# Temporal Optimization Planning for Fleet Repositioning

**Kevin Tierney** and **Rune Møller Jensen**
IT University of Copenhagen
Rued Langgaards Vej 7, DK-2300, Copenhagen S, Denmark
{kevt,rmj}@itu.dk

## Abstract

Fleet repositioning problems pose a high financial burden on shipping firms, but have received little attention in the literature, despite their high importance to the shipping industry. Fleet repositioning problems are characterized by chains of interacting activities, but state-of-the-art planning and scheduling techniques do not offer cost models that are rich enough to represent essential objectives of these problems. To this end, we introduce a novel framework called Temporal Optimization Planning (TOP). TOP uses partial order planning to build optimization models associated with the different possible activity scenarios and applies branch-and-bound with tight lower bounds to find optimal solutions efficiently. We show how key aspects of fleet repositioning can be modeled using TOP and demonstrate experimentally that our approach scales to the size of problems considered by our industrial collaborators.

## Introduction

Liner shipping networks transport containerized cargo through regularly scheduled shipping routes. Fleet repositioning problems consist of moving ships between services in a liner shipping network in order to better orient the overall network to the world economy and to ensure the proper maintenance of ships. Thus, fleet repositioning involves sailing and loading activities subject to complex handling and timing restrictions. As is the case for many industrial problems, the objective is cost minimization (including costs for $CO_2$ emissions and pollution), and it is important that all cost elements, including the ones that are only loosely coupled with activity choices, can be accurately modeled.

Optimization problems that involve chains of activities with complex interactions, like fleet repositioning problems, are hard to represent as mathematical programs. AI-planning and OR-scheduling offer a wide range of approaches to allocate interacting activities over time, but it has been observed in both fields (e.g., (Karger, Stein, and Wein 1997; Smith, Frank, and Jónsson 2000)) that the compound objectives of real-world problems often are hard to express in terms of the simple objective criteria like makespan and tardiness minimization considered in these approaches.

In this paper, we introduce a novel general framework called Temporal Optimization Planning (TOP) and show that it can model key aspects of fleet repositioning problems. The core idea of TOP is to associate durative planning actions with optimization model components and use planning algorithms to build and search through complete optimization models that are associated with the different activity scenarios of the problem. In contrast to advanced temporal planning languages (Fox and Long 2006), TOP does not enforce a strong semantic relation between planning actions and optimization components. The format of optimization components can be chosen freely by the model designer as long as the set of complete plans corresponds to the set of activity scenarios of the problem.

In this way, TOP allows general optimization models to be constructed, but at the same time makes it possible to naturally represent and explore complex interactions between durative activities with the expressive action models of AI-planning and its powerful search algorithms. In fact, TOP accommodates any optimization model over real-valued variables and thus is a simple way to reason about interacting durative activities in such models.

We solve TOP problems by an optimization version of partial-order planning (Penberthy and Weld 1992) based on a branch-and-bound and demonstrate the approach for linear optimization models. We define a general lower bound for partial plans in the naturally occurring case where the minimum costs of optimization components are non-negative. We show that this bound can be improved by an extension of the $h_{max}$ heuristic (Haslum and Geffner 2000) that makes it possible to estimate the cost of required actions not currently in the plan.

The TOP framework is validated experimentally on a set of fleet repositioning problems developed in collaboration with a liner shipping company. We provide a basic model of a fleet repositioning problem that gives a first step towards solving such problems in TOP. We

include key aspects such as the temporal interaction between ships and complex cost objectives such as sailing fuel consumption and the fixed cost of ships over time. Our experimental evaluation shows that the TOP approach scales to the size of problems faced by the industry and that our lower bounds are tight enough to reduce the total computation time by orders of magnitude.

The remainder of the paper is organized as follows. First we describe fleet repositioning problems, followed by a formalization of TOP along with a description of the $h^{cost}_{max}$ heuristic for estimating lower bounds of partial plans. Then we describe our model of a simple fleet repositioning problem and we present experimental results showing the performance of the TOP framework with several plan selection heuristics. Finally, we draw conclusions and discuss directions for future work.

## Fleet Repositioning

Liner shipping networks consist of multiple *services*, which are circular routes that connect a sequence of ports. Each service consists of one or more ships that maintain a weekly schedule. That is, each port on a service is visited by a ship on the same day each week, and a service is assigned as many ships as are necessary to maintain weekly frequency. Ships are periodically *repositioned*, in which they are moved from their current service to some other service. Shipping firms reposition their ships so that they can undergo regular maintenance, be returned to their owner, or to better match ships to markets.

Consider the example repositioning problem shown in Figure 1 in which a new service is being started and requires ships to sail on it. This example involves the temporal alignment of several ships. First a ship from the service on the left must sail to the dashed line service, but the ship on the dashed line service may not leave for its goal service until its replacement ship has arrived, and cargo has been transshipped. Second, the two ships sailing to the service on the right must have a time spacing of exactly one week. Complicating matters further is that shipping firms want to reposition their ships as cheaply as possible.

