WORKING PAPER L-2006-10

WP

Lars Relund Nielsen, Kim Allan Andersen &
Daniele Pretolani

Bicriterion a priori route choice in stochastic time-dependent networks

Logistics/SCM
Research Group

Bicriterion a priori route choice in stochastic time-dependent networks

LARS RELUND NIELSEN*
Research Unit of Statistics and Decision Analysis
DIAS
P.O. Box 50
DK-8830 Tjele
Denmark

KIM ALLAN ANDERSEN†
Department of Business Studies
Aarhus School of Business
Fuglesangs Allé 4
DK-8210 Aarhus V
Denmark

DANIELE PRETOLANI‡
Department of Sciences and Methods of Engineering
University of Modena and Reggio Emilia
Via Amendola 2
I-42100 Reggio Emilia
Italy

Abstract

In recent years there has been a growing interest in using stochastic time-dependent (STD) networks as a modelling tool for a number of applications within such areas as transportation and telecommunications. It is known that an optimal routing policy does not necessarily correspond to a path, but rather to a *time-adaptive strategy*. In some applications, however, it makes good sense to require that the routing policy corresponds to a loopless path in the network, that is, the time-adaptive aspect disappears and a priori route choice is considered.

In this paper we consider bicriterion a priori route choice in STD networks, i.e. the problem of finding the set of efficient paths. Both expectation and min-max criteria are considered and a solution method based on the two-phase approach is devised. Experimental results reveal that the full set of efficient solutions can be determined on rather large test instances, which is in contrast to previously reported results for the time-adaptive case.

Keywords: stochastic time-dependent networks; bicriterion shortest path; a priori route choice; two-phase method.

1 Introduction

Travel time between an origin and a destination is often the primary objective when routing data, commodities, vehicles etc. in a network. The problem of finding a minimal travel time path, if travel time is deterministic and time-independent, has been the subject of extensive research for many years. For an overview see e.g. Deo and Pang [4] or the textbook by Ahuja, Magnanti, and Orlin [1]. However, a transportation network in which travel times between locations are deterministic and time-independent is often

*Corresponding author, e-mail: lars@relund.dk.
†e-mail: kia@asb.dk.
‡e-mail: pretolani.daniele@unimore.it.

unrealistic. For instance, travel time between home and workplace is normally faster at midnight than during rush hour, and even during off-peak hours, travel times may vary substantially.

We say that a network is *time-dependent*, if the travel times on the arcs are functions of time, and *stochastic*, if the travel time is represented by probability distributions rather than simple scalars. It is evident that both the stochastic and time-dependent properties are appropriate in a transportation network model. As a result stochastic time-dependent networks[1] (*STD networks*) often provide a better modelling tool in e.g. transportation applications. These networks were first addressed by Hall [7], who considered the problem of finding a route between two nodes minimizing the expected travel time, when leaving the origin at a specific time. He pointed out several ways to formulate the route selection problem, and complications arising as a consequence of modelling both the stochastic and time-dependent properties.

If the driver is allowed to react to revealed (actual) arrival times at intermediate nodes, the best route is not necessarily a path, but rather a time-adaptive *strategy* that assigns optimal successor arcs to a node as a function of leaving time. This is referred to as *time-adaptive route choice*. Pretolani [14] presented a directed hypergraph model for STD networks with discrete travel time distributions and showed that a strategy corresponds to a hyperpath in a time-expanded hypergraph. Moreover, the best strategy under different criteria, such as minimizing expected or maximum possible travel time or cost, can be found by solving a minimum weight hyperpath problem using appropriate weights and weighting functions.

If a *loopless path* must be specified before travel begins, and no deviations from the route are permitted, the path is selected *a priori* on the basis of only the probability distributions of the arc travel-times. Thus, we seek a strategy that assigns the same successor arc for all leaving times for a specific node. This is referred to as *a priori route choice*, and may be the only possible model in several practical cases, e.g. for routing highly sensitive substances for which the path travelled must be preapproved, or when the driver does not have access to (or time to access) information while travelling. The problem of finding a minimal expected travel time path under a priori route choice is NP-hard [14].

