	1,466	3,610	7,554	12,630	703	1,586	5,862	8,665	16,113	700
20.20	565	2,519	2,657	5,742	537	622	4,035	2,711	7,368	742
25.15	1,274	5,404	8,251	14,929	1,108	1,522	9,270	9,145	19,937	972
25.20	1,458	6,289	5,615	13,363	950	2,244	11,736	7,385	21,365	882
30.6	255	1,399	895	2,548	565	406	2,617	1,506	4,529	587
30.10	486	3,119	2,193	5,798	748	724	5,275	3,240	9,239	896
30.15	2,878	7,479	9,182	19,539	1,298	3,034	10,787	12,469	26,291	1,353
40.10	1,080	8,459	5,381	14,921	1,252	3,113	13,477	8,701	25,291	1,306
50.6	379	4,540	2,109	7,028	900	713	8,023	3,402	12,137	1,257
60.4	234	3,063	1,283	4,580	570	470	5,744	2,377	8,617	1,087
70.3	167	1,155	844	2,166	367	359	2,429	1,733	4,521	800
70.5	369	6,593	2,399	9,361	889	752	12,418	4,108	17,278	1,509
80.5	518	8,147	3,567	12,232	1,030	1,036	14,825	5,928	21,790	1,900
90.5	495	10,631	4,101	15,227	1,059	1,094	21,150	6,945	29,189	2,174
100.5	518	15,628	4,895	21,042	674	1,331	29,914	8,979	40,224	2,404

Table 9: Average # of PLO cuts for transshipment with the SOS2 model

	Default + PLO cuts					**B&B + PLO cuts**				
$\eta.T$	(PLO1)	(PLO2)	(PLO5)	Total	Added	(PLO1)	(PLO2)	(PLO5)	Total	Added
15.20	738	2,835	3,996	7,569	771	658	3,372	4,848	8,878	620
15.25	1,483	4,770	7,293	13,546	889	907	4,918	5,227	11,052	569
15.30	3,848	8,249	17,334	29,431	749	1,293	5,303	9,909	16,505	682
20.20	608	4,273	3,210	8,091	1,035	670	4,809	3,434	8,912	627
25.15	974	7,556	12,004	20,534	1,253	1,264	8,646	11,932	21,842	971
25.20	1,487	9,977	8,109	19,574	1,183	2,394	13,039	9,603	25,036	750
30.6	385	2,291	1,326	4,002	887	386	2,353	1,393	4,132	679
30.10	545	4,656	2,907	8,108	961	561	5,252	3,171	8,984	803
30.15	2,383	7,779	10,570	20,732	1,524	3,200	10,478	11,517	25,195	1,160
40.10	2,568	12,122	8,353	23,044	1,493	2,159	12,382	8,726	23,268	1,384
50.6	737	7,003	3,296	11,036	1,616	702	8,234	3,523	12,459	1,398
60.4	523	5,206	2,244	7,974	1376	554	5,774	2,433	8,760	1,227
70.3	353	2,215	1,578	4,146	943	374	2,425	1,788	4,587	861
70.5	835	10,670	4,364	15,869	1,982	856	12,174	4,222	17,252	1,689
80.5	1,021	12,901	6,000	19,922	2,470	1,157	15,128	6,756	23,041	2,319
90.5	1,248	17,642	6,849	25,739	3,008	1,381	21,163	8,279	30,823	2,876
100.5	1,520	25,835	9,051	36,406	3,436	1,658	29,284	9,674	40,616	3,139

Table 10: Average # of GUROBI cuts for transshipment with the LOG model

	Default					**Default + PLO cuts**				
$\eta.T$	IB	Cv	FC	G	MIR	IB	Cv	FC	G	MIR
15.20	0	0	0	8	0	0	0	2	14	3
15.25	3	4	0	32	0	1	1	1	20	20
15.30	5	5	0	31	0	6	5	16	15	12
20.20	8	0	0	39	0	3	0	11	26	8
25.15	13	0	0	62	0	7	4	13	27	26
25.20	17	0	1	112	0	12	1	35	27	10
30.6	22	0	0	132	0	2	0	4	61	132
30.10	15	0	1	123	0	2	0	1	45	21
30.15	10	0	1	89	0	6	5	14	28	66
40.10	17	0	0	209	1	4	0	3	75	46
50.6	9	0	3	223	19	4	0	7	95	332
60.4	6	0	1	218	8	4		15	106	170
70.3	12	4	2	246	5	5	2	1	122	85
70.5	6	0	1	261	22	6	0	5	142	330
80.5	1	0	2	283	22	7	0	10	166	435
90.5	0	0	1	272	23	1	0	7	181	428
100.5	0	0	1	263	21	0	0	8	215	238

Table 11: Average # of GUROBI cuts for transshipment with the MIP model

	Default					**Default + PLO cuts**				
$\eta.T$	IB	Cv	FC	G	MIR	IB	Cv	FC	G	MIR
15.20	0	0	0	0	0	65	1	124	19	70
15.25	0	1	5	2	0	9	1	130	16	63
15.30	0	0	3	2	0	29	0	132	18	58
20.20	0	0	4	2	0	58	1	128	20	77
25.15	1	0	24	30	20	154	0	295	42	138
25.20	0	0	18	18	0	114	0	188	26	94
30.6	34	0	217	55	0	56	0	418	52	105
30.10	35	0	148	26	0	84	0	355	26	69
30.15	0	0	58	41	16	164	0	380	53	188
40.10	9	0	288	55	0	192	0	667	52	243
50.6	5	0	383	75	0	158	0	821	106	232
60.4	8	0	419	83	0	90	0	801	75	291
70.3	10	0	472	74	0	41	0	599	109	186
70.5	2	0	406	96	10	138	0	931	125	214
80.5	0	0	458	70	0	142	0	1,109	570	226
90.5	0	0	439	65	0	82	0	1,003	204	183
100.5	0	0	526	73	0	6	0	622	188	109

In Tables 12, 13, and 14 we compare the average number of enumeration nodes and solution time, and the number of instances solved to proven optimality for GUROBI in default setting against default with PLO cuts (PLO1) – (PLO4) added. In Tables 15, 16, and 17 we give the average number of cuts (PLO1) – (PLO4) separated and added. In Tables 18 and 19 we give the average number of MIP cuts used by GUROBI in the LOG and MIP formulations, respectively. In all tables each entry refers to 5 instances. As with transshipment, we computed both arithmetic and geometric averages, and they led to the same conclusions. So again we report only the arithmetic averages. Tables 12, 15, and 18 give results for the LOG formulation, Tables 13, 16, and 19 for the MIP formulation, and Tables 14 and 17 for the SOS2 approach. In Tables 15 – 19, the columns have the same meaning as the corresponding tables for transshipment.

