A New Subadditive Approach to Integer Programming: Theory and Algorithms

Diego Klabjan
Department of Mechanical and Industrial Engineering
University of Illinois at Urbana-Champaign
klabjan@uiuc.edu

1 Introduction

The LP duality was established a long time ago. With each primal problem there is an associated dual problem with the same objective value. Many algorithms compute both a primal and a dual solution, e.g. simplex and primal-dual algorithms. Given a dual vector we define the reduced cost of a column, which estimates how much would the addition of the column change the objective value, and the sensitivity analysis can be carried out using the dual information. Large-scale LPs can be efficiently solved with SPRINT, see e.g. Anbil et al. (1992). The idea of SPRINT is to solve many small LP subproblems and gradually add columns to the subproblem based on the reduced cost values. Columns with small reduced cost are more likely to improve the incumbent solution and therefore they are appended to the subproblem.

This paper addresses the questions of a dual function, reduced cost, and sensitivity analysis for IPs. The original motivation for this study was in designing an algorithm for large-scale IPs that mimics SPRINT. Such an approach solves at each iteration an IP consisting of a small subset of columns. Columns are then appended to the current IP and the problem is reoptimized. The key question related to the success of this approach is what columns to append, i.e. what is the equivalent notion to the LP reduced cost. Dual vectors are also used in the Benders decomposition algorithms, see e.g. Nemhauser and Wolsey (1988). Currently these approaches can be applied only to mixed integer linear programs since they require a dual vector. But if applied to IPs, they raise the question of a dual function for IPs. In LP it is known that all the alternative optimal solutions can be found among the columns with zero reduced cost. It would be very useful if we can produce alternative IP solutions, e.g. among several optimal solutions we can select the most robust one. Again, as shown in this work, an available dual function for IPs makes such a task much easier.

For integer programs the subadditive duality developed first by Johnson (1973) gives us a partial answer to these questions.

Definition 1. *A function* $F : \mathbb{R}^m \to \mathbb{R}$ *is subadditive on* $Z \subseteq \mathbb{R}^m$ *if* $F(x + y) \leq F(x) + F(y)$ *for*

all $x \in Z, y \in Z$ *such that* $x + y \in Z$.

If Z is not specified, we assume $Z = \mathbb{R}^m$. Johnson showed that for a feasible IP

$$\begin{array}{lcl} \min \ cx & & \max \ F(b) \\ Ax = b & = & F(a_i) \le c_i \qquad i = 1, \ldots, n \\ x \in \mathbb{Z}^n_+ & & F \ \text{subadditive}\,, \end{array} \tag{1}$$

where $A = (a_1, \ldots, a_n) \in \mathbb{Z}^{m \times n}, b \in \mathbb{Z}^m, c \in \mathbb{Z}^n$. We refer to the second problem as the *subadditive dual problem*. At least theoretically the answer to all of the raised questions are in the *optimal subadditive function* (OSF) F. In other words, the analog to the optimal dual vector in LP is the OSF. The reduced cost of a column i can be defined as $c_i - F(a_i)$ and most of the other properties from LP carry over to IP, e.g. complementary slackness, $F(b)$ provides a lower bound on the optimal IP value, and the alternative optimal solutions can be found only among the columns i with $c_i = F(a_i)$. However there are still two fundamental issues that need to be addressed; how to encode F and how to compute F. Theory tells us that an OSF can always be obtained as a composition of C-G inequalities, see e.g. Nemhauser and Wolsey (1988), but such a function would be hard to encode and hard to evaluate. Very little is known about how to compute an OSF. Llewellyn and Ryan (1993) show how an OSF can be constructed from Gomory cuts. Our work originates from the work done by Burdet and Johnson (1977), where an algorithm for solving an IP based on subadditivity is presented. Both of these two works do not present any computational experiments.

We give a new family of subadditive functions that is easy to encode and often easy to evaluate. We present an algorithm that computes an OSF. As part of the algorithm we give several new theorems that further shed light on OSFs. The contribution of this research goes beyond a novel methodology for computing an OSF. There are many implications of having an OSF: the reduced cost can be defined, the sensitivity analysis can be performed, we can obtain alternative optimal solutions, and new approaches for large-scale IPs can be developed ('integer' SPRINT, Benders decomposition for IPs).

