Custom-Design of FDR Encodings: The Case of Red-Black Planning

Daniel Fišer\textsuperscript{1,2*}, Daniel Gnad\textsuperscript{1}, Michael Katz\textsuperscript{3} and Jörg Hoffmann\textsuperscript{1}

\textsuperscript{1}Saarland University, Saarland Informatics Campus, Saarbrücken, Germany
\textsuperscript{2}Czech Technical University in Prague, Faculty of Electrical Engineering, Czech Republic
\textsuperscript{3}IBM Research, Yorktown Heights, NY, USA
danfis@danfis.cz, \{gnad,hoffmann\}@cs.uni-saarland.de, michael.katz1@ibm.com

Abstract

Classical planning tasks are commonly described in PDDL, while most planning systems operate on a grounded finite-domain representation (FDR). The translation of PDDL into FDR is complex and has a lot of choice points—it involves identifying so called mutex groups—but most systems rely on the translator that comes with Fast Downward. Yet the translation choice points can strongly impact performance. Prior work has considered optimizing FDR encodings in terms of the number of variables produced. Here we go one step further by proposing to custom-design FDR encodings, optimizing the encoding to suit particular planning techniques. We develop such a custom design here for red-black planning, a partial delete relaxation technique. The FDR encoding affects the causal graph and the domain transition graph structures, which govern the tractable fragment of red-black planning and hence affects the respective heuristic function. We develop integer linear programming techniques optimizing the scope of that fragment in the resulting FDR encoding. We empirically show that the performance of red-black planning can be improved through such FDR custom design.

1 Introduction

Classical planning tasks are usually defined in PDDL [McDermott, 2000], which is a lifted representation based on first-order logic. Yet, most state-of-the-are planners, in particular the systems based on Fast Downward (FD) [Helmert, 2006], use a finite-domain representation (FDR) encoding, where subsets of propositional facts are grouped into finite-domain state variables. This is possible only if reachable states make at most one of the grouped facts true, i.e., if they form a mutex group. FDR encodings are useful for a wide variety of purposes like abstraction based heuristics [Haslum et al., 2007; Katz and Domshlak, 2010; Helmert et al., 2014; Seipp and Helmert, 2018], or red-black planning heuristics [Domshlak et al., 2015; Gnad and Hoffmann, 2015; Katz, 2019], where a subset of (red) variables are delet-relaxed while the rest (black) keep their true semantics.

The translation of PDDL into FDR is complex—deciding whether a set of facts forms a mutex group is as hard as planning itself. There are many choice points in identifying mutex groups and grouping facts into state variables. Systems based on Fast Downward almost universally rely on its translator [Helmert, 2009] and do not question the choices made there. Yet the translation choices can impact performance. In prior work [Dvořák et al., 2013; Dvořák et al., 2015], a minimization of the number of FDR variables was proposed assuming fewer variables result in more information being aggregated into them. The results show that significantly reducing the number of variables can improve the performance of FDR-based techniques. The improvements are moderate though, and there is a lot of variance depending on domain and planning technique; a consistent picture does not emerge.

Here we show that this picture can change when going one step further in the target of FDR optimization: we propose to custom-design FDR encodings, optimizing the encoding to suit particular planning techniques.\footnote{Vallati et al. [2015; 2017] have explored domain model optimization tailored to specific planners before, but for PDDL not for FDR variable design, and changing only the ordering of artefacts in the input model (which had previously been observed to be relevant for the performance of some planners [Howe and Dahlman, 2002]).} We develop such a custom design for red-black planning, which distinguishes between relaxed (red) and non-relaxed (black) variables. We focus on a polynomial-time solvable fragment of red-black planning [Katz et al., 2013b] which uses a heuristic function based on generating a red-black plan for a given state [Katz et al., 2013a]. The tractability of red-black planning is determined by the decision which variables are painted red or black. The state-of-the-art approach uses painting strategies [Domshlak et al., 2015] assuming the FDR encoding is given and fixed, and decides only on variable colors. Here, we open up this design space and allow to custom-design the FDR variables and their painting at the same time.

The tractability of red-black planning is governed by the FDR encoding’s causal graph (whose projection onto the black variables must be acyclic) and domain transition graphs (that must be invertible for the black variables). We develop techniques optimizing these explicit syntactic criteria, to be able to choose from different viable FDR encodings.

\textsuperscript{*}Contact Author
of tractable red-black planning tasks. The optimization itself is then formulated as an integer linear program, where the allocation of black (non-relaxed) variables is controlled by the program’s objective function.

2 Background
A STRIPS planning task \( \Pi \) is specified by a tuple \( \Pi = (F, O, s_I, s_G) \), where \( F = \{ f_1, \ldots, f_n \} \) is a set of facts, and \( O = \{ o_1, \ldots, o_m \} \) is a set of grounded operators. A state \( s \in F \) is a set of facts, \( s_I \subseteq F \) is an initial state and \( s_G \subseteq F \) is a goal specification. A state \( s \) is a goal state iff \( s_G \subseteq s \).