In average, GUROBI added to the cutpool 15% of the PLO cuts separated for LOG, 10% for MIP, and 14% for SOS2. As with transshipment, the percentage reduction in the number of enumeration nodes due to PLO cuts was greater with CPLEX than GUROBI, but the percentage reduction in computational time was greater with GUROBI. The majority of GUROBI cutting planes added for LOG and MIP was flow cover, followed by Gomory. For LOG without PLO cuts, flow cover cuts were 54% and Gomory 42% of the GUROBI cuts; with PLO cuts, flow cover cuts were 69% and Gomory 30% of the GUROBI cuts. For MIP without PLO cuts, flow cover cuts were 52% and Gomory 32% of the GUROBI cuts; with PLO cuts, flow cover cuts were 64% and Gomory 27% of the GUROBI cuts. Because flow cover cuts are usually not as dense as Gomory, the more balanced use of flow cover and Gomory, with the majority being flow cover, may be one of the reasons MIP cuts were more efficient for transportation than for transshipment.

The impact of PLO cuts both on computational time and number of enumeration nodes was significant, but overall not as great as for transshipment. Out of the 180 instances tested, GUROBI, in default setting, was able to solve to proven optimality 94 (52%), as opposed to 115 (64%) with the PLO cuts, using the LOG model; 91 (51%), as opposed to 108 (58%), using the MIP model; and 43 (24%), as opposed to 92 (51%), using the SOS2 approach. The average reduction from default in computational time by adding the PLO cuts was 26% with the LOG model; 20% with the MIP model; and 31% with the SOS2 approach. The average reduction from default in the number of enumeration nodes by adding the PLO cuts was 95% with the LOG model, 93% with the MIP model, and 99% with the SOS2 approach.

We note, however, that while the PLO cuts were extremely efficient for the smaller values of T, that was not the case for the larger values of T. That is the reason the overall results for transportation are not as great as for transshipment. For the 60 instances with $T = 5$, default with the PLO cuts solved all instances to proven optimality (100%), as opposed to 26 (43%) without the PLO cuts for LOG; 59 (98%), as opposed to 46 (77%) for MIP; 57 (95%), as opposed to 26 (43%) for SOS2. The average reduction from default in computational time by adding the PLO cuts was 89%, with the LOG model; 78%, with the MIP model; and 84%, with the SOS2 approach. The average reduction from default in the number of enumeration nodes by adding PLO cuts was 99% with the LOG model, 92% with the MIP model, and 99% with the SOS2 approach.

For $T = 10$, default with the PLO cuts solved 48 instances to proven optimality (80%), as opposed to 39 (65%) without the PLO cuts for LOG; 36 (60%), as opposed to 35 (58%) for MIP; 30 (50%), as opposed to 16 (27%) for SOS2. The average reduction from default in computational time by adding the PLO cuts was 30%, with the LOG model; 7%, with the MIP model; and 17%, with the SOS2 approach. The average reduction from default in the number of enumeration nodes by adding the PLO cuts was 92% with the LOG model, 93% with the MIP model, and 98% with the SOS2 approach.

For $T = 20$, default with the PLO cuts solved 7 instances to proven optimality (12%), as opposed to 10 (16%) without the PLO cuts for LOG; 13 (22%), as opposed to 10 (16%) for MIP; 5 (8%), as opposed to 1 (2%) for SOS2. The average reduction from default in computational time by adding the PLO cuts was −3%, with the LOG model; 6%, with the MIP model; and 1%, with the SOS2 approach. The average reduction from default in the number of enumeration nodes by adding PLO cuts was 92% with the LOG and MIP models, and 98% with the SOS2 approach. In Section 5 we will analyze further the dependence on T of the performance of the PLO cuts for transportation.

We now discuss the effect of formulation on the computational results, first without the addition of the PLO cuts. Overall, MIP and LOG were the most efficient approaches, followed by SOS2. The MIP formulation solved to proven optimality 91 instances, as opposed to 94 for LOG, and 43 for SOS2. In number of nodes, MIP reduced by −6% the number of nodes for LOG, and by 91% for SOS2. In computational time, MIP reduced by 3% the computational time of LOG, and by 28% for SOS2. This rank remains the same for the different values of T.

With PLO cuts added, overall LOG was the most efficient, followed by MIP, followed by SOS2. The LOG formulation solved to proven optimality 115 instances, as opposed to 108 for MIP, and 92 for SOS2. In number of nodes, LOG reduced by 56% the number of nodes for MIP, and by 61% for SOS2. In computational time, LOG reduced by 3% the computational time of MIP, and by 22% for SOS2. This rank remains the same for $T \in \{5, 10\}$. For $T = 20$, MIP is the most efficient, followed by LOG, followed by SOS2. We leave for further study to analyze the reasons for such ranking, which is in disagreement with earlier studies, see for example [6, 9]. Such questions as in what ways MIP and SOS2 changed since the earlier studies, and how to improve SOS2 are relevant topics of research.

5 Large Special Ordered Sets

As pointed out in Section 4, the computational results for transportation indicated a degradation in performance of the PLO cuts for larger values of T. In this section we elaborate on this issue. We consider two sets of data, both transportation: the ones we generated and the data used in Vielma and Nemhauser [11]. Here, as in Section 4, the maximum computational time allowed was 3,600 seconds of CPU.

In Tables 20 – 25 we give the results for our data type. Each entry in the tables gives an average over 5 instances. The first two rows of the tables refer to small values of T (specifically, $T \in \{5, 10\}$), and the other rows, large values of T (specifically, $T \in \{20, 30, 40, 80, 120\}$). Tables 20 and 21 refer to the LOG formulation, Tables 22 and 23 to the MIP formulation, and Tables 24 and 25 to the SOS2 approach. In Tables 20, 22, and 24 we give the number of instances solved, the average computational time, and the average number of enumeration nodes with and without PLO cuts. In Tables 21, 23, and 25 we give the average number of PLO cuts separated and added. The asterisks in the last row of Tables 24 and 25 indicate that GUROBI crashed while trying to solve those instances (the same happened with CPLEX).

