Mathematical Programming Models for Optimizing Partial-Order Plan Flexibility

Buser Say[1] and **Andre A. Cire**[2] and **J. Christopher Beck**[1]

Abstract. A partial-order plan (POP) compactly encodes a set of sequential plans that can be dynamically chosen by an agent at execution time. One natural measure of the quality of a POP is its *flexibility*, which is defined to be the total number of sequential plans it embodies (i.e., its *linearizations*). As this criteria is hard to optimize, existing work has instead optimized proxy functions that are correlated with the number of linearizations. In this paper, we develop and strengthen mixed-integer linear programming (MILP) models for three proxy functions: two from the POP literature and a third novel function based on the *temporal flexibility* criteria from the scheduling literature. We show theoretically and empirically that none of the three proxy measures dominate the others in terms of number of sequential plans. Compared to the state-of-the-art MaxSAT model for the problem, we empirically demonstrate that two of our MILP models result in equivalent or slightly better solution quality with savings of approximately one order of magnitude in computation time.

1 Introduction

A *partial-order plan* (POP) is a plan that imposes only action orderings necessary for achieving a goal, as opposed to a total ordering of actions as enforced in sequential planning. Equivalently, a POP represents a *set* of sequential plans – or *linearizations* – all including the same actions but under different orderings. POPs provide flexibility to agents, who can dynamically commit to the sequence of actions during the real-time execution of the plan [18].

A key question in the area, however, concerns the criterion which best reflects the quality of a POP. Several different objectives have been proposed in the literature, such as the *makespan* of the plan (i.e., longest path from the initial state to the goal), total number of unordered actions, existence of possible action reorderings, among others [24, 20, 1]. This work extends recent research [18, 17] that combines two distinct criteria: a *cost* per action, and the *flexibility* of a plan, here measured as the total number of linearizations of the POP. This objective naturally incorporates the least-commitment principle of first executing as few (costly) actions as possible, and then improving the robustness of the system by placing as many sequential plans as possible at the disposal of the agent.

While total action cost has been traditionally tackled in sequential planning, enhancing the flexibility of a plan poses a much more challenging problem. Specifically, optimizing the number of linearizations of a POP is equivalent to maximizing the number of Hamiltonian paths in a directed acyclic graph, which is computationally impractical in general [16, 3]. To address this issue, Muise et al. [18, 17] optimize the number of ordering constraints in a POP, a metric that is correlated to the number of linearizations. Such a metric function is more computationally tractable and can be efficiently handled, e.g., by MaxSAT solvers.

Building on the work of Muise et al. and previous literature in planning and scheduling [7, 8], we address the problem of converting a valid sequential plan into a valid POP with minimum action cost and maximum number of linearizations. In particular, we consider the notion of *temporal flexibility* from the scheduling literature as a novel proxy function for the number of linearizations of a POP. We show that there is no dominance relation between our proxy function and two previous proxy functions in the literature: depending on the problem instance, the optimization of any of the proxy functions may lead to a greater number of linearizations than either of the others.

Nonetheless, a central benefit of the temporal flexibility criteria is the scaling of model size that is quadratic in the number of actions, rather than cubic as in Muise's model. Further, the linear relationships inherent in the temporal flexibility are amenable to mathematical programming techniques. We exploit this advantage and propose three mixed-integer linear programming (MILP) models for minimizing action cost and maximizing flexibility: a novel model of temporal flexibility, a model that linearizes an existing MaxSAT formulation [18], and a model that adapts an existing MILP formulation for temporal planning to POPs [8]. Going further, we derive a number of valid linear inequalities that can also be applied to the MILP models, substantially decreasing their solution times.

We compare our three MILP formulations to the current state-of-the-art MaxSAT model by Muise et al. [18]. Our empirical evaluation suggests that optimizing any of the three proxy functions results in equivalent solution quality, consistent with our theoretical results. Furthermore, the strengthened MILP models achieve approximately one order of magnitude speedup compared to the state-of-the-art MaxSAT model, solving significantly more problem instances to optimality. These results hold both when minimizing total action cost and maximizing flexibility and when only maximizing flexibility with a fixed set of actions.

Contributions. We present temporal flexibility as a novel proxy function for maximizing the number of linearizations in a partial-order plan and provide a novel mixed-integer linear program to optimize this criterion. We derive new valid linear inequalities that can be applied to the new and existing POP MILP formulations. Finally, we show that the modified MILP models achieve substantially better run-time performance than the current state-of-the-art without sacrificing solution quality.

[1] Department of Mechanical & Industrial Engineering, University of Toronto, Canada, email: {bsay,jcb}@mie.utoronto.ca
[2] Department of Management, University of Toronto Scarborough, Canada, email: acire@utsc.utoronto.ca

2 Preliminaries

A STRIPS planning problem is a tuple $\Pi = \langle F, A, I, G \rangle$ where F is a set of *fluent symbols*, A is a set of *actions*, $I \subseteq F$ is the *initial state*, and $G \subseteq F$ is the *goal state* [10, 9]. Each action $a \in A$ is associated with three sets of fluents, PRE_a, ADD_a, and DEL_a, representing the preconditions, add effects, and delete effects of a, respectively. An action a is executable in state $s \subseteq F$ if and only if $PRE_a \subseteq s$. The execution of an action a results in the state $s' = (s \cup ADD_a) \setminus DEL_a$. We denote by pre_f, add_f, and del_f the set of actions that have fluent f as precondition, that add f, and that delete f, respectively.

A *sequential plan* π is a solution to Π and corresponds to a sequence of executable actions $(a_1, a_2, \ldots, a_m)$ such that, when starting from the initial state I, executing each action in sequence results in a state s^* containing all fluents required in the goal G, i.e. $G \subseteq s^*$ [10, 19].

A *partial-order plan* (POP) relaxes the sequential nature of plans by only imposing orderings that are sufficient to satisfy the goal conditions [23]. Formally, a POP is a tuple $P = \langle \mathcal{A}, \mathcal{O} \rangle$ where $\mathcal{A} \subseteq A$ is the set of actions of the plan and $\mathcal{O} \subseteq \mathcal{A} \times \mathcal{A}$ is a set of *ordering constraints* [18]. We use the notation $a_1 \prec a_2 \in \mathcal{O}$ to indicate an ordering constraint between a_1 and a_2 in $\mathcal{O}$. A *linearization* of a POP P is a total ordering of $\mathcal{A}$ that is consistent with $\mathcal{O}$, i.e. a sequence $(a_1, a_2, \ldots, a_{|\mathcal{A}|})$ such that $a_i \prec a_j \in \mathcal{O}$ implies $i < j$. A POP P is *valid* for a planning problem Π if and only if all linearizations of P are valid sequential plans for Π [15].

Since enumerating linearizations is computationally expensive (see Section 1 and Muise [17]), the validity of a POP $P = \langle \mathcal{A}, \mathcal{O} \rangle$ is commonly established through *causal links* and *threats* [13]. Given actions $a_1, a_2 \in \mathcal{A}$ and a fluent $f \in F$, $\kappa(a_1, a_2, f)$ is a causal link for P if action a_1 adds the fluent f required by action a_2 in all linearizations of P [14]. Moreover, K is a set of causal links for P if, for every $\kappa(a_1, a_2, f) \in K$, we have $a_1 \in add_f$, $a_2 \in pre_f$, and there does not exist $\kappa(a_i, a_2, f) \in K$ for any $a_i \in add_f$ [14]. Note that $\kappa(a_1, a_2, f) \in K$ implies $a_1 \prec a_2 \in \mathcal{O}$. A causal link $\kappa(a_1, a_2, f) \in K$ is *threatened* by an action $a_3 \in \mathcal{A}$ if $a_3 \in del_f$, $a_3 \prec a_1 \notin \mathcal{O}$ and $a_2 \prec a_3 \notin \mathcal{O}$ [19].

