Efficient Computation and Informative Estimation of h^+ by Integer and Linear Programming

Masood Feyzbakhsh Rankooh and Jussi Rintanen

Department of Computer Science, Aalto University, Helsinki, Finland
jussi.rintanen@aalto.fi, masood.feyzbakhshrankooh@aalto.fi

Abstract

We investigate modeling cost optimal delete-free STRIPS Planning by Integer/Linear Programming (IP/LP). We introduce two IP models and their LP relaxations based on a recently formulated representation of relaxed plans, named *causal relaxed plan representation*. The new models are produced by enforcing acyclicity in so-called *causal relation graphs* using vertex elimination and time labeling methods. We empirically show that while the vertex elimination based method outperforms the time labeling based method and all previously introduced domain independent methods for computing the exact value of h^+, the time labeling based LP model is faster to solve compared to its vertex elimination based alternative, making it more suitable for using as heuristic function for optimal planning. We also theoretically analyze the admissible heuristic functions obtained by solving our LP models, and prove that the vertex elimination based heuristic is at least as informative as the time labeling based heuristic. Moreover, our empirical analysis shows that our vertex elimination based heuristic, which is a novel admissible estimation of h^+, often has information complementary to that of the LM-cut heuristic.

Introduction

The value of h^+ for a given planning problem is the optimal cost of the corresponding delete-relaxed planning problem, obtained from the original problem by removing delete effects of all actions. The value of h^+ is a lower bound of the optimal cost of the original problem. It has been shown that having h^+ can significantly improve the efficiency of optimal planning (Betz and Helmert 2009). Computing h^+, however, is NP-equivalent (Bylander 1994), and h^+ is also hard to approximate (Betz and Helmert 2009).

Computing h^+ is important because many admissible heuristics functions have been introduced to compute lower bounds of h^+ as admissible heuristics for cost optimal planning. Examples are the h^{max} heuristic (Bonet and Geffner 2001), the LM-cut heuristic (Helmert and Domshlak 2009), set-additive heuristic (Keyder and Geffner 2008), and cost-sharing approximations of h^{max} (Mirkis and Domshlak 2007). The value of h^+ can be considered as a measure for informativeness of such heuristic functions.

Moreover, methods for computing h^+ can also output an optimal plan for a given delete-free planning problem. Efficient solving of delete-free planning problems is important by itself. That is because there exist delete-free planning tasks that are of interest for the planning community. Examples are the minimal seed-set problem (Gefen and Brafman 2011), and the problem of determining join orders in relational database query plan generation (Robinson, McIlraith, and Toman 2014).

Another reason for the importance of efficient computation of h^+ is based on the fact that optimal plans for general planning problems can be produced by iterative solving and reformulating delete-free planning tasks (Haslum 2012). This can be done by repeatedly finding optimal plans for a delete-relaxed version of the original planning problem, and reducing the relaxation by reformulating the problem whenever a plan has been found for the delete-relaxed that is not a valid plan for the original problem.

The current efficient domain independent methods for computing h^+ are based on translating the delete-relaxed version of a given problem into a set of constraints and using of-the-shelf efficient constraint satisfaction solvers. One major approach of these methods is using Integer Programming (IP) (Haslum, Slaney, and Thiébaux 2012; Imai and Fukunaga 2015; Castro et al. 2020).

Another efficient approach for computing h^+ is using Boolean satisfiability (Rankooh and Rintanen 2022). In this method the concept of *causal partial functions* are utilized to translate the delete-relaxed version of a given problem into a propositional formula with an underlying directed graph. The vertex elimination method (Rose and Tarjan 1975) is then used to produce SAT formulas that ensure acyclicity in the mentioned underlying graph. By ensuring acyclicity in the underlying graphs, it is guaranteed that any solution will then be transformed to a *causal relaxed plan representation*, an alternative representation for relaxed plans of the given problem.

In this work we examine an IP adaptation of the method used in (Rankooh and Rintanen 2022). One important reason why an IP-based approach seems promising is that, unlike in propositional satisfiability, encoding and reasoning about action costs can be done very easily in it. Moreover, the Linear Programming (LP) relaxation of an IP model for optimal delete-free planning is an admissible heuristic function for the original problem. Unlike IP models, LP models can be solved in polynomial time, making them appropriate candidates for heuristic functions when search methods such as A* are employed. We provide the proofs for the soundness and completeness of this IP model, called IP-VE, and empirically show that this approach is considerably more efficient that all previously introduced methods of computing h^+ for conventional STRIPS planning problems.

We also investigate another IP model which improves the model introduced in (Imai and Fukunaga 2015) by considering the notion of causal relaxed plan representations and maintaining acyclicity in their underlying graphs. We call this model which incorporates time labels for atomic propositions to enforce acyclicity in the underlying graphs, the IP-TL model.

Furthermore, we consider LP-VE and LP-TL, the LP relaxations of our IP models. We prove that the admissible heuristic function obtained from solving LP-VE dominates the one obtained from solving LP-TL. We empirically show that both these heuristic functions are considerably more informative than the LM-cut heuristic function, but take more time to compute. Our results also show that our heuristic functions have information complementary to that of *LM-cut* in many cases.

Finally, we incorporate our heuristic functions into an A* search method. We show that while the LP-TL model demonstrates performance competitive with that of the LM-cut heuristic function in this setting, the LP-VE model does not scale as well, producing inferior results because of its costly computation.

Preliminaries and Background

A STRIPS planning problem is a 5-tuple $\Pi = (P, I, A, G, cost)$ where P is a finite set of Boolean state variables, also called *atomic propositions*. I, the initial state, and G, the set of goal conditions, are subsets of P. A is a finite set of actions. Each member a of A is a triple $(pre(a), add(a), del(a))$, where $pre(a)$, $add(a)$ and $del(a)$ are sets of atomic propositions denoting the set of preconditions, positive effects, and negative effects of a, which are the atomic propositions that a requires, adds, and deletes, respectively. The cost function $cost$ maps every member of A to a non-negative integer.

When states are represented as subsets of P, the successor $s' = exec_a(s)$ of a state s with respect to action $a \in A$ is defined if $pre(a) \subseteq s$, and it is $s' = (s \backslash del(a)) \cup add(a)$. An action sequence $a_1, ..., a_n$ is executable (in state s) if $exec_{a_1,...,a_n}(s) = exec_{a_n}(exec_{a_{n-1}}(...exec_{a_2}(exec_{a_1}(s))))$ is defined. A plan for Π is a sequence π of actions from A such that $G \subseteq exec_\pi(I)$. The cost of plan $\pi = \langle a_1, ..., a_n \rangle$ for Π, denoted by $cost(\pi)$, is defined by $\Sigma_{i=1,...,n} cost(a_i)$. An optimal plan for Π is a plan with minimal cost for Π.