Of the 10 instances with small value of T, regardless of whether the PLO cuts were used or not, all were solved to proven optimality (100%) for the LOG formulation, 8 (80%) for the MIP formulation, and all (100%) for the SOS2 approach. In average, the PLO cuts reduced the computational time of default by 79% for the LOG formulation, 48% for the MIP formulation, and

Table 12: Average solution time and node, and # of instances solved for transportation with the LOG model

	Default			**Default + PLO cuts**				
s.d.T	Node	Time	#Sol.	Node	%Red.	Time	%Red.	#Sol.
25.50.5	87,532	936	4	587	99.33	18	98.08	5
25.100.5	54,743	971	5	526	99.04	34	96.50	5
25.200.5	66,092	2,578	2	457	99.31	101	96.08	5
25.300.5	49,010	3,600	0	232	99.53	103	97.14	5
25.400.5	35,873	3,600	0	631	98.24	479	86.69	5
50.100.5	3,703	171	5	175	95.27	37	78.36	5
50.200.5	1,334	272	5	25	98.13	43	84.19	5
50.300.5	2,975	617	5	12	99.60	99	83.95	5
50.400.5	5,356	1,754	4	1	99.98	139	92.08	5
100.200.5	8,743	1,036	4	59	99.33	207	80.02	5
100.300.5	26	322	5	5	80.77	222	31.06	5
100.400.5	393	528	5	1	99.75	260	50.76	5
25.50.10	78,595	659	5	7,139	90.92	398	39.61	5
25.100.10	96,586	2,037	3	10,004	89.64	1,511	25.82	4
25.200.10	60,468	3,466	1	5,097	91.57	2,879	16.94	3
25.300.10	31,651	3,600	0	2,486	92.15	3,172	11.89	1
25.400.10	16,806	3,600	0	1,486	91.16	3,567	0.92	1
50.100.10	21,549	812	5	185	99.14	117	85.59	5
50.200.10	19,027	1,564	3	105	99.45	319	79.60	5
50.300.10	2,773	910	5	105	96.21	473	48.02	5
50.400.10	11,174	3,384	2	115	98.97	817	75.86	5
100.200.10	6,804	986	5	173	97.46	1,033	-4.77	4
100.300.10	150	590	5	2	98.67	595	-0.85	5
100.400.10	92	613	5	6	93.48	736	-20.07	5
25.50.20	206,308	3,150	2	16,025	92.23	3,600	-14.29	0
25.100.20	83,214	3,492	1	5,114	93.85	3,600	-3.09	0
25.200.20	19,508	3,600	0	1,372	92.97	3,600	0	0
25.300.20	9,143	3,600	0	625	93.16	3,600	0	0
25.400.20	3,677	3,600	0	377	89.75	3,567	0.92	1
50.100.20	27,569	1,962	3	1,783	93.53	3,504	-78.59	1
50.200.20	7,684	3,600	0	599	92.20	3,600	0	0
50.300.20	2,522	3,600	0	261	89.65	3,600	0	0
50.400.20	1,912	3,600	0	143	92.52	3,600	0	0
100.200.20	4,250	3,568	1	190	95.53	3,600	-0.9	0
100.300.20	1,070	3,327	2	66	93.83	3,062	7.97	2
100.400.20	559	2,880	2	13	97.67	2,241	22.19	3

Table 13: Average solution time and node, and # of instances solved for transportation with the MIP model

	Default			Default + PLO cuts				
s.d.T	Node	Time	#Sol.	Node	%Red.	Time	%Red.	#Sol.
25.50.5	146,664	1,381	4	25,343	82.72	356	74.22	5
25.100.5	145,207	1,710	3	2,686	98.15	131	92.34	5
25.200.5	68,845	2,642	2	5,873	91.47	890	66.31	4
25.300.5	44,531	3,300	2	630	98.59	324	90.18	5
25.400.5	39,032	3,600	0	991	94.46	837	76.75	5
50.100.5	5,656	252	5	1,172	79.28	167	33.73	5
50.200.5	660	161	5	80	87.88	59	63.35	5
50.300.5	1,115	318	5	56	94.98	104	67.30	5
50.400.5	1,626	704	5	7	79.57	78	88.92	5
100.200.5	1,107	413	5	99	91.06	175	57.63	5
100.300.5	9	61	5	2	77.78	76	-24.59	5
100.400.5	247	415	5	4	98.38	128	69.16	5
25.50.10	376,311	2,471	2	39,355	89.54	2,480	-0.36	2
25.100.10	507,188	3,600	0	20,001	96.06	3,600	0	0
25.200.10	47,534	3,600	0	5,507	88.41	3,600	0	0
25.300.10	29,901	3,600	0	2,577	91.38	3,600	0	0
25.400.10	17,788	3,600	0	1,424	91.99	3,600	0	0
50.100.10	3,714	396	5	308	91.71	218	44.95	5
50.200.10	7,061	1,078	4	209	97.04	497	53.90	5
50.300.10	2,804	1,259	5	390	86.09	1,469	-16.68	5
50.400.10	2,769	2,188	4	165	94.04	1,124	48.63	4
100.200.10	9	68	5	1	88.89	97	-42.65	5
100.300.10	3	96	5	0	100	112	-16.67	5
100.400.10	14	122	5	2	85.71	181	-48.36	5
25.50.20	186,226	3,600	0	14,066	72.45	3,600	0	0
25.100.20	27,271	3,600	0	4,96	98.18	3,600	0	0
25.200.20	8,113	3,600	0	1,379	83.00	3,600	0	0
25.300.20	2,739	3,600	0	591	78.42	3,600	0	0
25.400.20	2,745	3,600	0	273	90.05	3,600	0	0
50.100.20	9,962	3,207	1	2,096	78.96	3,600	-12.25	0
50.200.20	2,629	3,600	0	517	80.33	3,600	0	0
50.300.20	2,443	3,600	0	190	92.22	3,600	0	0
50.400.20	1,601	3,600	0	65	95.94	3,600	0	0
100.200.20	1,551	2,517	2	72	95.36	1,859	26.14	3
100.300.20	390	2,050	4	14	96.41	1,257	38.68	5
100.400.20	150	2,326	3	0	100	1,220	47.55	5

Table 14: Average solution time and node, and # of instances solved for transportation with the SOS2 model

	Default			**Default + PLO cuts**				
s.d.T	Node	Time	#Sol.	Node	%Red.	Time	%Red.	#Sol.
25.50.5	2,110,475	1,815	4	2,314	99.89	28	98.48	5
25.100.5	1,852,388	3,600	0	1,527	99.92	49	98.64	5
25.200.5	619,103	3,600	0	9,993	98.39	942	73.83	5
25.300.5	362,697	3,600	0	2,944	99.19	586	83.72	5
25.400.5	237,542	3,600	0	7,554	96.82	2,748	23.67	2
50.100.5	423,790	1,816	4	1,049	99.75	95	94.77	5
50.200.5	290,946	2,971	2	99	99.97	35	98.22	5
50.300.5	136,919	2,931	4	110	99.92	71	97.58	5
50.400.5	129,503	3,600	0	80	99.94	95	97.36	5
100.200.5	150,929	3,472	4	462	99.69	344	90.09	5
100.300.5	16,894	672	5	141	99.17	229	65.92	5
100.400.5	26,602	1,572	3	48	99.82	147	90.65	5
25.50.10	1,666,720	2,504	2	43,222	97.41	1,846	26.28	2
25.100.10	808,788	3,600	0	25,897	96.80	3,600	0	0
25.200.10	325,992	3,600	0	6,661	97.96	3,600	0	0
25.300.10	192,535	3,600	0	3,083	98.40	3,600	0	0
25.400.10	128,298	3,600	0	1,766	98.62	3,600	0	0
50.100.10	271,766	2,298	3	1,924	99.29	678	70.50	5
50.200.10	178,276	3,600	0	1,854	98.96	1,868	48.11	5
50.300.10	106,273	3,600	0	871	99.18	1,808	49.78	4
50.400.10	62,412	3,600	0	853	98.63	3,098	13.94	1
100.200.10	29,714	1,561	3	514	98.27	1,867	-19.60	3
100.300.10	10,105	760	4	1,077	89.34	1,001	-31.71	5
100.400.10	7,385	752	4	71	99.04	886	-17.82	5
25.50.20	876,301	3,600	0	17,196	98.04	3,600	0	0
25.100.20	371,069	3,600	0	5,327	98.56	3,600	0	0
25.200.20	152,110	3,600	0	1,576	98.96	3,600	0	0
25.300.20	83,838	3,600	0	769	99.08	3,600	0	0
25.400.20	52,835	3,600	0	448	99.15	3,600	0	0
50.100.20	176,752	3,600	0	1,941	98.90	3,600	0	0
50.200.20	53,197	3,600	0	649	98.78	3,600	0	0
50.300.20	34,203	3,600	0	325	99.05	3,600	0	0
50.400.20	21,153	3,600	0	197	99.07	3,600	0	0
100.200.20	25,147	3,600	0	212	99.16	3,600	0	0
100.300.20	14,627	3,600	0	106	99.28	3,230	10.28	4
100.400.20	8,044	2,898	1	69	99.14	3,237	-11.70	1

