Hitting Sets

Let $A = \{a_1, \ldots, a_n\}$ be a set and $F = \{F_1, \ldots, F_m\}$ a family of subsets of $A$. A subset $H \subseteq A$ has the hitting set property, or is a hitting set, iff $H \cap F_i \neq \emptyset$ for $1 \leq i \leq m$ (i.e., $H$ “hits” each set $F_i$). If we are given a cost function $c : A \rightarrow \mathbb{N}$, the cost of $H$ is $\sum_{a \in H} c(a)$. A hitting set is of minimum cost if its cost is minimal among all hitting sets.

The problem of finding a minimum-cost hitting set for family $F$ and cost function $c$ is denoted by $(F, c)$, and the cost of its solution by $\min(F, c)$. A relaxation for $(F, c)$ is a problem $(F', c')$ such that $c' \leq c$, and for all $F' \in F'$ there is $F \in F$ with $F \subseteq F'$. In words, $(F, c)$ can be relaxed by reducing costs, dropping sets from $F$, or enlarging elements of $F$. Determining the existence of a hitting set for a given cost bound is a classic problem in computer science, one of the first problems to be shown NP-complete (Kar72).

Lemma 1. If $(F', c')$ is a relaxation of $(F, c)$, then $\min(F', c') \leq \min(F, c)$. Furthermore, if $\{(F_i, c_i)\}$ is a collection of relaxations of $F$ such that $\sum_i c_i \leq c$, then $\sum_i \min(F_i, c_i) \leq \min(F, c)$.

Proof. The first claim is direct since a hitting set for $F'$ is also a hitting set for $F'$ and $c' \leq c$. For the second claim, consider a hitting set $H$ for $F$. Then,

$$c(H) \geq \sum_i c_i(H) \geq \sum_i \min(F_i, c_i).$$

The first inequality holds because $\sum_i c_i \leq c$ and the second because $H$ is a hitting set for each $F_i$.

Decomposition and Width

Let $F$ be a family that can be partitioned into $\Pi = \{F_1, \ldots, F_m\}$ satisfying $(\bigcup F_i) \cap (\bigcup F_j) = \emptyset$ for all $i \neq j$; i.e., the blocks in the partition are pairwise independent. Then, for any cost function $c$, $\min(F, c) = \sum_{i=1}^m \min(F_i, c)$ and the problem of finding a minimum-cost hitting set for $F$ can be decomposed into smaller subproblems. We call the maximum size of a block in $\Pi$ the width of $\Pi$, denoted by $width(\Pi)$. The width of $F$, denoted by $width(F)$, is the minimum width of $\Pi$ over all partitions $\Pi$ of $F$ into independent blocks. Finding a partition that minimizes the width is an easy problem similar to computing connected components of a graph, so $width(F)$ can be efficiently computed.

Indeed, for a family $\mathcal{F}$, define the (undirected) graph $G = (V, E)$ where $G = \cup \mathcal{F}$ and $\{a, a'\} \in E$ iff $a \neq a'$ and there is $F \in \mathcal{F}$ with $F \supseteq \{a, a'\}$. Then,

**Proposition 1.** The width of $\mathcal{F}$ equals the size of the largest connected component of $G$.

Proof. Let $\{G_1, \ldots, G_m\}$ be the connected components of $G$ and $w$ the size of a largest one. We need to show that $width(F) \leq w$ and $w \leq width(F)$.

For the first inequality, consider the partition $\Pi_G = \{F_1, \ldots, F_m\}$ of $\mathcal{F}$ defined by $F_i = \{F \in \mathcal{F} : F \cap G_i \neq \emptyset\}$. It is easy to check that this partition is well defined and with pairwise independent blocks. Therefore,

$$width(F) \leq width(\Pi_G) = \max_{i=1,...,m} |F_i| = w.$$

For the second inequality, let $\Pi = \{F_1, \ldots, F_k\}$ be a partition of $\mathcal{F}$ into pairwise independent blocks. It will be enough to show that each connected component $G_i$ is contained in some block of $\Pi$.