For a given STRIPS planning problem $\Pi = (P, I, A, G, cost)$, the delete relaxation (Bonet and Geffner 2001) $\Pi^+ = (P, I, A^+, G, cost)$ is defined, where A^+ is produced from A by replacing the set of negative effects of each member of A with the empty set. A plan for Π^+ is called a relaxed plan for Π. The optimal cost of Π^+ is denoted by $h^+(\Pi)$. If there is no relaxed plan for Π, we set $h^+(\Pi)$ to ∞.

Causal Relaxed Plan Representations

For encoding $h^+(\Pi)$ in this work, we use an alternative representation of relaxed plans, formulated in (Rankooh and Rintanen 2022).

Definition 1 (Causal partial functions)**.** *For a STRIPS planning problem $\Pi = (P, I, A, G, cost)$, we call a partial function f from $P \backslash I$ to A a causal partial function for Π iff the following conditions hold: 1) if $f(p) = a$ then a adds p; 2) if $f(p) = a$ then for every precondition q of a, either $q \in I$ or f is defined for q; 3) for every $p \in G$, either $p \in I$ or f is defined for p.*

Definition 2 (Causal relation graphs)**.** *Any causal partial function for Π induces a directed graph $G_f = (P, E_f)$, called the causal relation graph of f, such that (q, p) is a member of E_f iff for some a, $f(p) = a$ and q is a precondition of a. Causal relation graph of Π is $G_\Pi = (P, E_\Pi)$, such that (q, p) is a member of E_f iff for some action a, q and p is a precondition and effect of a, respectively.*

It should be clear that for every causal partial function f for Π, G_f is a subgraph of G_Π.

Definition 3 (Causal relaxed plan representations)**.** *Partial function f is called a causal relaxed plan representation for Π iff its causal relation graph is acyclic. The cost of f, denoted by $cost(f)$, is defined by the total cost of all actions to which some atomic proposition is mapped by f.*

In Definition 1, for each atomic proposition p, $f(p)$ is intended to represent the action that causes p to become *true*. Condition 1 of Definition 1 is thereby necessary. Condition 2 guarantees that if action a causes p to become *true*, then the preconditions of a become *true* by some action. Condition 3 provides that all goals must become *true*. The acyclicity of G_f is required to avoid causal cycles.

It has been shown that if Π is solvable, a causal relaxed plan representation exists for Π, and the minimal cost of causal relaxed plan representations equals to $h^+(\Pi)$. Also, an optimal relaxed plan for Π can be extracted from a minimal cost causal relaxed plan representation in polynomial time (Rankooh and Rintanen 2022).

Vertex Elimination Graphs

As it was mentioned above, for any given causal partial function f, the causal relation graph G_f is needed to be acyclicif f is a causal relaxed plan representation. One efficient way of ensuring acyclicity in directed graphs is using the concept of vertex elimination graphs.

Vertex elimination graph has originally been introduced in (Rose and Tarjan 1975). Let $G = (V, E)$ be a directed graph, $G^+ = (V, E^+)$ be the transitive closure of G, and $O = v_1, ..., v_{|V|}$ be an arbitrary ordering of members of V. According to ordering O, we produce a sequence of graphs $G_0 = G, ..., G_{|V|}$ by eliminating vertices of G. For each $i > 0$, G_i is produced from G_{i-1}, by eliminating v_i, and adding edges from all its in-neighbors to all its out-neighbors. Formally, $G_i =$

(V_i, E_i) is produced from $G_{i-1} = (V_{i-1}, E_{i-1})$ so that $V_i = V_{i-1} \setminus \{v_i\}$, and $E_i = E_{i-1} \setminus (\{(v_j, v_i) | (v_j, v_i) \in E_{i-1}\} \cup \{(v_i, v_k) | (v_i, v_k) \in E_{i-1}\}) \bigcup D_i$, where $D_i = \{(v_j, v_k) | (v_j, v_i) \in E_{i-1}, (v_i, v_k) \in E_{i-1}, j \neq k\}$. $G^* = (V, E^*)$ is the vertex elimination graph of G according to elimination ordering O, where $E^* = \bigcup_{i=1,\ldots,|V|} E_i$. It should be clear that G^* is a subgraph of G^+.

The directed elimination width (Hunter and Kreutzer 2007) of ordering O for graph G is defined by the maximum over number of out-neighbors of v_i in G_i for $i = 1, \ldots |V|$. The directed elimination width of G is the minimum width over all elimination orderings for G.

It has been shown that finding the ordering that produces the minimum width for a given digraph is NP-complete (Rose and Tarjan 1975). However, there are ad-hoc methods for producing orderings with low directed elimination widths in practice. An example is *minimum degree* heuristic, which chooses v_i with the minimum degree from G_{i-1}.

Related Works

Computing h^+ has been investigated before using Integer Programming. Most notably, Haslum, Slaney, and Thiébaux (2012) find the value of h^+ by using set-inclusion minimal disjunctive landmarks and solving the IP formulation of a hitting set problem. Imai and Fukunaga (2015) compute the value of h^+ by solving a Mixed Integer and Linear Programming (MILP) model of delete-free planning problems. Another notable work is computing h^+ by using relaxed Decision Diagram based heuristics (Castro et al. 2020).

Methods other than Integer Programming have also been used for finding the exact value of h^+. One such method is SAT encoding of delete-relaxed STRIPS problems done in (Rankooh and Rintanen 2022). This work introduces causal relaxed plan representations and proves that the cost of an optimal plan for a given problem is equal to the cost of an optimal causal relaxed plan representations for that problem.

LP encoding of h^+ using has also been extensively studied. Examples are extracting information from abstraction heuristics (van den Briel et al. 2007), using linear programming to compute heuristic estimates from landmarks for classical planning (Karpas and Domshlak 2009) and numeric planning (Scala et al. 2017), the state-equation heuristic (Bonet 2013), and post-hoc optimization heuristics (Pommerening, Röger, and Helmert 2013). A unified formulation for using different LP approaches has also been introduced (Pommerening et al. 2014).

IP Models for Causal Relaxed Plan Representations

In this section, we present an IP model for causal relaxed plan representations of Definition 3. We divide this into two tasks: encoding of causal partial functions, and encoding of acyclicity in induced causal relation graphs. For the second task, we use two different methods, vertex elimination and time labeling.