The above problems only consider a single objective. Nevertheless, due to the multi-objective nature of many transportation and routing problems, a single objective function is not sufficient to completely characterize most real-life problems. In a road network for instance, two parameters, travel time and cost, can be assigned to each arc. Clearly, often the fastest path may be too costly or the cheapest path may be too long. Therefore the decision maker must choose a solution among the set of *efficient* (Pareto optimal) paths. The problem of finding all efficient paths, commonly referred to as *bicriterion shortest path* (*bi-SP*) has been widely studied and is known to be NP-hard even if deterministic costs/travel times are used [6].

It is obvious that problems concerning bicriterion route choice in STD networks are relevant. For instance, when routing hazardous materials several criteria may be considered besides expected travel time, namely expected accident risk, population exposure, or travel costs. Risk and exposure (rather than travel time) may be the most relevant criteria, if materials must be routed through urban areas. Note that STD networks may be much more suitable in this case, due to their ability to capture the inherent fluctuations in these parameters. Moreover, the objective of the problem may vary; for example, a risk averse decision maker may be interested in minimizing the maximum risk, rather than its expected value. We remark that bicriterion route choice problems in STD networks show a much richer structure than bi-SP, for at least two reasons:

1. due to the time-dependent nature of the network, travel times turn out to be a quite particular criterion, as opposed to what happens in deterministic networks;

2. the purpose may be to minimize expected as well as maximum possible values.

The number of papers on multicriterion route choice in STD networks are rather limited. Miller-Hooks and Mahmassani [8] consider bicriterion a priori route choice in discrete STD networks, with the objectives being minimizing expected travel time and cost. They assume that the network only contains a single peak period and that the distributions are static after the peak period. A label-correcting procedure is described, which guarantees that all the efficient paths can be obtained. Computational results are presented on a single road network.

[1] Also known as random time-dependent networks, stochastic time-varying networks or stochastic dynamic networks.

Chang, Nozick, and Turnquist [2] consider multicriterion a priori route choice in a continuous time STD network where travel times are normally distributed. They devise a heuristic method based on the first two moments of the distributions, where an approximate stochastic dominance criterion is adopted to compare paths. Computational results are presented on an example network and a single road network.

Time-adaptive route choice has been presented in Nielsen, Andersen, and Pretolani [11], where an exact two-phase method is devised. Computational results are conducted on difficult STD grid networks, and the results indicate that the number of efficient strategies may grow exponentially with the network size. As a result, fast heuristic algorithms finding approximations of the efficient set are developed.

In this paper we consider bicriterion route choice problems in STD networks under a priori route choice. More specifically we consider the problem of finding the set of efficient paths between an origin and a destination node when leaving the origin at time zero. We assume that departure times are integer and that travel times are discrete random variables. The paper differ from previous work in the following aspects:

1. We propose a new algorithm using the two-phase method to determine the set of efficient paths as opposed to the labelling approach proposed by Miller-Hooks and Mahmassani [8].

2. In addition to expected time and cost (a somehow easier case, as we shall see) we address the case of two cost criteria, which allows us to evaluate the effect of uncorrelated and correlated costs; moreover, we consider expected as well as min-max criteria, and we address the issue of possible waiting at intermediate nodes.

3. We do not consider a "steady state" with deterministic travel time at the end of a peak period: throughout the route, times are always stochastic, and several peak periods are encountered. Thus we concentrate on the dynamic features of the STD network. Within this computational setting we perform a reasonably wide computational experience, concentrating on grid networks rather than random graphs.

4. Since our algorithms solve the bicriterion problem exactly on the set of instances addressed here, we are able to compare the efficient sets found under a priori and time-adaptive route choice.

The paper is organized as follows. In Section 2 we give the necessary definitions of STD networks and efficient paths. In Section 3 we give a short description of the two-phase method and describe the procedures we use for its implementation. Computational results are reported in Section 4, and conclusions are given in Section 5. Throughout the paper we illustrate several concepts by means of a running example, which is reported in details in Appendix A.

