Exact Algorithms for Combinatorial Optimization Problems with Submodular Objective Functions ⋆

Frank Baumann[1], Sebastian Berckey[2], and Christoph Buchheim[1]

[1] Fakultät für Mathematik, Technische Universität Dortmund
{frank.baumann, christoph.buchheim}@tu-dortmund.de
[2] INFORM Institut für Operations Research und Management GmbH
sebastian.berckey@inform-software.com

Abstract. Many combinatorial optimization problems have natural formulations as submodular minimization problems over well-studied combinatorial structures. A standard approach to these problems is to linearize the objective function by introducing new variables and constraints, yielding an extended formulation. We propose two new approaches for constrained submodular minimization problems. The first is a linearization approach that requires only a small number of additional variables. We exploit a tight polyhedral description of this new model and an efficient separation algorithm. The second approach uses Lagrangean decomposition to create two subproblems which are solved with polynomial combinatorial algorithms; the first subproblem corresponds to the objective function while the second consists of the constraints. The bounds obtained from both approaches are then used in a branch and bound-algorithm. We apply our general results to problems from wireless network design and mean-risk optimization. Our experimental results show that both approaches compare favorably to the standard techniques.

1 Introduction

Many combinatorial optimization problems can be naturally modeled as an integer program in which the objective function is not linear but submodular. Submodularity is a property of set functions. Given a set S, a function $f\colon 2^S \longrightarrow \mathbb{R}$ is called *submodular*, if for each pair of subsets $A, B \subseteq S$ the property

$$f(A \cup B) + f(A \cap B) \leq f(A) + f(B)$$

holds. It is easy to see that the class of submodular functions comprises the class of linear set functions. Submodularity can be interpreted as *diminishing returns*: Writing the above property as

$$f(A \cup B) \leq f(A) + f(B) - f(A \cap B)$$

shows that the profit generated by the combined set $A \cup B$ is less than (or equal to) the sum of the individual profits. A simple example of a submodular function is the maximum function. Given a weight w_s for each element s of S, the function

$$f(A) = \max_{s \in A} w_s$$

returns the weight of the heaviest element in the subset.

Integer programs with submodular objective functions are usually solved using one of the following approaches: (a) The problem is addressed by general nonlinear mixed-integer programming techniques such as second-order cone programming, or (b) The problem is reformulated as a linear program. It can then be solved with branch and cut-algorithms based on LP-relaxation. The latter technique has two disadvantages. Linearization of the objective function is usually achieved by introducing a large number of new binary variables and constraints linking them to the original variables. The resulting linear formulation tends to

⋆ This work was partially supported by the German Science Foundation, Priority Programme 1307 *Algorithm Engineering*, under contract BU 2313/1.

be much larger than the natural nonlinear formulation. Moreover, these reformulations are not generic but exploit the structure of the underlying combinatorial problem. Therefore, such linear formulations are known only for few combinatorial optimization problems with submodular costs.

In this paper, we aim at generic approaches for submodular combinatorial optimization problems. After describing the problem class under consideration and fixing notation in Section 2, we present two such general techniques: a polyhedral approach using a compact linear model in Section 3 and a Lagrangean decomposition approach in Section 4. In Section 5, we describe how to apply our results to range assignment problems and a problem from portfolio optimization, risk-averse capital budgeting. We present computational results and compare the effectiveness of the proposed techniques to existing state-of-the-art algorithms in Section 6.

An extended abstract covering some of the results presented in this paper has appeared in the proceedings of ISCO 2010 [4].

2 The Problem

In the following we consider combinatorial optimization problems of the form

$$\begin{aligned} &\min\ f(x) \\ &\text{ s.t. } x \in X \subset \{0,1\}^S, \end{aligned} \tag{1}$$

where $f\colon 2^S \longrightarrow \mathbb{R}$ is a submodular function on a set S. Without loss of generality we can assume $f(\emptyset) \geq 0$. We associate each binary variable x_i with an element of S. The set X contains all feasible solutions of the problem. In this paper, we generally assume that a *linear* objective function can be optimized over X in polynomial time or, equivalently, that the separation problem for $P := \operatorname{conv} X$ is polynomially solvable. From a practical point of view, the methods proposed can also be applied to problems where the linear optimization problem over X can be solved sufficiently fast or where a tight polyhedral description of P is known.

Unconstrained submodular function minimization has been proven to be polynomially solvable and several fully combinatorial algorithms for this problem exist [13, 22]. In the presence of linear constraints, however, the problem often becomes *NP*-hard. This is the case even if optimizing a linear objective function subject to the same constraints is easy.

Commonly, model (1) is either solved directly, as a nonlinear integer program, or it is reformulated as an integer linear program (ILP). ILP models have the advantage that they are well studied and state-of-the-art solvers are extremely efficient. A downside of considering an extended linear formulation is that the linearization often can only be achieved by introducing a large number of new variables and linear constraints to the model, reducing the advantage of using linear solvers considerably. Additionally, such reformulations often not only affect the objective function but also the original constraints, obscuring or even destroying the combinatorial structure of the problem.

3 A Cutting Plane Approach

In the following, we study the polyhedral structure of submodular combinatorial optimization problems. We describe a class of linear inequalities that gives a complete description of the corresponding polyhedron in the unconstrained case. Combined with the polyhedral description of the set of feasible solutions X we obtain an LP-relaxation of Problem (1). We also present an efficient separation algorithm for the linearization inequalities.

Starting from the unconstrained nonlinear model

$$\begin{aligned} &\min\ f(x) \\ &\text{ s.t. } x \in \{0,1\}^S \end{aligned}$$

we introduce a single new variable $y \in \mathbb{R}$ to replace the objective function. Clearly, the resulting model

$$\begin{aligned} \min\ & y \\ \text{s.t. } & y \geq f(x) \\ & x \in \{0,1\}^S \\ & y \in \mathbb{R} \end{aligned}$$

is equivalent to the original one, since we consider a minimization problem. Now consider the convex hull of feasible points:

$$P_f = \operatorname{conv}\left\{(x, y) \in \{0,1\}^S \times \mathbb{R} \mid y \geq f(x)\right\}$$

The following result by Edmonds [9] and Lovász [15] gives a complete polyhedral description of P_f.

Theorem 1. *Let $|S| = n$ and let $f\colon 2^S \to \mathbb{R}$ be a submodular function with $f(\emptyset) \geq 0$. Then the separation problem for P_f can be solved in $O(n \log n)$ time. The facets of P_f are either induced by trivial inequalities $0 \leq x_i \leq 1$, $i \in S$, or by an inequality $a^\top x \leq y$ with*

$$a_{\sigma(i)} = f(S_i) - f(S_{i-1}) \text{ for all } i \in \{1, \ldots, n\} \tag{2}$$

where $\sigma\colon \{1, \ldots, n\} \to S$ is any bijection and $S_i = \{\sigma(j) \mid j \in \{1, \ldots, i\}\}$.