Table 15: Average # of PLO cuts for transportation with the LOG model and default setting

$s.d.T$	(PLO1)	(PLO2)	(PLO3)	(PLO4)	Total	Added
25.50.5	563	100	9	71	743	193
25.100.5	589	72	6	29	696	146
25.200.5	540	138	10	57	745	180
25.300.5	532	173	7	70	782	232
25.400.5	1,074	273	22	90	1,458	298
50.100.5	180	41	8	25	254	101
50.200.5	126	45	2	7	181	44
50.300.5	173	45	0	6	224	51
50.400.5	207	65	3	7	282	52
100.200.5	175	79	14	30	298	84
100.300.5	67	36	10	18	131	21
100.400.5	70	31	4	6	110	21
25.50.10	1,971	298	30	527	2,826	230
25.100.10	3,992	168	4	134	4,299	259
25.200.10	1,052	186	3	105	1,346	200
25.300.10	1,353	282	2	133	1,770	251
25.400.10	1,925	358	6	180	2,470	233
50.100.10	127	135	26	44	333	89
50.200.10	356	38	0	7	401	123
50.300.10	227	24	1	8	260	60
50.400.10	459	35	2	3	499	137
100.200.10	141	168	10	27	346	43
100.300.10	49	46	3	9	108	11
100.400.10	26	32	5	5	68	10
25.50.20	747	1,073	896	547	3,262	304
25.100.20	324	310	13	150	797	137
25.200.20	537	318	5	218	1,078	106
25.300.20	809	536	0	303	1,649	82
25.400.20	952	557	7	260	1,775	256
50.100.20	873	394	27	130	1,424	150
50.200.20	729	190	7	96	1,022	227
50.300.20	1,135	152	1	101	1,389	275
50.400.20	950	197	0	103	1,249	268
100.200.20	254	313	15	23	604	167
100.300.20	245	57	7	9	317	82
100.400.20	171	36	0	6	213	41

Table 16: Average # of PLO cuts for transportation with the MIP model and default setting

$s.d.T$	(PLO1)	(PLO2)	(PLO3)	(PLO4)	Total	Added
25.50.5	9,476	1,119	11	257	10,862	269
25.100.5	424	104	5	31	564	147
25.200.5	793	215	11	83	1,102	166
25.300.5	837	309	10	105	1,260	94
25.400.5	1,350	376	19	115	1,861	269
50.100.5	339	80	7	43	469	90
50.200.5	148	54	3	10	216	57
50.300.5	214	60	0	8	283	67
50.400.5	265	93	3	11	371	65
100.200.5	140	87	11	33	270	74
100.300.5	75	41	9	19	144	18
100.400.5	86	57	5	8	156	24
25.50.10	1,445	437	25	314	2,221	261
25.100.10	1,616	279	5	152	2,052	189
25.200.10	1,869	231	2	93	2,195	213
25.300.10	2,084	288	2	159	2,533	282
25.400.10	3,090	359	6	170	3,635	256
50.100.10	197	242	32	58	529	65
50.200.10	488	78	4	9	578	70
50.300.10	496	42	2	16	556	26
50.400.10	575	57	3	7	642	108
100.200.10	38	149	11	23	221	18
100.300.10	12	29	2	7	51	8
100.400.10	15	43	8	10	76	6
25.50.20	608	858	622	489	2,577	232
25.100.20	320	417	7	130	874	121
25.200.20	475	212	4	172	863	178
25.300.20	703	359	0	191	1,253	206
25.400.20	1,086	562	6	247	1,901	269
50.100.20	352	500	24	65	941	207
50.200.20	638	195	6	87	925	183
50.300.20	1,119	152	1	117	1,388	247
50.400.20	1,080	236	0	134	1,450	166
100.200.20	216	776	19	41	1,053	104
100.300.20	97	248	9	12	366	23
100.400.20	159	299	0	7	464	22

Table 17: Average # of PLO cuts for transportation with the SOS2 model and default setting

$s.d.T$	(PLO1)	(PLO2)	(PLO3)	(PLO4)	Total	Added
25.50.5	2,269	246	50	590	3,155	291
25.100.5	480	142	9	43	673	274
25.200.5	927	307	23	108	1,365	467
25.300.5	1,424	641	29	195	2,289	514
25.400.5	1,974	726	42	192	2,943	715
50.100.5	327	92	13	39	470	240
50.200.5	194	97	7	13	311	174
50.300.5	275	106	0	12	393	223
50.400.5	353	141	7	18	520	283
100.200.5	296	153	39	74	562	252
100.300.5	126	76	27	48	276	132
100.400.5	132	63	10	16	220	113
25.50.10	9,709	1,083	872	12,824	24,487	232
25.100.10	1,330	278	11	125	1,745	437
25.200.10	2,251	445	3	151	2,850	424
25.300.10	2,314	702	4	212	3,232	567
25.400.10	3,054	891	14	252	4,210	565
50.100.10	438	259	83	121	900	278
50.200.10	1,327	138	3	15	1,483	246
50.300.10	766	75	1	18	860	197
50.400.10	1,090	131	13	22	1,255	284
100.200.10	389	250	23	72	735	166
100.300.10	170	80	6	23	279	127
100.400.10	87	49	14	16	167	78
25.50.20	7,938	2,411	1,211	5,631	17,191	265
25.100.20	498	561	5	170	1,234	258
25.200.20	581	788	10	278	1,657	267
25.300.20	470	669	1	180	1,320	538
25.400.20	700	1,015	11	171	1,897	663
50.100.20	693	621	66	147	1,528	346
50.200.20	842	523	31	115	1,510	202
50.300.20	804	269	0	68	1,142	690
50.400.20	827	356	0	42	1,226	593
100.200.20	257	491	25	31	804	240
100.300.20	287	137	17	19	461	180
100.400.20	175	54	0	3	232	114