We represent the initial and the goal states by actions a_I and a_G, where $a_I \in add_f, \forall f \in I$ and $a_G \in pre_f, \forall f \in G$, respectively. As shown previously [15, 14], a POP $P = \langle \mathcal{A}, \mathcal{O} \rangle$ is valid for a STRIPS planning problem Π if there exists a set of causal links, K, such that: (1) All the preconditions of each action $a \in \mathcal{A}$ are satisfied by exactly one causal link, $\kappa \in K$; (2) No causal link $\kappa \in K$ is threatened by any action $a \in \mathcal{A}$; and (3) The actions $a_I, a_G \in \mathcal{A}$ are respectively ordered before and after all the other actions $a \in \mathcal{A}$, $a \neq a_I, a \neq a_G$.

3 Least Commitment Flexible POPs

Given a sequential plan π to a STRIPS problem Π, our aim is to derive a POP P from π that is ideally optimal both in terms of its action cost and its flexibility. These two characteristics are embodied in the concept of *least commitment criteria* [15, 18].

The least commitment criteria evaluates POPs according to two metrics: the *total action cost* and the *number of linearizations* of the plan. The first is a natural and common objective in planning as agents would like to incur the minimum total action cost possible to achieve a goal. The second metric intuitively gives us a notion of how flexible the plan is, since more linearizations indicate more alternative ways to execute the actions to achieve the goal [18].

Equipped with these two notions, we formally define the structure of an optimal POP in our context.

Definition 1. *(Least Commitment Flexible POP (LCFP) of a Plan). Let $P = \langle \mathcal{A}, \mathcal{O} \rangle$ and $P' = \langle \mathcal{A}', \mathcal{O}' \rangle$ be two valid POPs for a planning problem Π where $\mathcal{A}' \subseteq \mathcal{A}$. Moreover, let c_a be a non-negative cost associated with each action $a \in \mathcal{A}$. P' is a least commitment flexible POP (LCFP) of plan P iff*

(i) For all valid POPs with action set $\mathcal{A}'' \subseteq \mathcal{A}$, we have $\sum_{a \in \mathcal{A}''} c_a \geq \sum_{a \in \mathcal{A}'} c_a$; and
(ii) For all valid POPs P'' with action set $\mathcal{A}'' \subseteq \mathcal{A}$ and $\sum_{a \in \mathcal{A}''} c_a = \sum_{a \in \mathcal{A}'} c_a$, the number of linearizations of P' is greater or equal to the number of linearizations of P''.

This definition differs from that of Muise et al. [18] as it contains a more general action cost structure and explicitly incorporates the number of linearizations, as opposed to the number of ordering constraints.

As in previous work, Definition 1 does not include solving the traditional POP planning problem of finding the set of actions and the ordering constraints that achieve a valid POP. Rather, we are concerned with the optimization of action cost and flexibility, given a valid POP. Thus, our primary goal in this work is to find the LCFP of a sequential plan, which is itself a POP. Since counting linearizations is computationally challenging, we investigate the optimization of alternative proxy functions that correlate with the number of linearizations of a POP.

4 Proxy Measures of POP Flexibility

We study three proxy functions for the number of linearizations of a POP, two of which are extracted from previous works, and one novel to this work: the minimization of the number of *open ordering* constraints [8], the minimization of the number of *closed ordering* constraints [18], and the maximization of temporal flexibility adapted from the scheduling literature [7], respectively.

4.1 Order Flexibility

Muise et al. [18] optimized what we term the *order flexibility* of a POP: the total number of ordering constraints in the plan. The more ordering constraints, the less order flexibility a POP has. We investigate two definitions of an ordering constraint.

Definition 2. *(Open Ordering Constraint): Given the set of actions $\mathcal{A}$ and the set of causal links K, the open ordering constraint $a_1 \prec a_2$ belongs to the set $\mathcal{O}$ for some POP $P = \langle \mathcal{A}, \mathcal{O} \rangle$ to a planning problem Π if:*

1. *There exists a causal link $\kappa(a_1, a_2, f) \in K$ from action a_1 to action a_2 on some fluent $f \in F$, or*
2. *There exists a causal link $\kappa(a_2, a_3, f) \in K$ from action a_2 to action a_3 on some fluent $f \in F$ and action $a_1 \in del_f$ is ordered before action a_2 to resolve the threat, or*
3. *There exists a causal link $\kappa(a_3, a_1, f) \in K$ from action a_3 to action a_1 on some fluent $f \in F$ and action $a_2 \in del_f$ is ordered after action a_1 to resolve the threat.*

Do and Kambhampati [8] minimize the equivalent of the number of open ordering constraints in *order constrained plans*, temporal plans which have a partial-order structure but allow concurrent execution of non-interfering actions. Concurrent execution semantics create a subtle but relevant difference in the meaning of open ordering constraints. In particular, Definition 2 relaxes Do and Kambhampati's definition by assuming a sequential execution of a POP.

We can now characterize the ordering constraint definition used by Muise et al. [18] as specifically the *transitive closure* of the open ordering constraints.

Definition 3. *(Closed Ordering Constraint): Given the set of actions $\mathcal{A}$ and the set of causal links K, the closed ordering constraint $a_1 \prec a_2$ belongs to the set $\mathcal{O}$ for some POP $P = \langle \mathcal{A}, \mathcal{O} \rangle$ to a planning problem Π if:*

1. *There exists an open ordering constraint between actions a_1 and a_2, or*
2. *There exists some other action a_3 such that: $a_1 \prec a_3 \in \mathcal{O}$ and $a_3 \prec a_2 \in \mathcal{O}$.*

We will refer to the set of open and closed ordering constraints as $\mathcal{O}_O$ and $\mathcal{O}_C$, respectively.

4.2 Temporal Flexibility

In scheduling, temporal flexibility refers to a schedule's ability to absorb temporal variation during execution [21]. We exploit the same property to define an analogous version of temporal flexibility for planning problems.

Given a POP $P = \langle \mathcal{A}, \mathcal{O} \rangle$ for Π and a duration d_a of each action $a \in A$, the *horizon* of P, H_P, is the sum of the action durations. Intuitively, since the actions will be executed in sequence, H_P is the (temporal) length of the plan. The *earliest start time* of an action a is $est_a = \max_{a' \prec a \in \mathcal{O}} (est_{a'} + d_{a'})$ and the *latest finish time of action* a is $lft_a = \min_{a \prec a' \in \mathcal{O}} (lft_{a'} - d_{a'})$, where $est_{a_I} = 0$ and $lft_{a_G} = H_P$ for the unique first and last actions. Let the *action slack* of a, T_a, be such that $T_a = lft_a - est_a - d_a$ [7]. The *temporal flexibility*, T, of P is given by $T = \sum_{a \in \mathcal{A}} T_a$. For classical (i.e. non-temporal) planning, we assume that all actions have a duration of one time unit.

4.3 Dominance Relations Among Proxy Functions

A proxy function obj_1 *dominates* another proxy function obj_2 if, for any planning problem instance Π, any POP P that optimizes obj_1 has at least as many linearizations as any POP P' that optimizes obj_2 and, for at least one instance Π^*, the number of linearizations in P is strictly greater than that of P'. We have the following result.