Fleet repositioning problems can be even more complicated than the example given, with several ships replacing one another in long chains. Aligning the ships temporally and finding a minimum cost plan of activities presents a significant challenge. The possible activities that a ship may undertake during repositioning is given as follows.

- Ships can leave or enter a service only at a port on days when the port is scheduled to be called.
- Ships must sail between their minimum and maximum speed, the cost of which is a function of the ship's speed and draft,
- Ships can idle at a port, incurring a fixed cost per hour (in shipping parlance, *hotel cost*).

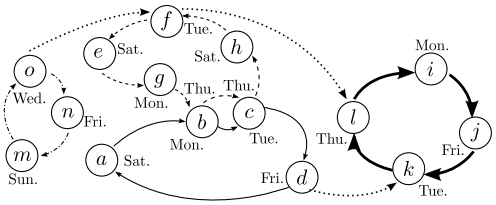


Figure 1: An example fleet repositioning problem involving three ships. Two ships are being repositioned from their initial services (solid, dashed) to a new service (bold solid). The ship on the dashed service is being replaced by a ship from the dashed-dotted service, which will cease operations. Circles represent ports and are labeled with the day the service calls the port.

- Cargo can be on/off loaded, resulting in a fee per container moved.
- Cargo can be directly transhipped between ships, incurring a reduced per container fee.
- Ships can move equipment (empty containers) to where it is needed, reducing the overall cost of repositioning.
- Ships may satisfy certain demands in the shipping network for a profit.

A number of constraints pose restrictions on when the given activities may take place. Ships must be replaced by another ship in order to leave their service, except for certain, designated ships. Ships may not load or unload cargo in certain ports due to cabotage restrictions, which are laws that prevent foreign ships from offering domestic services. If multiple ships are entering the same service, they must enter one week apart in distance or time from one another. In addition, multiple ships must alternate in size such that if there are several ships entering a service, no two ships of the same capacity should follow one another.

Given the high expense of repositioning, the goal of fleet repositioning problems is to find a scenario of activities, which involve continuous decisions regarding sailing time and cost configurations, associated with a lowest cost optimization model.

The fleet repositioning problem is difficult to solve for existing scheduling and planning methods.

Scheduling is concerned with the optimal allocation of scarce resources to activities over time (Karger, Stein, and Wein 1997), and scheduling research has focused on problems that only involve a small, fixed set of choices, while planning problems like fleet repositioning often involve cascading sets of choices that interact in complex ways (Smith, Frank, and Jónsson 2000). Another limitation is that mainstream scheduling research has focused predominately on optimization of selected, simple objective criteria such as minimizing makespan or minimizing tardiness (Smith 2005). More general objective criteria are required in order to solve fleet repositioning problems.

Integer Programming (IP) has successfully been applied to solve classical planning problems using competitive encodings based on the planning graph heuristic (Van Den Briel, Vossen, and Kambhampati 2005). An earlier version of this approach has also been used to extend classical planning with possibly continuous state variables over which linear constraints and objectives can be stated (Kautz and Walser 1999). A limitation of these encodings is, though, that they are unable represent continuous time. In general, it is difficult to represent partial-order planning with IP since the number of actions in a partial-order plan in principle is unbounded. A SAT encoding of actions with continuous duration (Shin and Davis 2005), however, shows that other encodings of continuous time using IP may be possible, but this encoding currently does not cover any objective criteria except minimizing the number of unique time points in the plan.

Within AI, there exists domain independent planners for temporal planning languages and specialized application planners.With respect to the former, while early approaches had limited scalability (e.g., (Ghallab and Laruelle 1994; Penberthy and Weld 1995)), a number of powerful solvers have recently been developed for planning languages with durative actions, real-valued state variables, and linear change of quantities during action execution (e.g. (Coles et al. 2009; 2010; Li and Williams 2008; Shin and Davis 2005)). These planning languages can model domains where activities depend on shared resources like electric power during execution, which is a typical situation for popular application domains within robotics and aerospace systems (e.g., (Frank, Gross, and Kurklu 2004; Muscettola 1993)). However, the fleet repositioning problem involves decoupled actions that intersect only temporally. Furthermore, most of these domain independent and application specific planning systems only allow simple objective criteria like makespan minimization. A notable exception is Sapa (Do and Kambhampati 2003), which can represent multi-criteria objectives covering any combination of makespan minimization and minimization of fixed action and resource costs. On the other hand, Sapa only handles discrete time and fixed action durations.

Recently, the 2008 International Planning Competition (Helmert, Do, and Refanidis 2008) featured a net benefit optimization category with several entries: HSP*, MIPS-XXL, and Gamer. These planners are unable to reason temporally, and only support fixed action costs in their objective, preventing them from handling many aspects of fleet repositioning problems, such as hotel costs and variable action cost. Linear Programming has been used to strengthen plan length estimation (e.g. (Bylander 1997; Van Den Briel et al. 2007)), however these approaches do not handle a cost-based objective or temporal setting, with the ordering relaxation in (Van Den Briel et al. 2007) being particularly troubling for a temporal planner.