In Section 2 we present a new subadditive function that is computationally tractable. We give several interesting properties of this function. In addition we generalize the concept of reduced cost fixing. Section 3 first outlines the algorithm that computes an optimal primal solution and an OSF. We show how to obtain an OSF given an OSF of the preprocessed problem and we show how to obtain an initial subadditive function from the LP formulation with clique inequalities. In this section we also present the entire algorithm that computes an optimal primal solution and an OSF. In the last section we report the computational experiments. We conclude with a brief description of the Burdet-Johnson algorithm.

The Burdet-Johnson Algorithm

Burdet and Johnson show that there is a subadditive function $\pi : \mathbb{R}^n \to \mathbb{R}$ such that $\pi(e_i) \le c_i$ for every column i, and the optimal value z^{IP} to the IP equals to $\min_{x \text{ feasible to IP}} \sum_{i \in N} \pi_i x_i$. Here e_i

is the ith unit vector. Note that such a function does not serve our purpose since it is defined on the set of all the columns and not rows. Based on π we do not see an appropriate way to define the reduced cost. However we do use such a π to obtain a good starting point for our desired F.

Their subadditive function is based on the concept of generator subsets. Given a subinclusive set $E \in \mathbb{Z}^n_+$, i.e. if $y \in E$ then for every $0 \leq x \leq y$ it follows that $x \in E$, and a vector $\delta \in \mathbb{R}^n$ they define a function $\pi(x) = \max_{y \in E \cap \underline{S}(x)} cy + \delta(x - y)$, where $\underline{S}(x) = \{y \in \mathbb{Z}^n_+ : y \leq x\}$. A candidate set H is defined as $H = \{x \in \mathbb{Z}^n_+ : x \notin E, \underline{S}(x) \setminus \{x\} \subseteq E\}$. They showed that if

$$\pi(x_1 + x_2) \leq cx_1 + cx_2 \tag{2}$$

for all $x_1 \in E, x_2 \in E$ and $x_1 + x_2 \in H$, then π is subadditive.

Their algorithm consists of two steps. The enumeration step selects an element from H and appends it to E. This operation must be followed by an update of H and δ has to be scaled to satisfy (2). The second step then further improves the dual objective value by adjusting δ. Given fixed E and H the maximum dual objective value is obtained by solving the LP

$$\begin{aligned}
&\max \ \pi_0 && \\
\pi_0 - \delta(x - y) &\leq cy && \text{for all } y \in E, x \text{ feasible to IP, and } y \leq x && (3)\\
\delta x &\leq cx && x \in H && (4)\\
\delta, \pi_0 &\ \text{unrestricted}\,. &&
\end{aligned}$$

It can be seen that (3) give the dual objective value and that (4) are equivalent to (2). These two steps are then iterated until an optimal solution is found.

We have tried to implement this algorithm but we could not solve even very small problems. We took their ideas and we have substantially enhanced them, e.g. among many elements in H they do not specify which one to select. In addition we chose δ in such a way that we get a good starting point for F. We present our algorithm in Section 3.3.

2 The Generator Subadditive Functions

Here we present the approach only for the set partitioning problems $\min\{cx : Ax = \mathbf{1}, x \ \text{ binary}\}$, where $\mathbf{1}$ is a vector with all components equal to 1. In addition, we assume that the problem is feasible. The extension to general IPs, the theory for infeasible IPs, and additional results are given in Klabjan (2001). All of the proofs can be found in this paper.

Given a vector $\alpha \in \mathbb{R}^m$, we define a generator subadditive function $F_\alpha : \mathbb{R}^m \to \mathbb{R}$ as

$$\begin{aligned}
F_\alpha(d) = \alpha d - \max & \sum_{i \in E} (\alpha a_i - c_i) x_i \\
& A_E x \leq d \\
& x \ \text{ binary}\,,
\end{aligned}$$

where $E = \{i \in N : \alpha a_i > c_i\}$ is a generator set and A_E is the submatrix of A corresponding to the columns in E. The generator set E depends on α but for simplicity of notation we do not show this dependence in our notation. Whenever an ambiguity can occur, we write $E(\alpha)$. In addition, for simplicity of notation we write $H = N \setminus E$.