An operator \( o \) is a tuple \( o = (\text{pre}(o), \text{add}(o), \text{del}(o)) \), where \( \text{pre}(o) \subseteq F \) is the set of preconditions of \( o \), and \( \text{add}(o) \subseteq F \) and \( \text{del}(o) \subseteq F \) are the sets of add and delete effects, respectively. All operators are well-formed, i.e., \( \text{pre}(o) \cap \text{add}(o) = \emptyset \) and \( \text{pre}(o) \cap \text{del}(o) = \emptyset \). We assume unit costs. An operator \( o \) is applicable in a state \( s \) if \( \text{pre}(o) \subseteq s \). The resulting state of applying an applicable operator \( o \) in a state \( s \) is the state \( s'[o] = \{ \text{del}(o) \cup \text{add}(o) \} \).

A sequence of operators \( \pi = \{ o_1, \ldots, o_n \} \) is applicable in a state \( s_0 \) if there are states \( s_1, \ldots, s_n \) such that \( o_i \) is applicable in \( s_{i-1} \) and \( s_i = o_i[s_{i-1}] \) for \( 1 \leq i \leq n \). The resulting state of this application is \( s_n = s_0 \).

A set of facts \( M \subseteq F \) is called a mutex if \( M \not\subseteq s \) for every reachable state \( s \). A set of facts \( M \subseteq F \) is called a mutex group if \( |M \cap s_i| \leq 1 \) for every reachable state \( s_i \) and \( M \) is called a fact-alternating mutex group (FAM-group) if \( |M \cap s_i| \leq 1 \) and \( |M \cap \text{add}(o) - \text{del}(o)| \) for every operator \( o \). An FAM-group is a mutex group. Every subset of a mutex group is a mutex group, but not every subset of a FAM-group is a FAM-group. Facts from every mutex group form pairwise mutexes [Fišer and Komenda, 2018].

Given a set of mutex groups \( G \), \( M_G \subseteq 2^F \) denotes an upper set of mutexes induced by \( G \), i.e., for every mutex group \( G \in G \) and every \( f, f' \in G \), \( f \neq f' \), it holds that \( \{ f, f' \} \in M_G \), and for every \( M \subseteq M_G \) and every \( f \in F \) it holds that \( \{ f \} \subseteq M \).

An FDR planning task \( \mathcal{P} \) is specified by a tuple \( \mathcal{P} = (V, O, \psi_I, \psi_G) \). A variable \( V \) is a finite set of variables, each variable \( v \in V \) has a finite domain \( \text{dom}(v) \). A fact \( (V, v) \) is a pair of a variable \( v \in V \) and one of its values \( v \in \text{dom}(V) \). A partial state \( p \) is a variable assignment over some variables \( \text{vars}(p) \subseteq V \). We write \( p[V] \) for the value assigned to the variable \( v \in \text{vars}(p) \) in the partial state \( p \). Given a set of variables \( U \subseteq V \), \( p[U] \) denotes a partial state restricted to \( U \). We also identify \( p \) with the set of facts contained in \( p \), i.e., \( p = \{ (V, v)[V] \mid v \in \text{vars}(p) \} \). A partial state \( s \) is a state if \( \text{vars}(s) = \psi_I \) is an initial state, \( \psi_G \) is a partial state called goal, and a state \( s \) is a goal state iff \( \psi_G \subseteq s \). \( O \) is a finite set of operators, each operator \( o \in O \) has a precondition \( \text{pre}(o) \) and effect \( \text{eff}(o) \), which are partial states. An operator \( o \) is applicable in a state \( s \) iff \( \text{pre}(o) \subseteq s \). The resulting state of applying an applicable operator \( o \) in a state \( s \) is the state \( o[s] \) where \( o[s][V] = \text{eff}(o)[V] \) for every \( v \in V \in \text{vars}(\text{eff}(o)) \), and \( o[s][V] = s[V] \) for every \( v \in V \in \text{vars}(\text{pre}(o)) \). Operator sequences and plans are defined analogously to STRIPS.

The domain transition graph (DTG) of a variable \( V \), denoted by \( D_V \), is an edge-labeled multi-digraph with vertices \( \text{dom}(V) \) and with an edge from \( d \) to \( d' \) induced by an operator \( o \in O \) and denoted by \( (d, q, d') \; \text{iff} \; d \neq d' \), \( \text{eff}(q)[V] = d' \), and either \( \text{pre}(q)[V] = d \) or \( V \not\in \text{vars}(\text{pre}(q)) \).

Let \( \mathcal{U} \subseteq \mathcal{V} \) denote the subset of variables. The causal graph \( CG_U \) of \( \mathcal{U} \) is a digraph with vertices \( \mathcal{U} \). An edge \( (V, V') \) is in \( CG_U \) if \( V \neq V' \) and there exists an operator \( o \in O \) such that \( (V, V') \in (\text{vars}(\text{pre}(o)) \cup \text{vars}(\text{eff}(o)) \times \text{vars}(\text{eff}(o))) \).

A red-black planning task \( \Pi \) is specified by a tuple \( \Pi = (V^R, F^B, O, \psi_I, \psi_G) \), where \( V^R \) and \( V^B \) are state variables, called red variables and black variables, respectively. Black variables have the value-switching semantics as in FDR. Red variables have the value-accumulating semantics, i.e., operators “extend” the value of \( V \in V^R \) from \( \{ x \} \) to \( \{ x, y \} \). Applicability of operators, sequences of operators, and red-black plans are defined accordingly [see Domshlak et al., 2015].