Proposition 4. *There are no dominance relations among the open ordering, closed ordering, and temporal flexibility proxy functions.*

Proof. For each pair of objective functions (obj_i, obj_j), $i \neq j$, it suffices to show the existence of two problem instances Π_1, Π_2, where, for Π_1, a POP P that optimizes obj_i has more linearizations than a POP P' that optimizes obj_j and vice versa for Π_2. The counter-examples are simple but tedious to verify and are presented in the Appendix of the paper. □

5 The $\mathcal{O}_C^{\text{MILP}}$ Model

Muise [17] empirically investigated equivalent MILP and MaxSAT models for the problem of minimum reordering: finding a POP with the minimum number of closed ordering constraints without removing actions from the initial set of actions. $\mathcal{O}_C^{\text{MILP}}$ is a minor extension of Muise's MILP encoding for the LCFP problem under the closed constraint proxy function. $\mathcal{O}_C^{\text{MILP}}$ does not use the input ordering of the initial sequential plan.

$$\min \quad (|A_N|^2 + 1) \sum_{a \in A_N} c_a Z_a + \sum_{a_i, a_j \in A_N} O_{a_i,a_j}$$

$$\text{s.t.} \sum_{a_i \in add_f} X^f_{a_i,a_j} = Z_{a_j} \qquad (1)$$

$$\forall a_j \in A, f \in PRE_{a_j}$$

$$(1 - X^f_{a_i,a_j}) + (O_{a_d,a_i} + O_{a_j,a_d}) \geq Z_{a_d} \qquad (2)$$

$$\forall a_i, a_d, a_j \in A, f \in PRE_{a_j} \cap DEL_{a_d} \cap ADD_{a_i}$$

$$O_{a_i,a_j} \geq X^f_{a_i,a_j} \qquad (3)$$

$$\forall a_i, a_j \in A, f \in ADD_{a_i} \cap PRE_{a_j}$$

$$O_{a_I,a} = Z_a \quad \forall a \in A \backslash \{a_I\} \qquad (4)$$

$$O_{a,a_G} = Z_a \quad \forall a \in A \backslash \{a_G\} \qquad (5)$$

$$O_{a_i,a_j} + O_{a_j,a_i} \leq \frac{Z_{a_i} + Z_{a_j}}{2} \quad \forall a_i, a_j \in A \qquad (6)$$

$$Z_{a_I} = Z_{a_G} = 1, \qquad (7)$$

$$O_{a,a} = 0 \quad \forall a \in A \qquad (8)$$

$$(1 - O_{a_i,a_j}) + (1 - O_{a_j,a_k}) + O_{a_i,a_k} \geq 1 \qquad (9)$$

$$\forall a_i, a_j, a_k \in A$$

Figure 1: The $\mathcal{O}_C^{\text{MILP}}$ model.

The parameters used in $\mathcal{O}_C^{\text{MILP}}$ are as follows.

- $A_N = A \backslash \{a_I, a_G\}$ is the set of non-dummy actions.
- $I_f = 1$ if dummy action a_I adds fluent f or equivalently if f is true in the initial state.
- $G_f = 1$ if dummy action a_G requires fluent f or equivalently if f is required to be true in the goal state.

The decision variables in the $\mathcal{O}_C^{\text{MILP}}$ model are as follows.

- $Z_a = 1$ iff action a is selected to be part of the POP.
- $X^f_{a_i,a_j} = 1$ iff action a_i supports fluent f for action a_j with the causal link $\kappa(a_i, a_j, f)$.
- $O_{a_i,a_j} = 1$ iff action a_i is ordered before action a_j.

The $\mathcal{O}_C^{\text{MILP}}$ MILP model is presented in Figure 1. The objective first minimizes the sum of action costs, then minimizes the number of closed ordering constraints. The first coefficient of the objective function, $|A_N|^2 + 1$, guarantees this property when all the actions have costs greater or equal to 1 unit. When this is not the case, we need to normalize the action costs so that all the actions have costs greater or equal to 1 unit. Constraint (1) ensures that if an action is selected, all of its preconditions are met exactly once. Constraint (2) orders the threatening actions either before or after the causal link. Constraint (3) is an order implication constraint that states that if action $a_1 \in A$ supports some other action $a_2 \in A$ with some fluent $f \in F$, a_1 must be ordered before a_2. Constraints (4)-(5) restrict all the included actions to be between the initial and the goal actions. Constraint (6) makes sure that the actions in an enforced ordering constraint are included in the plan. Constraint (7) includes the initial and goal actions. Constraint (8) disallows self-loops for ordering constraints. Constraint (9) produces a transitively-closed POP. Constraints (8)-(9) together forbid cycles in the action precedence graph.

Note that due to the ternary arity of Constraint (9), $\mathcal{O}_C^{\text{MILP}}$ grows cubically with the number of actions.

6 The $\mathcal{O}_O^{\text{MILP}}$ and $\mathcal{T}^{\text{MILP}}$ Models

Both $\mathcal{O}_O^{\text{MILP}}$ and $\mathcal{T}^{\text{MILP}}$ build on Do and Kambhampati's MILP model [8]. The main differences are that $\mathcal{O}_O^{\text{MILP}}$ does not enforce an ordering between all pairs of interfering actions, while also allowing for actions to be excluded when they are not relevant to the POP's validity.

$\mathcal{O}_O^{\text{MILP}}$ and $\mathcal{T}^{\text{MILP}}$ are identical aside from their objectives. The key difference with $\mathcal{O}_C^{\text{MILP}}$ is the use of start time variables to represent implied ordering constraints. With the addition of a linear number of variables, the models grow quadratically with the number of actions.

Compared to the formulation of $\mathcal{O}_C^{\text{MILP}}$, $\mathcal{O}_O^{\text{MILP}}$ and $\mathcal{T}^{\text{MILP}}$ introduce three additional variables: $Est_a = \max_{a' \prec a \in \mathcal{O}} (Est_{a'} + d_{a'})$ is the earliest start time of action a; $Lft_a = \min_{a \prec a' \in \mathcal{O}} (Lft_{a'} - d_{a'})$ is the latest finish time of a; and finally $T_a = Lft_a - Est_a - d_a$ is the slack of a.

$$\min \quad (|A_N| \sum_{a \in A_N} d_a + 1) \sum_{a \in A_N} c_a Z_a - \sum_{a \in A_N} T_a$$

s.t. Constraints (1)-(8)

$$Est_{a_i} + d_{a_i} O_{a_i,a_j} \leq Est_{a_j} + \sum_{a \in A} d_a (1 - O_{a_i,a_j}) \quad \forall a_i, a_j \in A \tag{10}$$

$$Lft_{a_i} + d_{a_j} O_{a_i,a_j} \leq Lft_{a_j} + \sum_{a \in A} d_a (1 - O_{a_i,a_j}) \quad \forall a_i, a_j \in A \tag{11}$$

$$Est_a + d_a Z_a + T_a = Lft_a \quad \forall a \in A \tag{12}$$

$$Est_{a_I} = 0 \tag{13}$$

$$Lft_{a_G} = \sum_{a \in A} d_a Z_a \tag{14}$$

Figure 2: The $\mathcal{T}^{\text{MILP}}$ model. The $\mathcal{O}_O^{\text{MILP}}$ model differs from $\mathcal{T}^{\text{MILP}}$ only in using the objective function in $\mathcal{O}_C^{\text{MILP}}$.

As $\mathcal{O}_O^{\text{MILP}}$ uses the same objective function as $\mathcal{O}_C^{\text{MILP}}$ and the only difference between $\mathcal{T}^{\text{MILP}}$ and $\mathcal{O}_O^{\text{MILP}}$ are the objectives, we present the model for $\mathcal{T}^{\text{MILP}}$ in Figure 2. Analogously to $\mathcal{O}_C^{\text{MILP}}$, in $\mathcal{T}^{\text{MILP}}$ the objective first minimizes the sum of action cost, then maximizes the sum of temporal slack of all the actions. Constraints (10)-(11) make sure that, if some action a_i is ordered before some other action a_j, the earliest start time and the latest finish time of a_j are not before the earliest start time and the latest finish time of a_i, respectively. Constraint (12) defines the temporal slack of an action. Constraints (13)-(14) set the plan horizon.

