Dominance in Pricing Problems with Stochasticity

N. Boland$^1$, S. Dickson$^2$, M. Savelsbergh$^1$, and K. Smilowitz$^3$

$^1$Georgia Institute of Technology
$^2$University of Melbourne and JetStar
$^3$Northwestern University

June 3, 2015

Abstract

Sequencing activities over time is a fundamental optimization problem. The problem can be modeled using a directed network in which activities are represented by nodes and pairs of activities that can be performed consecutively are represented by arcs. A sequence of activities then corresponds to a path in the directed network, and an optimal sequence of activities, assuming appropriately chosen costs on nodes and/or arcs, can then be determined by finding a shortest path in the directed network. In the presence of uncertainty, a stochastic variant of shortest path problem has to be solved, which is far more complex, particularly when the cost of a path depends on a probability distribution function at each node along the path. Calculating a path cost is more time consuming and the dominance between partial paths, which is at the heart of any shortest path algorithm, is no longer clear-cut. In this paper, we identify conditions to establish novel path dominance criteria that can be used to increase the efficiency of label setting algorithms for such problems. We show, using two activity sequencing applications, that these conditions are often satisfied naturally.

1 Introduction

Sequencing activities over time is a fundamental optimization problem. The problem can be modeled using a directed network in which activities are represented by nodes and pairs of activities that can be performed consecutively are represented by arcs. A sequence of activities then corresponds to a path in the directed network. An optimal sequence of activities, assuming appropriately chosen costs on nodes and/or arcs, can then be determined by finding a shortest path in the directed network.

In many environments, the situation is more complex because the activities consume or require one or more limited resources. In such environments, one needs to find a resource constrained shortest path. Unfortunately, whereas the shortest path can easily be solved
(Dijkstra, 1959), the (elementary) resource constrained shortest path is NP-hard (Dror, 1994). The resource constrained shortest path problem has been extensively studied and highly efficient approaches for its solution exist; see Irnich and Desaulniers (2005), Irnich (2008), and Pugliese and Guerriero (2013) for overviews.

In the presence of uncertainty, a stochastic variant of shortest path problem has to be solved, which is also far more complex. Stochastic variants of shortest path problems have received far less attention, even though they arise frequently in practical settings; see Sigal et al. (1980) and Bertsekas and Tsitsiklis (1991) for early work on stochastic variants of the shortest path problem.

We focus on settings in which a probability distribution function at each node of a path is required to calculate the cost of the path. This has an important implication: not only is calculating the cost or benefit of a given path more time consuming, but the dominance between partial paths, which is at the heart of any shortest path algorithm, is no longer clear-cut. Especially the latter, i.e., that dominance between partial paths is no longer clear-cut, may significantly, and negatively, affect the solution method efficiency.

Fortunately, under certain conditions, dominance between partial paths can still be established, which is the main result and contribution of this paper. Importantly, these conditions tend to be naturally satisfied in many applications.

The need to optimally sequence activities over time arises frequently as the pricing problem in column generation formulations for scheduling and routing problems. The usefulness of column generation formulations often increases when a problem setting has uncertainty, because it may be possible to move the complexity associated with calculating the objective into the pricing problem. This is the case, for example, in the two problem settings, quite different in nature, that motivated our investigation of dominance in pricing problems with stochasticity, and which will be used to demonstrate the applicability of our main results.

The remainder of the paper is organized as follows. In Section 2, we discuss the motivating problem settings. In Section 3, we briefly review label setting algorithms. In Section 4, we show how to recognize dominance among partial paths in the presence of stochasticity. Finally, in Section 5, we show that the conditions for stochastic dominance are satisfied in the motivating problem settings.

2 Problem settings

Below, we present two problem settings that involve the scheduling of activities in resource-constrained networks where a probability distribution function at each node of a path is required to calculate the cost of the schedule. Specifically, we consider a medical setting in which disease progression depends on the no-show probabilities associated with scheduled appointments, and an airline setting in which propagated delay of an aircraft depends on the delay probabilities associated with the connections in the aircraft’s schedule. In addition, to highlight the broad applicability of our ideas, we point to other settings where this situation arises naturally.
2.1 Patient appointment scheduling with stochastic disease progression

Savelsbergh and Smilowitz (2015) consider a patient scheduling problem arising in chronic disease management which explicitly models uncertainty in disease progression. Chronic disease management programs provide recurrent preventive services to manage disease progression, as disease control typically decreases as the time since the last service increases. This also implies that a patient’s failure to show up for an appointment will have a detrimental impact on the patient’s health. In the motivating chronic disease management program studied by Savelsbergh and Smilowitz (2015), a patient’s no-show probability is time-dependent (e.g., early morning and lunchtime appointments have greater schedule adherence) and there are limited opportunities to reschedule a missed appointment as the appointments are scheduled months in advance.

The Patient Appointment Scheduling (PAS) problem seeks to determine patient appointment schedules that maximize the aggregate health status of all patients over a planning horizon given limited physician time and recognizing the uncertainty caused by no-shows.

2.1.1 Patient disease progression

We focus on a special case with two control states (controlled and uncontrolled) and “perfect repair” (a patient returns to the controlled state after a visit, regardless of the state diagnosed at the visit). Disease progression is then characterized by the time since the last visit. Let $\alpha$ represent the probability that a controlled patient remains in the controlled health state during a period (a unit time length). Then the probability that a patient is in the controlled health state $\delta$ periods after his last visit is $\alpha^\delta$ and the probability that a patient is in the uncontrolled health state is $1 - \alpha^\delta$. The quality of a patient schedule (in terms of the patient being in a controlled state) thus depends on the realization of the no-show probabilities.

To calculate the expected time since the last visit, $l$, the probability density function (pdf) of all possible values of $l$ is needed. Given a planning horizon of $K$ periods, each with $T$ time slots in which a patient can be seen, and assuming that all patients are seen in period 0, there are $K + 1$ possible values. Let $p_{kl}$ denote the probability that the time since the last visit is $l$ immediately after an appointment in time slot $t$ in period $k$, i.e.,

$$
p_{kl} = \begin{cases} 
1 - n_t & \text{if } l = 0 \\
n_t p_{k-1,l-1} & \text{otherwise},
\end{cases} 
$$

where $n_t$ is the no-show probability for time slot $t$. Given that the patient is seen in period 0, we have $p_{00} = 1$ and $p_{0l} = 0$, for all $l > 0$.