In the presence of constraints the above theorem does not yield a complete polyhedral description anymore, but it still provides strong dual bounds on the LP-relaxation, as we will see later. The number of facets of P_f is exponential in $n = |S|$, but the separation problem can be solved efficiently by a simple greedy algorithm. Indeed, violation of the trivial facets is checked in linear time. The following algorithm produces a candidate for a separating hyperplane:

Given a fractional point $(x^\star, y^\star) \in [0,1]^S \times \mathbb{R}$, sort the elements of S in non-increasing order according to their value in $x^\star$. Starting with the empty set, iteratively construct a chain of subsets $\emptyset = S_0 \subset S_1 \subset \cdots \subset S_n = S$ by adding the elements in this order. The potentially violated inequality $a^\top x \leq y$ is then constructed by setting $a_i = f(S_i) - f(S_{i-1})$. Obviously this algorithm constructs an inequality of the form (2) that is most violated by the fractional point $(x^\star, y^\star)$. Either this inequality is a separating hyperplane or none such exists. A formal description of this separation procedure is given in Algorithm 1.

Algorithm 1 Separation Algorithm for P_f

input: a fractional solution $(x^\star, y^\star) = (x_1^\star, \ldots, x_n^\star, y^\star)$
output: a hyperplane $a^\top x \leq y$ separating $(x^\star, y^\star)$ from P_f, if one exists

sort the elements of S into a list $\{l_1, \ldots, l_n\}$ by non-increasing value of $x^\star$
$i \leftarrow 1$
$S_0 \leftarrow \emptyset$
repeat
  $S_i \leftarrow S_{i-1} \cup \{l_i\}$
  $a_i = f(S_i) - f(S_{i-1})$
  $i \leftarrow i + 1$
until $i = n$
if $y^\star < a^\top x^\star$ **then**
  return a
else
  return *no constraint found*
end if

In many applications, the submodular objective function f can be written as a conic combination of other submodular functions f_i, i.e., we have

$$f = \sum_{i=1}^{k} \alpha_i f_i, \quad \alpha_1, \ldots, \alpha_k \geq 0, \quad f_1, \ldots, f_k \text{ submodular.}$$

This situation can be exploited by modeling each function f_i separately, introducing a new continuous variable y_i modeling $f_i(x)$ for each $i \in \{1, \ldots, k\}$. Such an approach could be preferable if, e.g., the values $f_i(x)$ are used at other points in the model or if the functions f_i have much smaller domains than f. In the latter case, the total number of inequalities needed to describe the unconstrained problem can be reduced significantly.

We obtain

$$\begin{aligned} \min\ & \textstyle\sum_{i=1}^{k} \alpha_i y_i \\ \text{s.t. } & y_i \geq f_i(x) \text{ for all } i \in \{1, \ldots, k\} \\ & x \in \{0,1\}^S \\ & y \in \mathbb{R}^k. \end{aligned} \tag{3}$$

Our next aim is to show that the separation algorithm detailed above can still be used to generate a complete description for Problem (3). First note that Theorem 1 yields a complete description of the polytope P_{f_i} for each $i \in \{1 \ldots, k\}$. For the following, define

$$P = \bigcap_{i \in \{1,\ldots,k\}} P_{f_i} \ ,$$

where each P_{f_i} is trivially extended from $\{0,1\}^S \times \mathbb{R}$ to $\{0,1\}^S \times \mathbb{R}^k$. We will show that each vertex (x, y) of P satisfies $x \in \{0,1\}^S$ and $y_i = f_i(x)$, and hence is feasible for Problem (3). In other words, the separation problem corresponding to (3) can be reduced to the single separation problems for each P_{f_i}.

Lemma 1. *For any submodular function* $f\colon \{0,1\}^S \to \mathbb{R}$ *and* $j \in S$*, there is a submodular function* $g\colon \{0,1\}^{S\setminus\{j\}} \to \mathbb{R}$ *such that* $\{x \in P_f \mid x_j = 0\} = P_g$.

Proof. For $x \in \{0,1\}^{S\setminus\{j\}}$, let $\bar{x}$ be its extension to $\{0,1\}^S$, setting $\bar{x}_j = 0$. Defining $g(x) = f(\bar{x})$ yields the desired submodular function. □

Lemma 2. *For any submodular function* $f\colon \{0,1\}^S \to \mathbb{R}$ *and* $j \in S$*, there is a submodular function* $g\colon \{0,1\}^{S\setminus\{j\}} \to \mathbb{R}$ *such that* $\{x \in P_f \mid x_j = 1\} = e_j + P_g$*, where* e_j *denotes the unit vector corresponding to* x_j.

Proof. For $x \in \{0,1\}^{S\setminus\{j\}}$, let $\bar{x}$ be its extension to $\{0,1\}^S$, setting $\bar{x}_j = 1$. Defining $g(x) = f(\bar{x})$ yields the desired submodular function. □

Lemma 3. *If* $(x, y) \in P$ *with* $x \in (0,1)^S$*, then* (x, y) *is not a vertex of* P.

Proof. Let $\mathbf{1}_S$ denote the all-ones vector in $\mathbb{R}^S$ and choose $\varepsilon > 0$ such that $x \pm \varepsilon \mathbf{1}_S \in [0,1]^S$. Define $c \in \mathbb{R}^k$ by $c_i = f_i(S) - f_i(\emptyset)$ and consider

$$z_1 = (x - \varepsilon \mathbf{1}_S, y - \varepsilon c),\ \ z_2 = (x + \varepsilon \mathbf{1}_S, y + \varepsilon c)\ .$$

As $(x, y) = \frac{1}{2}(z_1 + z_2)$, it suffices to show $z_1, z_2 \in P$. This reduces to showing $(x \pm \varepsilon \mathbf{1}_S, y_i \pm \varepsilon c_i) \in P_{f_i}$ for all $i \in \{1, \ldots, k\}$. By Theorem 1, the polyhedron P_{f_i} is completely described by trivial inequalities and by inequalities of the type $a^\top x \leq y_i$ with

$$a_{\sigma(j)} = f_i(S_j) - f_i(S_{j-1}) \text{ for all } j \in \{1, \ldots, n\}$$

where $\sigma\colon \{1,\dots,n\} \to S$ is any bijection and $S_j = \{\sigma(1),\dots,\sigma(j)\}$. We obtain in particular that $a^\top \mathbf{1}_S = f_i(S) - f_i(\emptyset) = c_i$. As $(x,y) \in P$ and therefore $(x,y_i) \in P_{f_i}$, we derive

$$a^\top (x \pm \varepsilon \mathbf{1}_S) = a^\top x \pm \varepsilon a^\top \mathbf{1}_S \leq y_i \pm \varepsilon c_i \; .$$

Hence $z_1, z_2 \in P_{f_i}$. $\square$

Theorem 2. *The vertices of P are exactly the points $(x,y) \in \{0,1\}^S \times \mathbb{R}^k$ with $y_i = f_i(x)$ for all $i \in \{1,\dots,k\}$.*

Proof. It is clear that every such point is a vertex of P. We show that every vertex (x',y') of P is of this form. Since y_i is not bounded from above, every vertex must satisfy $y_i = f_i(x)$ for all $i \in \{1,\dots,k\}$. Now assume that at least one component of x' is fractional. Define

$$S_0 = \{j \in S \mid x'_j = 0\}, \quad S_1 = \{j \in S \mid x'_j = 1\}, \quad T = S \setminus \{S_0 \cup S_1\}$$

and consider the face

$$\begin{aligned} F &= \{(x,y) \in P \mid x_j = 0 \text{ for all } j \in S_0, \;\; x_j = 1 \text{ for all } j \in S_1\} \\ &= \cap_{i=1}^k \{(x,y) \in P_{f_i} \mid x_j = 0 \text{ for all } j \in S_0, \;\; x_j = 1 \text{ for all } j \in S_1\} \; . \end{aligned}$$

