Additive bounds [1]

Giovanni Righini
Università degli Studi di Milano

Operations Research Complements

UNIVERSITÀ DEGLI STUDI DI MILANO

[1] Based on M. Fischetti, P. Toth, “An additive bounding procedure for combinatorial optimization problems”, Operations Research 37, 2 (1989), 319-328

Linear programming

Consider an optimization problem P such as

$$\text{minimize } \{c^T x : x \in F(P)\}$$

where x a solution vector, c is a cost vector and $F(P)$ indicates the set of feasible solutions

$$F(P) \subset \{x \in \Re^n : x \geq 0\}.$$

Assume, for simplicity that P is bounded.

Bounding

To solve *NP*-hard optimization problems, it is common to resort to implicit enumeration techniques, which in turn rely upon relaxations, providing valid dual bounds.

These relaxations can be defined in many different ways. For several problems characterized by a combinatorial structure it happens that many possible relaxations can provide a dual bound.

For instance an ATSP can be relaxed into a linear assignment problem as well as into a shortest 1-arborescence problem.

We can choose one of the relaxations or we can solve all of them and select the tightest lower bound we obtain. In both cases, however, the information coming from the other relaxations is lost.

Additive bounds

Let $\mathcal{L}^{(1)}, \mathcal{L}^{(2)}, \ldots, \mathcal{L}^{(r)}$ be r lower bounding algorithms available for problem P.

Assume that each $\mathcal{L}^{(k)}$ for $k = 1, \ldots, r$, when executed on an instance of P with cost vector c, return a lower bound value $\delta^{(k)}$ and vector of reduced costs $c^{(k)}$ such that

$$c^{(k)} \geq 0$$

$$\delta^{(k)} + c^{(k)T} x \leq c^T x$$

for any $x \in F(P)$.

The additive bounding technique generates a sequence of instances of P, each obtained from the previous one by considering the corresponding reduced costs, computed each time with a different lower bounding procedure.

Additive bounds

Initially, procedure $\mathcal{L}^{(1)}$ is executed on problem P, yielding $\delta^{(1)}$ and $c^{(1)}$.
Then we consider the problem $R^{(1)}$:

$$\text{minimize } z^1 = \delta^{(1)} + c^{(1)T} x \;\; \text{s.t. } x \in F(P).$$

Problem $R^{(1)}$ is a relaxation of P, because:

- the feasible region is $F(P)$ in both cases;
- $z^1 = \delta^{(1)} + c^{(1)T} x \leq c^T x \;\; \forall x \in F(P)$

Problem $R^{(1)}$ is a residual instance of P.

Additive bounds

Since the optimal value of z^1 is non-negative, any lower bound of z^1 can be added to $\delta^{(1)}$ to obtain another (stronger) valid lower bound for P.

For this purpose, we can execute $\mathcal{L}^{(2)}$ to solve $R^{(1)}$, obtaining $\delta^{(2)}$ and $c^{(2)}$.

So we solve the problem $R^{(1)}$:

$$\text{minimize } z^1 = \delta^{(1)} + c^{(1)T} x \ \text{ s.t. } x \in F(P).$$

A new relaxation $R^{(2)}$ is obtained since

$$z^2 = \delta^{(1)} + \delta^{(2)} + c^{(2)T} x \leq c^T x \ \ \forall x \in F(P).$$

Additive bounds

The technique can be iterated for all the independent lower bounding techniques that are available for P.

The lower bound monotonically improves at each iteration.

It may be useful to perform some subgradient optimization iterations in one or more bounding procedures.

Variable fixing

After the execution of the last lower bounding procedure $\mathcal{L}^{(r)}$ we obtain a lower bound

$$\delta = \sum_{k=1}^{r} \delta^{(k)}$$

and a vector of non-negative reduced costs $c^{(r)}$. We know that

$$\delta + c^{(r)T} x \leq z^*.$$

If we know a heuristic solution of P whose cost is $\overline{z} \geq z^*$, then

$$\delta + c^{(r)T} x \leq \overline{z}.$$

In particular, if a binary variable x_j is such that $\delta + c_j^{(r)} > \overline{z}$, then we can fix x_j to 0.

Computing reduced costs

For this technique to work, we need lower bounding procedures that do not compute only lower bounds but also reduced costs, that are needed for the next iteration.