Table 18: Average # of GUROBI cuts for transportation with the LOG model

	Default					Default + PLO cuts				
$s.d.T$	IB	Cv	FC	G	MIR	IB	Cv	FC	G	MIR
25.50.5	6	0	117	95	12	0	0	47	8	0
25.100.5	3	0	133	143	0	0	0	55	12	0
25.200.5	9	1	184	223	12	3	0	74	35	0
25.300.5	0	0	206	276	27	0	0	124	23	0
25.400.5	0	0	253	292	13	1	0	134	96	0
50.100.5	33	1	131	38	0	0	0	63	6	0
50.200.5	8	2	67	38	0	0	0	41	6	0
50.300.5	11	0	107	70	0	0	0	54	10	0
50.400.5	6	0	119	70	0	0	0	69	15	0
100.200.5	1	3	86	12	0	0	0	52	3	0
100.300.5	0	0	57	2	0	0	0	25	3	0
100.400.5	11	10	63	15	0	0	0	26	5	0
25.50.10	6	1	66	29	9	2	0	61	21	0
25.100.10	4	1	142	102	27	3	0	96	72	0
25.200.10	0	0	200	154	34	1	0	156	134	0
25.300.10	0	0	317	278	0	0	0	188	146	0
25.400.10	0	0	358	277	0	0	0	230	193	0
50.100.10	14	6	52	15	12	0	0	47	2	0
50.200.10	0	0	65	13	0	0	0	54	1	0
50.300.10	0	0	45	16	0	0	0	48	0	0
50.400.10	0	0	106	59	0	0	0	60	1	0
100.200.10	0	4	41	13	0	0	0	37	5	0
100.300.10	0	4	25	2	0	0	0	21	0	0
100.400.10	0	0	11	0	0	0	0	10	0	0
25.50.20	0	0	48	23	0	1	0	68	35	0
25.100.20	0	0	112	95	0	0	0	126	103	0
25.200.20	0	0	185	178	0	0	0	178	164	0
25.300.20	0	0	244	238	0	0	0	214	199	0
25.400.20	0	0	295	264	0	0	0	374	1	0
50.100.20	0	0	25	7	0	1	1	47	26	0
50.200.20	0	0	40	35	0	0	0	64	6	0
50.300.20	0	0	57	50	0	0	0	102	0	0
50.400.20	0	0	76	68	0	0	0	113	0	0
100.200.20	0	0	1	0	0	0	0	3	2	0
100.300.20	0	0	0	0	0	0	0	1	5	0
100.400.20	0	0	0	0	0	0	0	6	1	0

Table 19: Average # of GUROBI cuts for transportation with the MIP model

	Default					**Default + PLO cuts**				
s.d.T	IB	Cv	FC	G	MIR	IB	Cv	FC	G	MIR
25.50.5	166	6	237	113	3	5	4	41	19	0
25.100.5	128	6	220	127	0	19	1	84	34	0
25.200.5	81	7	257	242	5	11	4	83	55	0
25.300.5	34	0	262	258	11	4	0	51	31	0
25.400.5	6	0	362	315	4	13	0	122	86	0
50.100.5	42	4	120	36	0	18	4	64	17	0
50.200.5	10	1	51	25	0	5	0	62	8	0
50.300.5	25	4	89	42	0	4	0	63	14	0
50.400.5	20	2	122	77	0	5	0	78	17	0
100.200.5	43	8	138	26	0	4	2	80	6	0
100.300.5	7	1	68	6	0	4	0	46	6	0
100.400.5	2	1	71	10	0	4	2	48	6	0
25.50.10	177	5	181	95	0	60	0	108	55	0
25.100.10	107	4	227	123	0	14	1	99	71	0
25.200.10	89	1	325	184	0	21	1	127	115	0
25.300.10	88	2	547	290	0	13	0	144	169	0
25.400.10	42	1	635	318	0	26	0	222	193	0
50.100.10	33	6	86	25	0	5	1	38	13	0
50.200.10	14	0	119	31	0	3	0	44	9	0
50.300.10	11	0	82	37	0	10	0	35	13	0
50.400.10	6	0	142	60	0	2	0	53	10	0
100.200.10	4	5	46	5	0	5	6	45	4	0
100.300.10	0	52	0	6	0	1	0	51	7	0
100.400.10	1	1	31	2	0	1	1	31	2	0
25.50.20	116	0	177	115	0	32	0	122	77	0
25.100.20	62	1	153	144	0	16	0	97	123	0
25.200.20	157	241	2	262	0	44	0	168	55	0
25.300.20	1	0	285	302	0	2	0	258	71	0
25.400.20	0	0	380	345	0	0	0	337	75	0
50.100.20	35	22	153	48	0	13	0	100	10	0
50.200.20	29	1	141	60	0	14	0	98	11	0
50.300.20	54	0	176	95	0	25	0	114	17	0
50.400.20	18	0	219	26	0	22	0	124	22	0
100.200.20	20	1	83	7	0	19	8	76	6	0
100.300.20	13	6	63	4	0	12	9	64	3	0
100.400.20	5	0	58	6	0	4	0	51	4	0

15% for the SOS2 approach. The number of nodes was reduced by 93% in the LOG formulation, 85% in the MIP formulation, and 91% in the SOS2 approach.

On the other hand, of the 25 instances with large value of T, default solved 20 (80%), as opposed to 13 (52%) with the PLO cuts, for the LOG formulation; 2 (8%), as opposed to 1 (4%), for the MIP formulation; and 8 (32%), as opposed to 7 (28%), for the SOS2 approach. In average, the use of PLO cuts increased the computational time over default by 203% for the LOG formulation, 7% for the MIP formulation, and 42% for the SOS2 approach. In number of nodes, the use of PLO cuts reduced the amount generated by default by 81% for the LOG formulation; 95% for the MIP formulation; and 96% for the SOS2 approach. So the performance of the PLO cuts was considerably inferior to default for the large values of T.

We note that the number of PLO cuts generated for the instances with large value of T was extremely large, and even more importantly, the percentage of PLO cuts added to the formulation was considerably smaller than for the other tests of this section and of the previous ones. In the particular case of the tests with small T of this section, 18% of the PLO cuts generated were used for the LOG formulation; 57% for the MIP formulation; and 0.1% for the SOS2 approach. For the tests with large T, 0.2% were used for the LOG formulation; 4% for the MIP formulation; and 0.001% for the SOS2 approach. So one possible explanation for the poor performance of the PLO cuts in our tests with large T was the time wasted separating a large number of cuts, which, in the end, did not pass GUROBI's user cut filter. With this in mind, we tried several alternatives for reducing the number of PLO cuts generated. For example, we tried cut-and-branch, or limiting the number of cuts by a fixed amount, or increasing the violation required for separation. In all our attempts, the performance of the PLO cuts continued poor. So, it may just be that our PLO cuts, for the type of data we used, are not efficient when T is large. We believe that further investigation of this issue is an interesting topic of further research.