Given a pdf $p_{k-1}$ immediately after a patient’s appointment in period $k - 1$, the pdf $p_k$ immediately after the patient’s appointment in time slot $t$ in period $k$ is $p_{k-1} Q^{kt}$ with
transition probability matrix \( Q^{kt} \) equal to

\[
\begin{pmatrix}
0 & 1 & 2 & \cdots & k + 1 & k + 2 & \cdots & K \\
0 & 1 - n_t & n_t & 0 & \cdots & 0 & 0 & \cdots & 0 \\
1 & 1 - n_t & 0 & n_t & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \ddots & \ddots & \ddots & \vdots & \vdots & \cdots & \vdots \\
k & 1 - n_t & 0 & 0 & \ddots & n_t & 0 & \cdots & 0 \\
k + 1 & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \ddots & \vdots & \vdots & \cdots & \vdots \\
K & 0 & 0 & 0 & \cdots & 0 & 0 & \cdots & 0 \\
\end{pmatrix}
\]

(2)

### 2.1.2 Patient appointment scheduling model formulation

The PAS problem is defined for a planning horizon of \( K \) periods, each with \( T \) time slots, and a population of \( P \) homogeneous patients. For ease of notation, we assume that periods are equally spaced in time. The appointment scheduling environment can be represented by means of a layered network. Each layer represents a period and each node within a layer represents a time slot, i.e., node \((k, t)\) represents time slot \( t \) in period \( k \), including an artificial appointment time slot \( T + 1 \) representing no appointment in that period. With this node for the artificial time slot (as well as the necessary arcs), a patient appointment schedule is represented by a path of exactly \( K + 1 \) arcs, each connecting one period to the next. We use disease control as an indicator of the quality of a schedule. With perfect schedule adherence, we measure the disease control between two consecutive real appointments \( \Delta \) periods apart using the following quantity:

\[
\sum_{\delta=1}^{\Delta} (1 - \alpha^\delta).
\]

(3)

We refer to this quantity as the aggregate probability that a patient is in an uncontrolled state (realizing that the value does not represent an actual probability). When patient schedule adherence is not perfect, computing the expected aggregate probability that a patient is in the uncontrolled state during the planning horizon becomes more involved, as will be seen in Section 2.1.3. The objective of the PAS problem is to minimize the sum over all patients of the expected aggregate probabilities that the patient is in an uncontrolled state.

The PAS problem can naturally be modeled with a column generation formulation. The stochastic objective function also motivates a column generation approach, because the complexity of calculating the objective can be moved to the subproblem. Let the set of all possible patient appointment schedules be denoted by \( \mathcal{R} \). Furthermore, let \( a_{kt}^r \) for \( k \in \{1, \ldots, K\} \), \( t \in \{1, \ldots, T + 1\} \), and \( r \in \mathcal{R} \) indicate whether or not a patient is scheduled in time slot \( t \) in period \( k \) in schedule \( r \) (\( a_{kt}^r = 1 \)) or not (\( a_{kt}^r = 0 \)), and let \( c_i^r \) for \( i \in \{1, \ldots, P\} \) and \( r \in \mathcal{R} \) denote the expected aggregate probability of patient \( i \) being in an uncontrolled state over the planning horizon when schedule \( r \) is assigned to the patient. Finally, let \( x_i^r \) for \( i \in \{1, \ldots, P\} \) and \( r \in \mathcal{R} \) be a binary variable representing whether or not schedule \( r \) is assigned to patient \( i \) (\( x_i^r = 1 \)) or not (\( x_i^r = 0 \)). The capacity of time slot \( T + 1 \) is \( C_{T+1} = P \); the capacity of all
other time slots is 1. The optimization model is defined as

\[
(PAS) \quad \min \sum_{r \in R} c_i^r x_i^r
\]

s.t.

\[
\sum_{r \in R} x_i^r \geq 1 \quad i \in \{1, ..., P\} \quad (4)
\]

\[
\sum_{r \in R} a_{k,t}^r x_i^r \leq C_t \quad k \in \{1, ..., K\}, t \in \{1, ..., T + 1\} \quad (5)
\]

\[
x_i^r \in \{0, 1\} \quad r \in R, i \in \{1, ..., P\}.
\]

Constraints (4) ensure that at least one appointment schedule is chosen for each patient. (Note that because \(c_i^r > 0\) for all \(i \in \{1, ..., P\}\) and \(r \in R\), we chose to use \(\geq\) rather than =, as it has computational advantages.) Constraints (5) ensure that the capacity of the time slots in a period is respected.

Rather than enumerating all possible patient appointment schedules upfront, we use column generation to solve the linear programming relaxation of PAS and iteratively add new appointment schedules to a restricted master problem.

2.1.3 Patient scheduling pricing problem

Let \(\sigma_i\) denote the dual variable associated with constraint (4) for patient \(i\) and let \(\pi_{kt}\) denote the dual variable associated with constraint (5) for period \(k\) and time slot \(t\). The reduced cost of an appointment schedule for patient \(i\) is given by the expected aggregate probability of being in an uncontrolled state of that schedule plus the sum of the dual values associated with the time slots in that schedule and the dual value associated with the constraint that ensures at least one schedule is selected for the patient. This is equivalent to the value of the corresponding path in the layered network plus the sum of the dual values associated with the nodes visited on that path and the (path-independent) dual value associated with the constraint that ensures at least one schedule is selected for the patient. Determining whether an appointment schedule with negative reduced cost exists can be done by finding a least reduced cost path problem in the layered network.