## Temporal Optimization Planning

In the absence of a suitable method for solving fleet repositioning problems, we introduce Temporal Optimization Planning (TOP). TOP diverges fundamentally from classical AI-planning approaches by introducing two sets of variables that decouple the planning problem from the optimization model. Thus, the optimization model is not tightly bound to the semantics of actions. Actions are merely used as handles to optimization components that are built together to complete optimization models using partial-order planning. This decoupling makes it possible to formulate any objective that can be expressed by the applied optimization model. Moreover, computationally expensive action models including real-valued state variables and general objective functions are avoided.

In contrast to the current trend in advanced temporal planning languages, TOP bypasses computationally expensive dense time models of shared resources like electric power consumption during activities. These models are important for the robotic or aerospace applications that often are targeted in AI-planning (e.g., (Frank, Gross, and Kurklu 2004; Muscettola 1993)), but TOP focuses on more physically separated activities where resources are exclusively controlled.

On the other hand, while this decoupling offers some new possibilities, it makes TOP less capable of solving traditional planning problems where a strong coupling is assumed as well as problems that fit within a classical scheduling model.

TOP is built off a state variable representation of propositional STRIPS planning (Fikes and Nilsson 1971). TOP utilizes partial-order planning (Penberthy and Weld 1992), and extends it in several ways. First, an optimization model is associated with each action in the planning domain. This allows for complex objectives and cost interactions that are common in real world optimization problems to be easily modeled. Second, instead of focusing on simply achieving feasibility, TOP minimizes a cost function. Finally, begin and end times can be associated with actions, making them durative. Such actions can have variable durations that are coupled with a cost function.

Formally, let $\mathcal{V} = \{v_1, \cdots, v_n\}$ denote a set of *state variables* with finite domains $D(v_1), \cdots, D(v_n)$. A *state variable assignment* $\omega$ is a mapping of state variables to values $\{v_{i(1)} \mapsto d_{i(1)}, \cdots, v_{i(k)} \mapsto d_{i(k)}\}$ where $d_{i(1)} \in D(v_{i(1)}), \cdots, d_{i(k)} \in D(v_{i(k)})$. We also define $vars(\omega)$ as the set of state variables used in $\omega$.

A TOP problem is represented by a tuple

$$P = \langle \mathcal{V}, \mathcal{D}, \mathcal{A}, \mathcal{I}, \mathcal{G}, pre, \mathit{eff}, \boldsymbol{x}, obj, con \rangle,$$

where $\mathcal{D}$ is the Cartesian product of the domains $D(v_1) \times \cdots \times D(v_n)$, $\mathcal{A}$ is the set of actions, $\mathcal{I}$ is a total state variable assignment (i.e. $vars(\mathcal{I}) = \mathcal{V}$) representing the initial state, $\mathcal{G}$ is a partial assignment (i.e. $vars(\mathcal{G}) \subseteq \mathcal{V}$) representing the goal states, $pre_a$ is a partial assignment representing the precondition of action $a$, $\mathit{eff}_a$ is a partial assignment representing the effect

of action $a$[1], $\boldsymbol{x} \in \mathbb{R}^m$ is a vector of optimization variables[2] that includes the begin and end time of each action, $x_b^a$ and $x_e^a$ respectively, for all actions $a \in A$, $obj_a : \mathbb{R}^m \to \mathbb{R}$ is a cost term introduced by action $a$, and $con_a : \mathbb{R}^m \to \mathbb{B}$ is a constraint expression introduced by action $a$ with $con_a \models x_b^a \leq x_e^a \wedge x_b^a \geq 0 \wedge x_e^a \geq 0$.

Let $S = \{\omega | vars(\omega) = \mathcal{V}\}$ denote the set of all the possible states. An action $a$ is applicable in $s \in S$ if $pre_a \subseteq s$ and is assumed to cause an instantaneous transition to a successor state defined by the total assignment

$$succ_{a,s}(v) = \begin{cases} eff_a(v) & \text{if } v \in vars(eff_a), \\ s(v) & \text{otherwise.} \end{cases}$$

We further define $M_a = min\{obj_a | con_a\}$, which is the optimization model component introduced by action $a$.

A temporal optimization plan is represented by a tuple $\langle A, C, O, M \rangle$, where $A$ is the set of actions in the plan, $C$ is a set of causal links, $O$ is a set of ordering constraints of the form $a \prec b$, where $a, b \in A$, and $M$ is an optimization model associated with the plan. $M$ is defined by

$$\begin{aligned} \min \quad & \sum_{a \in A} obj_a(\boldsymbol{x}) \\ \text{s.t.} \quad & x_e^{a_i} \leq x_b^{a_j} && \forall a_i \prec a_j \in O \quad (1) \\ & con_a(\boldsymbol{x}) && \forall a \in A \quad (2) \end{aligned}$$

The objective of $M$ is to minimize the sum of the costs introduced by actions. The first constraint ensures that the start and end time of actions are consistent with the plan, and second constraint requires that each action constraint is satisfied. Let $cost(\pi)$ denote the cost of an optimal solution to $M$ to a partial plan $\pi$.