7 Valid Inequalities

We now present valid linear inequalities to strengthen the MILP models. Unless noted, the constraints can be added to all formulations.

Mutual Threat Constraints As illustrated in Figure 3, a cycle is formed when action a_i supports fluent f for action a_j and fluent g for action a_j, where a_j and a_k threaten the causal links $X^g_{a_i,a_k}$ and $X^f_{a_i,a_j}$, respectively. Since a cycle is not allowed in the action precedence graph, $X^g_{a_i,a_k}$ and $X^f_{a_i,a_j}$ are mutually exclusive in any POP and are removed via Constraint (15). When $f = g$, all $X^f_{a_i,a_j}$ are mutually exclusive and are also removed by Constraint (15).

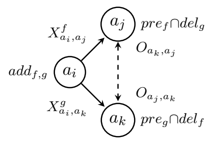

Figure 3: Mutual threat constraint example.

$$\max \left\{ X^g_{a_i,a_k} + X^f_{a_i,a_j}, \sum_{a_j \in (pre_f \cap del_f)} X^f_{a_i,a_j} \right\} \leq Z_{a_i} \tag{15}$$
$$\forall f \in ADD_{a_i}, a_i \in A$$

Action Relevance Constraint Constraint (16) states that a selected action must support at least one causal link.

$$\sum_{a_j \in pre_f, f \in ADD_{a_i}} X^f_{a_i,a_j} \geq Z_{a_i} \quad \forall a_i \in A_N. \tag{16}$$

Minimal Interference Constraint When an action $a_1 \in A$ adds only one fluent $f \in F$, there must exist another action $a_2 \in pre_f$ and the causal link $\kappa(a_1, a_2, f) \in K$. Further, if there exists a third action $a_3 \in del_f$, action a_1 must be ordered with respect to it, due to Constraint (2). Constraint (17)[3] enforces ordering constraints on all the pairs of actions, $a_1, a_2 \in A$, if, given $|ADD_{a_2}| = 1$, either $a_1 \in del_f$ and $a_2 \in (add_f \cup pre_f)$ or $a_2 \in del_f$ and $a_1 \in (add_f \cup pre_f)$.

$$O_{a_i,a_j} + O_{a_j,a_i} \geq Z_{a_i} + Z_{a_j} - 1 \tag{17}$$
$$\forall a_i, a_j \in A, \exists f \in (PRE_{a_i} \cup ADD_{a_i}) \cap DEL_{a_j},$$
$$\text{and } |ADD_{a_i}| = 1.$$

Counting Constraints Inspired by operator counting constraints [22], we observe the following: since all actions a_i that delete and require fluent f (i.e., $f \in F, a_i, a_j \in del_f \cap pre_f$) are sequentially ordered due to an interference on f, there must exist at least one action a_k that adds fluent f (i.e., $a_k \in add_f$) between each a_i. This observation gives rise to three constraints: Constraint (18) counts the total number of occurrences of both a_i and a_k while Constraints (19) and (20) each split the plan into two for each a_i and ensure there are more actions a_k than a_j succeeding and preceding a_i, respectively.

$$\sum_{a_k \in add_f} Z_{a_k} + I_f \geq \sum_{a_j \in (del_f \cap pre_f)} Z_{a_j} + G_f \quad \forall f \in F \tag{18}$$

$$\sum_{a_k \in add_f} O_{a_i,a_k} \geq \sum_{a_j \in (del_f \cap pre_f)} O_{a_i,a_j} + G_f Z_{a_i} \tag{19}$$
$$\forall a_i \in A, f \in PRE_{a_i} \cap DEL_{a_i}$$

$$\sum_{a_k \in add_f} O_{a_k,a_i} + I_f Z_{a_i} \geq \sum_{a_j \in (del_f \cap pre_f)} O_{a_j,a_i} \tag{20}$$
$$\forall a_i \in A, f \in PRE_{a_i} \cap DEL_{a_i}$$

Symmetry Breaking Constraints Two actions that are equivalent in their preconditions, add effects, and delete effects (denoted $a_i \equiv a_j$) introduce symmetrically identical solutions (and non-solutions)

[3] Based on the interference constraints of Do and Kambhampati [8].

$$Z_{a_i} \leq Z_{a_j} \quad \forall i < j, a_i \equiv a_j \in A \tag{21}$$

$$O_{a_i,a_j} = 0 \quad \forall i < j, a_i \equiv a_j \in A \tag{22}$$

$$Est_{a_j} \leq Est_{a_i} + \sum_{a \in A} d_a(1 - Z_{a_i}) \tag{23}$$

$$\forall i < j, a_i \equiv a_j \in A$$

$$O_{a_j,a_i} = Z_{a_i} \quad \forall i < j, a_i \equiv a_j \in A, \tag{24}$$

$$\exists f \in F, a_i, a_j \in del_f \cap pre_f$$

Figure 4: Symmetry breaking constraints.

into the search space. We can break this symmetry by enforcing a lexicographical ordering as shown in Figure 4. Constraints (21)-(23) disallow the inclusion, ordering, and start of the action with lower index value before its equivalent action with higher index value, respectively. Constraint (24) ensures that two equivalent actions, a_i, a_j, that delete and require fluent f (i.e. $f \in F, a_i, a_j \in del_f \cap pre_f$) are ordered with respect to their index values.

Note that Constraint (23) can only be applied to $\mathcal{O}_O^{\text{MILP}}$ and $\mathcal{T}^{\text{MILP}}$. Constraints (17), (19), (20), and (24) do not introduce any ordering constraints that are not relevant to the validity of a POP. However, their addition to $\mathcal{O}_O^{\text{MILP}}$ can change the optimal solution because the orderings due to threats may now be replaced with explicit ordering constraints. We add Constraints (15)-(22) and Constraint (24) to $\mathcal{O}_C^{\text{MILP}}$, $\mathcal{O}_O^{\text{MILP}}$ and $\mathcal{T}^{\text{MILP}}$, and add Constraint (23) only to $\mathcal{O}_O^{\text{MILP}}$ and $\mathcal{T}^{\text{MILP}}$. We refer to these strengthened models as $\mathcal{O}_C^{\text{MILP+S}}$, $\mathcal{O}_O^{\text{MILP+S}}$ and $\mathcal{T}^{\text{MILP+S}}$, respectively.

8 Computational Results

In this section, we present the results of two computational experiments. In Experiment 1, we investigate the empirical behaviour of the three proxy functions. While we discussed in Section 4.3 that theoretically none of the proxy functions dominate the others, this result does not speak to the average empirical behaviour: it is possible that, in practice, a proxy function often results in more linearizations than others. To test this possibility, we focus on the *minimum reordering problem*: a version of LCFP where the number of actions is fixed. This restriction ensures that we are comparing the proxy functions while controlling for the complicating factor of different action sets that arises in the LCFP models. We show that, generally, there is also no empirical domination among the proxy functions, with our only significant comparison being that $\mathcal{T}^{\text{MILP+S}}$ achieves a statistically significant higher mean logarithmic number of linearizations than $\mathcal{O}_C^{\text{maxSAT}}$. We also provide empirical evidence that the MILP models for temporal flexibility and open ordering constraints are substantially faster than the state-of-the-art MaxSAT model on the minimum reordering problems.