The objective of IP Models

For ensuring the optimality of the produced relaxed plan, we need the cost of the produced causal relaxed plan representation to be minimal. In order to do that, for each action $a \in A$ we use variable $f_a \in \{0, 1\}$ to indicate whether a has been chosen to be included in the final relaxed plan. We now introduce the objective of our IP models:

$$minimize \quad \sum_{a \in A} f_a cost(a) \tag{1}$$

Modeling Causal Partial Functions

Let $\Pi = (P, A, I, G, cost)$ be a STRIPS planning problem. Without loss of generality assume that all members of I have been removed from P, preconditions and effects of all actions, and G. In order to encode causal partial function f we use the following variables:

- for each $p \in P$, $f_p \in \{0, 1\}$ indicates whether f is defined for p.
- for each $a \in A$ and $p \in add(a)$, $f_{p,a} \in \{0, 1\}$ represents whether $f(p) = a$.

The following constraints model a causal partial function.

$$\forall p \in P, \quad f_p = \sum_{p \in add(a)} f_{p,a} \tag{2}$$

$$\forall p, q \in P, \quad \left(\sum_{q \in pre(a), p \in add(a)} f_{p,a} \right) \leq f_q \tag{3}$$

$$\forall p \in G, \quad f_p = 1 \tag{4}$$

$$\forall a \in A, p \in add(a) \quad f_{p,a} \leq f_a \tag{5}$$

Formula (2) guarantees that f is a partial function. Formulas (3) and (4) ensure conditions (2) and (3) of causal partial functions, respectively. Constraint (5) is added to enforce the semantics of f_a variables.

Modeling Acyclicity in Causal Relation Graphs

For any given causal partial function f, the graph G_f is needed to be acyclic if f is a relaxed plan representation. We use two different ways for ensuring acyclicity in causal relation graphs: vertex elimination and time labeling. The vertex elimination method has previously been shown to be quite efficient in encoding acyclicity as SAT in causal relation graphs (Rankooh and Rintanen 2022). The time labeling method has also been previously employed for enforcing order on actions and atomic propositions when modeling h^+ as IP (Imai and Fukunaga 2015).

Modeling Acyclicity Using Vertex Elimination We present an IP adaptation of the work done in (Rankooh and Rintanen 2022) for encoding acyclicity in causal relation graphs using vertex elimination. Let $G_\Pi = (P, E_\Pi)$ be the causal relation graph of the STRIPS planning problem $\Pi = (P, A, I, G, cost)$. Assume that $P = \{p_1, \ldots, p_{|P|}\}$. Let O be an elimination ordering for G_Π, and $G^*_\Pi = (P, E^*_\Pi)$ be the vertex elimination graph of G_Π according to O. Let Δ be the set of all triangles produced by elimination ordering O for graph G. Members of Δ are all ordered triples

(p_i, p_j, p_k) such that (p_i, p_k) is added to E^*_Π by eliminating p_j. The IP model of acyclicity in the causal relation graph G_f induced by the causal partial function f for Π using vertex elimination according to O is produced by using constraints (6) to (8). Note that setting variable $e_{i,j} \in \{0,1\}$ to 0 indicates that there is no edge from p_i to p_j in G^*_f, the vertex elimination graph of G_f according to elimination ordering O.

$$\forall a \in A, p_i \in pre(a), p_j \in add(a) \qquad f_{p_j,a} \le e_{i,j} \tag{6}$$

$$\forall (p_i, p_j) \in E^*_\Pi, \qquad e_{i,j} + e_{j,i} \le 1 \tag{7}$$

$$\forall (p_i, p_j, p_k) \in \Delta, \qquad e_{i,j} + e_{j,k} - 1 \le e_{i,k} \tag{8}$$

The number of variables used in this IP model of acyclicity is $\mathcal{O}(\delta|P|) \subseteq \mathcal{O}(|P|^2)$, and the number of constraints is $\mathcal{O}(\delta^2|P|) \subseteq \mathcal{O}(|P|^3)$, where δ is the directed elimination width of O for G_Π.

For a given STRIPS planning problem $\Pi = (P, A, I, G, cost)$ and order O on members of Π, we use IP-VE(Π, O) to refer to the problem of reaching objective (1) subject to constraints (2) to (8). IP-VE(Π, O) is our vertex elimination based IP model, using O as the vertex elimination order.

Modeling Acyclicity Using Time Labels Using time labels in the IP model of h^+ has been used before in (Imai and Fukunaga 2015) for ordering atomic propositions and actions in the produced relaxed plan. Here, we use IP-IF to denote the IP model of (Imai and Fukunaga 2015). We use the same method for guaranteeing acyclicity in causal relation graphs. This method simply works by assigning time labels to variables and ensuring that the time label of p is greater than the time label of q by at least 1, if there is an edge from q to p in the causal relation graph. To do so, we use variables $t_i \in \{1, ..., |P|\}$ to indicate the time label of p_i for each $p_i \in P$.

A standard approach in IP to encode $(y = 1) \rightarrow (Ax \le b)$ is to use $Ax \le b + u(1 - y)$, where u is an upper bound on $Ax - b$. In the case that y is equal to 0, the constraint becomes $Ax - b \le u$, which is always true. In our time label based model, for action a that requires p_i and adds p_j, we need to encode $(f_{p_j,a} = 1) \rightarrow (t_i - t_j \le -1)$. Therefore, we use constraint (9) to encode acyclicity using time labels.

$$\begin{aligned} &\forall a \in A, p_i \in pre(a), p_j \in add(a), \\ &\quad t_i - t_j + 1 \le |P|(1 - f_{p_j,a}) \end{aligned} \tag{9}$$

For a given STRIPS planning problem $\Pi = (P, A, I, G, cost)$, we use IP-TL(Π) to refer to the problem of reaching objective (1) subject to constraints (2) to (5), and (9). IP-TL(Π) is our time label based IP model.

The IP-TL model differs with IP-IF in three ways. First, since we know that time labels are used to order atomic propositions in causal relation graphs, we need to use no more than $|P|$ values for our labels, as no simple path of length greater than $|P|$ can exist in a graph with $|P|$ vertices. The upper bound used in constraint (9) is also $|P|$ for the same reason. In contrast, IP-IF uses variables with $|A| + 1$ values and an upper bound of $|A| + 1$ for producing constraints on time labels. However, for planning problems, $|A|$ is usually considerably greater than $|P|$. Greater numbers of values for the same variables can mean a larger state space for the solver, and as a result, a larger solving time. Also, a simple investigation can show that using a greater upper bound for producing constraints on time labels can loosen the constraints in the LP relaxation of the IP model, resulting in less informative heuristics.

Second, IP-IF also incorporates time labels for actions, which is redundant considering that acyclicity is needed to be enforced on a graph whose vertices are atomic propositions of the given problem. These redundant variables can also enlarge the search space of the IP solver.