By Lemma 1 and Lemma 2, the polyhedron F is an intersection of polyhedra $\mathbf{1}_{S_1} + P_{g_i}$ for suitable submodular functions $g_i\colon \{0,1\}^T \to \mathbb{R}$. Since $x'_i \in (0,1)$ for all $i \in T$, the point $(x' - \mathbf{1}_{S_1}, y')$ is not a vertex of $\cap_{i=1}^k P_{g_i}$ by Lemma 3. It follows that (x',y') is not a vertex of F and hence not a vertex of P. $\square$

Theorem 2 implies that the polyhedron P_f is a projection of P, given by the linear transformation $y := \sum_{i=1}^k \alpha_i y_i$. Moreover, it follows that each facet of P is obtained from a facet of one of the polyhedra P_{f_i}. In particular, the separation problem for (3) can be reduced to the respective separation problems for each polyhedron P_{f_i} as follows: To separate a point $x^\star$ from the polytope P_f it is sufficient to check if $x^\star$ violates any of the inequalities characterizing the polyhedra P_{f_i}. This can be done by applying Algorithm 1 to each P_{f_i} in turn.

So far we have only considered unconstrained submodular optimization problems. Recall that the original Problem (1) was given as

$$\begin{aligned} &\min f(x) \\ &\text{ s.t. } x \in X \subset \{0,1\}^S, \end{aligned}$$

where X is the set of feasible solutions. The problem can thus be formulated as

$$\begin{aligned} &\min y \\ &\text{ s.t. } (x,y) \in (X \times \mathbb{R}) \cap P_f \subset \{0,1\}^S \times \mathbb{R} \; . \end{aligned}$$

In this case our results remain applicable but do not necessarily give a complete polyhedral description of the problem anymore. Even if the complete linear description of X (or an exact separation algorithm for X) is available, the combination of the inequalities describing X and P_f in general does not yield a full description of the intersection $(X \times \mathbb{R}) \cap P_f$. For an exact algorithm the generation of cutting planes can be embedded in a branch and bound-approach. In each node of the branch and bound-tree the seaparation routines for X and P_f are used to generate cutting planes and then resolve the LP-relaxation until the solution is integral or no more violated inequalities are found. In this case a branching step on one of the binary variables is performed.

4 A Lagrangean Decomposition Approach

In this section we avoid the linearization of the submodular objective function and capitalize on the existence of polynomial time algorithms for submodular functions in the unconstrained case. We use Lagrangean decomposition to separate the objective function from its constraints. For this purpose we add a new variable set x_2 of the same size as the original variable set to Problem (1):

$$\begin{aligned} \min\ & f(x_1) \\ \text{s.t. } & x_1 = x_2 \\ & x_1 \in \{0,1\}^S \\ & x_2 \in X \subset \{0,1\}^S \end{aligned} \tag{4}$$

Our next aim is to eliminate the coupling constraint $x_1 = x_2$ and split the problem into two parts which are separately solvable. This is done by moving the constraint to the objective function through the introduction of Lagrangean multipliers $\lambda_i \in \mathbb{R}$. We obtain

$$Z(\lambda) = \begin{array}{l} \min\ f(x_1) - \lambda^T x_1 \\ \text{s.t. } x_1 \in \{0,1\}^S \end{array} + \begin{array}{l} \min\ \lambda^T x_2 \\ \text{s.t. } x_2 \in X \subset \{0,1\}^S. \end{array} \tag{5}$$

It is well known that $Z(\lambda)$ is a lower bound of Problems (1) and (4) for every $\lambda \in \mathbb{R}^S$. The second part of (5) is an integer linear problem which is assumed to be solvable in polynomial time in our context. For fixed λ, the first part is an unconstrained submodular minimization problem, as $-\lambda^T x_1$ is a modular function. Therefore Lagrangean decomposition enables us to calculate dual bounds for Problem (1) in polynomial time for given Lagrangean multipliers $\lambda \in \mathbb{R}^S$. To obtain the best dual bound in connection with Lagrangean relaxation we can use the so-called Lagrangean dual, i.e. the maximum over all possible Lagrangean mulipliers:

$$\begin{aligned} Z_D := \max\ & Z(\lambda) \\ \text{s.t. } & \lambda \in \mathbb{R}^S \end{aligned}$$

It is well-known that $Z(\lambda)$ is a concave function in λ which can be maximized by the subgradient method. To apply this method we need a supergradient of $Z(\lambda)$ for all $\lambda \in \mathbb{R}^S$.

Lemma 4. *For a given point $\lambda^\star \in \mathbb{R}^S$, let $x_1^\star$ and $x_2^\star$ be minimizers of the two components of (5), respectively. Then $x_2^\star - x_1^\star$ is a subgradient of Z in $\lambda^\star$.*

Proof. $Z(\lambda)$ is concave and for $\lambda \in \mathbb{R}^S$ we have

$$Z(\lambda^\star) + (\lambda - \lambda^\star)^T (x_2^\star - x_1^\star) = f(x_1^\star) + \lambda^T (x_2^\star - x_1^\star) \geq Z(\lambda). \quad \square$$

These dual bounds can replace the LP-based bounds in the branch and bound-approach described in Section 3. Furthermore, the second part of Problem (5) yields feasible solutions of Problem (1), which can be used to compute upper bounds. Because the second part is a stand-alone problem it is not necessary to solve it using linear optimization methods. If, for example, the set X contains all incidence vectors of spanning trees, the problem can be solved efficiently by Prim's or Kruskal's algorithm.

To solve the unconstrained submodular optimization problem one could use the linearization approach described above. However, for many submodular functions specialized combinatorial algorithms are available. Even if there are no specialized algorithms at hand there exist several combinatorial algorithms for submodular function minimization [13, 22], although these general approaches are not necessarily more efficient than solving the problem through linear optimization.

Theorem 3. *Let Z_D be defined as above, then*

$$\begin{aligned} Z_D = \min\ & y \\ \text{s.t. } & (x, y) \in P_f \\ & x \in \operatorname{conv} X\ . \end{aligned}$$

Proof. We can rewrite (4) as

$$\begin{aligned} \min\ & y \\ \text{s.t. } & x_1 = x_2 \\ & y \geq f(x_1) \\ & x_1 \in \{0,1\}^S \\ & x_2 \in X \subset \{0,1\}^S. \end{aligned}$$

By a general result for Lagrangean relaxation [11], Z_D is the minimum of y over the intersection of the two sets

$$\operatorname{conv}\{(x,y) \mid y \geq f(x),\ x \in \{0,1\}^S\} = P_f$$

and

$$\operatorname{conv}\{(x,y) \mid x \in X\} = (\operatorname{conv} X) \times \mathbb{R}. \quad \square$$

By Theorem 3, the best lower bound obtained by Lagrangean relaxation is the same as the LP bound discussed in Section 3. However, the Lagrange approach might allow a faster computation of this bound in practice, depending on the problem structure. We investigate this in Section 6.