This can be obtained in several ways:

- Linear Programming relaxation
- Variable decomposition
- Disjunction
- Lagrangean relaxation

LP relaxation

Let assume that $F(P) \subset \{x \in \Re^n : x \geq 0, Ax \geq b\}$.
Then a relaxation of P is

$$\text{minimize } z_{LP} = c^T x : Ax \geq b, x \geq 0.$$

To compute the optimal value δ and the reduced costs, we solve the dual problem

$$\text{maximize } w_{LP} = u^T b : c - u^T A \geq 0, u \geq 0.$$

Let u^* be an optimal dual solution, so that $\delta = u^{*T} b$.
The corresponding reduced costs are given by $\overline{c} = c - u^{*T} A$.

When the dual problem does not have a unique optimal solution, u^* can be chosen in order to determine the reduced costs which allow the next bounding procedures to be more effective.

Heuristic dual solutions can also be used; they are likely to yield weaker lower bounds at the current step, but they can provide reduced costs that yield stronger lower bounds in later iterations.

Variable decomposition

Let us suppose that we can define q sets $Y^{(1)}, Y^{(2)}, \dots, Y^{(q)}$ with

$$Y^{(h)} \subseteq \{y \in \Re^n : y \geq 0\} \; \forall h = 1, \dots, q$$

such that each feasible solution $x \in F(P)$ can be decomposed into q column vectors $y^{(1)}, y^{(2)}, \dots, y^{(q)}$ as

$$x = \sum_{h=1}^{q} y^{(h)}$$

with $y^{(h)} \in Y^{(h)} \; \forall h = 1, \dots, q$.

For instance, let $F(P)$ the set of all Hamiltonian circuits in a digraph. Then every feasible solution can be decomposed into $q = 2$ parts: a path from s to t and a path from t to s, being s and t two distinct nodes of the digraph. So $Y^{(1)}$ would be the set of all $s - t$ paths and $Y^{(2)}$ would be the set of all $t - s$ paths.

Variable decomposition

The initial problem P can be reformulated as

$$\text{minimize } z = c^T \sum_{h=1}^{q} y^{(h)} : y^{(h)} \in Y^{(h)} \ \forall h = 1, \ldots, q, \sum_{h=1}^{q} y^{(h)} \in F(P).$$

We can define a partial problem for each $h = 1, \ldots, q$ as

$$\text{minimize } z_h = c^T y^{(h)} : y^{(h)} \in Y^{(h)}.$$

This decomposition is useful when a lower bounding procedure is available for the partial problem, producing a lower bound $\theta^{(h)}$ and the required reduced costs vector $\gamma^{(h)} \geq 0$, such that

$$\theta^{(h)} + \gamma^{(h)T} y^{(h)} \leq c^T y^{(h)} \ \ \forall y^{(h)} \in Y^{(h)}.$$

Variable decomposition

In this case we can define the relaxed problem

$$\text{minimize } z_{ER} = \sum_{h=1}^{q} \theta^{(h)} +$$

$$\min_{h=1}^{q}\{\gamma^{(h)T} y^{(h)}\} : y^{(h)} \in Y^{(h)} \; \forall h = 1, \ldots, q, \sum_{h=1}^{q} y^{(h)} \in F(P).$$

This problem can be further relaxed into

$$\text{minimize } z_R = \delta + \min_{h=1}^{q}\{\overline{c}^T y^{(h)}\} : y^{(h)} \in Y^{(h)} \; \forall h = 1, \ldots, q, \sum_{h=1}^{q} y^{(h)} \in F(P),$$

where $\delta = \sum_{h=1}^{q} \theta^{(h)}$ and $\overline{c}_j = \min_{h=1,\ldots,q}\{\gamma_j^{(h)}\} \; \forall j = 1, \ldots, n$.
Finally the relaxed problem can be restated as

$$\text{minimize } z_R = \delta + \min_{h=1,\ldots,q}\{\overline{c}^T x\} : x \in F(P).$$

So, δ is a valid lower bound for the initial problem P and $\overline{c}$ is a corresponding vector of reduced costs.