An *open condition* $\xrightarrow{\mu} b$ is an unfulfilled precondition $\mu$ of action $b \in A$, that is, $\mu \in pre_b$ and $\forall a \in A$, $a \xrightarrow{\mu} b \notin C$. An *unsafe link* is a causal link $a \xrightarrow{\mu} b$ that is threatened by an action $c$ such that *i)* $vars(\mu) \in eff_c$, *ii)* $\mu \notin eff_c$, and *iii)* $\{a \prec c \prec b\} \cup O$ is consistent.

To deal with durative actions in TOP we need to keep track of another type of flaw called *interference*. We adopt an interference model based on the exclusive right to state variables (Sandewall and Rönnquist 1986). Thus, two actions $a$ and $b$ interfere if $vars(eff_a) \cap vars(eff_b) \neq \emptyset$ and $O$ implies neither $a \prec b$ nor $b \prec a$.

[1]In practice, is it often more convenient to represent actions in a more expressive form, e.g. by letting the precondition be a general expression on states $pre_a : S \to \mathbb{B}$ and represent conditional effects like resource consumption by letting the effect be a general transition function, depending on the current state of $S$, $eff_{a,s} : S \to S$. Such expressive implicit action representations may also be a computational advantage. We have chosen a ground explicit representation of actions because it simplifies the presentation and more expressive forms can be translated into it.

[2]In a slight abuse of notation, we sometimes let $\boldsymbol{x}$ denote a set rather than a vector.

**Algorithm 1** Optimization planning algorithm, based on (Williamson and Hanks 1996).

```
1: function TOP(I, G)
2:     Π ← {INITIALTOP(I,G)}
3:     π_best ← null
4:     u ← ∞ ▷ Cost of the incumbent (upper bound)
5:     while Π ≠ ∅ do
6:         π ← SELECTPLAN(Π)
7:         Π ← Π \ {π}
8:         if NUMFLAWS(π) = 0 ∧ COST(π) < u then
9:             u ← COST(π)
10:            π_best ← π
11:        else if ESTIMATECOST(π) < u then
12:            f ← SELECTFLAW(π)
13:            Π ← Π ∪ REPAIRFLAW(π, f)
14:    return π_best
```

An open condition flaw $\xrightarrow{\mu} b$ can be repaired by linking $\mu$ to an action $a$ such that $\mu \in eff_a$ and by posting an ordering constraint over $a$ and $b$. Thus, $C \leftarrow C \cup \{a \xrightarrow{\mu} b\}$ and $O \leftarrow O \cup \{a \prec b\}$. In the case that $a \notin A$, $A \leftarrow A \cup \{a\}$ and $O \leftarrow O \cup \{a_0 \prec a, a \prec a_\infty\}$.

An unsafe link $a \xrightarrow{\mu} b$ that is threatened by action $c$ can be repaired by either adding the ordering constraint $c \prec a$ (*demotion*) or $b \prec c$ (*promotion*) to $O$. Similar to unsafe links, an interference between actions $a$ and $b$ can be fixed by posting either $a \prec b$ or $b \prec a$ to $O$.

Together, open conditions, unsafe links and interferences constitute *flaws* in a plan. Let $flaws(\pi) = open(\pi) \cup unsafe(\pi) \cup interfere(\pi)$ be the set of flaws in the plan $\pi$, where $open(\pi)$ is the set of open conditions, $unsafe(\pi)$ is the set of unsafe links, and $interfere(\pi)$ is the set of interferences. We say that $\pi$ is a *complete plan* if $|flaws(\pi)| = 0$, otherwise $\pi$ is a *partial plan*. A plan $\pi^*$ is optimal if it is feasible and for all feasible solutions $\pi$, $cost(\pi^*) \leq cost(\pi)$.

## Linear Temporal Optimization Planning

To solve fleet repositioning problems, we introduce linear temporal optimization planning (LTOP). In LTOP, all of the optimization models associated with planning actions have a linear cost function and a conjunction of linear constraints. Thus, $obj_a$ is of the form $\mathbf{c^a}\boldsymbol{x}'$, where $\mathbf{c^a} \in \mathbb{R}^m$ and $con_a$ is of the form $\bigwedge_{1 \leq i \leq n_a} (\boldsymbol{\alpha}_i^a \boldsymbol{x}' \leq \beta_i)$, where $\boldsymbol{\alpha}_i^a \in \mathbb{R}^m$, $\beta_i \in \mathbb{R}$ and $n_a$ is the number of constraints associated with action $a$. Thus, $M_a$ and $M$ are linear programs (LPs).