Let \(p_0 = (1, 0, ..., 0)\) be the initial pdf, and let \(w = (1 - \alpha, 1 - \alpha^2, ..., 1 - \alpha^{K+1})\) be the “cost vector” representing the cost associated with each time-since-last-visit state. The pdf at node \((K, t_K)\), if it was reached via path \(((1, t_1), (2, t_2), ..., (K, t_K))\), is calculated by multiplying the transition matrices associated with the arcs in the path:

\[
p_K = p_0 Q^{t_1} Q^{t_2} \cdots Q^{t_K}.
\]

The total cost of the path \(((1, t_1), (2, t_2), ..., (K, t_K))\) for patient \(i\) is given by:

\[
\bar{c}_{Kt_K} = \sum_{k=1}^{K} (p_k w + \pi_{kt_k}) + \sigma_i,
\]
where $p_k$ is the pdf at node $(k, t_k)$ if node $(k, t_k)$ is reached via the path $((1, t_1), (2, t_2), \ldots, (k, t_k))$. Observe that $p_k = p_{k-1}Q^{kt_k}$ and $\bar{c}_k = \bar{c}_{k-1} + (p_kw + \pi_{kt_k})$ for $k = 1, \ldots, K$, where $\bar{c}_0 = \sigma_i$.

Finding such a minimum cost path is not a traditional shortest path problem, because the minimum cost of a path to node $j$ does not only depend on the cost of a minimum path to node $j-1$ and the connection $(j-1, j)$, but also on the path taken to reach $j$. However, it can be cast as a constrained shortest path problem, and we will show that it is possible to use a label setting algorithm to identify a minimum cost patient appointment schedule without having to generate labels corresponding to all possible paths. The network being searched is an acyclic, directed graph, and, as discussed in Section 4, one can employ a dominance criterion that allows some paths to be pruned.

2.2 Aircraft routing with stochastic delay propagation

Dickson (2013) considers flight re-timing and aircraft re-routing as a mechanism to improve the robustness of aircraft routes by ensuring larger turn times at airports where an aircraft is more likely to experience delays. In the Flight Re-timing and Re-routing (FRR) problem, an airline schedule, together with associated aircraft routes, designed to deliver optimal revenue and cost outcomes, are taken as given. The FRR problem is to identify slight modifications to flight departure times and revised aircraft routes so as to improve the robustness of the resulting schedule and routes. A small number of copies of each flight, each having slightly different departure time, are created, with the premise that by only allowing small changes, e.g., re-timing within 15 minutes of the original scheduled time, the quality of the original schedule will be largely unaffected. Importantly, accounting for delay only between consecutive flights in an aircraft’s route is too simplistic, as delays propagate through the route; its entire history must be considered.

2.2.1 Delay propagation

The robustness of an aircraft route, i.e., a sequence of flights, is measured using the delay probability distributions for each of the flights in the route. Dickson (2013) demonstrates, through analysis of historical airline data, that the delay of a flight in a route is not simply the sum of the propagated delay and an independent flight delay. In fact, the additional delay experienced by a flight depends on how delayed the flight is to begin with; the (further) flight delay is not independent of the delay propagated along the route up to that flight. Thus the FRR explicitly captures dependent delay propagation by introducing delay transition probability matrices associated with connections (pairs of consecutive flights) in a route, as follows.

Let $D$ be a set of delay categories (e.g., no-delay, minor-delay, moderate-delay, and major-delay). An entry $Q_{ij}^{ff'}$ of delay transition matrix $Q^{ff'}$ associated with the connection $(f', f)$ specifies the probability that flight $f$ will be in delay category $j$ if flight $f'$ is in delay category $i$. Thus, given a delay pdf $p^{ff'}$ for flight $f'$ and a delay transition matrix $Q^{ff'}$, the delay pdf
of flight $f$ will be

$$ p^f = p^f Q_{f'}. $$

### 2.2.2 Flight re-timing and re-routing formulation

The FRR problem can naturally be modeled with a column generation formulation. Let a *flight string* be a sequence of flights that can be feasibly operated by one aircraft. The goal, then, is to select a subset of flight strings covering all flights and minimizing the probability of having flights in unfavorable delay categories (or some other robust objective).

Let $L$ be the set of airports, $F$ be the set of flights, $R$ be the set of flight strings, $R^O_l \subseteq R$ be the set of flight strings that originate at airport $l$, and $R^T_l \subseteq R$ be the set of flight strings terminating at airport $l$. Let binary decision variable $x_r$ for all $r \in R$ indicate whether or not flight string $r$ is used in the solution. There are also parameters $c_r$, which represent the expected cost of $r$ (in terms of the probability of having flights in unfavorable delay categories), and $a_{rf}$, which is 1 if some copy of flight $f$ appears in flight string $r$ and 0 otherwise. The formulation can now be written as:

$$ (FRR) \quad \min \sum_{r \in R} c_r x_r \\
\text{s.t.} \quad \sum_{r \in R} a_{rf} x_r = 1 \quad \forall f \in F \quad (8) \\
\sum_{r \in R^O_l} x_r = N^O_l \quad \forall l \in L \quad (9) \\
\sum_{r \in R^T_l} x_r = N^T_l \quad \forall l \in L \quad (10) \\
x_r \in \{0, 1\} \quad \forall r \in R,$$

where $N^O_l$ and $N^T_l$ represent the number of aircraft originating at airport $l \in L$ and the number of aircraft whose final destination is airport $l \in L$, respectively. Constraints (8) ensure that each flight is covered exactly once. Constraints (9) ensure that there are only as many flight strings selected from each starting airport as there are available aircraft, and constraints (10) ensure that the correct number of aircraft finish at each airport.

As it not practical to enumerate all flight strings prior to solving FRR, flight string variables are dynamically generated using column generation.

### 2.2.3 Flight string pricing problem

Let $\pi_f$ be the dual variable associated with the cover constraint (8) for flight $f$, and let $\phi_l$ and $\delta_l$ be the dual variables associated with the aircraft count constraints (9) and (10), respectively. The reduced cost of a flight string $r$ originating at airport $o_r$ and terminating
at airport $t_r$ is:

$$
c_r = c_r - \sum_{f \in F_r} a_{fr}\pi_f - \phi_{or} - \delta_{tr} \tag{11}
$$

where $c_r$ is calculated as

$$
c_r = \sum_{f \in F_r} \left( \sum_{d \in D} w_{fd}p_{fd} \right) \tag{12}
$$

and where $F_r$ is the set of flights covered by flightstring $r$, and $w_{fd}$ represents the cost of having flight $f$ in delay category $d$, while $p_{fd}$ represents the probability that flight $f$ has delay in category $d$. Note that the probability $p_{fd}$ that flight $f$ has delay in category $d$ depends on flight string $r$. The goal of the pricing problem is to find a flight string with the minimum reduced cost.