It is easy to see that F_α is a subadditive function. It is also easy to see that $F_\alpha(a_i) \leq \alpha a_i \leq c_i$ for all $i \in H$ by taking $x = 0$ in $\max\{(\alpha A^E - c^E)x : A^E x \leq a_i, x \in \mathbb{Z}_+^{|E|}\}$ and that $F_\alpha(a_i) \leq c_i$ for all $i \in E$ by considering $x = e_i$ in $\max\{(\alpha A^E - c^E)x : A^E x \leq a_i, x \in \mathbb{Z}_+^{|E|}\}$ and therefore $F_\alpha(a_i) \leq c_i$ for all $i \in N$. This shows that F_α is a feasible subadditive function and therefore $F_\alpha(\mathbf{1})$ provides a lower bound on z^{IP}. The vector α is a generalization of dual vectors of the LP relaxation. Every dual feasible vector α to the LP relaxation has to satisfy $\alpha a_i \leq c_i$ for all $i \in N$, however α in the definition of F_α can violate some of these constraints. Indeed, if y^* is an optimal solution to the dual of the LP relaxation of the IP, then $E = \emptyset$ and F_{y^*} gives the value of the LP relaxation.

The equivalence (1) states that among all the subadditive dual functions there is one that attains the equality however it does not say anything for specially structured subadditive functions like F_α.

Theorem 1. *There exists an α such that F_α is an OSF, i.e. $F_\alpha(\mathbf{1}) = z^{IP}$.*

Since potentially we want to use F_α to compute reduced costs $c_i - F(a_i)$ for many columns i, e.g. the idea of an integral SPRINT, it is desirable that the cardinality of E is small. Our computational experiments show that this is indeed the case in practice. Even problems with 100,000 columns have only up to 300 columns in E.

Next we give two theorems that have a counterpart in LP and are used in our algorithm.

Theorem 2 (Complementary slackness). *Let x^* be an optimal primal solution. If $x_i^* = 1$, then $\alpha a_i \geq c_i$.*

This theorem easily follows from the general complementary condition $x_i^*(c_i - F(a_i)) = 0$.

Since F_α is always a valid subadditive function, $F(\mathbf{1})$ is a lower bound on z^{IP}. In IP the reduced cost fixing based on an LP solution is a commonly used technique for fixing the variables to 0. The next theorem establishes an equivalent condition based on a subadditive dual function.

Theorem 3 (Extended reduced cost fixing). *Let F be a feasible subadditive dual function, i.e. F is subadditive and $F(a_i) \leq c_i$ for all $i \in N$, and let $\bar{z}^{IP}$ be an upper bound on z^{IP}. If $c_i - F(a_i) \geq \bar{z}^{IP} - F(\mathbf{1})$ for a column $i \in N$, then there is an optimal solution with $x_i = 0$.*

If F is a subadditive valid function, then $\sum_{i \in N} F(a_i)x_i \geq F(\mathbf{1})$ is a valid subadditive function, Nemhauser and Wolsey (1988). Therefore for $F = F_\alpha$ we get that

$$\sum_{i \in E} c_i x_i + \sum_{i \in H} (\alpha a_i) x_i \geq F_\alpha(\mathbf{1}) \tag{5}$$

is a valid inequality. These inequalities are used in our computational experiments.

2.1 Basic and Minimal Generator Subadditive Functions

Note that the generator subadditive functions form an infinite family of functions since α is arbitrary. In linear programming extreme points suffice to solve the dual problem and there are only a finite number of them. A similar result holds for the generator subadditive functions. Namely, it suffices to consider only those generator subadditive functions, called *basic generator subadditive functions*, with α an extreme point of a certain polyhedra. These functions allows us to extend the traditional Benders decomposition algorithm for mixed integer programs to integer programs. The details are given in Klabjan (2001). In addition, if (5) is a facet, then F_α is a basic generator subadditive function.

A valid inequality is minimal if is not dominated by any other valid inequality. A generator subadditive function is *minimal* if (5) is a minimal valid inequality. Minimal generator subadditive functions have two interesting properties. Consider the set packing problem $\max\{(\alpha A^E - c^E)x : A^E x \leq \mathbf{1}, x \text{ binary}\}$. F_α is a minimal generator subadditive function if and only if this set packing problem has an optimal solution x^* such that there is a binary vector $x' \geq x^*$ with $A(x' + x^*) = \mathbf{1}$. Perhaps a more interesting property is the following. If F_α is minimal generator subadditive function, than for every $i \in E$ there is an optimal solution x^* to the set packing problem with $x^*_i = 1$. This statement shows that this set packing problem has a wide variety of optimal solutions. Since a generator OSF is minimal, these properties can facilitate us in designing algorithms for obtaining all optimal primal solutions.