Let $a$ be a vertex in $G_i$ and $F_j \in \Pi$ a block containing it; i.e., $a \in \cup F_j$. Consider an arbitrary vertex $a' \in G_i$. Since $G_i$ is connected, there is a path $(a = a_0, a_1, \ldots, a_\ell = a')$ in $G_i$. We perform induction on $\ell$ to show that $a'$ is also in $\cup F_j$. For $\ell = 0$, $a = a'$ and the claim is trivial. Assume that the claim holds for paths of length less than $\ell$. Then, $a_{\ell-1}$ is in $\cup F_j$ by inductive hypothesis. This implies that there is an $F \in F_j$ such that $a_{\ell-1} \in F$. On the other hand, the edge $\{a_{\ell-1}, a_\ell\}$ implies the existence of $F' \in \mathcal{F}$ with $\{a_{\ell-1}, a_\ell\} \subseteq F'$. Therefore, $F \cap F' \neq \emptyset$, $F' \in F_j$ (because the blocks of $\Pi$ are pairwise independent), and $a_\ell \in \cup F_j$.

Let $F = \{F_1, \ldots, F_k\}$ be a family over $A$ with $k$ subsets, but with no assumptions on the sizes of each $F_i$ or $A$. We show that $\min(F, c)$ and a hitting set achieving this cost can be computed in time bounded by $O(|\mathcal{F}| + k4^k)$. To see this, consider the hypergraph $H_F = (X, E)$ where $X = \{1, \ldots, k\}$ and there is a hyperedge $e(a) = \{i : a \in F_i\}$ with cost $c(a)$ for each $a \in A$. The hitting sets for $F$ are in one-to-one correspondence with the covers of the hypergraph $H_F$ (a cover is a set of hyperedges that “touch” every
vertex). Hence, finding \( \min(\mathcal{F}, c) \) is equivalent to finding a minimum-cost cover for \( H_F \). For the latter, observe that all hyperedges \( e(a) \) for which there is a hyperedge \( e(a') \) with \( e(a) = e(a') \) and \( e(a') < e(a) \) may be removed (and if \( e(a') = e(a) \), only one of the hyperedges needs to be kept).

Since a hyperedge is a subset of \( X \), this implies that we only need to consider hypergraphs with at most \( 2^k \) edges. Using dynamic programming, a minimum cost cover for such a hypergraph can be found in time \( O(2^k k^4) \). Combining this with the time required for constructing the hypergraph yields the overall \( O(\|\mathcal{F}\| + k^4) \) bound. This is an example of fixed-parameter tractability (FG06).

**Theorem 2.** The problem of computing \( \min(\mathcal{F}, c) \) is fixed-parameter tractable when considering the width of \( \mathcal{F} \) as the parameter. In particular, for any fixed bound \( k \), \( \min(\mathcal{F}, c) \) for families of width at most \( k \) can be computed in linear time.

**Proof.** We first show that a hitting set problem \( (\mathcal{F} = \{F_1, \ldots, F_l\}, c) \) of arbitrary width, can be solved in time \( O(\|\mathcal{F}\| + k^4) \), where \( \|\mathcal{F}\| \) denotes the input size for \( \mathcal{F} \).

Consider the hypergraph \( H_F = (X, E) \), where \( X = \{1, \ldots, k\} \) and there is a hyperedge \( e(a) = \{i : a \in F_i\} \) with cost \( c(a) \) for each \( a \in A \). The hitting sets for \( H_F \) are in one-to-one correspondence with the covers of \( H_F \) (a cover is a set of hyperedges that “touch” every vertex). Hence, finding \( \min(\mathcal{F}, c) \) is equivalent to finding a minimum-cost cover for \( H_F \). For the latter, observe that all hyperedges \( e(a) \) for which there is a hyperedge \( e(a') \) with \( e(a) = e(a') \) and \( e(a') < e(a) \) may be removed (if \( c(a') = c(a) \), only one of the hyperedges need to be kept). Thus, we only need to consider hypergraphs with at most \( 2^k \) edges. The following DP algorithm computes the cost of a minimum-cost cover.