In Experiment 2, we solve the full LCFP problems. Our findings show that the proposed models $\mathcal{O}_O^{\text{MILP+S}}$ and $\mathcal{T}^{\text{MILP+S}}$ can be solved to optimality faster on the majority of the tested instances, and scale better with the initial number of actions compared to $\mathcal{O}_C^{\text{maxSAT}}$. The solution quality across proxy functions is similar with no significant differences in the action cost or mean number of linearizations but with $\mathcal{T}^{\text{MILP+S}}$ and $\mathcal{O}_C^{\text{maxSAT}}$ finding a statistically significantly higher mean logarithmic number of linearizations than $\mathcal{O}_O^{\text{MILP+S}}$.

Experimental Details For both experiments, the initial plans are generated using *Fast-Forward* [12]. For the first experiment, we use eight domains from the International Planning Competition: `Depots`, `Driverlog`, `Freecell`, `Gripper`, `Logistics`, `Rovers`, `Tpp`, and `Zenotravel`, giving in total 144 instances. For the second experiment, we use the use the same experimental setup as Muise et al. [18] including the same domains: `Depots`, `Driverlog`, `Logistics`, `Rovers`, `Tpp`, and `Zenotravel`, giving in total 138 instances. The experiments ran on a MacBookPro computer with 2.66 GHz Intel Core i7. The MILP models were solved using IBM ILOG CPLEX 12.6.2 with 1 thread. For $\mathcal{O}_C^{\text{maxSAT}}$, we use the *SAT4j* MaxSAT 2.3.5 solver with a memory limit of 2GB. A time limit of 1,800 seconds was imposed on all models.

8.1 Experiment 1: Comparing Proxy Functions

To observe the effect of the optimization of each proxy objective on the number of linearizations in a POP, we fix the set of actions (i.e., $Z_a = 1, \forall a \in A$) and optimize the proxy objective functions.

Solution Quality In Table 1, we report the mean logarithmic number of linearizations considering the instances for which all three models return an optimal solution and for which we successfully count the number of linearizations: 99 instances. Values in boldface represent the maximum in each row. The number of linearizations is found through a simple depth-first search with a time limit of 30 minutes per instance. We note that the mean logarithmic number of linearizations is equivalent to the *geometric mean* of such numbers, which is more appropriate when comparing large-magnitude numbers (as the number of linearizations grows exponentially large with the number of actions). This measure is largely used in the optimization literature (see, e.g., [2, 4]).

We performed bootstrap paired t-tests [5] using two statistics: number of linearizations and logarithmic number of linearizations (base 10). Our results indicated that there are no statistically significant differences in the mean number of linearizations while for the mean logarithmic statistic the only significant difference ($p \leq 0.01$) is that $\mathcal{T}^{\text{MILP+S}}$ finds a higher mean than $\mathcal{O}_C^{\text{maxSAT}}$. Among the two models that optimized the closed ordering constraints, $\mathcal{O}_C^{\text{MILP+S}}$ and $\mathcal{O}_C^{\text{maxSAT}}$, $\mathcal{O}_C^{\text{maxSAT}}$ consistently dominated $\mathcal{O}_C^{\text{MILP+S}}$. Therefore the results for $\mathcal{O}_C^{\text{MILP+S}}$ are not presented in this section.

	Mean Log_{10} Number of Linearizations		
Dom	$\mathcal{T}^{\text{MILP+S}}$	$\mathcal{O}_O^{\text{MILP+S}}$	$\mathcal{O}_C^{\text{maxSAT}}$
`Dep`	**7.09**	6.97	6.93
`Dri`	5.84	**5.95**	5.83
`Fre`	7.55	6.80	**7.57**
`Gri`	**4.34**	**4.34**	**4.34**
`Log`	**11.91**	11.42	11.38
`Rov`	**8.53**	8.15	8.41
`Tpp`	**3.71**	**3.71**	**3.71**
`Zen`	**7.69**	7.55	**7.69**
Mean	**7.76**	7.51	7.63

Table 1: Solution quality in terms of linearizations (logarithmic) in Experiment 1.

In Figure 5, we plot the number of linearizations for which optimal solutions were found for both closed ordering flexibility and temporal flexibility and for which we successfully counted the number of linearizations. For 83% of the instances, optimization of both proxy objective functions resulted in POPs with the same number of linearizations.

Computational Effort We now compare the models with respect to the effort to optimize each proxy objective. Figure 6 shows a performance profile depicting the number of instances solved to optimality over the 30-minute time limit. Optimization of open order, temporal and closed ordering flexibility using $\mathcal{O}_O^{\text{MILP+S}}$, $\mathcal{T}^{\text{MILP+S}}$ and $\mathcal{O}_C^{\text{maxSAT}}$ models solve 126, 122 and 118 problem instances to optimality within 30 minutes, respectively. It can be observed that both MILP models outperform $\mathcal{O}_C^{\text{maxSAT}}$, while $\mathcal{O}_O^{\text{MILP+S}}$ is also superior to $\mathcal{T}^{\text{MILP+S}}$ across all time points.

8.2 Experiment 2: Solving LCFPs

Turning to LCFPs, we present results on three issues in this subsection: the quality of LCFPs produced by each model, the computational effort for each model, and the impact of the strengthening constraints that we introduced above.

Solution Quality In Table 2, we report the mean action cost for each planning domain. The table includes results from the 131 problems instances for which at least one of the approaches found a feasible solution. For instances for which an approach found no feasible

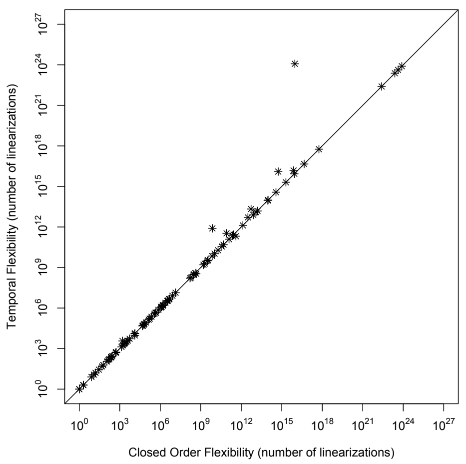

Figure 5: Number of linearizations between temporal flexibility and closed ordering flexibility (in logarithmic scale) in Experiment 1.

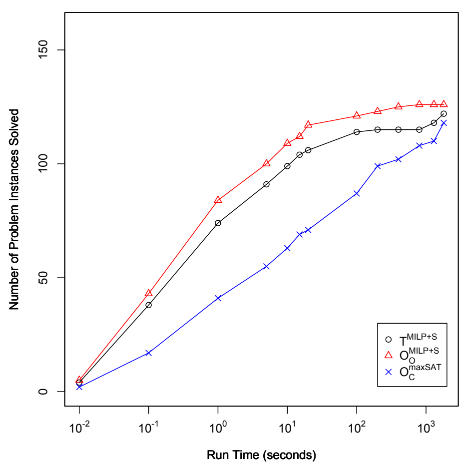

Figure 6: Performance profile (in log scale) for Experiment 1.

POP but another one did, we use the action cost of the input sequential plan for the former approach. Values in boldface are the minimum for each row. All four models perform similarly, a result reflected in the bootstrap paired t-tests showing no significant differences. The `Tpp` domain is the only one that appears to have variation, an observation that we further explore below (see Figure 10).

	Average Total Action Cost			
Dom	$\mathcal{O}_O^{\text{MILP+S}}$	$\mathcal{O}_C^{\text{MILP+S}}$	$\mathcal{T}^{\text{MILP+S}}$	$\mathcal{O}_C^{\text{maxSAT}}$
`Dep`	**42.30**	43.86	**42.30**	**42.30**
`Dri`	**26.80**	**26.80**	**26.80**	**26.80**
`Log`	**91.11**	91.94	**91.11**	91.14
`Rov`	**35.2**	**35.2**	**35.2**	**35.2**
`Tpp`	91.5	94.05	91.82	**85.95**
`Zen`	**33.1**	**33.1**	**33.1**	**33.1**
Mean	59.81	60.74	59.87	**58.95**

Table 2: Solution quality in terms of total action cost.