3 Solution Methodology

We first briefly describe the main ideas of our algorithm that finds a primal optimal solution and it simultaneously computes an OSF F_α.

We first preprocess the problem. We use all of the 9 preprocessing rules described in Borndorfer (1998) except that we do not check for dominated columns. In addition, we have detected 2 more preprocessing rules. In the next step we solve the LP relaxation of the preprocessed problem and we apply clique inequalities, Nemhauser and Wolsey (1988). They are separated with a standard heuristic. Next we form an initial E by considering the dual prices of the LP relaxation with cliques. We show in Section 3.2 that this is a feasible step.

Computational experiments have shown that finding in one 'attempt' an OSF is not computationally tractable. Instead we gradually remove columns that do not change the optimal value from the problem. This has an implication that the resulting OSF is not necessarily subadditive in $\mathbb{R}^m$ but it is only subadditive on a subset of still active columns. The algorithm proceeds in 3 stages. In stage 1 an optimal primal solution and an optimal subadditive function is found. However the subadditive function is only subadditive on a small subset of columns. In this stage we use the Burdet-Johnson framework. In stage 2 we improve subadditivity by obtaining a subadditive function that satisfies the complementary slackness conditions. The resulting function is still not necessarily subadditive on all the columns but it turns out that only few columns violate

subadditivity. In stage 3 we do the last correction to make the function subadditive on $\mathbb{R}^m$.

3.1 Preprocessing and the Generator Subadditive Functions

The problem is first preprocessed and then a generator OSF is found. Since we want to obtain an OSF to the original problem, we have to show how to get such a function given an OSF to the preprocessed problem. In preprocessing the preprocessing rules are iteratively applied. All of the 10 preprocessing rules are applied in several passes until at the last pass we do not further reduce the problem. To be able to construct an OSF of the original problem, we have to show how each preprocessing rule affects the subadditive function. If we want to easily construct the function, then we have to show how to obtain an OSF $F_{\bar{\alpha}}$ from an OSF F_α of the preprocessed problem, after one step of a given preprocessing rule. It is crucial here that the new function is again a generator subadditive function since otherwise we loose the structure and it would be hard to successfully unwind the preprocessing steps.

Fortunately for most of the preprocessing rules we can apply the following proposition. For an $i \in N$ let $A_i = \{j \in M : a_{ji} = 1\}$, where $M = \{1, \ldots, m\}$ is the set of all the rows, and similarly for $j \in M$ let $A^j = \{i \in N : a_{ji} = 1\}$.

Proposition 1. *Let F_α be an OSF for $\min\{cx : Ax = 1, x \text{ binary}\}$ and let r be a row of A. Let a_{n+1} be a new column with the cost c_{n+1}. Furthermore, let $r \notin A_{n+1}$, let $\sum_{i \in A^r} x_i + x_{n+1} \leq 1$ be a valid inequality for $\{(x, x_{n+1}) : Ax + a_{n+1}x_{n+1} \leq 1, x \text{ binary}, x_{n+1} \text{ binary}\}$, and let us define $F_\alpha(a_{n+1}) = \alpha a_{n+1}$. Then the LP*

$$\begin{aligned} \min \ & \mathbf{1}y & \\ & a_i y \leq c_i - F_\alpha(a_i) & i \in N - A^r \\ & (a_i - \mathbf{1})y \leq c_i - F_\alpha(a_i) & i \in A^r \text{ and } i = n+1 \\ & \mathbf{1}y \geq 0 & \end{aligned}$$

has an optimal solution y^* and $F_{\alpha+y^*}$ is an OSF for $\min\{cx + c_{n+1}x_{n+1} : Ax + a_{n+1}x_{n+1} = 1, x \text{ binary}, x_{n+1} \text{ binary}\}$.