Given a variable \( V \in V \), an edge \( (d, q, d') \) from \( V \) is relaxed side effects invertible (RSE-invertible) if there exists an edge \( (d', q', d) \) such that \( \text{pre}(q')[V \setminus V] \subseteq \text{pre}(q)[V \setminus V] \cup \text{eff}(q)[V \setminus V] \). \( V \) is RSE-invertible if every edge in \( V \) is RSE-invertible.

RSE-invertibility is a sufficient criterion for tractability in the case when the subgraph of the causal graph restricted to the black variables is acyclic [Katz et al., 2013a]. The actual algorithm presented in that work handles only the arc-less black causal graph case. Later, an algorithm for devising red-black plans for the acyclic causal graph case was presented [Katz and Hoffmann, 2014; Domshlak et al., 2015]. In what follows, we focus on the red-black heuristic \( h_{RB} \) for a state \( s \), that returns the length of a red-black plan for \( s \), computed for that tractable fragment.

3 Translation from STRIPS to FDR
A concise translation from STRIPS to FDR requires mutex groups that are used for creating FDR variables from STRIPS facts. Before we formalize the translation process, we formulate an intermediate STRIPS representation where we assume the mutex groups are already inferred and the STRIPS planning task is pruned with the mutex information contained in them. Moreover, we need to preserve the DTG structure of mutex groups from which black variables are created to preserve their RSE-invertibility. For this purpose, we introduce a mapping from mutex groups to fam-groups containing these mutex groups, called fam-group map. We choose specifically fam-groups because of their simpler structure [Fišer and Komenda, 2018]. It is not a limitation, since fam-groups are the most commonly used type of mutex groups anyway [Fišer, 2020].

Definition 1. Given a set of mutex groups \( G \), \( \gamma : G \rightarrow 2^F \) is a fam-group map if for every \( G \in G \) it holds that either \( \gamma(G) = \emptyset \) or \( \gamma(G) \) is a fam-group such that \( G \subseteq \gamma(G) \).

Definition 2. Given a STRIPS planning task \( \Pi = (F, O, s_I, s_G) \), and a set of non-empty mutex groups \( G \), and a fam-group map \( \gamma \) over \( G \), \( \Pi = (\Pi, G, \gamma) \) denotes a STRIPS planning task extended with mutex groups (MGE-STRIPS) if all of the following hold:

(S1) \( F = \bigcup_{o \in O} \text{add}(o) \cup \text{del}(o) \).
(S2) for every operator \( o \in \mathcal{O} \) it holds that \( \text{pre}(o) \notin \mathcal{M}_G \) and \( o[\text{pre}(o)] \notin \mathcal{M}_G \), and for every \( f \in \text{del}(o) \) it holds that \( \{f\} \cup \text{pre}(o) \notin \mathcal{M}_G \),

(S3) \( s_I \notin \mathcal{M}_G \) and \( s_G \notin \mathcal{M}_G \),

(S4) \( \bigcup_{G \in \mathcal{G}} G = \mathcal{F} \),

(S5) for every \( G, G' \in \mathcal{G} \), \( G \neq G' \) it holds that \( G \cap G' = \emptyset \),

(S6) \( \{f\} \in \mathcal{G} \) for every \( f \in \bigcup_{G \in \mathcal{G}} (\text{del}(G) \setminus \text{pre}(G)) \),

(S7) \( \gamma(G) \neq \emptyset \) for every \( G \in \mathcal{G} \) such that \( |G| \geq 2 \).

Condition (S1) makes sure that there are no “static” facts, i.e., facts that either do not appear in any reachable state, or that are set in the initial state and remain set in all reachable states. (S2) corresponds to pruning of unreachable operators and delete effects that are not applicable. (S3) can be false only for unsolvable tasks. (S4) ensures that the set of mutex groups cover all facts. Pairwise disjoint mutex groups (S5) ensure that each STRIPS fact is encoded only once. (S6) ensures a polynomial runtime without conditional effects (for a detailed explanation see [Helmert, 2009, Section 7.3]). The fam-group \( \gamma \) is used for encoding preconditions of operators in the translation to FDR, and, as we described before, we assume all mutex groups \( \mathcal{G} \) to be based on fam-groups, i.e., each mutex group from \( \mathcal{G} \) is either a singleton or it is mapped to some fam-group (S7).

Our formulation of the translation from MGE-STRIPS to an FDR planning task differs from the translation described by Helmert [2009] in the way preconditions of operators are encoded. We use the value \( \bot_G \) to express that none of the facts from the mutex group set \( G \) is set. Whenever the fam-group map \( \gamma \) maps \( G \) to a non-empty fam-group and \( \text{pre}(o) \cap (\gamma(G) \setminus G) \neq \emptyset \), we set the variable \( V_G \) to \( \bot_G \) in a precondition of operator \( q_o \), i.e., \( V_G \) is set to \( \bot_G \) only if \( \text{pre}(o) \) is mutex with \( G \) and we can infer this fact from the given fam-group map. Note also that if \( \gamma(G) = \emptyset \) for every \( G \in \mathcal{G} \), then we get the same FDR encoding as Helmert [2009].