	Mean Log_{10} Number of Linearizations			
Dom	$\mathcal{O}_O^{\text{MILP+S}}$	$\mathcal{O}_C^{\text{MILP+S}}$	$\mathcal{T}^{\text{MILP+S}}$	$\mathcal{O}_C^{\text{maxSAT}}$
`Dep`	9.24	8.55	**11.17**	10.95
`Dri`	6.66	**6.81**	6.79	**6.81**
`Log`	18.01	**18.76**	18.70	**18.76**
`Rov`	14.82	**15.05**	**15.05**	**15.05**
`Tpp`	9.75	8.06	10.53	**10.58**
`Zen`	7.64	**7.84**	7.82	**7.84**
Mean	11.06	10.97	**11.75**	11.72

Table 3: Solution quality in terms of linearizations (logarithmic).

In Table 3 we report the average logarithmic number of linearizations considering instances for which all four models return a feasible solution with the same action cost and for which we successfully count the number of linearizations: 93 instances in total. Values in boldface represent the maximum in each row. As in Experiment 1, we generate the linearizations through a depth-first search with a 30-minute time limit per instance. $\mathcal{T}^{\text{MILP+S}}$ performs at the same level as the closed ordering constraint models ($\mathcal{O}_C^{\text{MILP+S}}$ and $\mathcal{O}_C^{\text{maxSAT}}$) while $\mathcal{O}_O^{\text{MILP+S}}$ trails substantially. Bootstrap paired t-tests indicate no significant differences between any pair in terms of the mean number of linearizations while reflecting the pattern in the table in terms of the mean of the logarithm of the number of linearizations: both $\mathcal{T}^{\text{MILP+S}}$ and $\mathcal{O}_C^{\text{maxSAT}}$ have a significantly higher log mean number of linearizations than $\mathcal{O}_O^{\text{MILP+S}}$ ($p \leq 0.01$).

Computational Effort We compare the models with respect to solution times, focusing on the strengthened formulations – we evaluate the effect of the strengthening below (see Figure 11). Figure 7 shows a performance profile depicting the number of instances solved to optimality over time. $\mathcal{O}_O^{\text{MILP+S}}$ and $\mathcal{T}^{\text{MILP+S}}$ each solved 119 instances out of 138, while $\mathcal{O}_C^{\text{MILP+S}}$ and $\mathcal{O}_C^{\text{maxSAT}}$ each solved 111 instances, all of which were also solved by the other methods.

For the 111 instances solved by all methods, $\mathcal{O}_O^{\text{MILP+S}}$ and $\mathcal{T}^{\text{MILP+S}}$ were faster than the other approaches in all but one case. On average, $\mathcal{O}_O^{\text{MILP+S}}$ and $\mathcal{T}^{\text{MILP+S}}$ were approximately 27 times and 20 times faster than $\mathcal{O}_C^{\text{maxSAT}}$, respectively. A scatter plot comparing the run times of $\mathcal{T}^{\text{MILP+S}}$ and $\mathcal{O}_C^{\text{maxSAT}}$ for all instances is depicted in Figure 8. The plot comparing $\mathcal{O}_O^{\text{MILP+S}}$ and $\mathcal{O}_C^{\text{maxSAT}}$ on the same basis is similar. The speedups obtained both by $\mathcal{O}_O^{\text{MILP+S}}$ and $\mathcal{T}^{\text{MILP+S}}$ are likely due to a smaller formulation when compared to other models. As noted above, the number of constraints in $\mathcal{O}_C^{\text{MILP+S}}$ (which is derived directly from $\mathcal{O}_C^{\text{maxSAT}}$) grows cubically with the number of actions, while in $\mathcal{O}_O^{\text{MILP+S}}$ and $\mathcal{T}^{\text{MILP+S}}$ the growth is quadratic. This explanation is supported by Figure 9, which indicates that the difference in

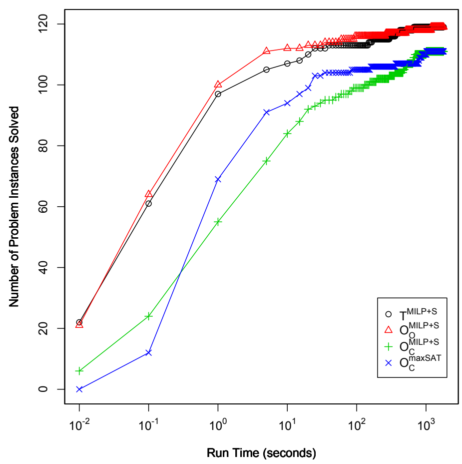

Figure 7: Performance profile (in log scale) for Experiment 2.

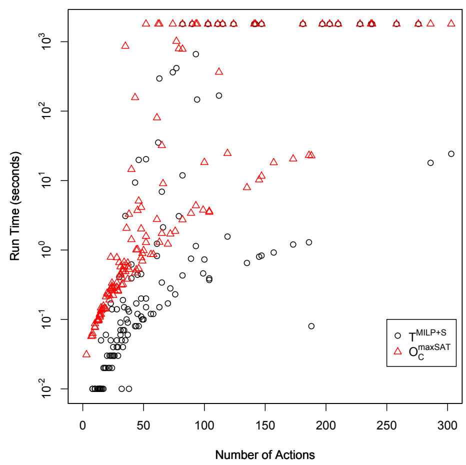

Figure 9: Run times of $\mathcal{O}_C^{\text{maxSAT}}$ and $\mathcal{T}^{\text{MILP+S}}$ and number of actions in the original plan (in logarithmic scale).

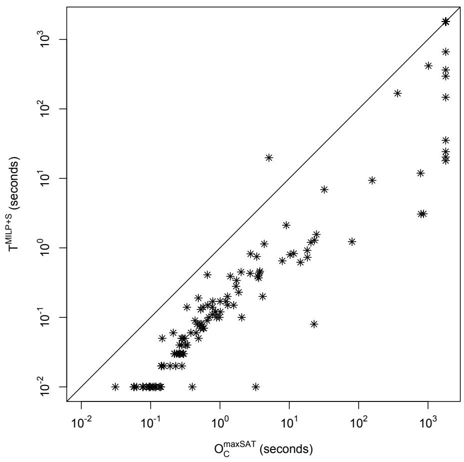

Figure 8: Run time comparison between $\mathcal{T}^{\text{MILP+S}}$ and $\mathcal{O}_C^{\text{maxSAT}}$ (in logarithmic scale).

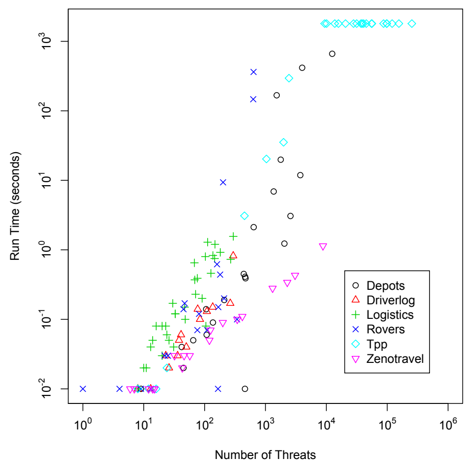

Figure 10: Run time performance of $\mathcal{T}^{\text{MILP+S}}$ and number of threat ordering constraints (in logarithmic scale).

run times between $\mathcal{T}^{\text{MILP+S}}$ and $\mathcal{O}_C^{\text{maxSAT}}$ is positively correlated with the number of actions in the original plan.