5 Applications

5.1 Range Assignment Problems

As a first application we study a class of problems from wireless network design, so-called *range assignment problems*. When designing an ad-hoc wireless network one main objective is to minimize transmission costs subject to certain requirements concerning the network topology. In traditional wired networks, these transmission costs are roughly proportional to the length of all connections installed, so that the aim is to minimize the total length of all connections. In wireless networks, the transmission costs depend on the transmission ranges assigned to the nodes. The main difference lies in the so-called *multicast advantage*: if a node v reaches another node w, then it also reaches each node u that is closer to v than w, at no additional cost. Accordingly, the objective function of range assignment problems, i.e. the overall transmission power of the network needed to establish the specified connections, is nonlinear as a function of the connections.

Range assignment problems have been studied intensively in recent years and several exact algorithms have been proposed. Fuchs [10] showed that the problem of finding a minimum-power connected network with bidirectional links (the *symmetric connectivity problem*) is *NP*-hard. Althaus et al. [1, 2] proposed an ILP formulation for this problem which is based on a linear extended formulation. For each node of the network they introduce a new binary variable for each value the transmission power of the node can take in an optimal solution and express the objective function in terms of these new variables. Montemanni and Gambardella [18] apply a very similar technique, modeling the transmission power levels of the nodes incrementally, as the sum of artificial binary variables. A comparison of different formulations for the symmetric connectivity problem can be found in [20]. Note that all models mentioned above are extended linear formulations of the original problem and do not exploit the submodularity of the objective function directly. We do not know of any approach in the literature that needs only a constant number of artificial variables.

A second important variant of the range assignment problem is the *minimum power multicast problem*. Here the objective is to construct a network that allows unidirectional communication from a designated source node to a set of receiving nodes. All nodes of the network, including the receiving stations, can function as relay nodes, thereby passing on a signal on its way from the source node to the receivers. Special cases are the *unicast problem* and the *broadcast problem*. In the former communication is directed to a single receiving node, in the latter all nodes except the source are addressed. The general minimum power multicast problem can be considered a nonlinear variant of the minimum Steiner arborescence problem and therefore also is *NP*-hard. The unicast and the broadcast problem, on the other hand, are efficiently solvable. With only a single receiving station the problem reduces to finding a shortest path through the directed network from the source to the destination node. The linear variant of the broadcast problem is also known as the *optimum branching problem*. Several authors independently presented efficient algorithms to compute an optimal solution [6, 8, 5]. The nonlinear variant we consider in this paper is known to be *NP*-hard [10].

Many of the algorithms for the symmetric connectivity case can be easily adapted to multicasting. Additionally, Leggieri et al. [14] investigate the multicasting problem specifically and present a set covering formulation, as well as preprocessing techniques to reduce the problem size [21]. There are also flow-based ILP-formulations. One example can be found in the paper by Min et al. [17]. In the same paper the authors present two exact iterative algorithms which use LP-relaxations to compute lower bounds. An overview over existing IP models for the multicast problem can be found in [7].

We model the general range assignment problem in graph theoretic terms. The communication stations correspond to the set of nodes V of the graph, links between the stations to the set of weighted edges E. For the symmetric connectivity problem, the graph $G = (V, E, c)$ is undirected with edge costs c, for the multicast problem it is directed. The objective is to compute a subset of the edges such that certain restrictions on the topology of the network are satisfied and the overall transmission costs are minimal. Common to both models is the objective function: Given a subset of edges, for each node only the most expensive incident/outgoing edge is taken into account. Summing up these values gives the overall costs. Associating a binary variable x_{vw} to each edge $e = (v, w) \in E$, the objective function can be written as

$$f(x) = \sum_{v \in V} \max \{c_{vw} x_{vw} \mid vw \in E\}. \tag{6}$$

A central property of this objective function is that it is submodular:

Theorem 4. *For each $v \in V$ and for arbitrary $c \in \mathbb{R}^E$, the function*

$$f_v(x) = \max \{c_{vw} x_{vw} \mid vw \in E\}$$

is submodular. In particular, the function $f(x) = \sum_{v \in V} f_v(x)$ is submodular.

Proof. By definition, f_v is submodular if

$$f_v(A \cup B) + f_v(A \cap B) \leq f_v(A) + f_v(B)$$

for arbitrary sets $A, B \subseteq E$. We distinguish two cases:

(a) if $f_v(A) \geq f_v(B)$, then $f_v(A \cup B) = f_v(A)$ and $f_v(A \cap B) \leq f_v(B)$
(b) if $f_v(A) \leq f_v(B)$, then $f_v(A \cup B) = f_v(B)$ and $f_v(A \cap B) \leq f_v(A)$

In both cases, it follows that $f_v(A \cup B) + f_v(A \cap B) \leq f_v(A) + f_v(B)$. Finally, the function f is submodular, because it is a conic combination of submodular functions. □

The desired network topology is described by a set of feasible vectors X. Combining objective function and constraints, the general IP formulation for range assignment problems reads

$$\begin{aligned} \min \quad & \sum_{v \in V} \max \{c_{vw} x_{vw} \mid vw \in E\} \\ \text{s.t.} \quad & x \in X \ . \end{aligned} \tag{7}$$

The Standard Model As mentioned earlier, the standard linearization for this model found in the wireless networking literature is due to Althaus et al. [1]. They introduce new binary variables which model the possible values of the nonlinear terms in optimal solutions and add constraints linking the original variables to the new ones. The resulting problem reads

$$\begin{aligned} \min \quad & \sum_{vw \in E} c_{vw} y_{vw} \\ \text{s.t.} \quad & \sum_{vw \in E} y_{vw} \leq 1 && \text{for all } v \in V \\ & \sum_{c_{vu} \geq c_{vw}} y_{vu} \geq x_{vw} && \text{for all } vw \in E \\ & x \in X \\ & y \in \{0,1\}^E \ . \end{aligned} \tag{8}$$

In this model, the binary variable y_{vw} is thus set to one if and only if the transmission power of node v is just enough to reach node w. Note that, depending on the network topology described by X, the first set of constraints can be strengthened to equations. This is the case when all feasible edge-induced subgraphs are connected. In this case, each node has to reach at least one other node. In general, this is not true, so that for some v all variables y_{vw} can be zero. The number of variables in this model is $2|E|$.

A closely related model appearing in the literature [19] uses binary variables in an incremental way: again, a variable $y'_{vw} \in \{0,1\}$ is used for each pair of nodes v and w, now set to one if and only if node v can reach node w. It is easy to see that the two models are isomorphic by the transformation

$$y'_{vw} = \sum_{c_{vu} \geq c_{vw}} y_{vu} \ .$$

Because of this, the two models are equivalent from a polyhedral point of view and it suffices to consider the first model in the following.

New Mixed-Integer Models The general formulation for range assignment problems we gave above is already very similar to the model we studied in Section 3. We can now introduce a single artificial variable $y \in \mathbb{R}$ to move the objective function into the constraints. The corresponding model is

$$\begin{aligned} \min \quad & y \\ \text{s.t.} \quad & y \geq \sum_{v \in V} \max\{c_{vw}x_{vw} \mid vw \in E\} \\ & x \in X \\ & y \in \mathbb{R} \ . \end{aligned} \tag{9}$$

From Theorem 4 we know that the objective is submodular; this means that Theorem 1 is applicable and we have an efficient separation algorithm to construct a strong LP-relaxation of (9).