Lastly, for every $a \in A$, $p \in add(a)$, and $q \in pre(a)$, IP-IF enforces a constraint in the form of $f_q \ge f_{p,a}$ for guaranteeing the truth of preconditions of an action, if the action has been chosen to be included in the final relaxed plan. In contrast we use constraint (3) for the same reason. While these two constraints are equivalent for IP models, constraint (3) can be tighter when considering the LP relaxation of the models. Example 1 shows how this can happen for a simple and abstract problem.

Example 1. *Let $\Pi = (P, A, I, G, cost)$ be a planning problem, where $P = \{x, g\}$, $A = \{a, b, c\}$, $I = \emptyset$, $G = \{g\}$, $pre(a) = \emptyset$, $pre(b) = pre(c) = \{x\}$, $add(a) = \{x\}$, $add(b) = add(c) = \{g\}$, and $cost(a) = cost(b) = cost(c) = 1$. Let us assume that the IP-IF model is exactly the same as IP-TL, except for the difference in constraint (3) explained above. The solution of both IP-IF and IP-TL models are forced to include a and exactly one of b and c in the produced plan. Therefore, the optimal objective value will be 2. However, a solution to the LP relaxation of IP-IF can assign 0.5 to f_a, f_b, and f_c, coming up with objective value 1.5. This cannot be true for a feasible solution of the LP relaxation of IP-TL. That is because assigning 0.5 to f_b and f_c, enforces $f_x \ge 1$ according to constraint (3), which causes $f_a = 1$ according to constraints (2) and (5). A straightforward investigation of this example can show that the optimal objective value for LP-TL is 2.*

Correctness Proofs

We here provide the proofs for soundness and completeness of our encodings. Since considering the discussion above about the relation between IP-TL and IP-IF, proofs for correctness of our IP-TL model can be given by applying minor modifications to proofs for Proposition 1 and 2 of (Imai and Fukunaga 2015), we only give the proofs for the IP-VE model.

Theorem 1. *Let $\Pi = (P, A, I, G, cost)$ be a STRIPS planning problem, and O be any order on members of $|P|$. If f is a causal relaxed plan representation for Π with cost c, then IP-VE(Π, O) has a feasible solution with objective value c.*

Proof. From f, we produce a feasible solution with objective value c for IP-VE(Π, O). For every $a \in A$ and $p \in P$, let the value of variables f_p, and $f_{p,a}$ be equal to 1 iff f is defined on p and $f(p) = a$, respectively. Also, for every $a \in A$

let $f_a = 1$ iff $f_{p,a} = 1$ for some $p \in P$. It should be obvious that with these assignments, constraints (2) to (5) are satisfied and the objective function in (1) equals to c. Assume that G_f is the causal relation graph of f and G_f^* is the vertex elimination graph of G_f according to elimination ordering O. Since f is a causal relaxed plan representation, G_f must be acyclic. Since G_f^* is a subgraph of the transitive closure graph of G_f, it is also acyclic. Let O' be a topological order of $|P|$ according to G_f^*. For every i and j, we let the value of variable $e_{i,j}$ be equal to 1 iff p_i is ordered before p_j according to O'. Constraint 6 is satisfied because if $f_{p_j,a} = 1$ and $p_i \in pre(a)$ then there is an edge in G_f from p_i to p_j, and therefore, p_i must be ordered before p_j according to O'. Constraint 7 holds trivially. Constraint 8 is also satisfied because if p_i is ordered before p_j and p_j is ordered before p_k according to O', then p_i must be ordered before p_k. $\square$

Theorem 2. *Let $\Pi = (P, A, I, G, cost)$ be a STRIPS planning problem, and O be any order on members of $|P|$. If IP-VE(Π, O) has a feasible solution with objective value c, then there exists a causal relaxed plan representation for Π with cost at most c.*

Proof. Let S be a feasible solution for p. From S We produce a partial function f from P to A, and prove that f is a causal relaxed plan representation for Π. For every $p \in P$, let $f(p) = a$ iff $S(f_{p,a}) = 1$. f is a well-defined partial function because constraint (2) is satisfied by S, and therefore, for $a' \neq a$ at least one of $f_{p,a}$ and $f_{p,a'}$ must equal zero. Moreover, it should be clear from constraint (2) that f is defined on $p \in P$ iff $S(f_p) = 1$. Consider conditions 1 to 3 of Definition 1. Condition 1 holds because IP-VE(Π, O) has the variable $f_{p,a}$ iff a adds p. For every $a \in A$, $p \in add(a)$, and $q \in pre(a)$, if $f(p) = a$, we have $S(f_q) = 1$ because S satisfies constraint (3). Therefore, since we have assumed that all members of I have been removed from preconditions of all actions before modeling the problem, condition 2 of Definition 1 holds. Condition 3 also holds because S satisfies constraint (4). Thus, f is a causal partial function for Π.

Consider $G_f = (P, E_f)$ of Definition 2. Let $G_f^* = (P, E_f^*)$ be the vertex elimination graph of G_f produced according to elimination ordering O. If $(p_i, p_j) \in E_f$, by Definition 2, for some a such that $p_i \in pre(a)$ and $p_j \in add(a)$, we have: $f(p_j) = a$, and by constraint (3), $S(e_{i,j}) = 1$. Now, because S satisfies (8), we can conclude that for every $(p_l, p_m) \in E_f^*$, we have $S(e_{l,m}) = 1$.

By induction on n, we show that G_f^* cannot have any cycle with length n, and therefore, G_f cannot have any cycle with length n. Base case: for $n = 2$, the conclusion holds because S satisfies constraint (7). Induction hypothesis: assume that for $k > 2$ the conclusion holds for $n = k$. Let $p_0, ..., p_{k+1} = p_0$ be a simple directed cycle in G_f^* of length $k + 1$. For the sake of simplicity, suppose that all indices are modulo $k + 1$. Assume that p_j is the first vertex in this cycle eliminated according to order O. Therefore, $(p_{j-1}, p_{j+1}) \in E_f^*$, and $p_0, ..., p_{j-1}, p_{j+1}, ..., p_{k+1} = p_0$ must be a simple directed cycle in G_f^* of length k, a contradiction.