Let us show how to use the theorem for dominated rows. If $r, n+1$ are two rows such that $A^r \subseteq A^{n+1}$, then we can fix to 0 all the variables in $A^{n+1} \setminus A^r$ and we can remove row $n+1$ from the problem. Suppose now that $F_{\bar{\alpha}}$ is an OSF for the problem without row $n+1$ and the columns in $A^{n+1} \setminus A^r$. First we can append back row $n+1$ without the columns in $A^{n+1} \setminus A^r$ to the matrix and define $\alpha = (\bar{\alpha}, 0) - (\bar{\alpha}_r/2)e_r + (\bar{\alpha}_r/2)e_{n+1}$. It is easy to see that F_α is an OSF for the new problem. Now we can handle the removed columns from $A^{n+1} \setminus A^r$ by Proposition 1 to get an OSF for the original problem.

3.2 Clique Inequalities and the Generator Subadditive Functions

Consider the LP relaxation together with some clique inequalities. The LP value gives a lower bound on z^{IP}. The next theorem shows how to obtain a generator subadditive function from such

an LP.

Theorem 4. *Let*

$$z^{clq} = \min\{cx : Ax = 1, \sum_{i \in C_j} x_i \leq 1 \quad j \in J, 0 \leq x\}\,,$$

where $C_j, j \in J$, are cliques in the conflict graph. Let α be the optimal dual vector corresponding to constraints $Ax = 1$. Then F_α has the value at least z^{clq}, i.e. $F_\alpha(\mathbf{1}) \geq z^{clq}$.

If we can solve a set partitioning problem to optimality by adding clique inequalities, then the generator OSF is readily available.

We show in Klabjan (2001) that the theorem holds for any IP $\min\{cx : Ax = b, x \geq 0, x \text{ integer}\}$, if we add valid inequalities for $\min\{cx : Ax = b, x \geq 0, x \text{ integer}\}$. Combining this theorem and Theorem 3 we obtain that we can fix $x_i = 0$ if $c_i - \alpha a_i \geq \bar{z}^{IP} - z^{\text{clq}}$, where α is as in the theorem. This is a strong result since it allows to reduce the size of an IP by reduced cost fixing. We are not aware of any such variable fixing in past literature.

In our implementation we first run a primal heuristic to obtain $\bar{z}^{IP}$. Then we solve the LP relaxation with cliques and we run the primal heuristic again but this time with the strengthen formulation with cliques. At the end we apply reduced cost fixing.

3.3 Stage 1: Obtaining an Optimal Primal Solution with the Use of Subadditive Functions

In this stage we find an optimal primal solution and an approximate subadditive dual function by using the Burdet-Johnson framework. Since we are interested in obtaining a subadditive function in $\mathbb{R}^m$, we choose $\delta = \alpha A$, where α is an unknown vector. This mapping also substantially simplifies (3) since now they read $\pi_0 - \mathbf{1}\alpha + \alpha(Ay) \leq cy$ for all $y \in E$.

We first enhance their algorithm with a concept that is the equivalent to pruning in branch-and-bound algorithms. Instead of searching for a subadditive function in all $\mathbb{R}^n$ we require subadditivity only on $\{x : Ax = 1, cx < \bar{z}^{IP}, x \text{ binary}\}$, where $\bar{z}^{IP}$ is the best known upper bound on z^{IP}. It means that we can remove from H all the elements that yield an objective function value larger than $\bar{z}^{IP}$. Namely, if $h \in H$ and

$$\min\{cx : Ax = 1, x_i = 1 \text{ for all } i \text{ with } h_i = 1, x \geq 0\} \tag{6}$$

is greater or equal to $\bar{z}^{IP}$, then h can be removed from H. However computing this LP for every element in H is too expensive and therefore we compute the exact value only for the element that is selected to be added to E. Note that the cardinality of H can double each iteration. For the candidate element h, if the objective value of (6) is greater or equal to $\bar{z}^{IP}$, then h is permanently removed from consideration and the selection process is repeated.

The second major enhancement we employ is the selection of an element from H that is added to E. In view of the above discussion, we would like to append to E such an element that introduces the fewer number of new elements in H. Note that the elements in H yield constraints (4) and

therefore it is desirable to have only a small number of them. It can be seen that this is equivalent to maximizing (6) and therefore selecting an element from H that has the highest objective value of (6) is beneficial for both goals, i.e. pruning and having a small H. We avoid computing (6) for every element in H by using pseudo costs, Nemhauser and Wolsey (1988). On the other hand, when an element is moved from H to E, the corresponding constraint (4) is relaxed into a constraint (3). Therefore we would like to relax the most binding constraint, i.e. a constraint with $\alpha Ah = ch$.