Now we analyze the results obtained from the data of Vielma and Nemhauser [11]. We tested 100 of their instances, specifically, their largest, with $s = d = 10$, and $T = 32$. Here we give the results of both CPLEX (denoted as CPX) and GUROBI (denoted as GRB). In Table 26 we give the number of instances solved to proven optimality by default, B&B, and B&B with PLO cuts. In Tables 27 – 29 we give the average computational time, number of enumeration nodes, and number of PLO cuts, respectively, again for default, B&B, and B&B with PLO cuts. First, we analyze the results without the PLO cuts.

With CPLEX, LOG was more efficient than SOS2, which was more efficient than MIP. LOG default and B&B solved all instances to proven optimality (100%), while SOS2 solved 96 (96%) in default and 97 (97%) in B&B, and MIP solved 91 (91%) in default and 92 (92%) in B&B. LOG reduced the computational time of SOS2 by 98% in default and by 97% in B&B; it reduced the computational time of MIP by 99% in default and B&B. LOG reduced the number of nodes of SOS2 by 99% in default and B&B; it reduced the number of nodes of MIP by 98% in default and B&B. These results are in agreement with the results of [11]. We note that B&B was slightly better than default for LOG, SOS2, and MIP. Therefore, CPLEX's facilities, such as MIP cutting planes, did not play a role.

With GUROBI, LOG was slightly more efficient than SOS2, which was more efficient than MIP, but the difference now was much less significant. All of them solved all instances to proven optimality (100%) in both default and B&B. LOG reduced the computational time of SOS2 by 60% in default and by 73% in B&B. However, the difference is of just a few seconds. As for MIP, LOG reduced its computational time by 90% in default and by 93% in B&B, but again the absolute difference is considerably smaller. The small difference in absolute value is in disagreement with the much higher difference in [11]. Because B&B was better than default for CPLEX and equivalent for GUROBI, and B&B GUROBI was considerably more efficient than B&B CPLEX, we raise the possibility that, in the case of the data of [11], the difference in performance of LOG and SOS2 may be due to a better implementation of variable dichotomy branching than SOS2 branching. LOG reduced the number of nodes of SOS2 by 88% in default and by 90% in B&B; it reduced the number of nodes of MIP by 89% in default and by 92% in B&B.

With the PLO cuts, all instances were solved to proven optimality (100%). Now the difference in performance for LOG, MIP, SOS2, CPLEX, and GUROBI became of just a few seconds. The PLO cuts reduced the computational time in all cases. The reduction was particularly significant in the case of CPLEX SOS2 and MIP. In the first case, the PLO cuts reduced the computational time of default by 96% and of B&B by 95%. In the second case, they reduced the computational time of default and B&B by 97%. This means that for the data of [11], the PLO cuts were efficient even with the value of T being large. We note that for the data of [11], the percentage of PLO cuts that passed GUROBI's user cut filter was higher, 24% for LOG, 3% for SOS2, and 27% for MIP.

Table 20: Average solution time and # of nodes, and # of instances solved for transportation with the LOG model

	Default			**Default + PLO cuts**				
$s.d.T$	Node	Time	#Sol.	Node	%Red.	Time	%Red.	#Sol.
10.20.5	14,542	18	5	951	93	3	83	5
10.20.10	22,418	46	5	1,873	92	12	74	5
10.20.20	35,044	48	5	8,664	75	121	−152	5
10.20.30	85,893	173	5	32,563	62	846	−389	5
10.20.40	165,599	440	5	39,757	76	2,074	−371	3
10.20.80	278,037	1,924	3	13,624	95	3,600	−87	0
10.20.120	282,304	3,097	2	4,059	99	3,600	−16	0

Table 21: Average # of PLO cuts for transportation with the LOG model in default setting

$s.d.T$	(PLO1)	(PLO2)	(PLO3)	(PLO4)	Total	Added
10.20.5	601	196	40	331	1,168	181
10.20.10	361	123	10	341	835	170
10.20.20	209	312	41	807	1,369	217
10.20.30	929	1,138	318	13,682	16,068	401
10.20.40	2,445	5,051	1,838	23,873	33,207	539
10.20.80	9,615	114,502	149,690	478,712	752,519	861
10.20.120	15,120	266,337	233,558	686,685	1,201,700	1,292

Table 22: Average solution time and # of nodes, and # of instances solved for transportation with the MIP model

	Default			Default + PLO cuts				
s.d.T	Node	Time	#Sol.	Node	%Red.	Time	%Red.	#Sol.
10.20.5	81,933	129	5	1,907	98	7	95	5
10.20.10	909,081	1,463	3	258,833	72	1,452	1	3
10.20.20	1,702,314	2,326	2	208,367	88	3,177	−36	1
10.20.30	2,064,999	3,600	0	103,104	95	3,600	0	0
10.20.40	1,070,054	3,600	0	52,286	95	3,600	0	0
10.20.80	337,078	3,600	0	933	98	3,600	0	0
10.20.120	37,606	3,600	0	539	99	3,600	0	0

Table 23: Average # of PLO cuts for transportation with the MIP model in default setting

s.d.T	(PLO1)	(PLO2)	(PLO3)	(PLO4)	Total	Added
10.20.5	330	99	29	227	685	262
10.20.10	292	150	17	382	841	303
10.20.20	256	629	38	723	1,646	353
10.20.30	323	1,593	136	2,254	4,306	765
10.20.40	636	7,263	1,029	9,109	18,037	1,291
10.20.80	1,784	7,078	26,619	72,257	107,738	2,982
10.20.120	1,757	9,880	37,237	135,497	184,731	8,924

Table 24: Average solution time and # of nodes, and # of instances solved for transportation with the SOS2 approach

	Default			Default + PLO cuts				
s.d.T	Node	Time	#Sol.	Node	%Red.	Time	%Red.	#Sol.
10.20.5	119,886	23	5	1,754	99	2	91	5
10.20.10	237,369	70	5	39,827	83	113	−61	5
10.20.20	861,870	433	5	87,644	90	855	−97	5
10.20.30	2,028,894	1,591	3	109,731	95	2,707	−70	2
10.20.40	3,442,864	3,600	0	66,138	98	3,600	0	0
10.20.80	1,618,481	3,600	0	12,243	99	3,600	0	0
10.20.120	***	***	****	***	***	***	***	***

Table 25: Average # of PLO cuts for transportation with the SOS2 approach in default setting