2 Preliminaries

In this section we present the basic definitions used in this paper. We introduce stochastic time-dependent networks and formally define the concept of a strategy and a path-strategy. Finally, we recall some basic facts from multicriterion analysis. Definitions are illustrated by means of a running example, discussed in details in Appendix A, where we adopt (after a short introduction) the hypergraph representation of the STD network given by Pretolani [14].

2.1 Stochastic time-dependent networks

We consider discrete STD networks, where departure and arrival times are integer, and travel times are independent integer-valued discrete random variables with time-dependent probability density functions.

Let $G = (N, A)$ be a directed graph with node set N and arc set A, referred to as the *topological network*. The forward star of $u \in N$ is $FS(u) = \{(u, v) \in A\}$. Let $o, d \in N$ denote two different nodes which represent the *origin* and the *destination* node in G, respectively.

Assume that departure and arrival times belong to a finite *time horizon*, i.e. a set $H = \{0, 1, ..., t_{\max}\}$. This is done by discretizing the relevant time period into time intervals of length δ, that is, the time horizon H corresponds to the set of time instances $0, \delta, 2\delta, ..., t_{\max}\delta$.

For each arc $(u,v) \in A$ let $L(u,v) \subseteq H$ be the set of possible leaving times from node u along arc (u,v). Moreover, let $L(u)$, $u \neq d$ denote the set of possible leaving times from node u, i.e

$$L(u) = \bigcup_{(u,v) \in FS(u)} L(u,v)$$

and let $L(d)$ denote the set of possible arrival times at node d. For each arc $(u,v) \in A$ and $t \in L(u,v)$, let $X(u,v,t)$ denote the arrival time at node v, when leaving node u at time t along arc (u,v). The arrival time $X(u,v,t)$ is a discrete random variable with density

$$\Pr(X(u,v,t) = t_i) = \theta_{uvt}(t_i), \quad t_i \in I(u,v,t)$$

where

$$I(u,v,t) = \left\{t_1, \ldots, t_{\kappa(u,v,t)}\right\}$$

denotes the set of $\kappa(u,v,t)$ possible arrival times at node v. That is, for each $t_i \in I(u,v,t)$ the probability of arriving at node v at time t_i when leaving node u at time t is $\theta_{uvt}(t_i)$. We assume that travel times are positive, and that the traveller cannot wait at an intermediate node. Some issues related to waiting will be discussed later.

Under time-adaptive route choice the best route is not necessarily a path but rather a strategy S, i.e. a function which provides routing choices for travelling from all nodes and leaving times in its domain towards the destination d. In particular, a traveller leaving node u at time t travels along arc $S(u,t)$. More formally, following [12], the definition of a strategy can be stated as follows

Definition 1 A *strategy* is a function S with domain

$$Dm(S) \subseteq \{(u,t) : u \in N \setminus \{d\}, \ t \in L(u)\}$$

assigning to each pair $(u,t) \in Dm(S)$ a successor arc $(u,v) \in FS(u)$. Furthermore, strategy S must satisfy the following conditions

1. If $(u,t) \in Dm(S)$ and $S(u,t) = (u,v) \Rightarrow t \in L(u,v)$.
2. If $(u,t) \in Dm(S)$ and $S(u,t) = (u,v)$, $v \neq d \Rightarrow (v,t') \in Dm(S), \forall t' \in I(u,v,t)$.

Note that Condition (2) ensures that a traveller following strategy S cannot get stuck in an intermediate node, that is, he/she will arrive at the destination within the time interval. Throughout this paper we consider routing from an origin node o towards a destination node d when leaving node o at time zero. Hence we are interested in the particular case defined below.

Definition 2 An $(o,0)$-*strategy* is a *minimal* strategy S such that $(o,0) \in Dm(S)$. Here, minimality means that no other strategy with domain strictly contained in $Dm(S)$ exists.

For the sake of simplicity, in the remaining part of the paper we use the term strategy to denote a $(o,0)$-strategy. Since we leave the origin at time zero the arrival time to the destination will be equal to the travel time. As a result we consider arrival time and travel time as equivalent in this paper.