The pricing problem can be cast as a constrained shortest path problem in a connection network. This is achieved by allocating each component of the reduced cost for the string to an arc so that the reduced cost for the entire string is calculated by summing these components along the string. More specifically, consider the following network. Each flight-copy is represented by a node, and an arc $(i, j)$ exists between two nodes if an aircraft can feasibly operate flight-copy $i$ and then flight-copy $j$. Dummy nodes are created for each possible originating airport ($O_l$) and each possible terminating airport ($T_l$) for an aircraft. An arc $(O_l, i)$ exists if the departure airport of flight-copy $i$ is $l$, similarly an arc $(i, T_l)$ exists if the arrival airport of flight-copy $i$ is $l$. Denote the set of nodes relating to flight-copies $N_F$, the set of nodes relating to originating airports $N_O$ and the set of nodes relating to terminating airports $N_T$. The set of all nodes is denoted $N$. Note that $N_F$, $N_O$ and $N_T$ are mutually exclusive sets and $N = N_F \cup N_O \cup N_T$. Let $flight(i)$ refer to the flight associated with node $i$ for all $i$ in $N_F$, and let $airport(i)$ refer to the airport that dummy node $i$ represents for all $i$ in $N_O \cup N_T$. Associate with each node a cost $\varsigma_i$, where

$$
\varsigma_i = -\pi_i - \phi_i - \delta_i \quad \forall i \in N.
$$

The values for $\pi_i$, $\phi_i$ and $\delta_i$ are related to the dual variables as follows:

$$
\pi_i = \begin{cases} 
\pi_{flight(i)}, & \text{if } i \in N_F, \\
0, & \text{otherwise}
\end{cases},
$$

$$
\phi_i = \begin{cases} 
\phi_{airport(i)}, & \text{if } i \in N_O, \\
0, & \text{otherwise}
\end{cases}, \text{ and}
$$

$$
\delta_i = \begin{cases} 
\delta_{airport(i)}, & \text{if } i \in N_T, \\
0, & \text{otherwise}
\end{cases}.
$$

Consider also pdf $p_0 = (1, 0, ..., 0)$, where the first state represents delay category “no delay”, and a cost vector $w_k$ representing the cost of having delay in each delay category for the flight represented by node $k$. Note that $w_k$ is zero for all dummy nodes $k$ in $N_O \cup N_T$. Then
the pdf at node \( j \) if it was reached via path \((i_0, i_1, \ldots, i_m = j)\) is calculated by multiplying the delay transition matrices along the arcs in the path:

\[
p_j = p_0 Q^{i_0,i_1} Q^{i_1,i_2} \cdots Q^{i_{m-1},i_m}.
\]  

(13)

The total cost of the path \((i_0, i_1, \ldots, i_m)\) is given by:

\[
\bar{c}_j = \sum_{l=0,\ldots,m} (p_{i_l} w_{i_l} + \varsigma_{i_l}),
\]

(14)

where \( p_{i_l} \) is the pdf at node \( i_l \) if node \( i_l \) is reached via the path \((i_0, i_1, \ldots, i_l)\). Observe that \( p_{i_l} = p_{i_{l-1}} Q^{i_{l-1},i_l} \) and \( \bar{c}_{i_l} = \bar{c}_{i_{l-1}} + (p_{i_l} w_{i_l} + \varsigma_{i_l}) \) for \( l = 1, \ldots, m \) (with \( p_{i_0} = p_0 \)).

It is easy to verify that a path through this connection network from a dummy node representing an originating airport to a dummy node representing a terminating airport with minimum cost is a solution to the pricing problem. As with the PAS, we will show that a label setting algorithm can be used to identify a minimum cost flight string without having to generate labels corresponding to all possible paths. The connection network being searched is an acyclic, directed graph, and, as discussed in Section 4, we can employ a dominance criterion that allows some paths to be pruned.

### 2.3 Other settings

Other natural settings include the allocation of buffer times in liner shipping networks. Mulder et al. (2012) use transition probability matrices for each ship position in the network, and each possible choice of buffer time, to model how the distribution of (discrete) delay states at that position evolve under that buffer time choice. Then a Markov decision process model is formulated and solved, to find buffer time choice policies that minimize expected cost of delays. The transition matrices and buffer choice settings considered by Mulder et al. (2012) could readily be used in a finite time horizon setting, instead, which would allow transition probabilities to be conditioned on current weather forecast or port congestion information, for example. In this case, or in extensions that consider column generation (or Lagrangian relaxation) approaches to robust ship route design, the stochastic shortest path model we discuss here would apply.

The stochastic shortest path models we address here also arise in vehicle routing problems with stochastic travel and/or service times (e.g., Laporte et al. (1992), Kenyon and Morton (2003)). Vehicle routing problems are often tackled with column generation methods. If the stochastic times are taken to follow a discrete distribution, then, depending on the how the constraints or objectives depend on the probability distributions of delays on a route, column generation subproblems may be cast in the form we study here. This may allow the development of exact approaches for vehicle routing problems with stochastic travel and/or service times, to complement the (meta)heuristics that are prevalent (e.g., Lei et al. (2012), Yan et al. (2014)).
3 Labeling algorithms

As mentioned in the introduction, resource constrained shortest path problems arise naturally and frequently in the pricing problems associated with column generation formulations of scheduling and routing problems. For an introduction to and overview of column generation techniques for solving integer programming problems, see Barnhart et al. (1998) and Desaulniers et al. (2006). Resources typically model quantities such as the time available until a must crew return to their home base, or the number of take-offs and landings an aircraft may perform before their next maintenance check. Algorithms for solving resource constrained shortest path problems typically store *labels* at each node, where each label corresponds to a path to the node, and consists of a pair \( (c, v) \), with \( c \) giving the cost of the path and \( v \) the vector of resources consumed on the path. A key operation in algorithms is to *extend* a path to node \( i \) along the arc \((i, j)\), making use of an *extension function*, \( \zeta_{ij}(c, v) \), to compute the label on \( j \) corresponding to the extended path. Comprehensive reviews of these concepts, and algorithms for resource constrained shortest path problems, are provided by Irnich and Desaulniers (2005), and Irnich (2008). In the applications we discuss here, the graphs are acyclic, which make *label setting* algorithms a natural choice (see for example, Dumitrescu and Boland (2003)).