$s.d.T$	(PLO1)	(PLO2)	(PLO3)	(PLO4)	Total	Added
10.20.5	3,495	1,487	628	3,411	9,021	263
10.20.10	100,856	70,349	6,521	101,339	279,065	93
10.20.20	104,522	119,745	1,649	169,601	395,517	46
10.20.30	514,441	340,105	293,215	2,460934	3,508,695	46
10.20.40	690,242	725,494	626,750	3,104,560	5,147,046	46
10.20.80	312,393	1,051,639	1,516,723	3,380,750	6,261,505	46
10.20.120	***	***	***	***	***	***

Table 26: # of Vielma-Nemhauser instances solved

	LOG			**MIP**			**SOS2**		
Solver	Def.	B&B	PLO	Def.	B&B	PLO	Def.	B&B	PLO
CPX	100	100	100	91	92	100	96	97	100
GRB	100	100	100	100	100	100	100	100	100

Table 27: Average solution time for the Vielma-Nemhauser instances

	LOG			**MIP**			**SOS2**		
Solver	Def.	B&B	PLO	Def.	B&B	PLO	Def.	B&B	PLO
CPX	6	8	4	684	545	21	266	244	10
GRB	6	4	3	56	56	8	15	15	7

Table 28: Average # of nodes for the Vielma-Nemhauser instances

	LOG			**MIP**			**SOS2**		
Solver	Def.	B&B	PLO	Def.	B&B	PLO	Def.	B&B	PLO
CPX	5,063	4,446	57	254,910	303,039	258	709,249	710,774	270
GRB	4,403	4,215	71	41,520	49,369	155	37,800	41,254	455

Table 29: Average # of PLO cuts separated and added for the Vielma-Nemhauser instances

	CPLEX			**GUROBI**		
	LOG	MIP	SOS2	LOG	MIP	SOS2
(PLO1)	34	60	82	63	80	796
(PLO2)	413	578	624	965	1,102	6,217
(PLO3)	252	346	337	480	589	3,294
(PLO4)	425	747	878	823	953	7,454
Total	1,124	1,731	1,921	2,331	2,724	17,760
Added	1,124	1,731	1,912	549	734	582

6 Performance of SC-PLO Cuts for Transportation with Semi-Continuous Constraints

We now give the results of our initial computational testing of the SC-PLO cuts on transportation instances with semi-continuous constraints. In all tests we used only the LOG formulation and GUROBI in default setting. Table 30 gives the computational time and number of enumeration nodes for default, default with PLO cuts, and default with SC-PLO cuts. The columns "%Red." for default with PLO cuts refer to number of nodes and computational time reduction over default. However, "%Red." for default with SC-PLO cuts refer to number of nodes and computational time reduction over default with PLO cuts. Table 31 gives the number of PLO cuts separated and added and Table 32 gives the number of SC-PLO cuts separated and added. Unlike in the other sections, each entry of Tables 30, 31, and 32 refers to a single instance. Because these instances turned out to be considerably more difficult than the ones without semi-continuous constraints, we allowed for a maximum computational time of 7,200 seconds of CPU.

Of the 25 instances tested, default solved to proven optimality 15 instances, as opposed to 17 with PLO cuts and 18 with SC-PLO cuts. The average reduction in computational time over default for the 17 instances that default with PLO cuts solved to proven optimality was of 31%. Of these 17 instances, default with PLO cuts was faster for 13 instances (in almost all cases by more than 50%) and slower for 4 instances. The average reduction in number of nodes was of 81%.

The average reduction in computational time over default with PLO cuts for the 18 instances that default with SC-PLO cuts solved to proven optimality was of 33%. Of these 18 instances, default with SC-PLO cuts was faster for 16 instances (in most cases by more than 50%) and slower for 2 instances. The average reduction in number of nodes was of 45%.

So, PLO cuts were considerably useful for solving the instances, and SC-PLO cuts were even more useful. We note that only 0.03% and 0.04% of the PLO and SC-PLO cuts, respectively, separated were used. It is then possible that by separating less PLO and SC-PLO cuts, their performance on these instances may be improved. We leave this question for further research.

7 Summary of Conclusions and Further Research

We conclude that both the PLO and SC-PLO cuts given in [12] can be of great use in solving difficult instances of SPLO and SPLO with semi-continuous constraints. As already expected from previous studies, e.g. [5, 6, 9], SOS2 performed better than MIP in our tests. In addition, LOG performed better than SOS2. An important open question is why this is so, and whether it is possible to improve the SOS2 approach to perform as well as or better than LOG.

We find the following question to be also interesting:

- How should PLO and SC-PLO cuts be filtered?
- When are MIP cuts, for the LOG and MIP models, efficient?
- How can we make sure PLO cuts are efficient when T is large?

We suggest the study of these questions as topics for further research.

Table 30: Solution time and # of nodes for transportation with semi-continuous constraints

	Default		**Def. + PLO cuts**				**Def. + SC-PLO cuts**			
s.d.T	# Nodes	Time	# Nodes	%Red.	Time	%Red.	# Nodes	%Red.	Time	%Red.
5.20.10	26,720	13	1,360	95	5	62	1,113	18	5	0
5.20.10	131,228	108	27,992	79	64	41	6,741	76	24	63
5.20.10	50,160	27	3,542	93	8	70	2,077	47	8	0
5.20.10	42,766	26	1,555	96	5	81	927	40	3	40
5.20.10	145,656	134	2,558	98	9	93	1,933	24	9	0
7.14.10	67,495	32	5,543	92	16	50	2,304	58	8	50
7.14.10	254,140	171	38,223	85	83	51	12,480	67	45	46
7.14.10	572,431	386	37,593	93	87	77	19,276	49	54	38
7.14.10	204,731	183	79,948	61	185	−1	40,034	50	120	35
7.14.10	32,341	19	8,788	73	24	−26	11,878	−35	31	−29
8.16.10	2,814,189	2,442	1,330,216	53	3,788	−55	300,325	77	1,111	71
8.16.10	156,997	114	107,716	31	321	−182	14,586	86	65	80
8.16.10	3,209,494	2,518	725,697	77	1,966	22	635,064	10	2,280	−16
8.16.10	613,659	368	53,172	91	157	57	25,906	51	100	36
8.16.10	2,051,311	1,468	445,506	78	1,197	18	154,669	65	551	54
10.20.5	2,828,338	7,200	2,001,681	29	7,200	0	1,728,372	14	7,200	0
10.20.5	6,315,328	7,200	161,549	97	587	92	63,803	61	244	58
10.20.5	4,600,864	7,200	2,392,827	48	7,200	0	2,316,737	3	7,200	0
10.20.5	2,744,003	7,200	1,971,728	28	7,200	0	1,969,959	0	7,200	0
10.20.5	3,229,417	7,200	1,380,925	57	7,200	0	1,315,588	5	7,200	0
10.20.10	2,221,766	7,200	1,050,939	53	7,200	0	989,279	6	6,895	4
10.20.10	3,350,263	7,200	762,793	77	7,200	0	740,642	3	7,200	0
10.20.10	2,607,928	7,200	798,658	69	7,200	0	772,746	3	7,200	0
10.20.10	2,904,149	7,200	386,021	87	2,236	69	128,515	67	863	61
10.20.10	3,003,894	7,200	1,153,694	82	7,200	0	1,605,999	8	7,200	0