Under a priori route choice we must travel along a loopless path in G. That is, we only consider strategies, where the successor arcs for a node are time-independent.

Definition 3 A *path-strategy* is a strategy S satisfying

$$S(u,t) = S(u,t'), \quad \forall (u,t), (u,t') \in Dm(S)$$

It is easy to see that a path-strategy defines a unique o-d loopless path in G. The converse is not necessarily true, that is, an o-d path in G may not correspond to a strategy, since it may be impossible to reach the destination within the time horizon traveling along the path. From now on, we shall consider path-strategies rather than paths; in some cases, we may use the term "path" as synonymous with a path-strategy. We denote by $\mathscr{S}$ the set of all strategies and with $\mathscr{S}_P$ the set of all path-strategies. Clearly, $\mathscr{S}_P \subseteq \mathscr{S}$.

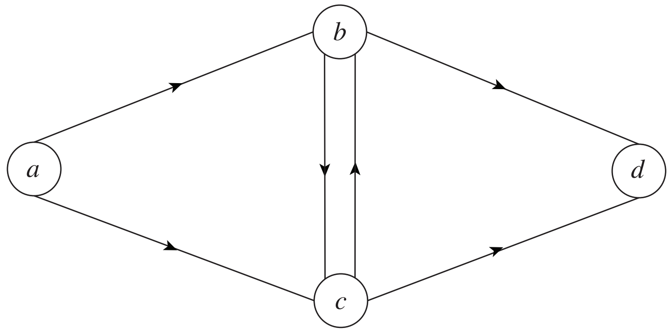


Figure 1: The topological network G.

$(u,v),t$	$(a,b),0$	$(a,c),0$	$(b,c),1$	$(b,c),3$	$(b,d),1$	$(b,d),3$	$(c,b),2$	$(c,d),2$	$(c,d),4$	$(c,d),5$
$I(u,v,t)$	$\{1,3\}$	$\{2\}$	$\{2\}$	$\{4,5\}$	$\{3\}$	$\{6,7\}$	$\{3\}$	$\{3,4\}$	$\{5,6\}$	$\{6\}$
$c(u,v,t)$	$(1,1)$	$(3,0)$	$(2,2)$	$(0,4)$	$(3,8)$	$(3,6)$	$(2,1)$	$(4,2)$	$(3,3)$	$(1,5)$

Table 1: Input parameters.

Example 1 Consider the topological network $G = (N,A)$ in Figure 1, where a is the origin node and d is the destination node. For each arc in G, the possible departure and arrival times are listed in Table 1. Here a pair $((u,v),t)$ corresponds to a possible leaving time t from node u along arc (u,v). For the sake of simplicity, we assume that $X(u,v,t)$ has a uniform density, i.e., for each $t' \in I(u,v,t)$, we have $\theta_{uvt}(t') = 1/|I(i,j,t)|$. For example, if we leave node b at time 3 along arc (b,c), we arrive at node c at time 4 or 5 with the same probability $1/2$. Two possible strategies are

$$S_1 : S_1(a,0) = (a,b),\ S_1(b,1) = (b,d),\ S_1(b,3) = (b,d)$$
$$S_2 : S_2(a,0) = (a,b),\ S_2(b,1) = (b,d),\ S_2(b,3) = (b,c),\ S_2(c,4) = (c,d),\ S_2(c,5) = (c,d)$$

Strategy S_1 is a path-strategy and corresponds to the path $a-b-d$ while for strategy S_2 we travel different routes depending on the leaving time from node b. ■

Costs can be included in the model by letting $c(u,v,t)$, $t \in L(u,v)$ denote the travel cost of leaving node u at time t along arc (u,v). Moreover let $g(t)$ be the penalty cost of arriving at node d at time t. Different criteria described by Pretolani [14] are considered in this paper. Given $S \in \mathscr{S}$, the *expected cost* $E_C^S(u,t)$ of S for each $u \neq d$ and $t \in L(u)$ is defined by the recursive equations:

$$E_C^S(u,t) = c(u,v,t) + \sum_{t' \in I(u,v,t)} \theta_{uvt}(t') E_C^S(v,t'), \tag{1}$$

where $S(u,t) = (u,v)$ and $E_C^S(d,t) = g_d(t)$, for each $t \in H$. $E_C^S(u,t)$ is the expected cost incurred, when leaving node u at time t following strategy S towards d, i.e. the expected cost of strategy S is equal to $E_C^S(o,0)$. Here cost is formulated in general terms, e.g. cost may be a risk measure or the economic travel cost and one objective might be to determine a strategy with minimum expected cost (*MEC*).