Theorem 4 showed that in the case of range assignment problems the objective function is not only submodular itself but also the sum of submodular functions. We can thus use the slightly larger mixed-integer model (3), which in our application reads

$$\begin{aligned} \min \quad & \textstyle\sum_{v \in V} y_v \\ \text{s.t.} \quad & y_v \geq f_v(x) \ \text{ for all } v \in V \\ & x \in X \\ & y \in \mathbb{R}^V \ . \end{aligned} \tag{10}$$

We know from Theorem 2 that we can again separate efficiently when ignoring the problem-specific constraints $x \in X$.

Polyhedral Relations In the following, we investigate the polyhedral properties of the standard model and the new mixed-integer models. First, we show how the corresponding polyhedra are related to each other. For this, let $P_1(X)$, $P_2(X)$, and $P_3(X)$ denote the polyhedra given as the convex hulls of feasible solutions in the models (8), (9), and (10), respectively. Note that $P_1(X)$ is a convex hull of binary vectors, so in particular it is a polytope and all its integral points are vertices. On the other hand, the polyhedra $P_2(X)$ and $P_3(X)$ are unbounded by definition. It is easy to see that $P_3(X)$ arises from the convex hull of

$$\left\{(x,y) \in X \times \mathbb{R}^V \mid y_v = \max\{c_{vw}x_{vw} \mid vw \in E\} \text{ for all } v \in V\right\}$$

by adding arbitrary nonnegative multiples of unit vectors for the variables y_v. Similarly, $P_2(X)$ arises from the convex hull of

$$\left\{(x,y) \in X \times \mathbb{R} \mid y = \sum_{v \in V} \max\{c_{vw}x_{vw} \mid vw \in E\}\right\}$$

by adding arbitrary nonnegative multiples of the unit vector for y.

Theorem 5. *The convex hull of all vertices of $P_3(X)$ is a projection of an integer subpolytope of $P_1(X)$.*

Proof. Consider the projection π_1 given by

$$y_v := \sum_{vw \in E} c_{vw} y_{vw} \ .$$

Let $(x, y) \in X \times \mathbb{R}^V$ be a vertex of $P_3(X)$. Then $y_v = \max\{c_{vw}x_{vw} \mid vw \in E\}$ for all $v \in V$. Thus setting $y_{vw} = 1$ for exactly one w with $y_v = c_{vw}$ yields a vertex of $P_1(X)$ that is mapped to (x, y) under π_1. ☐

In Section 3, we have shown that $P_2(X)$ is a projection of the polyhedron $P_3(X)$, so that Theorem 5 also holds if $P_3(X)$ is replaced by $P_2(X)$. These results show that for every reasonable objective function the optimal faces of all three polyhedra are projections of each other. The first model can thus be considered an extended formulation of the second and third one, and the second model can be considered an extended formulation of the third one.

Note that in general $P_1(X)$ contains vertices that are not mapped to the convex hull of vertices of $P_2(X)$ or $P_3(X)$. These vertices cannot be optimal for any of the considered objective functions.

5.2 Risk-Averse Capital Budgeting

As a second application we study the risk-averse capital budgeting problem. In portfolio theory an important concept is to not only consider the expected return when choosing a set of investments but also take into account the risk associated with investments. Such *mean-risk optimization problems* can be modeled using stochastic objective functions. Potential investment decisions are represented by independent random variables that have an associated mean value μ as well as a variance σ^2. The mean value stands for the expected return of the investments, σ^2 models the uncertainty inherent in the investment, i.e. the risk that the real return deviates from the expected. The case of continuous variables is well studied whereas the case of discrete variables has received relatively little attention yet.

We concentrate on the *risk-averse capital budgeting problem* with binary variables [3]. In this variant of the mean-risk optimization problem a set of possible investments characterized by their costs, expected return values and variances and a number ε are given as input. The number $\varepsilon > 0$ characterizes the level of risk the investor is willing to take. Investment decisions are binary, this means one can choose to make a certain investment or not. The only constraint in the risk-averse capital budgeting problem is a limit on the available budget. An optimal solution of the problem is a set of investment decisions and a solution value z. The choice of investments guarantees that with probability $1 - \varepsilon$ the portfolio will return at least a profit of z.

The corresponding nonlinear IP-model is

$$\begin{aligned} z = \max \sum_{i \in I} \mu_i x_i - \sqrt{\frac{1-\varepsilon}{\varepsilon} \sum_{i \in I} \sigma_i^2 x_i^2} \\ \text{s.t.} \sum_{i \in I} a_i x_i \le b \\ x \in \{0,1\}^I, \end{aligned}$$

where I is the set of available investments, a_i the cost of investment $i \in I$, and b the amount of capital that can be invested. The vector μ represents the expected returns of the investments and σ^2 the variance of the expected returns.

To apply the polyhedral results from Section 3 we need to rewrite the above model as a minimization problem and show that the objective function is submodular. Note that since the x-variables are binary we have $x_i^2 = x_i$. The problem now reads

$$\begin{aligned} z = -\min -\sum_{i \in I} \mu_i x_i + \sqrt{\frac{1-\varepsilon}{\varepsilon} \sum_{i \in I} \sigma_i^2 x_i} \\ \text{s.t.} \sum_{i \in I} a_i x_i \le b \\ x \in \{0,1\}^I \ . \end{aligned} \tag{11}$$

The first part of the objective function

$$f(A) = -\sum_{i\in A}\mu_i + \sqrt{\frac{1-\varepsilon}{\varepsilon}\sum_{i\in A}\sigma_i^2}$$

is obviously modular. The second part is the composition of a nondecreasing modular function and a nondecreasing concave function. It is easy to prove submodularity for a slightly more general class of functions:

Theorem 6. *Let $f\colon \mathbb{R} \longrightarrow \mathbb{R}$ be a concave function and $g\colon 2^S \longrightarrow \mathbb{R}$ a submodular function on the set S. If both f and g are nondecreasing, the composition*

$$h\colon 2^S \longrightarrow \mathbb{R}, \quad h(A) = f(g(A))$$

is submodular and nondecreasing.

Proof. The composition $f \circ g$ obviously is nondecreasing. To see that it is submodular, note that

$$\begin{aligned} A\cap B \subseteq B \subseteq A\cup B &\Rightarrow g(A\cap B) \le g(B) \le g(A\cup B) \\ &\Rightarrow \exists t \in [0,1] : g(B) = tg(A\cap B) + (1-t)g(A\cup B). \end{aligned}$$

We have

$$\begin{aligned} f(g(A)) &\ge f(g(A\cup B) + g(A\cap B) - g(B)) \\ &= f(tg(A\cup B) + (1-t)g(A\cap B)) \\ &\ge tf(g(A\cup B)) + (1-t)f(g(A\cap B)) \\ &= f(g(A\cup B)) + f(g(A\cap B)) - (tf(g(A\cap B)) + (1-t)f(g(A\cup B))) \\ &\ge f(g(A\cup B)) + f(g(A\cap B)) - f(tg(A\cap B) + (1-t)g(A\cup B)) \\ &= f(g(A\cup B)) + f(g(A\cap B)) - f(g(B)), \end{aligned}$$

since f is concave. □

Corollary 1. *The objective function of model (11) is submodular.*

Corollary 1 shows that we can adress the risk-averse capital budgeting problem (11) by the techniques described above.