**Definition 3.** Given an MGE-STRIPS planning task \( \text{E}_{\Pi} = \langle \Pi = \langle \mathcal{F}, \mathcal{O}, s_I, s_G \rangle, \mathcal{G}, \gamma \rangle \), \( \mathcal{P}(\text{E}_{\Pi}) = \langle \mathcal{V}, \mathcal{O}, \psi_I, \psi_G \rangle \) is an FDR planning task such that:

1. \( \mathcal{V} = \{V_G \mid G \in \mathcal{G}\} \) where \( \text{dom}(V_G) = G \cup \{\bot_G\} \) for every \( V_G \in \mathcal{V} \),
2. \( \mathcal{O} = \{q_o \mid o \in \mathcal{O}\} \) where
   - \( \text{pre}(q_o) = \{(V_G, f) \mid G \in \mathcal{G}, G \cap \text{pre}(o) \neq \emptyset, f \in G \cap \text{pre}(o) \cup \{V_G, \bot_G\} \} \) if \( G \in \mathcal{G}, G \cap \text{pre}(o) = \emptyset, \gamma(G) \cap \text{pre}(o) \neq \emptyset \),
   - \( \text{eff}(q_o) = \{(V_G, f) \mid G \in \mathcal{G}, G \cap \text{add}(o) \neq \emptyset, f \in G \cap \text{add}(o) \cup \{V_G, \bot_G\} \} \) if \( G \in \mathcal{G}, G \cap \text{add}(o) = \emptyset, G \cap \text{del}(o) = \emptyset, G \cap \text{add}(o) = \emptyset \),
3. \( \psi_I = \{(V_G, f) \mid G \in \mathcal{G}, G \cap s_I \neq \emptyset, f \in G \cap s_I \} \cup \{(V_G, \bot_G) \mid G \in \mathcal{G}, G \cap s_I = \emptyset \} \),
4. \( \psi_G = \{(V_G, f) \mid G \in \mathcal{G}, G \cap s_G \neq \emptyset, f \in G \cap s_G \} \),

Although encoding \( \bot_G \) in preconditions of operators is not necessary, because this fact is true implicitly, we use it to ensure that the structure of a projection to the mutex group in STRIPS is preserved in the corresponding DTG in FDR. Why we need this will become clear in the next section where we deal with RSE-invertibility. For now, see the example depicted in Figure 1. The projection to the fam-group \( F = \{f_1, \ldots, f_3\} \) is shown in Figure 1a. Now suppose we decide to encode only a subset of \( F \) as an FDR variable, namely \( G = \{f_1, f_2\} \). If we set \( \gamma(G) = \emptyset \), then the resulting DTG of \( V_G \) will be as depicted in Figure 1c, i.e., it will have additional edges between \( f_1 \) and \( f_2 \) not present in the original projection, because the operators \( o_{31} \) and \( o_{42} \) have \( f_3 \) and \( f_4 \), respectively, in their preconditions. Therefore \( V_G \) is not set in \( \text{pre}(q_{o31}) \) and \( \text{pre}(q_{o42}) \). However, if we set \( \gamma(G) = F \), then we preserve the structure, as depicted in Figure 1b, because \( V_G \) will be set to \( \bot_G \) in the preconditions of \( q_{o31} \) and \( q_{o42} \).

**Proposition 4.** Let \( \text{E}_{\Pi} = \langle \Pi = \langle \mathcal{F}, \mathcal{O}, s_I, s_G \rangle, \mathcal{G}, \gamma \rangle \) denote an MGE-STRIPS planning task. \( \pi = \{o_1, \ldots, o_n\} \) is a plan for \( \text{E}_{\Pi} \) iff \( \pi' = \{\psi_{o1}, \ldots, \psi_{o_n}\} \) is a plan for \( \mathcal{P}(\text{E}_{\Pi}) \).

**Proof Sketch.** \( \mathcal{P}(\text{E}_{\Pi}) \) is well-defined, because (T3) assigns one value to every variable, i.e., \( \psi_I \) is a state. Similarly, (T4) ensures that \( \psi_G \) is a partial state, and (T2a-b) produce partial states, because only one value is assigned to each variable. Note that \( f \in s_I \) iff (\( V_G, f \) \( \in \psi_I \) for every \( G \) s.t. \( G \cap s_I = \{f\} \), and (\( V_G, \bot_G \) \( \in \psi_I \) only if \( s_I \cap G = \emptyset \), similarly, \( f \in s_G \) iff (\( V_G, f \) \( \in \psi_G \); and for every \( o \in \mathcal{O} \) holds that \( f \in \text{pre}(o) \) iff (\( V_G, f \) \( \in \text{pre}(o) \); and (\( V_G, \bot_G \) \( \in \text{pre}(o) \) only if \( \text{pre}(o) \cap G = \emptyset \) and \( \text{pre}(o) \cap \gamma(G) \neq \emptyset \)). Hence \( o \) is applicable in \( s_I \) iff \( q_o \) is applicable in \( \psi_I \) for every \( o \in \mathcal{O} \). Furthermore, from (T2b) it follows that \( f \in o[s_I] \) iff (\( V_G, f \) \( \in q_o[\psi_I] \) for every \( G \) s.t. \( G \cap o[s_I] = \{f\} \), and (\( V_G, \bot_G \) \( \in q_o[\psi_I] \) only if \( G \) is mutex with \( o[s_I] \)). The rest follows by induction.