Critical to the efficiency of labeling algorithms are effective *dominance criteria* to control, as much as possible, the number of labels that are kept at a node. Dominance criteria compare two labels and identify if one dominates another in the sense that any path extending the partial path of the dominant label to the destination node is at least as good as any path to the destination node from the dominated label. The importance of dominance criteria to practical algorithm efficiency cannot be understated. For example, Engineer et al. (2012) provide a highly sophisticated and effective label setting algorithm – for the fixed-charge shortest path problem – that relies heavily on the exploitation of dominance criteria for its success. For the shortest path problem with forbidden paths, Smith and Savelsbergh (2014) give an analysis of the effect of dominance criteria on the number of labels that have to be generated. Compelling evidence is also provided by Hutson and Shier (2009), in the context of minimizing the sum of an increasing convex function and an increasing concave function, where each is a function of the sum, over the arcs in a path, of a given (different) arc cost measure. This is applied to find robust shortest paths in the case that arc lengths are random, using the mean arc length as one cost measure and the arc length variance as the other. The dominance criteria deduced are able to make enormous differences to the solution times, of around four orders of magnitude.

A key insight for the problems we study here, in which a (discrete) probability distribution function at each node of a path is required to calculate the cost of the path, is that each discrete value can be viewed as a resource, with multiplication by transition probability matrices providing the extension functions. In Section 4, we develop novel dominance criteria for such settings.
4 Dominance in stochastic settings

In general, we consider paths in a directed network \((N, A, s, t)\) with node set \(N\), arc set \(A \subset N \times N\) and distinct nodes \(s, t \in N\), defined to be the path start and end nodes respectively. We also take as given a partition of the nodes into (disjoint) ordered sets, \(N = N^1 \cup N^2 \cup \cdots \cup N^K\), with \(N^k = \{i^k_1, \ldots, i^k_{\eta_k}\}\) for each \(k = 1, \ldots, K\), having the property that if \((i^k_h, j) \in A\) for some \(h \in \{2, \ldots, \eta_k\}\), then \((i^k_{h-1}, j) \in A\) also, i.e., \(\Delta^+(i^k_{h-1}) \supseteq \Delta^+(i^k_h)\) for all \(h = 2, \ldots, \eta_k\), where \(\Delta^+(i) \triangleq \{j \in N : (i, j) \in A\}\). Note that one may always take the partition to consist of a set for each node.

We imagine a path in the network as representing the progress of an entity. We are also given an ordered set of possible states \(S = \{1, \ldots, |S|\}\) that the entity can be in at any point along the path, at the time that the path is executed in operation. At the start of the path, we are given an initial probability vector, \(p^0 \in [0, 1]^{|S|}\) with \(1p^0 = 1\), where 1 is the (row) vector of all ones, which for each state \(s \in S\) denotes the probability, \(p_s\), that the entity is initially in that state. In other words, \(p\) defines a discrete probability density function (pdf) over \(S\). For each arc \(a \in A\), there is a given transition matrix, \(Q^a\), that defines how the pdf changes as a result of traversal of the arc. So \(Q^a_{rs}\) gives the probability that the entity is in state \(s\) when it reaches node \(j\), given that it was in state \(r\) at node \(i\) and then traverses arc \(a = (i, j)\). Thus if \(p\) is the pdf on node \(i\), \(pQ^i_j\) gives the pdf on node \(j\). A path \((s = i_0, i_1, \ldots, i_h = t)\) in the network has final pdf given by

\[
p^0Q^{i_0i_1}Q^{i_1i_2}\cdots Q^{i_{h-1}i_h}.
\]

We assume the cost of a path can be calculated via a label on each node of the form \((c, p)\), consisting of a vector of cost components \(c \in \mathbb{R}^R\) and a pdf vector, \(p\), together with an extension function \(\zeta_a\) for each arc \(a = (i, j) \in A\) which transforms a label \((c, p)\) on \(i\) into a label \(\zeta_a(c, p)\) on \(j\), and a final cost function \(F : \mathbb{R}^R \times [0, 1]^{|S|} \to \mathbb{R}\) that calculates the cost of a path from its label on \(t\). In other words, there is a given initial cost vector \(c^0\) so that the cost of a path \((s = i_0, i_1, \ldots, i_h = t)\) can be computed via

\[
(c^t, p^t) = \zeta_{i_{h-1}i_h}(\ldots \zeta_{i_1i_2}(\zeta_{i_0i_1}(c^0, p^0))) \ldots
\]

to calculate the path’s final label, and then evaluate \(F(c^t, p^t)\). The probability vector component of the extension function is assumed to be given by the transition matrix, i.e., \(\zeta_a(c, p) = (d, pQ^a)\) for some \(d\). In such a general setting, one would expect that to find a path in the network that minimizes its cost, i.e., minimizes \(F(c^t, p^t)\) where \((c^t, p^t)\) is its final label, one would have to enumerate all possible labels. However, as we shall see, for the price of some assumptions that are naturally satisfied in motivating applications, we can in fact eliminate labels using dominance relationships.

To enable dominance between labels on the same node, we ask for two key properties. First, we note the definition of stochastic dominance.