6 Computational Results

In the following, we report results of branch and bound-algorithms based on the cutting plane approach of Section 3 and the Lagrangean approach of Section 4, respectively. For the implementation, we used the exact optimization software library SCIL [23]. The LP-relaxations at each node of the enumeration tree were solved with CPLEX 12.1. The subgradient method for the Lagrangean relaxation approach is implemented using the ConicBundle library v0.3.8 [12]. To calculate the subgradients, i.e. to optimize the second partial problem, we used an implementation of Edmonds' algorithm [25] for the broadcast problem and the Boost Graph Library 1.46.1 for graph modelling and basic graph algorithms [24].

All experiments were run on a 2.6 GHz AMD Opteron 252 processor. We set a time limit of one hour for each instance.

6.1 Symmetric Connectivity

As mentioned earlier the symmetric connectivity problem is a range assignment problem on an undirected graph G. To establish a connection between nodes u and v the transmission range of node u must be large enough to reach node v and vice versa. The set X in the general nonlinear model (1) specializes to the set of spanning subgraphs of G.

In this case, all three IP-formulations (8), (9), and (10) can be significantly strengthened. First of all, the set X can be restricted to the set of spanning trees in G without loss of generality. This is equivalent to introducing an additional constraint $\sum_{e \in E} x_e = |V| - 1$. This stronger formulation does not change the optimum of our problem, but improves the quality of the bounds obtained from the LP-relaxations and thus reduces running time. In our experiments we used the subtour formulation of the spanning tree polytope.

Another way to strengthen the model is related to the fact that in a connected subgraph (on at least two nodes) each node has at least one adjacent edge. For the standard model, this means that the constraints $\sum_{uv \in E} y_{uv} \leq 1$ can be strengthened to equations $\sum_{uv \in E} y_{uv} = 1$, for all $u \in V$. In the mixed-integer models (10) and (9) we can eliminate one variable from each maximum term. As the transmission power for each node v has to be at least the smallest weight $c_v^{\min}$ of the adjacent edges, this constant can be extracted from the corresponding maximum term. The constraints of model (10) become

$$y_v \geq c_v^{\min} + \max\left\{(c_{vw} - c_v^{\min})x_{vw} \mid vw \in E\right\} \text{ for all } v \in V \ ,$$

so that at least one entry in the maximum can be removed. In the compact model (9), the constraint that bounds the overall transmission power from below can be strengthened analogously. Both replacements lead to stronger LP-relaxations if the separation algorithms derived in Section 3 are now applied to the remaining maximum terms.

Turning to the Lagrangean relaxation approach, the structure of the set X, i.e. the set of all spanning trees, allows to apply fast combinatorial algorithms like Kruskal's or Prim's to the second problem in (5). The first problem is a special submodular function minimization problem. Even though currently no specialized combinatorial algorithm for this kind of submodular function is available in the case of undirected graphs, there exists one for directed graphs first described by Miller [16]. The algorithm is based on the fact that for directed graphs the minimization of the corresponding submodular function can be decomposed into $|V|$ smaller minimization problems

$$\sum_{v \in V} \min_{x \in \{0,1\}^{\delta(v)}} \left(\max_{e \in \delta(v)} \{c_e x_e\} - \sum_{e \in \delta(v)} \lambda_e x_e \right) ,$$

where $\delta(v) = \{vw \in E \mid w \in V\}$. This is due to the fact that each variable x_{vw} appears in only one of the minima and the variables are not linked by any constraints. The partial problems can be solved by Algorithm 2, which for each $e \in \delta(v)$ computes the optimal solution x satisfying $x_e = 1$ and $x_f = 0$ for $c_f > c_e$.

We mention that Algorithm 2 can also be implemented to run in linear time after sorting the coefficients c_e; the latter can be done in a preprocessing step, as it does not depend on the Lagrangean multipliers λ. To take advantage of this algorithm we will consider a directed version of the symmetric connectivity problem. To gain an equivalent directed formulation we double the variables and introduce new constraints $x_{vw} = x_{wv}$ for all $vw \in E$ where E is now the set of all directed edges between nodes in V. These new constraints will become part of the second problem in (5), so that a spanning tree algorithm can still be applied to the corresponding undirected graph where the weights (Lagrangean multipliers) of two edges vw and wv are summed up.

As an alternative, one could use an algorithm for general submodular function minimization or a linear programming approach to solve the first problem in (5) directly on the undirected instance. However, experiments show that the directed approach achieves similar or even slightly better bounds than the undirected one while being much faster in each step of the subgradient method. The following results for the Lagrangean decomposition approach are therefore based on the directed version of the symmetric connectivity problem.

Algorithm 2 Solution of partial problem

---

input: objective function $f_v(x) := \max_{e \in \delta(v)} \{c_e x_e\} - \sum_{e \in \delta(v)} \lambda_e x_e$

output: optimal solution of $\min_{x \in \{0,1\}^{\delta(v)}} f_v$

```
x* ← 0
opt ← 0
for e ∈ δ(v) do
    x ← 0
    sum ← c_e
    for f ∈ δ(v) do
        if c_f ≤ c_e and λ_f > 0 then
            x_f ← 1
            sum ← sum − λ_f
        end if
    end for
    if sum < opt then
        opt ← sum
        x* ← x
    end if
end for
return x*
```

To speed up the subgradient method, we use a warm start approach, using the best Lagrangean multipliers from the corresponding parent node as starting points. This leads to much lower number of iterations in general. Note that in most instances over 50% of the total time was spent in the root node to compute a good initial set of Lagrangean multipliers.

We generated random range assignment instances by randomly placing points on a 10000×10000 grid, as proposed by Althaus et al. [1]. For each size, 50 instances were created. The transmission power needed for node u to reach node v was chosen as $d(u,v)^2$, where $d(u,v)$ is the Euclidian distance between u and v. Table 1 summarizes our results for the symmetric connectivity problem. The first column shows the size of the instances, the second the average number of subproblems computed in the branch and cut-tree. The column LPs/LCs contains the average number of linear programs solved (for the cutting plane approaches) and the average number of times the Lagrangean dual was solved (for the decomposition approach), respectively. t_{tot}/s is the average overall time needed to solve the instance. For the cutting plane approaches we also give the time spent on separation (t_{sep}/s). The last column shows how many of the 50 instances of each size could be solved within the time limit of one hour. For the computation of averages only instances that could be solved to optimality were considered.

It turned out that the compact model (9) is not competitive. Because only a single inequality of the description of the objective function can be separated per iteration, the number of LPs grows quickly in comparison to the other models. The medium-sized model (10) gives the best results for instances up to 15 nodes, also compared to the Lagrangean decomposition approach. The number of subproblems is significantly smaller than for the standard model, which compensates the larger number of LPs. For instance size 20 the decomposition approach performs best. For the largest instances the standard model gives the best results, because the time spent per node in the other models becomes too large. It is remarkable that several instances could not be solved at all within the time limit, whereas the average solution time for the other instances is relatively small and only grows moderately with the instance size.