The overall selection can be summarized as follows. Among all the elements h in H that satisfy $\alpha Ah = ch$, we select the element with the largest pseudo cost since such an element is most likely to be pruned and it keeps H small. Next we solve (6) and if the element can be pruned, we permanently delete it and we repeat the selection procedure. Otherwise the element is added to E. If the solution to (6) is integral and better than the best known primal solution so far, we update the best known solution.

At every iteration we apply the extended reduced cost fixing and we add to the problem (6) the subadditivity cut resulting from the current subadditive function.

Note that at the end we assert that there are no feasible solution, i.e. the current solution is optimal. We are optimal when $\{x : Ax = 1, cx < \bar{z}^{IP}, x \text{ binary}\}$ is empty, which implies that there is a subadditive function with infinity objective value. Therefore the stopping criteria is when the dual objective π_0 becomes greater or equal to the best optimal primal value $\bar{z}^{IP}$.

3.4 Stage 2: Improving the Feasibility of the Subadditive Dual Function

The goal of this stage is to obtain a subadditive function that is subadditive on the set $\{x : Ax = 1, cx \leq z^{\text{IP}}, x \text{ binary}\}$. Such a function has to satisfy the complementary slackness conditions.

This stage is very similar to stage 1. In stage 1 we record E and H every time we obtain a better primal solution. The last recorded E and H are used as a warm start for stage 2. The algorithm follows closely the algorithm in stage 1 with a few changes. The selection of an element from H that is added to E is now more targeted toward improving the dual objective.

3.5 Stage 3: Computing a Generator Optimal Subadditive Function

In this stage we obtain a generator OSF. We take α from stage 2 as a starting point. F_α is an OSF on all the columns that have not been pruned in stage 2. This F_α is not necessarily an OSF since the columns i that have been pruned may have $\alpha a_i > c_i$. In this case we would have to add i to E, which can potentially decrease $F(\mathbf{1})$. We call the columns that have been pruned in stage 2 and have $\alpha a_i > c_i$ infeasible columns. Typically we have several thousand infeasible columns. Stage 3 makes these columns feasible by modifying E and α.

First we try to increase the number of feasible columns by slightly adjusting α. The procedure is based on the following theorem.

Theorem 5. *Let F_α be a generator OSF for the set partitioning problem with the input data c and A, let c_{n+1}, a_{n+1} be a column with $\alpha a_{n+1} > c_{n+1}$, and assume that* $\min\{\sum_{i \in N}(c_i - F_\alpha(a_i))x_i +$

$(c_{n+1} - \alpha a_{n+1})x_{n+1} : Ax + a_{n+1}x_{n+1} = 1, x \geq 0, x_{n+1} \geq 0\} = 0$. *If* $\bar{\alpha}$ *is an optimal dual vector to this LP, then* $F_{\alpha+\bar{\alpha}}$ *is a generator OSF for the set partitioning problem with the input data* (c, c_{n+1}) *and* (A, a_{n+1}).

Note that if $x_{n+1} = 0$ in the above LP, then the objective value is 0 since all of the coefficients are nonnegative and the optimal IP solution gives a solution with 0 objective value. Therefore the condition in the theorem is likely to hold. We apply this theorem iteratively for all the infeasible columns. If the condition is not met, then we leave the column infeasible. This procedure reduces the number of infeasible columns from several thousand to only a dozen.

Obtaining feasibility for remaining infeasible columns is the most computationally intensive part of the entire algorithm. For each infeasible column i we proceed as follows. We add i to E to obtain a generator subadditive function that satisfies the complementary slackness but it is not necessarily optimal since the addition of a new column to E can reduce the value of $\max\{\sum_{i \in E}(\alpha a_i - c_i)x_i^E : A_E x^E \leq 1, x^E \text{ binary}\}$. Given E we maximize the dual value by solving the LP

$$\begin{aligned} \max \ & \pi_0 & & \\ \pi_0 - \mathbf{1}\alpha + \alpha A_E x &\leq c_E x & & \text{for all binary } x, A_E x \leq \mathbf{1} \\ \alpha a_i &\leq c_i & & i \in H\ . \end{aligned} \tag{7}$$

Given a solution to this LP, we choose a column from H with the largest dual value and we move it from H to E. The process is repeated until the objective value π becomes z^{IP}. This LP is solved by row generation.