We conclude that G_f is acyclic, hence f is a causal relaxed plan representation for Π. If f maps p to action a, then according to constraint 5, $f_a \geq f_{p,a} = 1$. Thus, $cost(f)$ is at most c. $\square$

LP Relaxation of the IP Models

An advantage of IP modeling of h^+ is that admissible heuristics can be obtained by solving LP relaxation of the produced models. Given a STRIPS planning problem Π and elimination order O on all atomic propositions of Π, let LP-VE(Π, O) and LP-TL(Π) be respectively the LP relaxation of IP-VE(Π, O) and IP-TL(Π) explained above. Also, let $h_{VE}(\Pi, O)$, and $h_{TL}(\Pi)$ be the value of the objective function of the solution of LP-VE(Π, O) and LP-TL(Π), respectively. Both h_{VE} and h_{TL} are admissible heuristics. In this section, we theoretically investigate the relation between these two heuristics. More specifically, we prove that h_{VE} is at least as informative as h_{TL}.

Relation Between h_{VE} and h_{TL}

We show that for any elimination ordering O, given feasible solution S_{VE} for LP-VE(Π, O), feasible solution S_{TL} for LP-TL(Π) with the same objective value can be produced. All variables in the form of $f_{p,a}$, f_p, and f_a are present in both LP-VE and LP-TL models. Let S_{TL} be equal to S_{VE} for all those variables. We only need to provide values for variables t_i such that constraint (9) is satisfied.

The idea of our proof comes from the observation that IP-TL guarantees acyclicity by enforcing arc consistency, while IP-VE enforces path consistency for the same purpose. The outline of the proof is as follows: given a model for LP-VE, we construct a weighted graph named the VE-graph. We prove an upper bound on the total weight of any simple cycle of this graph. We then construct another weighted graph named the TL-graph from the VE-graph, and show that the TL-graph, while possibly having edges with negative weights, cannot have any cycle with negative total weight. We consider a dummy vertex which has outgoing edges with weights zero to all other vertices of the TL-graph. A shortest path must exist from this dummy vertex to every other vertex. We set t_i to the shortest distance from the dummy vertex to p_i, and show that these values satisfy constraint (9).

Definition 4 (Induced VE-graphs)**.** *Let $\Pi = (P, A, I, G, cost)$ be a STRIPS planning problem, $G_\Pi = (P, E_\Pi)$ be the causal relation graph of Π, O be any ordering on members of P, $G_\Pi^* = (P, E_\Pi^*)$ be the vertex elimination graph of G_Π according to elimination order O, and S_{VE} be a feasible solution for LP-VE(Π, O). The VE-graph induced by S_{VE}, denoted by $G_{VE} = (P, E_\Pi^*)$, is a weighted directed graph with weight function w, such that for all $(p_i, p_j) \in E_\Pi^*$, $w(p_i, p_j) = S_{VE}(e_{i,j})$.*

Lemma 1. *If $p_0, ..., p_n = p_0$ is a simple directed cycle in G_{VE} of length n, then $\sum_{i=0,...,n-1} w(p_i, p_{i+1}) \leq n - 1$.*

Proof. We give the proof by induction on n. Base case: for $n = 2$, the conclusion holds because S_{VE} satisfies constraint (7). Induction hypothesis: assume that for $k > 2$ the conclusion holds for $n = k$. Let $p_0, ..., p_{k+1} = p_0$ be a simple directed cycle in G_{VE} of length $k+1$. For the sake of simplicity, suppose that all indices are modulo $k+1$. Assume that p_j is the first vertex in this cycle eliminated according to order O. Then $p_0, ..., p_{j-1}, p_{j+1}, ..., p_{k+1} = p_0$ must be a simple directed cycle in G_{VE} of length k. Therefore, $\sum_{i=0,...,j-2,j+1,,n-1} w(p_i, p_{i+1}) + w(p_{j-1}, p_{j+1}) \leq k - 1$. However, since according to constraint (8), $w(p_{j-1}, p_j) + w(p_j, p_{j+1}) - 1 \leq w(p_{j-1}, p_{j+1})$, we have $\sum_{i=0,...,j-2,j+1,,k} w(p_i, p_{i+1}) + w(p_{j-1}, p_j) + w(p_j, p_{j+1}) - 1 \leq k - 1$, and therefore, $\sum_{i=0,...,k} w(p_i, p_{i+1}) \leq k$. □

Definition 5 (Induced TL-graphs)**.** *Let* $\Pi = (P, A, I, G, cost)$ *be a STRIPS planning problem,* $G_\Pi = (P, E_\Pi)$ *be the causal relation graph of* Π*,* O *be any elimination ordering on members of* P*, and* S_{VE} *be a feasible solution for LP-VE(*Π, O*). The TL-graph induced by* S_{VE}*, denoted by* $G_{TL} = (P, E_\Pi)$*, is a weighted directed graph with weight function* w'*, such that for all* $(p_i, p_j) \in E_\Pi$*,* $w'(p_i, p_j) = 1 - 1 \;/\; |P| - w(p_i, p_j)$*, where* w *is the weight function of VE-graph induced by* S_{VE}*.*

Lemma 2. *If* $p_0, ..., p_n = p_0$ *is a simple directed cycle in* G_{TL} *of length* n*, then* $\sum_{i=0,...,n-1} w'(p_i, p_{i+1}) \geq 0$*.*

Proof. If $p_0, ..., p_n$ is a simple directed cycle in G_{TL}, then it is also a simple directed cycle in G_{VE}, and by Lemma 1, $\sum_{i=0,...,n-1} w(p_i, p_{i+1}) \leq n - 1$. However, we have: $\sum_{i=0,...,n-1} w'(p_i, p_{i+1}) = n - n \;/\; |P| - \sum_{i=0,...,n-1} w(p_i, p_j)$. Thus, $\sum_{i=0,...,n-1} w'(p_i, p_{i+1}) \geq n - n \;/\; |P| - n + 1 = 1 - n \;/\; |P|$. Since $n \;/\; |P| \leq 1$, we have $\sum_{i=0,...,n-1} w'(p_i, p_{i+1}) \geq 0$. □

Theorem 3. *Let* $\Pi = (P, A, I, G, cost)$ *be a STRIPS planning problem and* O *be any order on members of* P*.* $h_{VE}(\Pi, O) \geq h_{TL}(\Pi)$*.*