Table 1. Results for the symmetric connectivity range assignment problem.

n	subs	LPs/LCs	t_{sep}/s	t_{tot}/s	# solved
		standard IP model (8)			
10	29.48	32.20	0.00	0.08	50
15	1147.96	1217.28	0.18	12.28	50
20	4048.22	4461.83	3.63	114.65	46
25	2248.40	2513.51	4.35	117.99	43
		MIP model (10)			
10	23.28	70.70	0.01	0.08	50
15	823.04	2597.38	0.98	7.29	50
20	2820.51	11049.13	16.17	100.79	45
25	3353.15	14001.85	45.21	281.17	41
		Lagrangean decomposition			
10	18.84	282.26	-	0.63	50
15	180.00	3007.10	-	20.43	50
20	749.83	11871.20	-	106.39	48
25	1668.22	26067.50	-	324.14	41

6.2 Multicast

We next investigate the min-power multicast problem. Recall that its objective is to send signals wireless from a designated source node to a set of receiving stations at minimum cost. Transmissions are unidirectional and all stations can relay signals through the network. Treating this problem as a graph optimization problem, there obviously is a one-to-one correspondence between feasible solutions and Steiner Arborescences in the directed graph. The multicast advantage can again be expressed by (6), this time for directed graphs. We used a separation routine for the cut formulation of the Steiner arborescence polytope to model the network topology in both cutting plane models.

The given connectivity constraints can again be used to strengthen the three different formulations, however to a lesser extent than in the symmetric case. Only the fact that at least one edge has to leave the source node provides a way to strengthen the models.

Table 2. Results for the multicast range assignment problem with $|T| = \lfloor \frac{n-1}{2} \rfloor$.

n	subs	LPs	t_{sep}/s	t_{tot}/s	# solved
		standard IP model (8)			
10	32.68	57.10	0.00	0.08	50
15	117.92	241.84	0.09	0.93	50
20	2991.52	6444.50	8.43	71.18	50
25	5773.97	27788.03	64.02	383.07	39
		MIP model (10)			
10	28.92	111.92	0.01	0.10	50
15	88.24	556.38	0.28	1.14	50
20	951.32	7815.54	11.39	33.28	50
25	5650.17	73373.59	208.51	571.88	46

Table 2 shows the results for the multicast problem. The number of terminal nodes is $\lfloor \frac{n-1}{2} \rfloor$. As mentioned before the decomposition approach is not applicable here, because there is no efficient algorithm for the

Steiner tree problem. We used the same instances as for the symmetric connectivity problem. The source and terminal nodes were determined randomly. For this kind of problem exploiting the submodularity of the objective function clearly pays off. While for small instances both models give similar results, the better polyhedral description in the MIP model significantly reduces running times for larger instances. 46 of the largest instances could be solved to proven optimality within the time limit of one hour, compared to 39 with the standard model.

6.3 Broadcast

Since the problem of finding a minimal Steiner Arborescence is *NP*-hard the Lagrangean decomposition approach is inefficient for general multicast problems. However, the set of feasible solutions for the broadcast problem corresponds to the set of s-Arborescences for which the minimization problem can be solved in polynomial time [8]. The first problem in (5) can then again be solved by the algorithm described for the symmetric connectivity problem. Table 3 shows that the Lagrangean decomposition approach is able to solve the highest number of the large instances, while remaining competitive for the smaller instances. The MIP approach is slowed down by the large number of LPs and the resulting high number of calls to the separation routines.

Table 3. Results for the broadcast range assignment problem.

n	subs	LPs/LCs	t_{sep}/s	t_{tot}/s	# solved
		standard IP model (8)			
10	36.16	45.02	0.00	0.09	50
15	167.24	243.04	0.07	1.31	50
20	1519.40	2801.68	4.24	36.53	50
25	7117.19	16238.07	32.57	375.87	43
		MIP model (10)			
10	32.88	120.70	0.02	0.12	50
15	142.52	776.78	0.43	1.79	50
20	1051.76	8796.32	13.79	42.87	50
25	6101.98	69896.47	200.10	598.74	43
		Lagrangean decomposition			
10	25.72	350.14	-	0.53	50
15	447.32	3674.34	-	7.08	50
20	2437.36	20767.40	-	55.22	50
25	32657.40	245163.00	-	875.73	44

6.4 Risk-Averse Capital Budgeting

In contrast to the range-assignment problem presented before, no linear model for this problem is known. Atamtürk and Narayanan [3] present a solution approach that solves the above model using second-order cone programming embedded in a branch and bound-algorithm. They use the inequalities of Theorem 1 to strengthen the relaxation in each node of the enumeration tree. Table 4 shows our results for the risk-averse capital budgeting problem. We solved the set of random instances from Atamtürk and Narayanan [3], which have between 25 and 100 variables. The larger instances with up to 1000 variables were generated using the same method as for the smaller instances. The expected returns μ and the costs a are independent random numbers between 0 and 100. The variances σ are chosen as the expected returns multiplied by an independent random number between 0 and 1. The available budget is $\frac{1}{2}\sum_{i\in I} a_i$. This ensures the existence of feasible solutions and at the same time excludes the trivial case where the budget is large enough to make all investments.

Table 4. Results for the risk-averse capital budgeting problem

n	ε	subs	LPs	time/s	n	ε	subs	LPs	time/s
25	.10	52.60	147.00	0.07	500	.10	515.00	1847.80	3.93
	.05	43.40	182.40	0.10		.05	1083.40	6768.60	14.38
	.03	21.40	151.80	0.06		.03	895.00	8714.00	20.61
	.02	6.20	74.60	0.03		.02	1479.00	20914.60	54.44
	.01	4.60	37.60	0.02		.01	2420.60	73099.80	267.04
50	.10	145.40	409.80	0.14	600	.10	907.00	3295.80	8.34
	.05	107.80	528.00	0.15		.05	1277.80	7710.80	20.65
	.03	67.00	409.40	0.12		.03	658.20	6830.40	20.38
	.02	77.80	634.60	0.20		.02	2207.40	28467.80	96.58
	.01	19.00	513.00	0.20		.01	1378.60	38536.80	197.85
100	.10	210.60	625.60	0.28	700	.10	1531.40	7823.20	23.47
	.05	132.20	852.40	0.42		.05	1064.20	6240.80	19.90
	.03	251.40	2276.00	1.21		.03	1391.80	18167.40	65.54
	.02	252.60	2015.80	1.19		.02	1970.20	32083.20	135.11
	.01	205.00	4877.40	3.34		.01	1616.60	51970.80	336.48
200	.10	315.40	1041.60	0.91	800	.10	922.20	3778.20	13.20
	.05	387.80	2012.60	1.78		.05	1648.60	11625.00	43.78
	.03	323.40	2928.20	2.78		.03	1623.80	14574.20	59.50
	.02	415.40	4644.60	5.10		.02	1612.60	22405.00	98.44
	.01	407.00	18369.20	29.35		.01	2330.20	83383.80	553.18
300	.10	623.80	3103.40	3.86	900	.10	690.20	1710.80	7.03
	.05	324.60	2377.20	3.10		.05	456.60	1715.60	7.53
	.03	391.40	3475.80	4.81		.03	1049.40	5700.20	27.65
	.02	411.40	6940.00	11.24		.02	3505.80	20868.00	117.43
	.01	364.20	20375.60	59.27		.01	6004.60	120758.40	929.79
400	.10	682.20	2884.40	4.83	1000	.10	1601.40	3730.20	17.77
	.05	990.60	9454.00	16.33		.05	1313.80	4198.80	21.11
	.03	1094.60	12793.20	23.64		.03	480.60	2496.60	14.31
	.02	311.40	6138.20	12.83		.02	1968.20	12609.60	77.33
	.01	2409.40	56058.00	165.37		.01	2925.40	71134.20	636.85

We generated five instances of each size and solved each instance for the values of ε given in the table. All values given (number of subproblems, number of linear programs and the running time in seconds) are averages over these five instances.