All models grow linearly with the number of threats in the plan, here encoded by Constraints (2), which now becomes more relevant to the size of both $\mathcal{O}_O^{\text{MILP+S}}$ and $\mathcal{T}^{\text{MILP+S}}$. Figure 10 depicts the run time of $\mathcal{T}^{\text{MILP+S}}$ as a function of the number of threat ordering constraints for each domain, and strongly suggests a direct correlation. The 19 instances that were unsolved by $\mathcal{O}_O^{\text{MILP+S}}$ and $\mathcal{T}^{\text{MILP+S}}$ are from the `Tpp` domain and have more than 10,000 threats.

The Effect of the Strengthening Constraints In Figure 11 we plot run time comparisons between the pairs of base and strengthened models. The effect is significant for the instances that take longer than one second to solve. On average, the strengthened models are one order of magnitude faster than their corresponding base models.

Summary of Results The performance profiles in Figures 6 and 7 clearly show the superior performance of $\mathcal{T}^{\text{MILP+S}}$ and $\mathcal{O}_O^{\text{MILP+S}}$ over $\mathcal{O}_C^{\text{maxSAT}}$ in terms of problem solving efficiency. Figure 11 demonstrates that the run time advantage is largely a result of the valid inequalities that we derived. The solution quality results are more nuanced, showing no significant differences in action cost in Experiment 2 or in mean number of linearizations in either experiment. The mean logarithmic number of linearizations does show superiority for $\mathcal{T}^{\text{MILP+S}}$ over $\mathcal{O}_C^{\text{maxSAT}}$ in Experiment 1 and for both $\mathcal{T}^{\text{MILP+S}}$ and $\mathcal{O}_C^{\text{maxSAT}}$ over $\mathcal{O}_O^{\text{MILP+S}}$ in Experiment 2.

9 Related Work

The problem of generating a flexible POP from an initial set of actions has been theoretically [1] and empirically [8, 18] investigated. Do and Kambhampati [8] studied the problem of generating a flexible order-constrained plan by minimizing the number of ordering constraints in a plan. Unlike POPs, order-constrained plans allow for concurrent execution of multiple actions, which can invalidate the plan if two interfering actions (i.e., $a_1 \in del_f$ and $a_2 \in add_f \cup pre_f$) overlap during execution. Do and Kambhampati [8] ordered all pairs of interfering actions with ordering constraints. Because we assume

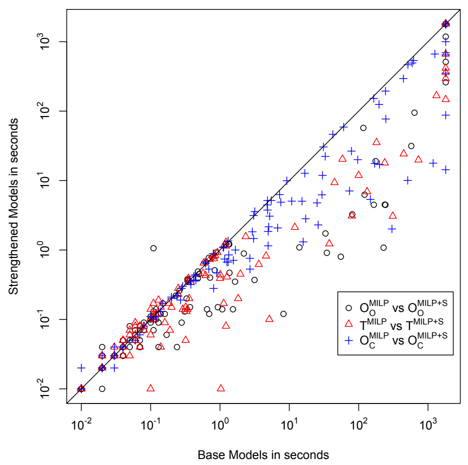

Figure 11: Effect of Constraints (15)-(24) on base models.

a sequential execution, we do not need to enforce the ordering of all such interfering actions unless a causal link is threatened. Therefore, we relax Do and Kambhampati's [8] ordering constraint definition to form our open ordering constraint definition.

Muise at al. [18] focused on the problem of finding a POP with minimum total action cost and, among all such minimum POPs, the one that minimizes the number of ordering constraints under transitive closure, given an initial sequential plan. The authors showed that minimizing the number of ordering constraints under transitive closure is positively correlated with the number of linearizations in the POP and developed the MaxSAT model denoted as $\mathcal{O}_C^{\text{maxSAT}}$ above.

10 Conclusion

We presented three mixed-integer programming models for converting a sequential plan to a POP by optimizing a combination of action cost and one of three different proxy functions: the number of open ordering constraints [8], the number of closed ordering constraints [18] and a novel proxy, temporal flexibility. We proved, through a set of counter-examples, that none of these functions dominates the others in terms of the number of linearizations in the resulting POPs. We then added valid strengthening constraints to these models, resulting in approximately an order of magnitude improvement in performance. Finally, we demonstrated that the two MILP models based on open ordering constraints and temporal flexibility achieve solution quality equal to that of the previous state-of-the-art MaxSAT approach [18] with a decrease in run time of approximately one order of magnitude.

An obvious direction for future work is to investigate the improvement of the MaxSAT model through the encoding of temporal variables rather than closed ordering constraints. The ideas of Crawford & Baker [6] and Frausto-Solis & Cruz-Chavez [11] may be useful here. We would also like to investigate other proxy functions and their relationship to POP flexibility.

A Appendix: Dominance Relations

To formally prove Proposition 4, Figures 12 to 15 present counter-examples showing that there does not exist a dominance relation between any pair of the three proxy functions in terms of the resulting number of linearizations in a POP. That is, it is not the case that optimizing one proxy function will always lead to more POP linearizations than optimizing one of the other two.

Each counter-example presents a planning problem with fluents indexed by integers and two graphs presenting the POP found by optimizing two of the objective functions, respectively. The nodes represent the actions and arcs represent the ordering constraints under the $\mathcal{O}_O$ definition. For clarity, we do not show the causal links, the action variables a_I and a_G, or the transitive closure. We use the notation $obj_1 \not\Rightarrow obj_2$ to represent the non-dominance of obj_1 over obj_2 with respect to the number of linearizations, where obj_1 and obj_2 represent a pair of proxy objectives. In order to show $obj_1 \not\Rightarrow obj_2$, we optimize the set of actions with respect to both obj_1 and obj_2, and show that $L_1 > L_2$, where L_i is the number of linearizations of objective i which counted through exhaustive enumeration. For each POP example, we report the proxy objective functions optimized, and the resulting $|\mathcal{O}_O|$, $|\mathcal{O}_C|$, T and L values.

Actions	Pre	Add	Del
a_I	-	0	-
a_1	0	1,7	-
a_2	1	2,3,8	-
a_3	0	6,9	4
a_4	6	4,5,10	-
a_5	4	2,3,11	-
a_6	2,3,5	12	-
a_G	7,8,9,10,11,12	-	-

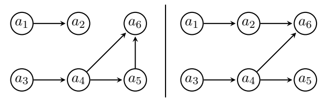

Figure 12: $min.\ |\mathcal{O}_C| \not\Rightarrow max.\ T$. **Left:** min. $|\mathcal{O}_C|$, $|\mathcal{O}_O| = 5$, $|\mathcal{O}_C| = 7, T = 16, L = 15$. **Right:** max. T, $|\mathcal{O}_O| = 5$, $|\mathcal{O}_C| = 8$, $T = 18, L = 16$.

Actions	Pre	Add	Del
a_I	-	8,9,10	-
a_1	8	0,3	-
a_2	0	1,2,4	-
a_3	0,1,2	5	-
a_4	9	1,6	-
a_5	10	2,7	-
a_G	3,4,5,6,7	-	-

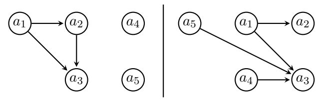

Figure 13: $max.\ T \not\Rightarrow min.\ |\mathcal{O}_C|$ and $max.\ T \not\Rightarrow min.\ |\mathcal{O}_O|$. **Left:** min. $|\mathcal{O}_C|$ or min. $|\mathcal{O}_O|$, $|\mathcal{O}_O| = 3$, $|\mathcal{O}_C| = 3$, $T = 14$, $L = 20$. **Right:** max. T, $|\mathcal{O}_O| = 4$, $|\mathcal{O}_C| = 4$, $T = 15$, $L = 18$.