\( \square \)

4 RSE-Invertible Variables

Now, we show how to identify mutex groups in \( \text{E}_{\Pi} \) whose counterparts form variables in \( \mathcal{P}(\text{E}_{\Pi}) \) that are RSE-invertible. This condition is required for black variables in the tractable fragment of red-black planning we consider. First, we formulate conditions for RSE-invertibility of facts and mutex groups. Then, we show that, given a fam-group map, these conditions are both sufficient and necessary for RSE-invertibility of the resulting FDR variables.

For the rest of this section, let \( \text{E}_{\Pi} = \langle \Pi = \langle \mathcal{F}, \mathcal{O}, s_I, s_G \rangle, \gamma \rangle \) denote an MGE-STRIPS planning task, and let \( \mathcal{P}(\text{E}_{\Pi}) = \langle \mathcal{V}, \mathcal{O}, \psi_I, \psi_G \rangle \) denote an FDR planning task constructed from \( \text{E}_{\Pi} \) according to Definition 3.
Definition 5. A fact \( f \in F \) is RSE-invertible if
(1) there exist \( o, o' \in O \) such that \( f \in \text{add}(o) \) and \( f \in \text{del}(o') \); and
(2) for every operator \( o \in O \) such that \( f \in \text{del}(o) \) it holds that there exists an operator \( o' \in O \) such that \( f \in \text{add}(o') \) and \( \text{pre}(o') \subseteq \text{pre}(o) \cup \text{add}(o) \),
(3) for every operator \( o \in O \) such that \( f \in \text{add}(o) \) it holds that there exists an operator \( o' \in O \) such that \( f \in \text{del}(o') \) and \( \text{pre}(o') \subseteq \text{pre}(o) \cup \text{add}(o) \).

We also call a set of facts (or mutex group) \( F \subseteq F \) RSE-invertible if every \( f \in F \) is RSE-invertible.

For the main result of this section (Theorem 9), we need to prove several auxiliary lemmas first. We start by showing that the condition \( \text{pre}(o) \subseteq \text{pre}(o') \cup \text{add}(o') \) from (12-3) corresponds to the RSE-invertibility condition on edges in DTGs as long as we have transitions in both directions.

Lemma 6. Let \( G \in G \) be a mutex group and \( o, o' \in O \) be two operators with edges \((x, q_o, x')\) and \((x', q_{o'}, x)\) in the DTG of \( V_G \). Then \( \text{pre}(o) \subseteq \text{pre}(o') \cup \text{add}(o') \) iff \( \text{pre}(q_o) \subseteq \text{pre}(q_{o'}) \cup \text{add}(q_{o'}) \) in the DTG of \( V_G \).

Proof. \( \Rightarrow \) \text{: It is easy to see that if } \text{pre}(o) \sqsubseteq \text{pre}(o') \cup \text{add}(o') \text{ iff } \text{pre}(q_o) \subseteq \text{pre}(q_{o'}) \cup \text{add}(q_{o'}) \text{ in the DTG of } V_G. \text{ Therefore } \text{pre}(o) \subseteq \text{pre}(o') \cup \text{add}(o'). \text{ Moreover, since there is also an edge } (x', q_{o'}, f) \text{ in the DTG of } V_G, \text{ it follows that } \text{eff}(q_o)[V_G] = f \text{ and } \text{f}(q_o)[V_G] = f \text{ and } \text{f}(q_{o'})[V_G] = f \text{ or } \text{f}(q_{o'})[V_G] = f. \text{ So, it remains to show that the claim holds also for every } H \in G, H \neq G, \text{ such that } H \cap \text{pre}(o) = \emptyset \text{ and } H \cap \text{add}(o') = \emptyset \text{ and } H \cap \text{pre}(o') = \emptyset \text{ and } H \cap \text{add}(o) = \emptyset. \text{ If } \text{pre}(q_o)[V_H] = H \cap \text{pre}(o) = \emptyset \text{ and } \text{pre}(q_{o'})[V_H] = H \cap \text{pre}(o') = \emptyset \text{ and } \text{pre}(q_{o'})[V_H] = H \cap \text{add}(o') = \emptyset \text{ and } \text{pre}(q_o)[V_H] = H \cap \text{add}(o) = \emptyset. \text{ Then the conditions (11-3) are satisfied.} \text{ \( \Box \)}

A mutex group \( M \in G \) such that \(|M| \geq 3 \) requires the fam-group map \( \gamma \) to map \( M \) to its superset fam-group (S7).

Recall the example in Figure 1. The facts \( f_1 \) and \( f_2 \) are RSE-invertible, but \( f_3 \) and \( d_4 \) are not, because of the operator \( o_{31} \).

So, we would like to construct an RSE-invertible variable from \( G = \{ f_1, f_2 \} \).

Recall the example in Figure 1. The facts \( f_1 \) and \( f_2 \) are RSE-invertible, but \( f_3 \) and \( d_4 \) are not, because of the operator \( o_{31} \).

So, we would like to construct an RSE-invertible variable from \( G = \{ f_1, f_2 \} \).