Disjunction

Let us suppose that we can define p sets $W^{(1)}, W^{(2)}, \ldots, W^{(p)}$ with

$$W^{(h)} \subseteq \{x \in \Re^n : x \geq 0\} \; \forall h = 1, \ldots, p$$

such that

$$F(P) \subseteq \bigcup_{h=1}^{p} W^{(h)}.$$

This means that for every $x \in F(P)$ the following disjunction holds:

$$(x \in W^{(1)}) \vee (x \in W^{(2)}) \vee \ldots \vee (x \in W^{(p)}).$$

Disjunction

We can formulate a restricted problem for each $h = 1, \ldots, p$:

$$\text{minimize } z_h = c^T x : x \in F(P) \cap W^{(h)}.$$

This is useful when a lower bounding procedure is available for the reduced problem, producing a lower bound $\theta^{(h)}$ of z_h and the corresponding reduced costs vector $\gamma^{(h)} \geq 0$, such that

$$\theta^{(h)} + \gamma^{(h)T} x \leq c^T x \;\; \forall x \in F(P) \cap W^{(h)}.$$

Disjunction

Then a valid lower bound for P is

$$\delta = \min_{h=1,\dots,p} \{\theta^{(h)}\}$$

and the corresponding reduced costs are given by

$$\overline{c}_j = \min_{h=1,\dots,p} \{\gamma_j^{(h)}\} \ \ \forall j = 1, \dots, n.$$

In fact, for any feasible solution $\overline{x} \in W^{(\overline{h})}$ we have

$$\delta + \overline{c}^T \overline{x} \leq \theta^{(\overline{h})} + \gamma_j^{(\overline{h})T} \overline{x} \leq c\overline{x}.$$

Lagrangean relaxation

Suppose that $F(P) \subset \{x \in \Re^n : Ax \geq b\}$ with $b \in \Re^m$.
Let $\tilde{u} \in \Re^m$ be a vector of non-negative multipliers, defining the Lagrangean relaxation in which the cost vector is $\tilde{c} = c - \tilde{u}^T A$.
Then

$$\tilde{u}^T b + \tilde{c}^T x \leq c^T x \ \ \forall x \in F(P).$$

Now assume a lower bounding procedure is available for problem P with cost vector $\tilde{c}$, providing a lower bound θ and the corresponding reduced cost vector γ. Then

$$\tilde{u}^T b + (\theta + \gamma^T x) \leq \tilde{u}^T b + \tilde{c}^T x \leq c^T x \ \ \forall x \in F(P).$$

Therefore $\delta = \tilde{u}^T b + \theta$ is a valid lower bound for P defined by cost vector c and γ is the corresponding reduced cost vector.
A convenient value for $\tilde{u}$ can be determined by iterating the procedure to search for a large value of δ, with subgradient optimization techniques.

The ATSP with Precedence Constraints

Given a digraph $G = (N, A)$ with n nodes, a cost function $c \geq 0$ on the arcs and a set of precedence relations defined through node subsets $P_2, P_3, \ldots, P_n$, find a minimum cost Hamiltonian circuit starting from node 1 and visiting all nodes in P_h before node h, for each $h = 2, \ldots, n$.

$$\text{minimize } z = \sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij} x_{ij}$$

$$\begin{aligned}
\text{s.t. } & \sum_{i=1}^{n} x_{ij} = 1 && \forall j = 1, \ldots, n \\
& \sum_{j=1}^{n} x_{ij} = 1 && \forall i = 1, \ldots, n \\
& \sum_{i \in N \setminus S} \sum_{j \in S} x_{ij} \geq 1 && \forall S \subset N : 1 \notin S, S \neq \emptyset \\
& \text{precedence constraints for node } h && \forall h = 2, \ldots, n \\
& x_{ij} \geq 0, \ \text{integer} && \forall (i, j) \in A
\end{aligned}$$

Relaxations from LP

Any relaxation for the ATSP is also a relaxation for the ATSP with PC.
In particular:

- the Shortest Spanning 1-Arborescence Problem
- the Linear Assignment Problem (Minimum Cost Bipartite Matching Problem)

are two combinatorial relaxations of the ATSP, which possess the integrality property, so that they can be solved as LP problems.

Relaxation from variable decomposition

Relaxations for the ATSP neglect precedence constraints. To take them into account, consider a pair of distinct nodes a and b different from 1, such that $a \in P_b$. For each node $h = 2, \ldots, n$, define the set $S_h = \{i \in N : h \in P_i\}$ of its successors.