Algorithm 1 shows a branch-and-bound algorithm that finds an optimal plan to an LTOP problem. First, an initial plan $\pi_{init}$ is created by the INITIALTOP function (line 2). We define $\pi_{init} = \langle \{a_0, a_\infty\}, \emptyset, \{a_0 \prec a_\infty\}, M_{init} \} \rangle$, where $a_0$ is an action representing $\mathcal{I}$ with $pre_{a_0} = \emptyset$ and $eff_{a_0} = \mathcal{I}$, $a_\infty$ is an action representing $\mathcal{G}$ with $pre_{a_\infty} = \mathcal{G}$ and $eff_{a_\infty} = \emptyset$, and $M_{init}$ is an optimization model with no objective and two constraints, $con_{a_0}$ and $con_{a_\infty}$, which are special constraints on the dummy actions $a_0$ and $a_\infty$ such that $con_{a_0} = (x_b^{a_0} =$

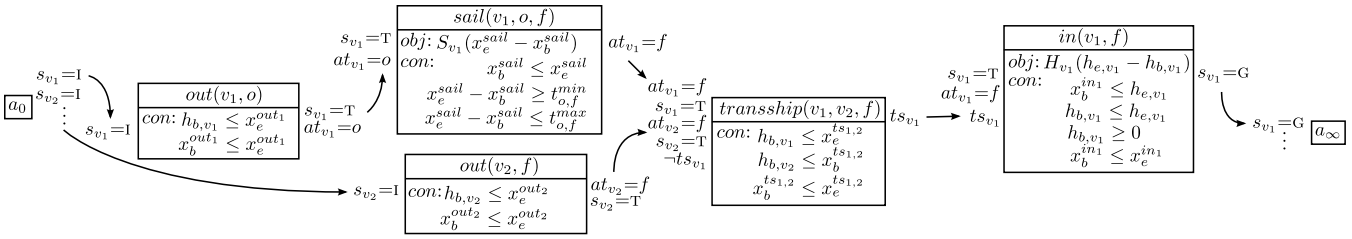


Figure 2: A partial temporal optimization plan from Figure 1 showing the ship sailing from the dash-dotted service to the dashed service ($v_1$) and the ship sailing from the dashed service to the bold, solid service. Boxes represent actions and contain associated optimization models. Arrows between boxes show causal links. The optimization variables $x_b^a$ and $x_e^a$ represent the begin and end time of action $a$, and $h_{b,v}$, $h_{e,v}$ are the begin and end hotel time of vessel $v$ respectively. Each vessel $v$ is associated with a state variable $s_v$ with a domain of $\{\text{I}, \text{T}, \text{G}\}$ which indicate that $v$ is on its initial service, in transit or at its goal service, respectively.

$x_e^{a_0} \wedge x_b^{a_0} \geq 0$) and $con_{a_\infty} = (x_b^{a_\infty} = x_e^{a_\infty} \wedge x_b^{a_\infty} \geq 0)$. The optimization variables $x_b^{a_0}, x_e^{a_0}, x_b^{a_\infty}$ and $x_e^{a_\infty}$ represent the begin and end times of actions $a_0$ and $a_\infty$ respectively.

The algorithm then selects a plan from $\Pi$ (line 6) and checks to see if it is a complete plan. If the plan is complete, its cost is compared with the current upper bound ($u$), and if the cost is lower, the *incumbent* $\pi_{best}$ is replaced with the current plan $\pi$ and the upper bound is updated (lines 9 and 10). When $\pi$ is a partial plan, an estimated lower bound of the plan is compared with the cost of the incumbent solution (line 11). If the estimated cost of the plan is higher, the plan is discarded. Otherwise, a flaw is selected and repaired (lines 12 and 13). This process is repeated until $\Pi$ is empty, at which point the current incumbent is returned, if there is one.

Algorithm 1 is guaranteed to find the optimal solution (if there is one) as long as EstimateCost does not overestimate the cost of completing a partial plan. To prune as much of the branch-and-bound tree as possible, we need tight lower bounds. If we require that the cost of each action subject to its constraints is non-negative, we can prove that $cost(\pi)$ is such a lower bound.

**Proposition 1.** *Given any valid partial plan* $\pi = \langle A, C, O, M \rangle$ *where* $M_a \geq 0 \ \forall a \in A$, $cost(\pi) \leq cost(\bar{\pi})$ *for any completion* $\bar{\pi}$ *of* $\pi$.

*Proof.* Let $\pi'$ be $\pi$ with a single flaw repaired. The flaw is either *i*) an unsafe link, *ii*) an interference, *iii*) an open condition being satisfied by an action in the plan, or *iv*) an open condition being satisfied by an action not in the plan.