We make use of a table \( \text{best cover} \) of size \( 2^k \) that maps subsets of \( X = \{1, \ldots, k\} \) into \([0, \infty) \cup \{\infty\} \). At the end of the algorithm, the entry \( \text{best cover}[X] \) contains the cost of a minimum-cost cover for \( (\mathcal{F}, c) \), and a minimum-cost cover can be recovered from the table in linear time. The algorithm initializes the table as \( \text{best cover}[\emptyset] := 0 \) and \( \text{best cover}[X'] := \infty \) for all \( \emptyset \subset X' \subset X \). Then, it updates the table using DP as shown in Fig. 1. The initialization loop takes \( O(2^k) \) time. The two nested loops make a total of \( O(4^k) \) iterations, each taking time \( O(k) \). The running time of the DP algorithm is \( O(\|\mathcal{F}\|) \) time for constructing the hypergraph, and \( O(k4^k) \) time for finding a minimum-cost cover of the hypergraph.

Now, we show that a hitting set problem \( (\mathcal{F}, c) \) of width at most \( k \) can be solved in linear time. By Proposition 1, one can compute in linear time a partition \( \Pi = \{F_1, \ldots, F_m\} \) of \( \mathcal{F} \) into independent blocks. Then, since the blocks are independent, \( \min(\mathcal{F}, c) = \sum_{i=1}^{m} \min(F_i, c) \). Each subproblem \( (F_i, c) \) can be solved in time \( O(\|F_i\| + k4^k) \). Therefore, \( (\mathcal{F}, c) \) can be solved in time

\[
\sum_{i=1}^{m} O(\|F_i\| + k4^k) = O(m\|\mathcal{F}\| + mk4^k) = O(\|\mathcal{F}\| + \|\mathcal{F}\|k4^k)
\]

which is linear in \( \|\mathcal{F}\| \) since \( k \) is fixed and bounded.

**Landmarks**

A STRIPS problem with action costs is a tuple \( P = \langle \mathcal{F}, O, I, G, c \rangle \) where \( \mathcal{F} \) is the set of fluents, \( O \) is the set of actions or operators, \( I \) and \( G \) are the initial and goal description, and \( c : \mathcal{O} \rightarrow \mathbb{N} \) is the cost function. We are interested in delete relaxations, so we assume that the operators have empty delete lists, and thus ‘plan’ and ‘relaxed plan’ shall denote the same. For a definition of the basic concepts underlying delete relaxations, such as the \( h^\mathsf{max} \) and \( h^+ \) functions, we refer to the literature (HDD09). We also assume from now on that all fluents have finite \( h^\mathsf{max} \) values, which implies that the problem has finite \( h^+ \) value. As additional simplifying assumptions, we require that all operators have nonempty preconditions, that there are two fluents \( s, t \in F \) such that \( I = \{s\} \) and \( G = \{t\} \), and that there is a unique operator \( fin \) that adds \( t \). When these simplifying assumptions are not met, they can be achieved through simple linear-time transformations. We denote the precondition and effects of \( a \in O \) by \( \text{pre}(a) \) and \( \text{post}(a) \). The \( h^+ \) value for state \( I \) is denoted by \( h^+(P) \).

An (action) landmark for \( P \) is a disjunction \( a_1 \vee \cdots \vee a_n \) of actions such that every plan for \( P \) must contain at least one such action. Such a landmark is denoted by the set \( \{a_1, \ldots, a_n\} \).

Recall that a pcf \( D \) assigns a precondition \( \forall a \in O \) to each action \( a \in O \). Our first result relates cuts in the justification graph \( G(D) \) with landmarks for \( P \).

**Lemma 3.** Let \( D \) be a pcf and \( C \) an \( s \cdot t \)-cut of \( G(D) \). Then, the labels of the edges in the cut-set of \( C \) form a landmark.

**Proof.** A relaxed plan defines an \( s\cdot t \)-path on \( G(D) \) that must cross every \( s \cdot t \)-cut.

Given a pcf \( D \), we denote the set of landmarks associated with the cut-sets of \( G(D) \) by \( \text{Landmarks}(D) \). By considering all pcf and all cuts in the justification graphs, we obtain the hitting set problem \( \mathcal{F}_L = \bigcup \{\text{Landmarks}(D) : D \text{ is a precondition-choice function}\} \).

**Theorem 4.** If \( H \) is a plan for \( P \), then \( H \) is a hitting set for \( \mathcal{F}_L \). Conversely, if \( H \) is a hitting set for \( \mathcal{F}_L \), then \( H \) contains a plan for \( P \). Therefore, \( \min(\mathcal{F}_L, c) = h^+(P) \).

**Proof.** The first claim is direct, since by Lemma 3, every element of \( \mathcal{F}_L \) is hit by every plan. The last claim follows from the first two.