4 Computational Experiments

The computational experiments were carried out on the set partitioning instances used by Hoffman and Padberg (1993) and Eso (1999). They were performed on an IBM Thinkpad 570 with a 333 MHz Pentium processor and 196 MBytes of main memory.

The results are presented in Table 1. Instances with an integral solution at the root node are left out as are some other hard instances that we are currently working on. All the times are CPU execution times in seconds. The column 'inf' shows the number of infeasible columns before starting the final step of the feasibility improvement. We observe that there are only a few instances where this final adjustment is needed, however, stage 3 is computationally expensive. It is important to note that the cardinality of E is always small and therefore evaluating $F_\alpha(d)$ should not be computationally hard, which makes approaches such as the integer SPRINT algorithm potentially computationally tractable. In the last two instances we have exceeded the maximum execution time of 2 hours in stage 3. The instance denoted by † is infeasible and our algorithm establishes this property by finding an unbounded LP (7).

The overall computational times are acceptable for a methodology that reveals much more information about an IP instance, e.g. we can perform the sensitivity analysis, alternative optimal solutions can be found only among the columns with $F_\alpha(a_i) = c_i$. It is unreasonable to expect

that the computational times are lower than branch-and-cut computational times since the latter algorithm finds only a primal optimal solution. Nevertheless this computational results show that obtaining an optimal subadditive dual is possible.

size		preprocessing			time					CPLEX
rows	cols	rows	cols	time	stage 1	stage 2	stage 3	$\mid E \mid$	inf	time
825	8627	537	6695	30	25	7	164	142	0	88
55	7479	47	5915	49	12	63	523	93	13	14
59	43749	53	38958	6	29	0	0	19	0	48
50	6774	37	5964	45	10	0	0	58	0	8
124	10757	81	7861	90	100	0	0	292	0	13
22	685	22	536	1	1	0	0	13	0	0
19	711	15	416	1	4	0	0	52	0	2
19	1366	19	926	2	1	0	0	52	0	0
18	2540	18	2034	6	3	5	14	50	4	0
26	2653	26	1877	22	1	0	0	27	0	0
26	2662	26	1728	14	1	0	0	13	0	0
19	294	17	250	1	2	0	0	28	0	0
23	3068	23	2308	26	2	2	3	16	1	0
20	1783	20	1244	15	5	0	14	67	12	0
23	1079	20	791	4	1	0	0	30	0	0
100	13635	44	5197	289	247	0	0	287	0	17
163	28016	95	4080	53	3	0	0	25	0	68
†104	2775	63	519	47	438					240
173	3686	150	3528	59	39	0	0	223	0	5
111	1668	73	967	26	3	0	0	97	0	18
801	8308	521	6236	21	106	19	?	?	1	97
646	7292	486	5858	23	52	23	?	?	5	39

Table 1: Computational Results

References

Anbil, R., Johnson, E. and Tanga, R. 1992. A global approach to crew pairing optimization. *IBM Systems Journal* **31**, 71–78.

Borndorfer, R. 1998. *Aspects of Set Packing, Partitioning, and Covering*. PhD thesis. Technical University of Berlin.

Burdet, C. and Johnson, E. 1977. A subadditive approach to solve integer programs. *Annals of Discrete Mathematics* **1**, 117–144.

Eso, M. 1999. *Parallel Branch and Cut for Set Partitioning*. PhD thesis. Cornell University.

Hoffman, K. and Padberg, M. 1993. Solving airline crew scheduling problems by branch-and-cut. *Management Science* **39**, 657–682.

Johnson, E. 1973. Cyclic groups, cutting planes and shortest path. *In* T. Hu and S. Robinson (eds), *Mathematical Programming*. Academic Press. 185–211.

Klabjan, D. 2001. A new subadditive approach to integer programming: Theory and algorithms. *Technical report*. University of Illinois at Urbana-Champaign. Available from http://www.staff.uiuc.edu/∼klabjan/professional.html.

Llewellyn, D. and Ryan, J. 1993. A primal dual integer programming algorithm. *Discrete Applied Mathematics* **45**, 261–275.

Nemhauser, G. and Wolsey, L. 1988. *Integer and combinatorial optimization*. John Wiley & Sons.