In cases *i* and *ii* the flaw is repaired by adding an ordering constraint to $\pi$, which further constrains $\pi$, thus $cost(\pi) \leq cost(\pi')$. Case *iii* results in a new causal link and an ordering constraint, and is therefore the same as cases *i* and *ii*. In case *iv*, the action's optimization model is added to $\pi$, but since the cost function of the action must be non-negative under its constraints, $cost(\pi')$ cannot be less than $cost(\pi)$. By applying this argument inductively on the complete branch-and-bound subtree grown from $\pi$, we get $cost(\pi) \leq cost(\bar{\pi})$ for any completion $\bar{\pi}$ of $\pi$. □

## Heuristic Cost Estimation

Although $cost(\pi)$ provides a reasonable lower bound for $\pi$, the bound is only computed over actions in the plan. It can be strengthened by also reasoning over actions that are not in the plan. We present an extension of the $h_{max}$ heuristic (Haslum and Geffner 2000), $h_{max}^{cost}$, which estimates the cost of achieving the open conditions of a plan $\pi$. Let

$$h_{max}^{cost}(\omega, \pi) = \begin{cases} 0 & \text{if } \omega \subseteq \mathit{effs}_\pi, \text{ else} \\ f(\omega, \pi) & \text{if } \omega = \{\mu\}, \text{ else} \\ g(\omega, \pi) & \text{if } |\omega| > 1, \end{cases}$$

$$f(\omega, \pi) = \min_{\{a \in \mathcal{A} \setminus A | \mu \in \mathit{eff}_a\}} \{M_a + h_{max}^{cost}(\mathit{pre}_a, \pi)\},$$
$$g(\omega, \pi) = \max_{\mu \in \omega} \{h_{max}^{cost}(\{\mu\}, \pi)\},$$

where $\omega$ is a partial state variable assignment, $\mu$ is a single state variable assignment $v \mapsto d$, and $\mathit{effs}_\pi = \bigcup_{a \in A} \mathit{eff}_a$. The heuristic takes the max over the estimated cost of achieving the elements in the given assignment $\omega$. The cost is zero if the elements are already in $\pi$, otherwise the minimum cost of achieving each element is computed by finding the cheapest way of bringing that element into the plan.

**Proposition 2.** *Given any valid partial plan* $\pi = \langle A, C, O, M \rangle$ *where* $M_a \geq 0 \ \forall a \in A$, $cost(\pi) + h_{max}^{cost}(open(\pi), \pi) \leq cost(\bar{\pi})$ *for any completion* $\bar{\pi}$ *of* $\pi$.

*Proof.* We have $h_{max}^{cost}(\omega, \pi) = \sum_{a \in R} M_a$, where R is a set of actions not currently in $\pi$ ($R \cap A = \emptyset$) that are required to resolve $\omega$ and among such sets has the minimum cost. Thus, any completion $\bar{\pi}$ of $\pi$ as described in Proposition 1 must at least increase $cost(\pi)$ by $h_{max}^{cost}(\omega, \pi) = \sum_{a \in R} M_a$. □

## Modeling Fleet Repositioning

We describe an LTOP model of fleet repositioning problems that represents a first step towards modelling and

solving fleet repositioning problems. Our model represents a subset of fleet repositioning that captures its difficult elements, while excluding excessive detail. We focus on several key components of the fleet repositioning problems that form the basis for more difficult versions including *i*) the temporal interaction between ships in a sequence of replacements, *ii*) variable ship sailing durations, *iii*) sailing costs that vary linearly with the sailing duration, and *iv*) ship hotel costs that must span multiple actions.

Given a set of services, the goal is to carry out a sequence of repositionings in which a ship on each service is moved to the next service in the sequence at minimal cost. A ship may not be moved from its service until replaced by another ship, except in the case of the first service in the sequence.

Ships cease operations on their initial service through the action *out* and begin operations on a service through the *in* action. During the time in between *out* and *in*, ships are charged their hotel cost, which is represented in the objective of the *in* action. The hotel cost objective is given in the *out* action for ships that do not have a target service, and will therefore not use the *in* action, such as a ship being returned to its owner or heading for repairs.

Ships must *sail* between ports within their minimum and maximum speed, incurring a cost that varies linearly with the speed. In order for a ship to replace another ship on a service, they must transship their cargo. Transshipments in our model are instantaneous, free actions that require both ships to simultaneously be at a port where it is lawful for the ships to perform a *transshipment*. Transshipments are only allowed at ports in which a ship is lawfully able to transfer cargo, which we are able to represent by simply not including transshipment actions between ships at ports where it would be unlawful.

Implicit in our model are wait actions. Such an action does not need to be explicitly given, because actions, even those that are ordered or linked, are not required to start directly after one another. The objective of this implicit action is given by the hotel cost computed in the *out* and *in* actions.

Figure 2 shows a fleet repositioning partial plan in the LTOP framework in which two ships are moved off of their initial services (*out* action), and meet at port $f$ where they transship cargo (*transship* action). Ship $v_1$ then begins service at port $f$ (*in* action), freeing $v_2$ to continue to a different service or undergo maintenance. Notice how the objective for the hotel cost for $v_1$, represented by $h_{b,v_1}$ and $h_{e,v_1}$, is computed in the $in(v_1, f)$ action, but the bounds for the hotel cost are updated throughout the partial plan. This allows the LP to compute a more accurate bound throughout the planning process. Multiple actions are therefore contributing to the hotel cost, meaning that the interaction of actions can have interesting cost consequences.