If we set \( \gamma(G) = (\emptyset, \emptyset) \), then the resulting DTG (Figure 1c) will have edges between \( f_1 \) and \( f_2 \) that are not RSE-invertible, because the operator \( o_{31} \) has \( f_3 \) as its precondition, which is not part of \( \text{pre}(o_{31}) \cup \text{add}(o_{31}) \).

However, \( \gamma(G) = F \) results in the RSE-invertible DTG (Figure 1b).

Lemma 8. Let \( G \in G \) be a mutex group with \(|G| \geq 2 \) and \( \gamma(G) \neq \emptyset \).

Then \( G \) is RSE-invertible if \( V_G \) is RSE-invertible.

Proof. \( \Rightarrow \) : Since \( F \) is a fam-group s.t. \( G \subseteq \gamma(G) \) and \(|\gamma(G)| \geq 2 \) and (S6), for every operator \( o \in O \) and every \( f \in G \) it holds that \( f \in \text{del}(o) \) implies \( f \in \text{pre}(o) \).
Let \( q_o, d_o \) denote an edge in the DTG of \( V_G \).
Then \( \text{eff}(q_o)[V_G] = d_o \) and either \( \text{pre}(q_o)[V_G] = d_o \) or \( V_G \not\subseteq \text{vars}(q_o) \).
Now, if we show that there also exist an edge \((d', q_o, d)\) for some \( d' \in O \) s.t. \( \text{pre}(d') \subseteq \text{pre}(o) \cup \text{add}(d) \), then the rest follows from Lemma 6. So three cases need to be investigated.

(1) If \( d, d' \in G \), then \( d' \in \text{add}(o) \). And since \( \gamma(G) \) is a fam-group, we have \( V_G \subseteq \text{vars}(q_o) \) and \( d \in \text{pre}(o) \cap \text{del}(o) \). Therefore, there exist \( o \in O \) s.t. \( d \in \text{add}(o) \) and \( \text{pre}(o') \subseteq \text{pre}(o) \cup \text{add}(o) \). Thus, \( d' \in \text{pre}(o') \), because \( d \notin \text{pre}(o) \) and \( \gamma(G) \cap (\text{pre}(o) \cup \text{add}(o)) = \{d', d}\). Therefore, there exists an edge \((d', q_o, d)\).
(2) If \( d \in G \) and \( d' = \bot_G \), then \( \text{add}(o) \cap G = \emptyset \) and \( \text{del}(o) \cap G \neq \emptyset \), therefore \( d \in \text{del}(o) \), because otherwise there would exist \( x \in G, x \neq d \), s.t. \( x \in \text{del}(o) \), therefore \( x \in \text{pre}(o) \), therefore pre\((q_0)[V_G] = x \) which is in contradiction. So we have \( d \in \text{del}(o) \) and also \( d \in \text{pre}(o) \). Therefore, there exists \( o' \) s.t. \( d \in \text{add}(o') \) and \( \text{pre}(o') \subseteq \text{pre}(o) \cup \text{add}(o) \). So since \( d \in \text{pre}(o) \) and \( \text{add}(o) \cap G = \emptyset \), then \( \text{pre}(o') \cap G = \emptyset \), therefore we have \( \text{eff}(q_0)[V_G] = d \) and either \( \text{pre}(q_0)[V_G] = \bot_G \) or \( V_G \not\subseteq \text{vars}(\text{pre}(q_0)) \). Therefore, we have an edge \((\bot_G, q_0, d')\).

(3) If \( d' = \bot_G \) and \( d'' \in G \), then \( d'' \in \text{add}(o) \), therefore there exists \( f \in \gamma(G) \setminus G \) s.t. \( f \in \text{add}(o) \) and \( \text{pre}(o) \subseteq \text{pre}(o) \cup \text{add}(o) \). Hence, \( d'' \in \text{pre}(o') \cap \text{del}(o) \) because \( \gamma(G) \cap \text{pre}(o') \cap \text{del}(o') \neq \emptyset \) (\( \gamma(G) \) is a fam-group) and \((\text{pre}(o) \cup \text{add}(o)) \cap \gamma(G) = \emptyset \). Therefore, \( \text{pre}(q_0)[V_G] = d'' \) and \( \text{eff}(q_0)[V_G] = \bot_G \), and thus we have \((d'', q_0', \bot_G)\).

\[ \Rightarrow \]: Let \((x, q_o, x')\) denote an edge in \( D_{V_G} \). If \( x' \in G \), then \( x' \in \text{add}(o) \) (T2b). Since \( G \) is a fam-group, either \( \text{pre}(q_0)[V_G] = \emptyset \text{ or } \text{pre}(q_0)[V_G] = x \) s.t. \( x \in G \). Therefore if \( x \in G \), then \( x \in \text{pre}(o) \cap \text{del}(o) \). Therefore, for every such \( q_o \), it holds that either \( G \cap \text{del}(o) \neq \emptyset \) or \( G \cap \text{add}(o) \neq \emptyset \). Since \( V_G \) is RSE-invertible, for every edge \((x, q_o, x')\) there exist an edge \((x', q_o, x)\) s.t. \( \text{pre}(q_0)[V_G \setminus V_G] \subseteq \text{pre}(q_0)[V_G \setminus V_G] \cup \text{eff}(q_0)[V_G] \), therefore it follows from Lemma 6 that \( \text{pre}(o) \subseteq \text{pre}(o) \cup \text{add}(o) \). Hence, the conditions (I-1-3) are satisfied.