**Definition 1** Probability vector \(p\) is said to stochastically dominate probability vector \(\hat{p}\), written \(p \succeq_{st} \hat{p}\), if the cumulative density function for \(p\) lies above and to the left of the cumulative density function for \(\hat{p}\), i.e., if \(\sum_{r=1}^s p_s \geq \sum_{r=1}^s \hat{p}_s\) for all \(s \in S\).
**Property 1** We say the problem has stochastically monotone transitions if all transition matrices $Q^a$ are stochastically monotone, i.e., if row $s$ of $Q^a$ stochastically dominates row $s + 1$ for all $s = 1, \ldots, |S| - 1$, and for all $a \in A$.

**Property 2** We say the problem has a dominance-compatible objective if $F$ and the extension functions $\zeta_a$ for $a \in A$ satisfy

$$F(c, p) \leq F(\hat{c}, \hat{p}) \quad \text{and} \quad d \leq \hat{d}$$

whenever

$$c \leq \hat{c} \quad \text{and} \quad q \geq_{st} \hat{q},$$

where $(d, q) = \zeta_{ij}(c, p)$ and $(\hat{d}, \hat{q}) = \zeta_{i'j}(\hat{c}, \hat{p})$ for some $i, i'$ in the same set of the node partition and some $j \in \Delta^+(i) \cap \Delta^+(i')$.

The following well-known lemma (Conlisk (1990); Keilson and Kester (1977)) is critical to establishing dominance properties between labels on the same node.

**Lemma 1** $Q$ is stochastically monotone if and only if $pQ \geq_{st} \hat{p}Q$ whenever $p \geq_{st} \hat{p}$.

**Lemma 2** If the problem has stochastically monotone transitions and a dominance-compatible objective then

$$F(\zeta_{i_0i_1}(\ldots \zeta_{i_1i_2}(\zeta_{i_0i_1}(c, p))\ldots)) \leq F(\zeta_{i_0i_1}(\ldots \zeta_{i_1i_2}(\zeta_{i_0i_1}(\hat{c}, \hat{p}))\ldots))$$

whenever

$$c \leq \hat{c} \quad \text{and} \quad p \geq_{st} \hat{p},$$

where $(i_0, i_1, \ldots, i_h = \ell)$ is any path in the network from some node to the final node.

**Proof.** Assume $c \leq \hat{c}$ and $p \geq_{st} \hat{p}$. Let $(d^1, q^1) = \zeta_{i_0i_1}(c, p)$ and $(\hat{d}^1, \hat{q}^1) = \zeta_{i_0i_1}(\hat{c}, \hat{p})$. Since the problem has stochastically monotone transitions, and by Lemma 1, we know $q^1 = pQ^1i_1 \geq_{st} \hat{p}Q^1i_1 = \hat{q}^1$. Thus, and since the objective is dominance-compatible, we know $d^1 \leq \hat{d}^1$. By induction, it must be that $d^h \leq \hat{d}^h$ and $q^h \geq_{st} \hat{q}^h$ where

$$(d^h, q^h) = \zeta_{i_{h-1}i_h}(\ldots \zeta_{i_1i_2}(\zeta_{i_0i_1}(c, p))\ldots) \quad \text{and} \quad (\hat{d}^h, \hat{q}^h) = \zeta_{i_{h-1}i_h}(\ldots \zeta_{i_1i_2}(\zeta_{i_0i_1}(\hat{c}, \hat{p}))\ldots).$$

Finally, since the objective is dominance-compatible, we have

$$F(d^h, q^h) \leq F(\hat{d}^h, \hat{q}^h)$$

as required. \(\square\)

To enable dominance between nodes in the same set of the node partition, we ask for one additional property.
**Property 3** We say a node partition is a dominance-compatible partition if for nodes $i_h$ and $i_l$ in the same set of the partition, with $h < l$, we have that each row of $Q^{i_hj}$ stochastically dominates the corresponding row of $Q^{i_lj}$ for all $j \in \Delta^+(i_l)$.

To establish dominance between labels on different nodes in the same set of the partition, we require the following result, which is easily proved.

**Lemma 3** If each row of $Q^1$ stochastically dominates the corresponding row of $Q^2$, then $pQ^1 \succeq_{st} pQ^2$ for all nonnegative (probability) vectors $p$.

It is also convenient to have a definition of dominance between labels on different nodes.

**Definition 2** We say that label $(c, p)$ on node $i$ dominates label $(\hat{c}, \hat{p})$ on node $j$ if no path from $j$ to $t$ can extend $(\hat{c}, \hat{p})$ to a final label with lower objective value than one from $i$ to $t$ extending $(c, p)$. In this case, any labeling algorithm can safely remove the label $(\hat{c}, \hat{p})$ on $j$ from the set of labels stored.

We now give our main result, which establishes the conditions under which a label can safely be eliminated from consideration.

**Theorem 1** If the problem has stochastically monotone transitions, a dominance-compatible objective, and its node partition is also dominance-compatible, then for any pair of labels $(c, p)$ on node $i_h$ and $(\hat{c}, \hat{p})$ on node $i_l$, where $i_h$ and $i_l$ lie in the same set of the node partition and $h \leq l$, if $c \leq \hat{c}$ and $p \succeq_{st} \hat{p}$ then $(c, p)$ on $i_h$ dominates $(\hat{c}, \hat{p})$ on $i_l$.

**Proof.** Let $(i_l = j_0, j_1, \ldots, j_u = t)$ be a path from $i_l$ to $t$ which, out of all such paths, extends $(\hat{c}, \hat{p})$ to a final label with lowest objective value. Observe $j_1 \in \Delta^+(i_l)$. Since $h \leq l$, we have $\Delta^+(i_h) \supseteq \Delta^+(i_l)$, so $j_1 \in \Delta^+(i_h)$ and $(i_h, j_1, \ldots, t)$ is also a path in the network. Now by Lemma 3 and since the partition is dominance-compatible, we have that

$$pQ^{i_hj_1} \succeq_{st} pQ^{i_lj_1}$$

and by Lemma 1 and since the problem has stochastically monotone transitions, we also have