Even this simple version of the fleet repositioning problem is not solvable with existing scheduling or planning approaches like ZENO (Penberthy and Weld 1995) due to the lack of general optimization criteria or Sapa (Do and Kambhampati 2003) due to the lack of continuous time. But even if it was, an advantage of TOP from an OR-perspective is that it is easy to use any optimization model within the framework without changing the representation of activities.

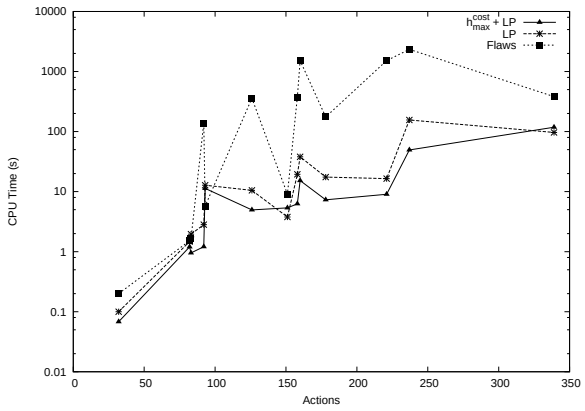


Figure 3: A plot of the performance of LTOP with the planning heuristics $h^{cost}_{max} + LP$ (*solid line*), $LP$ (*dashed line*) and the number of flaws, $Flaws$ (*dotted line*).

## Experimental Evaluation

We created a dataset of sample problems based on discussions with a liner shipping company. The instances range in size from 4 to 12 ports with 2 or 3 services over time frames of 2 to 3 weeks.

Table 1 and Figure 3 display the performance of our LTOP solver using several plan selection heuristics on our dataset. Results were computed on dual six-core 2 GHz AMD Opteron 2425 HE processors, and each execution was allowed a maximum of 4 GB of RAM. In addition, our LTOP solver uses the linear programming solver in COIN-OR 1.5.0 (Lougee-Heimer 2003). The number of actions in the optimal plan varies with the number of ships being repositioned. Instance CR1's optimal plan has five actions, while CR13 has 8 actions.

The $h^{cost}_{max} + LP$ plan selection heuristic selects the cheapest plan available using the sum of the real cost of a plan and the estimated cost of the plan's completion, the $LP$ heuristic only uses the real cost of the plan, and the $Flaws$ heuristic selects the plan with the lowest number of flaws. The $LP + h^{cost}_{max}$ heuristic performs the best, taking on average 61% of the time of the $LP$ heuristic, and only 3.3% of the time of the $Flaws$ heuristic. The geometric mean of $h^{cost}_{max} + LP$ is over 12 times as fast as $Flaws$ and almost twice as fast as $LP$, indicating that $h^{cost}_{max} + LP$ performs well across the entire dataset, and not just on a few instances. Thus, the superior search guidance provided by the $h^{cost}_{max}$ heuristic

| Instance | Actions | $h^{cost}_{max} + LP$ | $LP$ | $Flaws$ |
|---|---|---|---|---|
| CR1 | 32 | **0.07** | 0.10 | 0.20 |
| CR2 | 82 | **1.20** | 1.47 | 1.52 |
| CR3 | 83 | **0.95** | 1.96 | 1.64 |
| CR4 | 92 | **1.21** | 2.82 | 138.53 |
| CR5 | 93 | 11.32 | 12.80 | **5.74** |
| CR6 | 126 | **4.96** | 10.50 | 358.81 |
| CR7 | 151 | 5.36 | **3.78** | 9.06 |
| CR8 | 158 | **6.27** | 19.40 | 376.84 |
| CR9 | 160 | **15.25** | 38.00 | 1,519.92 |
| CR10 | 178 | **7.30** | 17.44 | 174.89 |
| CR11 | 221 | **9.02** | 16.46 | 1,519.92 |
| CR12 | 237 | **49.45** | 156.02 | 2,339.51 |
| CR13 | 339 | 118.20 | **96.67** | 382.12 |
| Mean | | **17.74** | 29.03 | 525.28 |
| Geo. Mean | | **4.94** | 8.55 | 61.35 |
| Std. Dev. | | 32.81 | 46.03 | 762.30 |

Table 1: Results from our LTOP solver on a crafted dataset for our sample fleet repositioning problem with several plan selection heuristics. All times are CPU times given in seconds.

is worth the extra computation time. The TOP framework is therefore able to scale to solve real world sized problems with the $LP$ and $h^{cost}_{max}$ heuristics.

## Conclusion

We presented a novel framework called Temporal Optimization Planning (TOP) for modeling and solving fleet repositioning problems with compound objectives spread throughout interconnected activities. We introduced an extension to the domain independent $h_{max}$ heuristic, $h^{cost}_{max}$, and showed that by using this heuristic to estimate the costs of actions required to complete a plan, the TOP framework is capable of scaling to the size of real fleet repositioning problems.

We gave a model of a fleet repositioning problem that represents the first step to understanding and solving problems in the fleet repositioning domain. The model captured key aspects of fleet repositioning that cannot be modeled with current methods, such as hotel cost, variable sailing durations and duration linked costs.