For the second claim, let \( H \) be a hitting set for \( \mathcal{F}_L \) and let \( R \) be the set of fluents that can be reached by only using operators in \( H \). If \( R \) contains the goal \( t \), then \( H \) contains a plan and there is nothing to prove. So, assume \( t \notin R \). We construct a pcf \( D \) such that \( G(D) \) contains an \( s \cdot t \)-cut whose cut-set is not hit by \( H \), thus reaching a contradiction. We classify operators into three types and define \( D \):

T1. If \( \text{pre}(a) \subseteq R \) and \( \text{post}(a) \subseteq R \), then set \( D(a) \) arbitrarily to some \( p \in \text{pre}(a) \).

T2. If \( \text{pre}(a) \subseteq R \) and \( \text{post}(a) \notin R \), then set \( D(a) \) arbitrarily to some \( p \in \text{pre}(a) \).

T3. If \( \text{pre}(a) \notin R \), then set \( D(a) \) to some \( p \in \text{pre}(a) \setminus R \).
Now consider the cut $(R, R')$ of $G(D)$, where $R'$ is the set of all fluents not in $R$. It is a cut since $s \in R$ and $t \notin R$. We show that $H$ does not hit the cut-set, i.e., there exists no operator $a \in H$ that labels an edge going from some fluent in $R$ to some fluent not in $R$.

Assume that $a \in H$ were such an operator. It cannot be of type T1, because edges labeled by type T1 operators go from $R$ into $R$. It cannot be of type T2, because pre$(a) \subseteq R$ and $a \in H$ implies post$(a) \subseteq R$ (by definition of $R$). Finally, it cannot be of type T3, as edges labeled by type T3 operators do not start in $R$. Hence, no such operator exists.

In practice, computing $F_L$ according to the definition above is infeasible because there are usually exponentially many pcfs. However, if we can compute and solve, in polynomial time, a relaxation of $F_L$, then this provides a polytime admissible approximation of $h^+$.

**Corollary 5.** Let $\langle F, c' \rangle$ be a polynomial-time computable relaxation of $\langle F_L, c \rangle$ (possibly additive\(^1\)) whose solution is polynomial-time computable. Then the heuristic $h = \min(\langle F, c' \rangle)$ is a polytime admissible approximation of $h^+$.

**Proof.** Direct from the assumptions: the admissibility because $\langle F, c' \rangle$ is a relaxation of $\langle F_L, c \rangle$, which defines $h^+$, and the polynomial-time computability because $\langle F, c' \rangle$ is solvable in polynomial time. \(\square\)

An important special case covered by the corollary are landmark heuristics based on cost partitioning, including LM-cut (see below) and the heuristics of Karpas and Domshlak (KD09). In general, given a set $L = \{L_1, \ldots, L_n\}$ of landmarks, a cost partitioning for $L$ is a collection $C = \{c_1, \ldots, c_n\}$ of cost functions such that $\sum_{i=1}^n c_i(a) \leq c(a)$ for each action $a$. The partitioning defines the heuristic $h_C = \sum_{i=1}^n \min_{a \in L_i} c_i(a)$, which is an additive relaxation of $F_L$ when $L \subseteq F_L$.

Karpas and Domshlak studied *uniform* cost partitioning, defined as $c_i(a) = 0$ if $a \notin L_i$ and $c_i(a) = c(a)/|\{i : a \in L_i\}|$ if $a \in L_i$, and optimal cost partitioning, which maximizes $h_C$ through linear programming (LP). Interestingly,

\(^1\)In the additive case, we slightly abuse notation since $\langle F, c' \rangle$ should be replaced by a collection $\{(F_i, c_i)\}_i$. there is a close connection between the optimal cost partitioning LP and the hitting set ILP for $L$.

**Theorem 6.** Let $L$ be a collection of landmarks, and let $c$ be the cost function for the actions. Then, the LP that defines the optimal cost partitioning is the dual of the LP relaxation of the ILP for $\langle L, c \rangle$.