$$pQ^{i_lj_1} \succeq_{st} \hat{p}Q^{i_lj_1}.$$

Thus

$$pQ^{i_hj_1} \succeq_{st} \hat{p}Q^{i_lj_1}.$$ Setting $(d, q) = \zeta_{i_hj_1}(c, p)$ and $(\hat{d}, \hat{q}) = \zeta_{i_lj_1}(\hat{c}, \hat{p})$, we thus have $q = pQ^{i_hj_1} \succeq_{st} \hat{p}Q^{i_lj_1}\hat{q}$. Since the objective is dominance-compatible, we also have $d \leq \hat{d}$. By Proposition 2, it must be

$$F(\zeta_{j_u-1j_u}(\ldots \zeta_{j_1j_2}(d, q) \ldots)) \leq F(\zeta_{j_u-1j_u}(\ldots \zeta_{j_1j_2}(\hat{d}, \hat{q}) \ldots))$$

and hence

$$F(\zeta_{j_u-1j_u}(\ldots \zeta_{j_1j_2}(\zeta_{i_hj_1}(c, p)) \ldots)) \leq F(\zeta_{j_u-1j_u}(\ldots \zeta_{j_1j_2}(\zeta_{i_lj_1}(\hat{c}, \hat{p})) \ldots)).$$

$\square$
In what follows, we apply the above results in settings such as those of our two motivating applications, in which the following problem needs to be solved:

$$\min_{(a_1,\ldots,a_{|P|})\in P} \sum_{i=1}^{|P|} \left( c_{a_i} + \left( (p^0)^T \prod_{j=1}^i Q^{a_j} \right) w_{a_i} \right),$$

where $P$ is the set of all (possibly resource constrained) paths in $(N,A)$ from $s$ to $t$, $p^0$ is a given probability vector, $Q^a$ are given transition probability matrices for all arcs $a$, and $w_a$ is a given nondecreasing vector for all arcs $a$.

Dominance in stochastic settings is also explored by Wellman et al. (1995), who present a priority-first search dominance algorithm to find the optimal path in a transportation network with uncertain, time-dependent edge costs. Their algorithm makes use of stochastic consistency (monotonicity) and stochastic dominance in a dynamic programming approach to solving a shortest path problem. As in the settings we consider, the utility of a node depends on the entire path taken to the node, not just a pairwise comparison of connections into the node. Unfortunately, their work is not widely known in the broader operations research community. We demonstrate the value of the underlying ideas in the context of pricing problems with stochasticity and extend their results so that node partitions and general cost functions can be handled.

The ability to handle general cost functions $w$ can yield objective penalties on different kinds of statistics for the selected path. For example, for the airline application with four delay categories, indicating delays of up to 5, 10, 15 and 20 minutes, setting $w_j = (0,0,1,1)$ for all flights $j$ will result in minimizing the expected number of flights that are more than 15 minutes late, while setting $w_{js}$ to the expected delay minutes for flight $j$ in delay category $s$ will minimize the average flight delay.

In the case that arc lengths are random, with a discrete probability distribution, Hastings and Shier (2013) develop an algebraic approach for computing the distribution of shortest path lengths. This is the distribution observed \textit{a posteriori}, in the sense that the shortest path is found after the arc length random variables are realized. The problem addressed is that of finding, over all realizations of arc lengths, the probability that the shortest path using the realized arc lengths will be of a given length, for each such length possible. Both exact and approximating algorithms are explored.

This differs from our setting in that we seek a path that, \textit{a priori}, minimizes a function of the probability distribution associated with each node on the path. However some of the concepts developed by Hastings and Shier have parallels with those we discuss here. Primary among them is that our observation that the discrete values in the probability distribution can be viewed as resources is analogous to the observation of Hastings and Shier that discrete values can be used as powers in a polynomial function.
5 Problem settings revisited

The two problem settings discussed in Section 2 have extension functions and an objective of the form

$$\zeta_{(i,j)}(c,p) = (c + pQ^j w_j + \gamma_j, pQ^j)$$

and

$$F(c,p) = c, \quad (15)$$

with $c \in \mathbb{R}$, and where $w_j$ and $\gamma_j$ are given. A useful observation is that, provided the vector $w$ is nondecreasing, i.e., $w_1 \leq w_2 \leq \cdots \leq w_{|S|}$, and the transitions are stochastically monotone, this form results in a dominance-compatible objective. This observation follows from a fundamental property of stochastic dominance, details of which can be found in Conlisk (1990) or Keilson and Kester (1977): if $w$ is nondecreasing and $p \geq \mathfrak{p}$ then $wp \leq \mathfrak{w}p$.

This observation applies in the two problem settings, since the states $S$ are ordered so that states with a higher index are “worse” (longer time since last patient visit and longer flight delay, respectively), and this ordering is reflected in the cost vectors $w_j$, which are nondecreasing (disease control decreases with time since last visit and longer delays are less desirable, respectively).

Both problem settings also have dominance-compatible node partitions, as we discuss in Sections 5.1 and 5.2, respectively.

Formally, we establish the following dominance criterion for these problem settings.

**Proposition 1** If the problem has stochastically monotone transitions, a dominance-compatible node partition, and extension functions and objective of the form given in (15), with $w_j$ non-decreasing for all $j \in N$, then for any pair of labels $(c,p)$ on node $i_h$ and $(\hat{c}, \hat{p})$ on node $i_l$, where $i_h$ and $i_l$ lie in the same set of the node partition and $h \leq l$, if $c \leq \hat{c}$ and $p \geq \mathfrak{st} \hat{p}$ then $(c,p)$ on $i_h$ dominates $(\hat{c}, \hat{p})$ on $i_l$.

**Proof.** First, we show that the problem has a dominance-compatible objective. Suppose $c \leq \hat{c}$, $(d,q) = \zeta_{ij}(c,p)$, $(\hat{d}, \hat{q}) = \zeta_{ij}(\hat{c}, \hat{p})$, and $q \geq \mathfrak{st} \hat{q}$ for some $i, i'$ in the same set of the node partition and some $j \in \Delta^+(i) \cap \Delta^+(i')$. Obviously $F(c,p) = c \leq \hat{c} = F(\hat{c}, \hat{p})$.