The TOP framework shows promise as a new method for tackling many industrial cost minimization problems that are difficult to solve using state-of-the-art AI or OR methods. We intend to further investigate TOP on more realistic versions of the fleet repositioning problem and related problems, including problems that are non-linear and problems that do not have non-negativity restrictions on the optimization models. In addition, we will investigate forward-chaining methods for TOP, similar to those in (Coles et al. 2010). Finally, we also intend to see if it is possible to implement TOP within a mixed integer programming framework.

## Acknowledgements

We would like to thank our industrial collaborators Mikkel Muhldorff Sigurd and Shaun Long at Maersk Line for their support and detailed description of the fleet repositioning problem. This research is sponsered in part by the Danish Council for Strategic Research as part of the ENERPLAN research project.

## References

Bylander, T. 1997. A linear programming heuristic for optimal planning. In *Proceedings of the National Conference on Artificial Intelligence*, 694–699.

Coles, A.; Coles, A.; Fox, M.; and Long, D. 2009. Temporal planning in domains with linear processes. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI)*.

Coles, A. J.; Coles, A. I.; Fox, M.; and Long, D. 2010. Forward-chaining partial-order planning. In *Proceedings of the Twentieth International Conference on Automated Planning and Scheduling (ICAPS-10)*.

Do, M., and Kambhampati, S. 2003. Sapa: A multi-objective metric temporal planner. *Journal of Artificial Intelligence Research* 20(1):155–194.

Fikes, R., and Nilsson, N. 1971. STRIPS: A new approach to the application of theorem proving to problem solving. *Artificial intelligence* 2(3-4):189–208.

Fox, M., and Long, D. 2006. Modelling mixed discrete-continuous domains for planning. *Journal of Artificial Intelligence Research* 27(1):235–297.

Frank, J.; Gross, M.; and Kurklu, E. 2004. SOFIA's choice: an AI approach to scheduling airborne astronomy observations. In *Proceedings of the National Conference on Artificial Intelligence*, 828–835. Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999.

Ghallab, M., and Laruelle, H. 1994. Representation and control in IxTeT, a temporal planner. In *Proceedings of AIPS*, volume 94, 61–67.

Haslum, P., and Geffner, H. 2000. Admissible heuristics for optimal planning. In *Proc. AIPS*, 140–149. Citeseer.

Helmert, M.; Do, M.; and Refanidis, I. 2008. The sixth international planning competition, deterministic track. http://ipc.informatik.uni-freiburg.de

Karger, D.; Stein, C.; and Wein, J. 1997. Scheduling algorithms. *CRC Handbook of Computer Science*.

Kautz, H., and Walser, J. 1999. State-space planning by integer optimization. In *Proceedings of the National Conference on Artificial Intelligence*, 526–533.

Li, H., and Williams, B. 2008. Generative planning for hybrid systems based on flow tubes. In *Proc. 18th Intl. Conf. on Automated Planning and Scheduling (ICAPS)*.

Lougee-Heimer, R. 2003. The Common Optimization INterface for Operations Research: Promoting open-source software in the operations research community. *IBM J. Res. Dev.* 47:57–66.

Muscettola, N. 1993. HSTS: Integrating planning and scheduling. In Zweben, M., and Fox, M., eds., *Intelligent Scheduling*. Morgan Kaufmann. 169–212.

Penberthy, J., and Weld, D. 1992. UCPOP: A sound, complete, partial order planner for ADL. In *Proceedings of the Third International Conference on Knowledge Representation and Reasoning*, 103–114.

Penberthy, J., and Weld, D. 1995. Temporal planning with continuous change. In *Proceedings of the national conference on Artificial Intelligence*, 1010–1015.

Sandewall, E., and Rönnquist, R. 1986. A Representation of Action Structures. In *Proceedings of 5th (US) National Conference on Artificial Intelligence*, 89–97. American Association for Artificial Intelligence, Morgan Kaufmann.

Shin, J., and Davis, E. 2005. Processes and continuous change in a SAT-based planner. *Artificial Intelligence* 166(1-2):194–253.

Smith, D.; Frank, J.; and Jónsson, A. 2000. Bridging the gap between planning and scheduling. *The Knowledge Engineering Review* 15(1):47–83.

Smith, S. 2005. Is scheduling a solved problem? *Multidisciplinary Scheduling: Theory and Applications* 3–17.

Van Den Briel, M.; Benton, J.; Kambhampati, S.; and Vossen, T. 2007. An LP-based heuristic for optimal planning. In *Principles and Practice of Constraint Programming–CP 2007*, 651–665. Springer.

Van Den Briel, M.; Vossen, T.; and Kambhampati, S. 2005. Reviving integer programming approaches for AI planning: A branch-and-cut framework. In *Proceedings of the Fifteenth International Conference on Automated Planning and Scheduling (ICAPS 2005)*, 310–319.

Williamson, M., and Hanks, S. 1996. Flaw selection strategies for value-directed planning. In *Proceedings of the Third International Conference on Artificial Intelligence Planning Systems*, volume 23, 244.