It can be observed that the value of ε has a strong impact on the running times. For decreasing ε the problem becomes harder to solve. This was already observed by Atamtürk and Narayanan [3]. A direct comparison of our results with the results for the second-order cone programming approach in [3] shows that the number of nodes in the branch and bound-tree is much smaller when our MIP model is used. It is also remarkable that in our model the number of violated inequalities separated is much higher.

As can be seen from Table 4 we were able to solve instances of size 50 in 0.2 seconds on average. The times reported by Atamtürk and Narayanan [3] for the same instances vary between 2 and 79 seconds for different values of ε. Also for $n = 100$ our algorithm is significantly faster for all values of ε. While with the second-order cone programming approach only instances of up to 100 variables could be solved within half an hour, our cutting plane approach easily solves instances of size 1000. Especially remarkable is the fact that the number of subproblems grows only moderately with the instance size.

We did not apply the Lagrangean relaxation approach to risk-averse capital budgeting problems, as we do not know of any fast algorithm for solving the first problem of the decomposition (5) in this case. Using a general purpose submodular function minimizer did not yield satisfactory results.

7 Conclusion

We propose two exact algorithms for solving combinatorial optimization problems with submodular objective functions. Both approaches are tailored for problems that become tractable whenever the submodular objective function is replaced by a linear function. Our algorithms are based on a branch and bound-scheme, where bounds are computed by either a cutting plane approach or by Lagrangean relaxation. The performance of both approaches depends on the underlying problem structure and on the given submodular objective function. If the latter can be minimized very efficiently (ignoring the problem constraints), as is typically the case for range assignment problems, the Lagrangean approach turns out to be very effective. The LP-based approach is applicable to general submodular functions; it yields a flexible and fast solution method, as demonstrated by our results for the risk-averse capital budgeting problem. Our experiments show that treating the objective function and the underlying constraints separately still yields tight relaxations.

Bibliography

[1] E. Althaus, G. Calinescu, I. I. Mandoiu, S. Prasad, N. Tchervenski, and A. Zelikovsky. Power efficient range assignment in ad-hoc wireless networks. In *In Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC '03)*, pages 1889–1894, 2003.

[2] E. Althaus, G. Calinescu, I. I. Mandoiu, S. Prasad, N. Tchervenski, and A. Zelikovsky. Power efficient range assignment for symmetric connectivity in static ad hoc wireless networks. *Wirel. Netw.*, 12(3): 287–299, 2006.

[3] A. Atamtürk and V. Narayanan. Polymatroids and mean-risk minimization in discrete optimization. *Oper. Res. Lett.*, 36(5):618–622, 2008.

[4] F. Baumann and C. Buchheim. Submodular formulations for range assignment problems. In M. Haouari and A. R. Mahjoub, editors, *Proceedings of ISCO 2010*, volume 36 of *Electronic Notes in Discrete Mathematics*, pages 239–246, 2010.

[5] F. Bock. An algorithm to construct a minimum directed spanning tree in a directed network. In *Developments in operations research*, pages 29–44, 1971.

[6] Y. Chu and T. Liu. On the shortest arborescence of a directed graph. *Sci. Sin.*, 14:1396–1400, 1965.

[7] A. K Das, R. J Marks, M. El-Sharkawi, P. Arabshahi, and A. Gray. Minimum power broadcast trees for wireless networks: Integer programming formulations. In *INFOCOM 2003*, volume 2, pages 1001–1010, 2003.

[8] J. Edmonds. Optimum branchings. *Journal Of Research Of The National Bureau Of Standards*, 71B: 233–240, 1967.

[9] J. Edmonds. Submodular functions, matroids, and certain polyhedra. In Michael Jünger, Gerhard Reinelt, and Giovanni Rinaldi, editors, *Combinatorial Optimization – Eureka, You Shrink!*, volume 2570 of *Lecture Notes in Computer Science*, pages 11–26. Springer Berlin / Heidelberg, 2003.

[10] B. Fuchs. On the hardness of range assignment problems. *Netw.*, 52(4):183–195, 2008.

[11] A. M. Geoffrion. Lagrangean relaxation for integer programming. *Mathematical Programming Study*, 2: 82–114, 1974.

[12] C. Helmberg. The ConicBundle Library for Convex Optimization. `www-user.tu-chemnitz.de/~helmberg/ConicBundle`, 2011.

[13] S. Iwata and J. B. Orlin. A simple combinatorial algorithm for submodular function minimization. In *Proceedings of the twentieth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '09, pages 1230–1237, 2009.

[14] V. Leggieri, P. Nobili, and C. Triki. Minimum power multicasting problem in wireless networks. *Math. Meth. Oper. Res.*, 68:295–311, 2008.

[15] L. Lovász. Submodular functions and convexity. In *Mathematical programming: the state of the art (Bonn, 1982)*, pages 235–257. Springer, Berlin, 1983.

[16] P.M. Miller. Exakte und heuristische Verfahren zur Lösung von Range-Assignment-Problemen. Master's thesis, Universität zu Köln, 2010.

[17] M. Min, O. Prokopyev, and P. Pardalos. Optimal solutions to minimum total energy broadcasting problem in wireless ad hoc networks. *J. Comb. Opt.*, 11:59–69(11), 2006.

[18] R. Montemanni and L. M. Gambardella. Minimum power symmetric connectivity problem in wireless networks: A new approach. In *MWCN*, pages 497–508, 2004.

[19] R. Montemanni and L. M. Gambardella. Exact algorithms for the minimum power symmetric connectivity problem in wireless networks. *Comp. Oper. Res.*, 32:2891–2904, 2005.

[20] R. Montemanni, L. M. Gambardella, and A. Das. Mathematical models and exact algorithms for the min-power symmetric connectivity problem: an overview. In Jie Wu, editor, *Handbook on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless, and Peer-to-Peer Networks*, pages 133–146. CRC Press, 2006.

[21] P. Nobili, S. Oprea, and C. Triki. Preprocessing techniques for the multicast problem in wireless networks. In *Proceedings of the Conference MTISD 2008*, pages 131–134, 2008.

[22] J. B. Orlin. A faster strongly polynomial time algorithm for submodular function minimization. *Mathematical Programming*, 118(2):237–251, November 2007.
[23] SCIL. SCIL – Symbolic Constraints in Integer Linear programming. `scil-opt.net`, 2011.
[24] J. G. Siek, L. Lee, and A. Lumsdaine. *The Boost Graph Library: User Guide and Reference Manual (C++ In-Depth Series)*. Addison-Wesley Professional, December 2001. ISBN 0201729148.
[25] E. Sjölund and A. Tofigh. Edmonds' algorithm. `edmonds-alg.sourceforge.net`, 2010.