**Proof.** Let $L = \{L_j\}_j$ be a collection of landmarks over actions $A$, and $c$ a costs functions for the actions. The ILP corresponding to $\langle L, c \rangle$ is:

\begin{align*}
\text{minimize} & \quad \sum_{a \in A} c(a)x_a \\
\text{subject to} & \quad \sum_{a \in L_j} x_a \geq 1, \\
& \quad x_a \in \{0, 1\}.
\end{align*}

The variables $x_a$ are indicator variables that define the hitting set. The LP relaxation of the ILP is:

\begin{align*}
\text{minimize} & \quad \sum_{a \in A} c(a)x_a \\
\text{subject to} & \quad \sum_{a \in L_j} x_a \geq 1, \\
& \quad 0 \leq x_a \leq 1.
\end{align*}

The dual of the LP is:

\begin{align*}
\text{maximize} & \quad \sum_j y_j \\
\text{subject to} & \quad \sum_{a \in L_j} y_j \leq c(a), \\
& \quad y_j \geq 0.
\end{align*}

The variables $y_j$ are the dual variables corresponding to the constraints of the LP. This LP attains the same value of the LP that defines the optimal cost partitioning for $L$. Indeed,
the $y_i$ variables can be interpreted as the cost of each landmark for the optimal cost assignment. The constraints avoid cost partitionings $c_j$ such that $\sum_j c_j(a) \geq c(a)$. □

The LM-Cut Heuristic

**Theorem 7.** Given a set of landmark $L \subseteq F_L$ and cost partitioning $C$, $h_C$ is an additive relaxation of $F_L$. LM-cut is one such relaxation.

**Proof.** The first claim is direct by definition. For the second claim, we show that there is a collection $\{F_i\}_{i=1}^n$ of relaxations of $F_L$ such that

$$h_{LM-cut}(P) = \sum_{i=1}^n m_i = \sum_{i=1}^n \min(F_i, c_i) \leq \min(F_L, c).$$

Let $L_1, \ldots, L_n$ and $c_1, \ldots, c_n$ be the landmarks and cost functions computed by LM-cut at each stage. Define $F_i = \{L_i\}$ and $c'_i(a) = m_i$ if $a \in L_i$ and $c'_i(a) = 0$ otherwise. Clearly, each $F_i$ is a relaxation of $F_L$ and $\min(F_i, c_i) = \min(F_i, c'_i) = m_i$. By Lemma 1, it remains to show that $\sum c'_i \leq c$.

Let $I(a, k) = \{i : a \in L_i, 1 \leq i \leq k\}$ be the set of indices for the landmarks in $\{L_1, \ldots, L_k\}$ that contain $a$. Using induction it is not difficult to show that $c_{k+1}(a) = c(a) - \sum_{i \in I(a, k)} m_i$. On the other hand,

$$c'_1(a) + c'_2(a) + \cdots + c'_k(a) = \sum_{i \in I(a, k)} c'_i(a) = \sum_{i \in I(a, k)} m_i.$$

We show using induction on $k$ that $c'_1(a) + \cdots + c'_k(a) \leq c(a)$ for every action $a$ and $1 \leq k \leq n$. The base of the induction is easy. If $a \notin L_1$, then $c'_1(a) = 0 \leq c(a)$. If $a \in L_1$, then $c'_1(a) = m_1 = \min_{a' \in L_1} c_1(a') \leq c(a)$. Suppose that the claim holds up to $k$. We need to show it for $k+1$. If $a \notin L_{k+1}$, then $\sum_{i \in I(a, k+1)} c'_i(a) = \sum_{i \in I(a, k)} c'_i(a) \leq c(a)$ by inductive hypothesis. If $a \in L_{k+1}$, then

$$\sum_{i \in I(a, k+1)} c'_i(a) = \sum_{i \in I(a, k)} c'_i(a) + c'_{k+1}(a)$$

$$= \sum_{i \in I(a, k)} m_i + m_{k+1}$$

$$= \sum_{i \in I(a, k)} m_i + \min_{a' \in L_{k+1}} c_{k+1}(a')$$

$$\leq \sum_{i \in I(a, k)} m_i + c_{k+1}(a)$$

$$= \sum_{i \in I(a, k)} m_i + c(a) - \sum_{i \in I(a, k)} m_i$$

$$= c(a).$$

□

**Theorem 8.** For any fixed $p \geq 1$ and $k \geq 1$, $h_{\text{LM-cut}}$ is computable in polynomial time and dominates $h_{\text{LM-cut}}$.

**Proof.** Direct. □

**References**