Proof. Let S_{VE} be a feasible solution for LP-VE(Π, O). We construct solution S_{TL} for LP-TL(Π). For every variable x in form of f_p, $f_{p,a}$, and f_a, let $S_{TL}(x) = S_{VE}(x)$. Consider G_{VE} to be the VE-graph induced by S_{VE} with weight function w, and G_{TL} to be the TL-graph induced by S_{VE} with weight function w'. We add a dummy vertex v to G_{TL} and an edge with weight zero from v to every vertex of G_{TL}. By Lemma 2, G_{TL} does not have any cycle with negative total weight. Therefore, a shortest distance from v to every vertex p_i of G_{TL}, denoted by $d(p_i)$, exists. Let $S_{TL}(t_i) = -d(p_i)$. Since there exists an edge with weight zero from v to p_i, we have $S_{TL}(t_i) \geq 0$. On the other hand, since w' is always greater than -1, we have $S_{TL}(t_i) \leq |P|$. We only need to show that constraint (9) is satisfied. Let $a \in A$, $p_i \in pre(a)$, and $p_j \in add(a)$. There must exist an edge (p_i, p_j) in G_{TL} with weight $1 - 1 \;/\; |P| - w(p_i, p_j) = 1 - 1 \;/\; |P| - S_{VE}(e_{i,j})$. Since $d(p_i)$ and $d(p_j)$ are the shortest distances from v to p_i and p_j, respectively, we have: $d(p_j) \leq d(p_i) + 1 - 1 \;/\; |P| - S_{VE}(e_{i,j})$. Therefore, $S_{TL}(t_i) - S_{TL}(t_j) \leq 1 - 1 \;/\; |P| - S_{VE}(e_{i,j})$. Because $|P| \geq 1$, by multiplying the right-hand side by $|P|$ we have $S_{TL}(t_i) - S_{TL}(t_j) \leq |P| - 1 - S_{VE}(e_{i,j})|P|$, and therefore, $S_{TL}(t_i) - S_{TL}(t_j) + 1 \leq |P|(1 - S_{VE}(e_{i,j}))$. Since S_{VE} is a feasible solution for LP-VE(Π, O), it must satisfy (6). Thus $S_{TL}(f_{p_j,a}) = S_{VE}(f_{p_j,a}) \leq S_{VE}(e_{i,j})$, and $S_{TL}(t_i) - S_{TL}(t_j) + 1 \leq |P|(1 - S_{TL}(f_{p_j,a}))$. We conclude that S_{TL} is a feasible solution for LP-TL(Π). Because the value of objective function for S_{TL} is equal to that of S_{VE}, we have: $h_{VE}(\Pi, O) \geq h_{TL}(\Pi)$. □

Theorem 3 shows that h_{VE} dominates h_{TL}, no matter what ordering has been used for variable elimination. it can be shown that there are problems for which h_{VE} is strictly more informative. However, since concrete examples of such problems are quite complicated, we show this using our empirical analysis. In fact, our empirical results show that h_{VE} can be considerably more informative than h_{TL}.

Empirical Results

We have implemented our IP and LP models inside the HSP* planner (Haslum 2021). All experiments have been run on a cluster of Linux machines, using a timeout of 1800 seconds per problem, and, if not stated otherwise, a memory limit of 4 GB. In versions that vertex elimination is used, for determining the order of vertex elimination, we have implemented the *minimum degree* heuristic, i.e., eliminating a vertex with minimal total number of incoming and outgoing edges in the graph produced after the elimination of previously eliminated vertices. As the optimizer, we have used IBM ILOG CPLEX Optimization Studio 20.1[1].

All models also use some of the preprocessing methods presented in (Imai and Fukunaga 2015). These methods are 1) finding *fact landmarks* (Gefen and Brafman 2011) and adding them to the goal conditions; 2) doing action relevance analysis and removing non-relevant actions; and 3) dominated action elimination. The implementation of these methods was part of the base HSP* package.

As our benchmark problem set, we have used the STRIPS planning problem sets found in the *planning repository*[2]. From IPC domains, domains from both satisficing and optimal tracks have been considered. In total, 2212 problem instances from 84 problem sets have been used for comparison.

Computing the Exact Value of h^+

Our IP-VE and IP-TL models can be solved to compute the exact value of h^+. Henceforth, We call our solvers by their corresponding model name, IP-VE, and IP-TL. To evaluate their efficiency, we have compared our solvers with 1) SAT, the boolean satisfiability based encoding used in (Rankooh and Rintanen 2022); 2) IF, the solver with IP model introduced in (Imai and Fukunaga 2015); and 3) HST, the minimum-cost hitting set based method introduced in

[1]https://www.ibm.com/products/ilog-cplex-optimization-studio

[2]https://github.com/AI-Planning/classical-domains

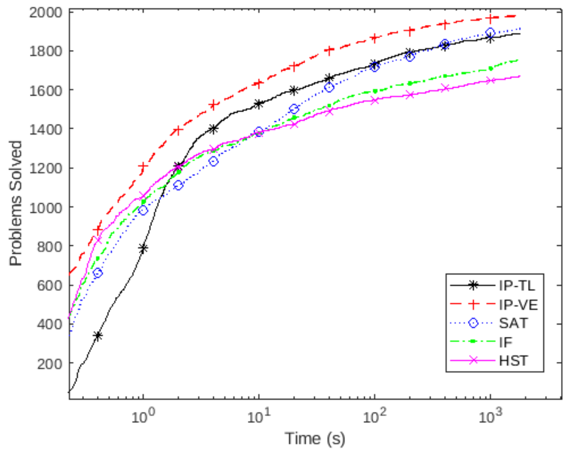

Figure 1: Cumulative number of delete-relaxed problems solved by the competing methods

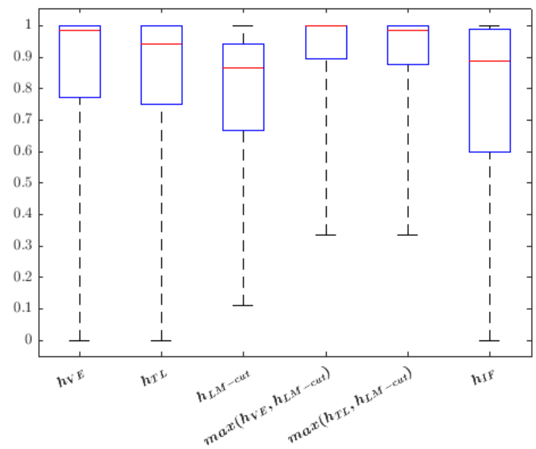

Figure 2: Informativeness of competing admissible heuristics

(Haslum, Slaney, and Thiébaux 2012). Although the Decision Diagram based method (Castro et al. 2020) is more recent than HST and IF, its results are strongly dominated by other methods and therefore, not presented here. Since the original implementation of IF is not publicly accessible, we have used the implementation of IF that is included in HST solver. For the SAT method we have used *Kissat* (Biere et al. 2020) as the SAT solver. Because the encodings of SAT need large amount of memory in some cases, we have given this solver 16GB of memory. All solvers have been run on the delete-relaxed versions of the benchmark problems. Cumulative number of problems solved by all methods are presented in Figure 1.