**Theorem 9.** Let \( G \in \mathcal{G} \) be a mutex group with \(|G| = 1 \) or \( \gamma(G) \neq \emptyset \). Then \( G \) is RSE-invertible iff \( V_G \) is RSE-invertible.

**Proof.** It follows directly from Lemma 7 and Lemma 8.

5 Causal Graph

Besides RSE-invertibility of black variables, the tractable fragment of red-black planning requires the black causal graph to be acyclic. To achieve that, we show how to find a causal link between mutex groups that is translated into FDR.

**Definition 10.** Given a mutex group \( G \in 2^F \), two distinct facts \( f, f' \in G \) and \( f' \notin G \), and a fam-group map \( \gamma \) on \( 2^F \), we say that there is a **causal link from** \((G, f)\) to \((G', f')\) if there exists an operator \( o \in O \) such that (i) \( f' \in \text{del}(o) \cup \text{add}(o) \), and (ii) \( f \in \text{pre}(o) \cup \text{del}(o) \cup \text{add}(o) \), or \( \gamma(G) \cap \text{pre}(o) \neq \emptyset \).

**Theorem 11.** Let \( \mathcal{E}_I = (\Pi, G, \gamma) \) be an MGE-STRIPS task with operators \( O, A, B \in \mathcal{G} \) be two distinct mutex groups, and \( CG_V \) be the causal graph of \( \mathcal{E}_I \). There is an edge \((V_A, V_B)\) in \( CG_V \) iff there exist \( f \in A \) and \( f' \in B \) such that there is a causal link from \((A, f)\) to \((B, f')\).

**Proof.** \( \Leftarrow \): Let \( o \in O \) be the operator inducing the causal link from \((A, f)\) to \((B, f')\). If \( f \in \text{pre}(o) \), then \( \text{pre}(q_0)[V_A] = f \). If \( f \in \text{add}(o) \), then \( \text{eff}(q_0)[V_A] = f \). If \( f \in \text{del}(o) \), then either \( \text{eff}(q_0)[V_A] = \bot_A \) or there exist \( x \in A \) s.t. \( \text{eff}(q_0)[V_A] = x \). If \( \gamma(A) \neq \emptyset \) and \( \gamma(A) \cap \text{pre}(o) \neq \emptyset \), then either \( \text{pre}(q_0)[V_A] = x \) for some \( x \in A \), or \( \text{pre}(q_0)[V_A] = \bot_A \). Therefore, \( V_A \in \text{vars}(\text{pre}(q_0)) \cup \text{vars}(\text{eff}(q_0)) \), and for similar reasons \( V_B \in \text{vars}(\text{eff}(q_0)) \).
Algorithm 1: Inference of RSE-invertible mutex groups forming acyclic causal graph.

Input: A STRIPS planning task $\Pi = (F, O, s_I, s_G)$, a set of fam-groups $H \subseteq 2^F$, a set of RSE-invertible facts $N \subseteq F$.

Output: A set of RSE-invertible mutex groups $K$, a fam-group map $\gamma$ over $K$.

1. $H' \leftarrow \{\{f\} \mid f \in N \setminus \bigcup_{G \in H} G\}$;
2. Construct a fam-group map $\gamma'$; $\gamma'(G) = 0$ for every $G \in H'$, and $\gamma'(G \cap N) = G$ for every $G \in H$ s.t. $G \cap N \neq \emptyset$;
3. $H^* \leftarrow H' \cup \{G \cap N \mid G \in H, G \cap N \neq \emptyset\}$;
4. Construct the digraph $\mathcal{X}(\Pi, H^*, \gamma') = (V, E)$;
5. Find vertices $V' \subseteq V$ such that for every cycle $c$ in the subgraph of $\mathcal{X}(\Pi, H^*, \gamma')$ induced by $V'$ it holds that $|\mathcal{C}(c)| = 1$;
6. $Z \leftarrow \{(G, \langle f \mid (G, f) \in V'\rangle) \mid (G, f') \in V'\}$;
7. Construct a fam-group map $\gamma$. For every $(G, X) \in Z$, set $\gamma(X) = 0$ if $|X| = 1$ and $\gamma(X) = G$ otherwise;
8. $K \leftarrow \{X \mid (G, X) \in Z\}$;

map its RSE-invertible subset $G \cap N$ to $G$. In step 3, $H^*$ is constructed as a set of RSE-invertible mutex groups covering all RSE-invertible facts. In step 4, the graph defined in Definition 12 is constructed, and in step 5, we find a subset of vertices having cycles only within each mutex group. Finally, in the last three steps we extract the solution.

Note that the step 5 is a variant of an NP-complete feedback vertex set problem [Karp, 1972]. We decided to use a naive approach and solve this problem by an integer linear program (ILP), where each vertex from $\mathcal{X}(\Pi, H^*, \gamma')$ corresponds to a binary variable, and each constraint corresponds to a cycle that has to be avoided. Instead of listing all cycles, we first add all cycles between pairs of vertices from different mutex groups, then we solve the problem, and check whether the resulting solution has a cycle. If we find a cycle, we add it as another constraint, solve the problem, and continue in this manner until we find a feasible solution.