Actions	Pre	Add	Del
a_I	-	7	-
a_1	7	2,3	0
a_2	7	0,1,4	-
a_3	0,7	1,2,5	-
a_4	1,2,7	6	-
a_G	3,4,5,6	-	-

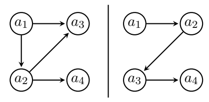

Figure 14: $min.\ |\mathcal{O}_O| \not\Rightarrow min.\ |\mathcal{O}_C|$ and $min.\ |\mathcal{O}_O| \not\Rightarrow max.\ T$. **Left:** min. $|\mathcal{O}_C|$ or max. T, $|\mathcal{O}_O| = 4$, $|\mathcal{O}_C| = 5$, $T = 4$, $L = 2$. **Right:** min. $|\mathcal{O}_O|$, $|\mathcal{O}_O| = 3$, $|\mathcal{O}_C| = 6$, $T = 0$, $L = 1$.

Actions	Pre	Add	Del
a_I	-	10	-
a_1	10	3,4,5	-
a_2	3,4,10	0,1,2,6	-
a_3	10	1,4,7	3
a_4	1,10	2,8	-
a_5	2,10	3,9	-
a_G	5,6,7,8,9	-	-

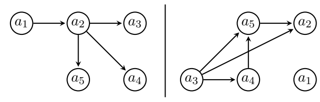

Figure 15: $min.\ |\mathcal{O}_C| \not\Rightarrow min.\ |\mathcal{O}_O|$. **Left:** min. $|\mathcal{O}_O|$, $|\mathcal{O}_O| = 4$, $|\mathcal{O}_C| = 7, T = 10, L = 6$. **Right:** min. $|\mathcal{O}_C|$, $|\mathcal{O}_O| = 5$; $|\mathcal{O}_C| = 6$, $T = 8, L = 5$.

REFERENCES

[1] Christer Bäckström, 'Computational aspects of reordering plans', *Journal of Artificial Intelligence Research*, 99–137, (1998).

[2] Dimitris Bertsimas, Iain Dunning, and Miles Lubin, 'Reformulation versus cutting-planes for robust optimization', *Computational Management Science*, **13**(2), 195–217, (2016).

[3] Sudip Biswas, Stephane Durocher, Debajyoti Mondal, and RahnumaIslam Nishat, 'Hamiltonian paths and cycles in planar graphs', in *Combinatorial Optimization and Applications*, ed., Guohui Lin, volume 7402 of *Lecture Notes in Computer Science*, 83–94, Springer Berlin Heidelberg, (2012).

[4] Christina Büsing, Arie M. C. A. Koster, and Manuel Kutschka, 'Recoverable robust knapsacks: the discrete scenario case', *Optimization Letters*, **5**(3), 379–392, (2011).

[5] Paul R. Cohen, *Empirical Methods for Artificial Intelligence*, MIT Press, Cambridge, MA, USA, 1995.

[6] James M. Crawford and Andrew B. Baker, 'Experimental results on the application of satisfiability algorithms to scheduling problems', in *Proceedings of the Twelfth National Conference on Artificial Intelligence (Vol. 2)*, AAAI 1994, pp. 1092–1097, Menlo Park, CA, USA, (1994). American Association for Artificial Intelligence.

[7] Andrew J. Davenport, Christophe Gefflot, and J. Christopher Beck, 'Slack-based techniques for robust schedules', in *Proceedings of the Sixth European Conference on Planning (ECP 2001)*, (2001).

[8] Minh B. Do and Subbarao Kambhampati, 'Improving the temporal flexibility of position constrained metric temporal plans', in *Proceedings of the Thirteenth International Conference on Automated Planning and Scheduling (ICAPS 2003)*, (2003).

[9] Richard E. Fikes, Peter E. Hart, and Nils J. Nilsson, 'Learning and executing generalized robot plans', *Artificial Intelligence*, **3**, 251 – 288, (1972).

[10] Richard E. Fikes and Nils J. Nilsson, 'Strips: A new approach to the application of theorem proving to problem solving', in *Proceedings of the Second International Joint Conference on Artificial Intelligence*, IJCAI 1971, pp. 608–620, San Francisco, CA, USA, (1971). Morgan Kaufmann Publishers Inc.

[11] Juan Frausto-Solis and Marco A. Cruz-Chavez, 'A reduced codification for the logical representation of job shop scheduling problems', in *Computational Science and Its Applications ICCSA 2004*, eds., Antonio Lagan, MarinaL. Gavrilova, Vipin Kumar, Youngsong Mun, C.J.Kenneth Tan, and Osvaldo Gervasi, volume 3046 of *Lecture Notes in Computer Science*, 553–562, Springer Berlin Heidelberg, (2004).

[12] Jörg Hoffmann and Bernhard Nebel, 'The FF planning system: Fast plan generation through heuristic search', *Journal of Artificial Intelligence Research*, **14**(1), 253–302, (May 2001).

[13] Subbarao Kambhampati and James A. Hendler, 'A validation-structure-based theory of plan modification and reuse', *Artificial Intelligence*, **55**(2), 193–258, (1992).

[14] Subbarao Kambhampati and Smadar Kedar, 'A unified framework for explanation-based generalization of partially ordered and partially instantiated plans', *Artificial Intelligence*, **67**(1), 29–70, (1994).

[15] David McAllester and David Rosenblitt, 'Systematic nonlinear planning', in *Proceedings of the Ninth National Conference on Artificial Intelligence*, volume 2 of *AAAI 1991*, pp. 634–639. AAAI Press, (1991).

[16] Steven Minton, John Bresina, and Mark Drummond, 'Total-order and partial-order planning: A comparative analysis', *Journal of Artificial Intelligence Research*, **2**(1), 227–262, (January 1995).

[17] Christian Muise, *Exploiting Relevance to Improve Robustness and Flexibility in Plan Generation and Execution*, Ph.D. dissertation, Graduate Department of Computer Science, University of Toronto, 2014.

[18] Christian Muise, Sheila A. McIlraith, and J. Christopher Beck, 'Optimally relaxing partial-order plans with maxsat', in *Proceedings of the Twenty Second International Conference on Automated Planning and Scheduling (ICAPS 2012)*, pp. 358–362, (2012).

[19] Dana Nau, Malik Ghallab, and Paolo Traverso, *Automated Planning: Theory & Practice*, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004.

[20] XuanLong Nguyen and Subbarao Kambhampati, 'Reviving partial order planning', in *Proceedings of the Seventeenth International Joint Conference on Artificial Intelligence (IJCAI2001)*, pp. 459–464, (2001).

[21] Nicola Policella, Stephen F. Smith, Amedeo Cesta, and Angelo Oddi, 'Generating robust schedules through temporal flexibility', in *Proceedings of the Fourteenth International Conference on Automated Planning and Scheduling (ICAPS 2004)*, (2004).

[22] Florian Pommerening, Gabriele Roger, Malte Helmert, and Blai Bonet, 'LP-based heuristics for cost-optimal planning', in *Proceedings of the Twenty Fourth International Conference on Automated Planning and Scheduling (ICAPS 2014)*, (2014).

[23] Earl D. Sacerdoti, 'The nonlinear nature of plans', in *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, volume 1 of *IJCAI 1975*, pp. 206–214, San Francisco, CA, USA, (1975). Morgan Kaufmann Publishers Inc.

[24] Fazlul Hasan Siddiqui and Patrik Haslum, 'Block-structured plan deordering', in *Proceedings of the Twenty Fifth Australasian Joint Conference on Artificial Intelligence*, pp. 803–814, (2012).