Out of the 2212 problems under study, the exact value of h^+ was computed in 1800 seconds for 1980, 1913, 1889, 1715, and 1669 problems by IP-VE, SAT, IP-TL, IF, and HST, respectively. As it can be seen in Figure 1, IP-VE significantly outperforms all of the other solvers. In fact, no matter what the time limit is, IP-VE computes h^+ for more problems compared to any other solver. The IP-TL, despite performing rather slowly in the beginning, outperforms IF and HST, and is competitive with SAT when given enough time.

Informativeness of the LP-based Heuristics

To evaluate the informativeness of heuristics obtained by solving our LP-VE and LP-TL methods, we have computed their accuracy in estimating the value of h^+. This can be done by dividing the value of the heuristics to the exact value of h^+ obtained by our IP-VE solver. The accuracy of our LP-based heuristics are compared with that of the LM-cut heuristic (Helmert and Domshlak 2009), and also the LP relaxation of the IP model used in (Imai and Fukunaga 2015), denoted by LP-IF. We have used the implementation of the LM-cut heuristic which is a part of the HSP* planner.

Figure 2 shows the spread of obtained accuracies by separate box plots. For the sake of clarity, we have not included the problems for which all mentioned heuristics were able to produce the exact value of h^+. Each plot illustrates the minimum, the maximum, the median, and the first and third quartile of the accuracy of the corresponding heuristic function on the 1319 remaining problems.

As it can be seen in Figure 2, both our heuristics are quite informative when compared to LM-cut and LP-IF. Our empirical results also confirm the result of Theorem 3, showing that $h_{VE} \geq h_{TL}$ in all cases. Furthermore, it can be deduced from Figure 2 that h_{VE} is strictly greater than h_{TL} for some problems. In fact, the median accuracy of h_{VE} is 98.5 percent for the problems under study, considerably higher than median accuracy of h_{TL}, which is 94.2 percent.

We have also investigated how complementary our heuristics are to the LM-cut heuristic. For this purpose, we have provided the box plots for the maximum of h_{VE} and $h_{LM\text{-}cut}$, and also the maximum of h_{TL} and $h_{LM\text{-}cut}$ in Figure 2. Our results show that in more than half of the problems for which not all heuristics under study can compute the exact value of h^+, $max(h_{VE}, h_{LM\text{-}cut})$ gives the exact value. Moreover, in half of the problems under study, $max(h_{TL}, h_{LM\text{-}cut})$ has an accuracy of 98.5 or more. These results show that our heuristics can do quite a good job in making up for the lack of information of the LM-cut heuristic in relation to h^+.

Using h_{VE} and h_{TL} for Optimal Planning

The informativeness of h_{VE} and h_{TL} comes at a price. Computation of the LM-cut heuristic takes a time at most quadratic in the size of the input planning problem (Helmert and Domshlak 2009). Our LP-TL model, on the other hand, uses a number of variables linear in size of the planning problem, which makes it impossible to be solved in quadratic time using the current algorithms (Jiang et al. 2021). The situation for our LP-VE heuristic is even worse, as it may use a quadratic number of variables in the size of the input problem.

We have compared the time needed to compute h_{TL} with the time needed for computing $h_{LM\text{-}cut}$ (Figure 3), and h_{VE}

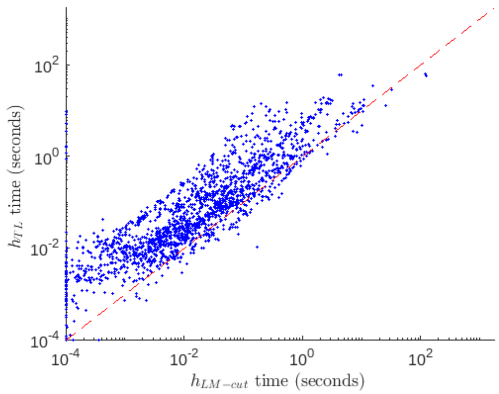

Figure 3: Comparison of time (in seconds) needed to copmute h_{TL} and $h_{LM\text{-}cut}$

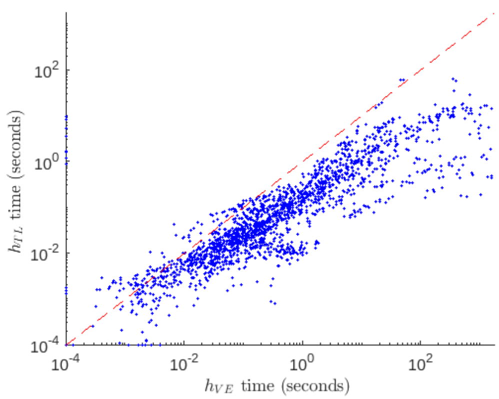

Figure 4: Comparison of time (in seconds) needed to copmute h_{TL} and h_{VE}

(Figure 4), for all benchmark problems. As it can be seen $h_{LM\text{-}cut}$ is computed in many cases almost one order of magnitude faster than h_{TL}. Also, computing h_{TL} is considerably faster than computing h_{VE}.

To see how h_{VE}, h_{TL}, and our IP based implementation of h^+ work as heuristic in state space search for cost optimal planning, we have used these functions as the heuristic for A* search provided inside the HSP* planner. The cumulative numbers of problems solved by A* when using $h_{LM\text{-}cut}$, h_{VE}, h_{TL}, h_{IF} (by solving LP relaxation of the IP model introduced in (Imai and Fukunaga 2015)), h^+_{IP} (by solving our IP-VE model), and $max(h_{TL}, h_{LM\text{-}cut})$ are shown in Figure 5. Note that out of the mentioned heuristic functions, the time complexity of computing h^+_{IP} is exponential.

It can be seen in Figure 5 that fast computation of $h_{LM\text{-}cut}$ pays off. Using LM-cut, A* solves 822 problems, 39 problems more than the 783 problems solved when using h_{TL}. However, when using the maximum of h_{TL} and $h_{LM\text{-}cut}$, 813 problems are solved, which is quite close to the coverage of $h_{LM\text{-}cut}$, despite its costly heuristic computation. Moreover, the number of problems solved when using h_{VE} and h^+_{IP} is 709 and 717, respectively, considerably lower

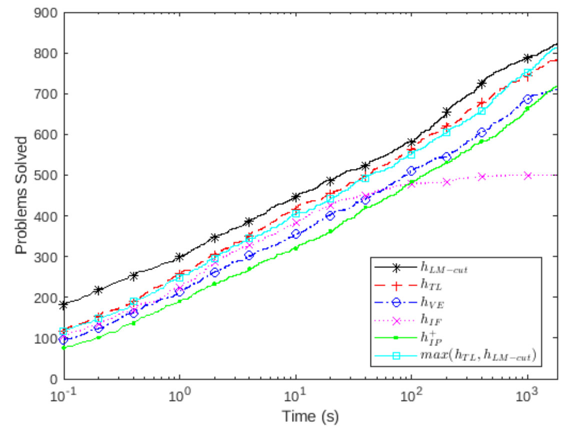

Figure 5: Cumulative number of problems solved by A* using different heuristics

than when h_{TL} is used.