Table 31: # of PLO cuts separated and added for transportation with semi-continuous constraints

$s.d.T$	(PLO1)	(PLO2)	(PLO3)	(PLO4)	Total	Added
5.20.10	179	136	29	218	562	129
5.20.10	390	203	52	466	1,111	352
5.20.10	573	310	59	556	1,498	195
5.20.10	407	387	132	479	1,405	176
5.20.10	361	359	88	617	1,425	360
7.14.10	401	260	139	746	1,546	445
7.14.10	1,923	1,363	970	3,567	7,823	414
7.14.10	7,704	8,751	3,989	9,866	30,310	442
7.14.10	1,510	771	174	5,603	8,058	409
7.14.10	295	124	43	440	902	384
8.16.10	65,502	68,420	20,051	96,846	251,819	523
8.16.10	15,593	25,755	4,836	25,771	71,955	547
8.16.10	55,266	29,805	6,585	111,467	203,123	551
8.16.10	1,835	1,318	696	5,119	8,968	530
8.16.10	16,465	11,378	7,092	39,236	74,171	496
10.20.5	3,260,590	877,494	215,140	1,420,026	5,773,250	608
10.20.5	65,549	26,758	22,659	90,763	205,729	674
10.20.5	5,123,158	1,815,653	394,197	3,862,792	11,195,800	665
10.20.5	3,611,342	1,189,307	339,585	2,427,493	7,567,727	611
10.20.5	4,069,597	1,292,393	213,768	1,905,675	7,481,433	592
10.20.10	289,480	192,425	17,386	413,021	912,312	853
10.20.10	334,344	455,386	24,296	699,177	1,513,203	775
10.20.10	245,125	83,558	86,466	229,752	644,901	777
10.20.10	11,301	27,258	1,178	10,003	49,740	836
10.20.10	59,630	15,636	2,409	91,714	169,389	809

Table 32: # of SC-PLO cuts separated and added

$s.d.T$	(SC-PLO1)	(SC-PLO2)	(SC-PLO3)	(SC-PLO4)	Total	Added
5.20.10	190	147	323	311	971	227
5.20.10	592	221	436	1,500	2,749	393
5.20.10	239	252	262	303	1,056	328
5.20.10	146	162	121	203	632	382
5.20.10	394	339	146	381	1,260	355
7.14.10	255	169	149	611	1,184	416
7.14.10	1,159	1,068	1,635	5,347	9,209	426
7.14.10	9,570	7,962	8,640	13,715	39,887	444
7.14.10	4,531	3,466	3,643	18,367	30,007	417
7.14.10	8,596	2,753	5,858	27,415	44,622	207
8.16.10	54,388	82,035	67,184	325,938	529,545	563
8.16.10	2,172	2,880	14,461	7,949	27,462	568
8.16.10	268,415	144,488	326,430	748,802	1,488,135	554
8.16.10	3,264	1,296	9,702	8,986	23,248	567
8.16.10	31,432	20,465	19,580	191,868	263,345	539
10.20.5	5,581,150	1,314,618	2,518,922	5,926,822	15,341,512	642
10.20.5	14,883	7,963	21,208	16,840	60,894	663
10.20.5	5,270,687	2,435,068	5,055,210	4,654,387	17,415,352	677
10.20.5	4,625,886	1,477,862	2,603,142	3,259,608	11,966,498	683
10.20.5	5,262,177	2,159,272	1,867,008	3,694,172	12,991,629	624
10.20.10	2,731	802	3,380	6,238	13,151	3,225
10.20.10	808	473	813	653	2,747	1,922
10.20.10	9,753	1,960	22,345	13,497	47,555	4,385
10.20.10	2,985	961	2,995	3,988	11,929	3,344
10.20.10	2,894	962	1,932	2,650	8,438	3,624

Acknowledgement This research was partially supported by the National Science Foundation and the Office of Naval Research through grants CMMI-0620755 and N000140910332, respectively. Their support is gratefully acknowledged. We are grateful to George Nemhauser and Juan-Pablo Vielma for making available to us the instances of their paper [11].

References

[1] E.M.L. Beale and J.A. Tomlin, "Special Facilities in a General Mathematical Programming System for Nonconvex Problems Using Ordered Sets of Variables," in J. Lawrence (Ed.), *Proceedings of the Fifth International Conference on Operations Research*, Tavistock Publications, 1970, pp. 447-454.

[2] G.B. Dantzig, "On the Significance of Solving Linear Programming Problems with some Integer Variables," *Econometrica* 28, 30-44 (1960).

[3] I.R. de Farias JR., "Semi-Continuous Cuts for Mixed-Integer Programming," in D. Bienstock and G.L. Nemhauser (Eds.), *Integer Programming and Combinatorial Optimization (IPCO)*, Lecture Notes in Computer Science, Vol. 3064, Springer, 2004, pp. 163-177.

[4] I.R. de Farias JR., "Semi-Continuous Cuts for Mixed-Integer Programming," Department of Industrial Engineering, Texas Tech (2011).

[5] I.R. de Farias JR., E.L. Johnson, and G.L. Nemhauser, "A Generalized Assignment Problem with Special Ordered Sets: A Polyhedral Approach," *Mathematical Programming* 89, 187–203 (2000).

[6] I.R. de Farias, E.L. Johnson, and G.L. Nemhauser, "Branch-and-Cut for Combinatorial Optimization Problems without Auxiliary Binary Variables," *Knowledge Engineering Review* 16, 25-39 (2001).

[7] I.R. de Farias JR. and G.L. Nemhauser, "A Polyhedral Study of the Cardinality Constrained Knapsack Problem," *Mathematical Programming* 96, 439-467 (2003).

[8] http://www.hpcc.ttu.edu/index.php.

[9] A.B. Keha, I.R. de Farias JR., and G.L. Nemhauser, "A Branch-and-Cut Algorithm without Binary Variables for Nonconvex Piecewise Linear Optimization," *Operations Research* 54, 847-858 (2006).

[10] J.P. Vielma, S. Ahmed, and G.L. Nemhauser, "Mixed-Integer Models for Nonseparable Piecewise Linear Optimization: Unifying Framework and Extensions," *Operations Research* 58, 303-315 (2010).

[11] J.P. Vielma and G.L. Nemhauser, "Modeling Disjunctive Constraints with a Logarithmic Number of Binary Variables and Constraints," *Mathematical Programming*, in press.

[12] M. Zhao and I.R. de Farias JR., "Branch-and-Cut for Separable Piecewise Linear Optimization: New Inequalities and Intersection with Semi-Continuous Constraints," Department of Industrial Engineering, Texas Tech University (2010).