Furthermore, from (15), we have

$$d = c + pQ^j w_j + \gamma_j = c + qw_j + \gamma_j, \quad \text{and} \quad \hat{d} = \hat{c} + \hat{p}Q^{i'}j w_j + \gamma_j = \hat{c} + \hat{q}w_j + \gamma_j,$$

and the result that $d \leq \hat{d}$ follows trivially from the abovementioned property of stochastic dominance, and since each $w_j$ is nondecreasing. The result now follows by Theorem 1. \square

5.1 Dominance in patient appointment scheduling with stochastic disease progression

In the patient appointment scheduling problem, the state of a patient represents the number of periods elapsed since the patient was last seen, which implies that the number of possible states is $K+1$, since all patients are seen in period 0. The cost vector is $(1 - \alpha, 1 - \alpha^2, \ldots, 1 - \alpha^{K+1})$ independent of the period and the time slot. As the cost vector is nondecreasing, the objective is dominance-compatible.
It is easily seen from (2) that the transition matrices associated with arcs in the layered network are stochastically monotone.

The nodes are partitioned into sets \( N_k \), corresponding to the appointment time slots in a period \( k \), for \( k = 1, \ldots, K \). This gives a partition of the required form, since every appointment time slot in one period may be followed by any appointment time slot in the next, i.e., \( \Delta^+(k, t) = \{(k + 1, 1), \ldots, (k + 1, T), (k + 1, T + 1)\} \) for all \( k = 1, \ldots, K - 1 \). The node partition can be made dominance-compatible by ordering the time slots in each period in increasing order of no-show rate, as opposed to their actual time, so that \( n_1 \leq n_2 \leq \cdots \leq n_T \leq n_{T+1} = 1 \) (if no appointment is scheduled then the patient will definitely not show up). Then for nodes \((k - 1, t)\) and \((k - 1, t')\) in the same set of the node partition, with \( t < t' \), row \( l \leq k \) in \( Q^k t \) is \((1 - n_t, 0, \ldots, 0, n_t, 0, \ldots, 0)\), with zero in all but the first and \((l + 1)\)th entries, which stochastically dominates \((1 - n_{t'}, 0, \ldots, 0, n_{t'}, 0, \ldots, 0)\), the corresponding row in \( Q^k t' \), with zero in all but the same positions, since \( n_{t'} \geq n_t \). Rows \( l > k \) are zero in both matrices.

Thus, the conditions of Theorem 1 are satisfied and dominance can be employed to prune paths in the label-setting algorithm that solves the pricing problem.

When implementing a label setting algorithm, it is possible to exploit the structure of the node partitions for efficiency. For any partial path ending at node \((k, t)\) in period \( k \), there are \( T + 1 \) possible path extensions to the next layer (representing the scheduling decisions for the next time period). Notably, the extension to the node in period \( k + 1 \) associated with the time slot with the lowest no-show probability is always nondominated as it yields the smallest (unadjusted) cost. To see this, rewrite the expected cost at node \((k, t)\) as

\[
c_k = c_{k-1} + (1 - n_t)(1 - \alpha) + n_t \sum_{l=1}^{k} p_{k-1, t-1}(1 - \alpha^{l+1}),
\]

and observe that \((1 - \alpha^{l+1}) > (1 - \alpha)\) for \( l = 1, \ldots, k \), which implies

\[
n_t \sum_{l=1}^{k} p_{k-1, t-1}(1 - \alpha^{l+1}) > n_t \sum_{l=1}^{k} p_{k-1, t-1}(1 - \alpha) = n_t(1 - \alpha) \sum_{l=0}^{k-1} p_{k-1, l-1} = n_t(1 - \alpha),
\]

which, in turn, implies that the expected cost is smallest for the time slot with the lowest no-show probability. Extensions to other nodes in period \( k + 1 \) will be nondominated only if the dual value for the time slot associated with the node leads to a lower adjusted expected cost. Thus, it pays to first extend to the node associated with the time slot with the lowest no-show probability and then to check if any of the extensions to other nodes are nondominated because of their adjusted expected cost, in which case these extensions should be created as well.
5.2 Dominance in aircraft routing with stochastic delay propagation

It is not difficult to argue that the objective of the flight re-timing and re-routing optimization problem is dominance-compatible. The components of the cost vector $w$ are associated with the discrete categories of flight delay, with the delay categories sorted in decreasing order of the delay durations contained within each category. From both a passenger and airline perspective, the shorter the delay the better. Therefore the cost vector will be non-decreasing to ensure that the delay categories with shorter delays are valued more highly than the categories with longer delays.

It is also reasonable to expect that the delay transition functions are stochastically monotone; when an aircraft is already delayed, it is more likely that additional delays will occur. Furthermore, it is difficult for the ground staff and the flight crew to make up any time during a turn around, so things will at least stay the same if they do not get worse. (Analysis of the delay transition functions of one airline confirmed that they are indeed stochastically monotone; Dickson (2013).)

Finally, the sets in the node partition contain the nodes representing the copies of a single flight, in order of their departure time. It is reasonable to assume that if an incoming flight can connect to an outgoing one, then so can any earlier copy of the incoming flight. Thus, the node partition has the required form. It is dominance-compatible since a flight-copy departing later will have a shorter connection time to the next flight, which can only increase the probability of additional delays.

As a consequence, the conditions of Theorem 1 are satisfied and dominance can be employed to prune paths in the label-setting algorithm that solves the pricing problem.

As with PAS, it is possible to exploit the structure of the network in the label setting algorithm. The algorithm begins by creating a path from the source to any node that can be the first node in a path (practically speaking, this relates to the nodes representing flight-copies that depart from an originating port).

We note that Borndörfer et al. (2010) also explore robustness of aircraft rotations, in the form of the Most Robust Rotation Problem, which seeks a path with the smallest probability of delay (either primary or propagated delay) using a label setting algorithm that finds efficient labels for each node using stochastic dominance on discretized probability density functions. However, Borndörfer et al. (2010) assume independence of primary and propagated delay, simplifying the costs in their network into additive costs on the arcs in the network rather than multiplicative costs as is the case in the aircraft routing problem with stochastic delay propagation we have considered. As a consequence, the dominance criteria employed in their algorithm are more straightforward.

References