As objective function of the ILP, we tried two variants. The first variant is simply a maximization of the number of facts in the resulting mutex groups. The second variant weights facts using the number of conflicts in a relaxed plan $\pi^+$ obtained from the FF heuristic [Hoffmann and Nebel, 2001] computed on the STRIPS representation. Following prior work [Domshlak et al., 2015], we say a fact $f$ has a conflict if there is an operator $o$ in $\pi^+$ such that $f \in pre(o)$ and $f$ is not satisfied when executing $\pi^+$ with the non-relaxed semantics. For each mutex group $M$, we sum the number of conflicts from all $f \in M$, and we set the weight of each fact $f \in M$ as the sum over $|M|$. This way we prioritize mutex groups containing facts with most conflicts in the relaxed plan.

Note that Algorithm 1 describes only the inference of RSE-invertible variables. The rest of the FDR variables, i.e., the red variables, are constructed from the remaining facts not covered by the resulting black variables by greedy approach described by Helmert [2009], i.e., we greedily maximize the size of red variables.

Figure 2: Per-task comparison of the translation time (in seconds).

7 Experimental Evaluation

We implemented Algorithm 1 in C, used fam-groups inferred from PDDL [Fišer, 2020]², and used the red-black heuristic $h_{RB}^{RB}$ [Domshlak et al., 2015] implemented in FD [Helmert, 2006]. The experiments were performed on Intel(R) Xeon(R) Scalable Gold 6146 machines with CPLEX solver v12.6, and time and memory limits of 30min and 8GiB. Out of all benchmarks from International Planning Competitions of 1998 to 2018 satisfying planning tracks, we selected all tasks where no conditional effects were created and at least one variable could be painted black.

We compare the following configurations: the baseline configuration B runs the painting strategy denoted as $A^3$ by Domshlak et al. [2015] on the default FDR encoding [Helmert, 2009]; $M$ maximizes the number of black facts using Algorithm 1; $C$ is the variant of Algorithm 1 using conflicts in a relaxed plan; and the “oracle” 0 picks the best FDR encoding and painting for each tested task. It considers B, and five best encodings (by objective value) from each of $M$ and $C$, which we obtain by repeatedly solving the corresponding ILP, each time disallowing previous solutions by additional constraints. The encodings are chosen in terms of coverage, preferring tasks solved in the initial state by the oracle 0 serves to illustrate the potential of our customized encodings, assuming perfect knowledge about which encoding works best for a given task.

The most time-demanding part of Algorithm 1 is solving the ILP (step 5). The average and median time spent in this step was 3.3 seconds and 12 milliseconds, respectively, for both $C$ and $M$. The maximum was more than six minutes, but it took more than a minute only for 15 tasks in transport, 7 tasks in visits1, and 1 task in the satellite domain. Figure 2 shows a comparison of running times (in seconds) of the whole translation process. Although Algorithm 1 often requires more time in comparison to the greedy approach of the baseline, it, for most cases, leaves enough time for the planner, as we describe below.

The scatter plots in Figure 3 show that our methods are able to increase the number of black facts by several orders of magnitude. The oracle often uses significantly less black facts than B, indicating that it is not necessarily beneficial to have more black facts. In fact, having more black variables (facts) can negatively impact heuristic computation time, without improving informativeness.

²https://github.com/danifis/cpddl, branch icai21-fdr-red-black

³The painting strategy greedily paints variables red until the black causal graph is acyclic, preferring keeping variables with larger number of incident edges to black variables.
Table 1 shows the number of tasks solved (coverage) in the left, and the number of tasks solved by $h_{RB}$ in the initial state in the center. B clearly outperforms M, which, again, shows that maximizing the number of black facts is not necessarily the best painting strategy. C and B are very similar in coverage, with strengths in different domains. C can solve six more tasks in the initial state resulting from only two domains, blocks and satellite.

The oracle variant O demonstrates the potential of our approach if we better understand how to select the best encoding for a given task. B already solves a large portion of tasks, but here we show that it is still possible to create an encoding and painting that could further increase this number by at least 25 tasks, and 14 more can be solved in the initial state. Unfortunately, it is unclear how to directly compare the quality of individual encodings for a given task without running the search. Maximizing the number of black facts turned out to be too simplistic as an optimization criterion. Considering conflicts in a relaxed plan improves the results in some domains, but it is still unclear what properties identify a good FDR encoding and painting for red-black planning.

The rightmost column in Table 1 shows the number of tasks with a single maximal set of black variables. That is, it shows the number of tasks where different paintings can only be achieved by painting black a subset of the facts that were already painted black by C and M. In other words, both C and M created exactly the same set of black variables, but B could paint only less variables black. This is exactly what happened in the depot domain where painting less variables black proved to generate a more suitable FDR for the red-black heuristic.

Table 1: Numbers of tasks solved (left), number of tasks solved in the initial state (center), and number of tasks with a single maximal set of black variables (right). B, M, C, and O are described in the text. Domains with identical results are summarized in “others”.

### Acknowledgements

Daniel Gnad was supported by the German Research Foundation (DFG), under grant HO 2169/6-2, “Star-Topology Decoupled State Space Search”. Jörg Hoffmann’s research group has received support by DFG grant 389792660 as part of TRR 248 (see perspicuous-computing.science).

### References