Despite the observation of inefficiency of using h_{VE} in the mentioned setting, it cannot be concluded that h_{VE} is altogether without benefit. h_{VE} could be helpful in the settings where more effort is done when expanding a search node. One obvious example of such settings would be when a portfolio of many different heuristic functions is used for evaluating a given search node, a prominent approach for efficient optimal planning. We leave investigating such settings to future research.

Conclusion

We introduced new IP/LP models for computing h^+. The new models, IP/LP-VE and IP/VE-TL, were based on enforcing acyclicity in causal relation graphs of causal relaxed plan representations for given STRIPS planning problems, using vertex elimination and time labeling methods. We provided proofs for correctness of our models, and proved that the LP-VE based heuristic dominates the LP-TL based one. Our empirical results show that while the IP-VE model computes h^+ faster than IP-TL and all previously introduced methods, the LP-TL model is faster to solve compared to LP-VE, and is more suitable for using as heuristic function for optimal planning. Our results also show that the new admissible heuristics are often more informative compared to the LM-cut heuristic. Our LP-VE and LP-TL based heuristics were also shown to be effective in compensating for the lack of informativeness of the LM-cut heuristic.

References

Betz, C.; and Helmert, M. 2009. Planning with h^+ in theory and practice. In *KI 2009: Advances in Artificial Intelligence, 32nd Annual German Conference on AI, Paderborn, Germany, September 15-18, 2009. Proceedings*, volume 5803 of *Lecture Notes in Computer Science*, 9–16. Springer-Verlag.

Biere, A.; Fazekas, K.; Fleury, M.; and Heisinger, M. 2020. CaDiCaL, Kissat, Paracooba, Plingeling and Treengeling entering the SAT competition 2020. In *Proceedings of SAT COMPETITION 2020*, 50–54.

Bonet, B. 2013. An admissible heuristic for SAS+ planning obtained from the state equation. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 2268–2274. IJCAI/AAAI.

Bonet, B.; and Geffner, H. 2001. Planning as heuristic search. *Artificial Intelligence*, 129(1-2): 5–33.

Bylander, T. 1994. The computational complexity of propositional STRIPS planning. *Artif. Intell.*, 69(1-2): 165–204.

Castro, M. P.; Piacentini, C.; Ciré, A. A.; and Beck, J. C. 2020. Solving delete free planning with relaxed decision diagram based heuristics. *Journal of Artificial Intelligence Research*, 67: 607–651.

Gefen, A.; and Brafman, R. I. 2011. The minimal seed set problem. In *Proceedings of the 21st International Conference on Automated Planning and Scheduling, ICAPS 2011*, 319–322. AAAI Press.

Haslum, P. 2012. Incremental lower bounds for additive cost planning problems. In *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling, ICAPS 2012*, 74–82. AAAI Press.

Haslum, P. 2021. HSP* code and documentation http://users.cecs.anu.edu.au/patrik/un-hsps.html.

Haslum, P.; Slaney, J. K.; and Thiébaux, S. 2012. Minimal landmarks for optimal delete-free planning. In *Proceedings of the Twenty-Second International Conference on Automated Planning and Scheduling, ICAPS 2012*, 353–357. AAAI Press.

Helmert, M.; and Domshlak, C. 2009. Landmarks, critical paths and abstractions: what's the difference anyway? In *Proceedings of the 19th International Conference on Automated Planning and Scheduling, ICAPS 2009*, 162–169. AAAI Press.

Hunter, P.; and Kreutzer, S. 2007. Digraph measures: Kelly decompositions, games, and orderings. In *Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2007*, 637–644. SIAM.

Imai, T.; and Fukunaga, A. 2015. On a practical, integer-linear programming model for delete-free tasks and its use as a heuristic for cost-optimal planning. *Journal of Artificial Intelligence Research*, 54: 631–677.

Jiang, S.; Song, Z.; Weinstein, O.; and Zhang, H. 2021. A faster algorithm for solving general LPs. In *STOC'21: 53rd Annual ACM SIGACT Symposium on Theory of Computing, Virtual Event, Italy, June 21-25, 2021*, 823–832. ACM.

Karpas, E.; and Domshlak, C. 2009. Cost-optimal planning with landmarks. In *IJCAI 2009, Proceedings of the 21st International Joint Conference on Artificial Intelligence*, 1728–1733.

Keyder, E.; and Geffner, H. 2008. Heuristics for planning with action costs revisited. In *ECAI 2008 - 18th European Conference on Artificial Intelligence*, volume 178 of *Frontiers in Artificial Intelligence and Applications*, 588–592. IOS Press.

Mirkis, V.; and Domshlak, C. 2007. Cost-sharing approximations for h+. In *Proceedings of the Seventeenth International Conference on Automated Planning and Scheduling, ICAPS 2007*, 240–247. AAAI Press.

Pommerening, F.; Röger, G.; and Helmert, M. 2013. Getting the most out of pattern databases for classical planning. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence*, 2357–2364. IJCAI/AAAI.

Pommerening, F.; Röger, G.; Helmert, M.; and Bonet, B. 2014. LP-based heuristics for cost-optimal planning. In *Proceedings of the Twenty-Fourth International Conference on Automated Planning and Scheduling, ICAPS 2014*, 226–234. AAAI Press.

Rankooh, M. F.; and Rintanen, J. 2022. Efficient encoding of cost optimal delete-free planning as SAT. In *Proceedings of the 36th AAAI Conference on Artificial Intelligence (AAAI-12)*. AAAI Press.

Robinson, N.; McIlraith, S. A.; and Toman, D. 2014. Cost-based query optimization via AI planning. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, 2344–2351. AAAI Press.

Rose, D. J.; and Tarjan, R. E. 1975. Algorithmic aspects of vertex elimination. In *Proceedings of the 7th Annual ACM Symposium on Theory of Computing*, 245–254. ACM.

Scala, E.; Haslum, P.; Magazzeni, D.; and Thiébaux, S. 2017. Landmarks for numeric planning problems. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI 2017*, 4384–4390. ijcai.org.

van den Briel, M.; Benton, J.; Kambhampati, S.; and Vossen, T. 2007. An LP-based heuristic for optimal planning. In *Principles and Practice of Constraint Programming - CP 2007, 13th International Conference, CP 2007*, volume 4741 of *Lecture Notes in Computer Science*, 651–665. Springer-Verlag.