Every feasible solution can be decomposed into three paths:

- a path from 1 to a not visiting any node in S_a;
- a path from a to b not visiting any node in $P_a \cup S_b \cup \{1\}$
- a path from b to 1 not visiting any node in P_b.

This generates three partial problems, all of them being instances of the shortest path problem on a reduced digraph.

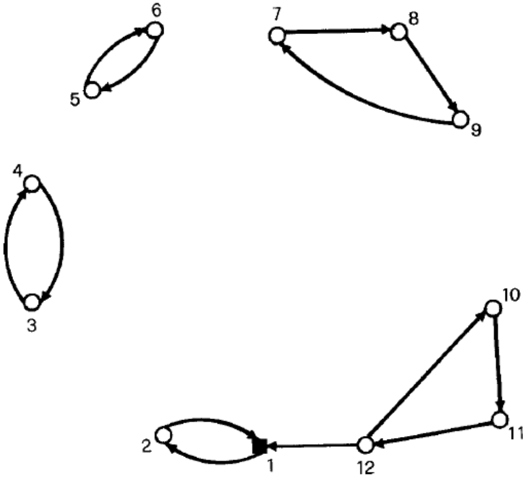
Relaxation from disjunction

Let a, b and r be three distinct nodes in N. In any Hamiltonian circuit two paths exist, one from a to b and the other from b to a and one of them does not visit r.

This generates two restricted problems (for each chosen triplet), requiring to compute a shortest path between two nodes and not visiting a third node.

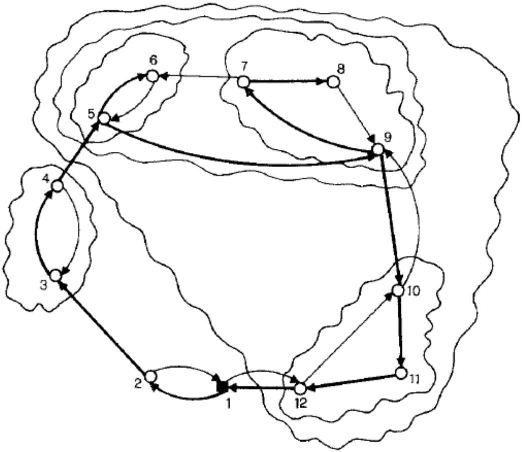
This is equivalent to the problem with precedences where r is the starting vertex and a must precede b or vice versa.

Step 1: assignment



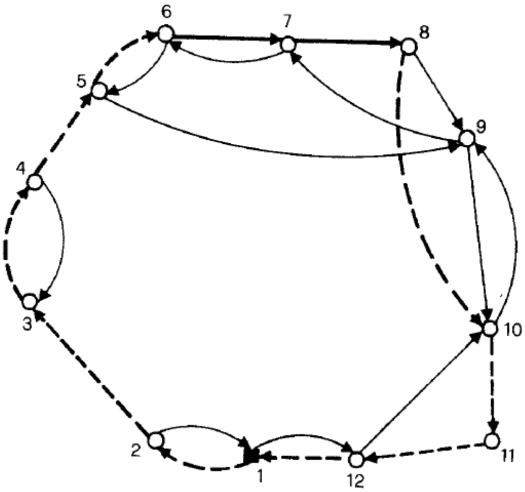
Not all nodes are reachable from node 1.

Step 2: arborescence



Consider the triplet $(6, 8, 9)$. Any feasible solution must contain either a path from 6 to 8 or a path from 8 to 6 not visiting 9.

Step 3: disjunction



A Hamiltonian tour now exists but 3 is visited before 8.

Step 4: variable decomposition

Assume we have precedence constraints such that:
$P_8 = \{5, 7\}$, $S_8 = \{3, 10, 12\}$.
$P_3 = \{5, 7, 8\}$, $S_3 = \{10\}$.
Every feasible solution can be decomposed into three paths:

- a path from 1 to 8 not visiting any node in S_8 (i.e. not visiting $\{3, 10, 12\}$);
- a path from 8 to 3 not visiting any node in $P_8 \cup S_3 \cup \{1\}$ (i.e. not visiting $\{1, 5, 7, 10\}$);
- a path from 3 to 1 not visiting any node in P_3 (i.e. not visiting $\{5, 7, 8\}$).

Step 4: variable decomposition