Instead of considering expectation criteria worst cases may be of primary concern. That is, finding the strategy minimizing maximum possible cost (*MMC*). Given $S \in \mathscr{S}$ the *maximum cost* $M_C^S(u,t)$ of S for each $u \neq d$ and $t \in L(u)$ is defined by the recursive equations:

$$M_C^S(u,t) = c(u,v,t) + \max_{t' \in I(u,v,t)} M_C^S(v,t') \tag{2}$$

where $M_C^S(d,t) = g(t)$, for each $t \in H$.

Similarly, we can define minimum expected travel time (*MET*) and the problem of minimizing maximum travel time (*MMT*). In fact finding the optimal strategy under the MET (MMT) criterion may be formulated as finding the optimal strategy under the MEC (MMC) criterion by using specific costs and penalty costs [14].

Theorem 1 *Finding the optimal MET (MMT) strategy is equivalent to finding the optimal MEC (MMC) strategy using costs $c(u,v,t)=0$ for all $(u,v)\in A$ and $t\in L(u,v)$ and penalty costs $g(t)=t$ for all $t\in L(d)$.*

We shall generically denote by $W(S)$ the *cost* of a strategy $S\in\mathscr{S}$ with respect to one of the four criteria described above, i.e., MEC, MET, MMC and MMT.

2.2 Bicriterion concepts

Let $W(S)=(W_1(S),W_2(S))$, denote the cost of strategy $S\in\mathscr{S}_{\mathscr{P}}$ using some of the previously described criteria. In this paper we face the following problem:

$$\begin{array}{ll}\min & W(S)=(W_1(S),W_2(S))\\ \text{s.t.} & S\in\mathscr{S}_{\mathscr{P}}\end{array}\tag{3}$$

Only path-strategies $S\in\mathscr{S}_{\mathscr{P}}$ are considered since we are routing under a priori route choice. Path-strategies are defined in the *decision space* $\mathscr{S}_{\mathscr{P}}$ and correspond to points in the *criterion space* $\mathscr{W}=\{W(S)\in\mathbb{R}^2 \mid S\in\mathscr{S}_{\mathscr{P}}\}$.

Minimizing a vector-valued objective function, such as (3), requires some explanation since there is no complete order defined in $\mathbb{R}^2$. Path-strategy S is *efficient* (Pareto optimal) if and only if

$$\nexists\tilde{S}\in\mathscr{S}_{\mathscr{P}}: W_1(\tilde{S})\le W_1(S) \text{ and } W_2(\tilde{S})\le W_2(S)$$

with at least one strict inequality; otherwise S is *inefficient*.

A point $W(S)\in\mathscr{W}$ is a *nondominated* point if and only if S is an efficient strategy. Otherwise $W(S)$ is a *dominated* point. Let

$$\mathscr{S}_E=\{S\in\mathscr{S}_{\mathscr{P}} \mid S \text{ is efficient}\},\quad \mathscr{W}_E=\left\{W(S)\in\mathbb{R}^2 \mid S\in\mathscr{S}_E\right\}$$

denote the set of efficient path-strategies and nondominated points, respectively. Nondominated points can be partitioned into two sets, namely supported and unsupported. The supported ones can be further subdivided into extreme and nonextreme. To this aim, let us define the following set

$$\mathscr{W}^{\ge}=conv(\mathscr{W}_E)\oplus\left\{\mathbf{w}\in\mathbb{R}^2 \mid \mathbf{w}\ge 0\right\},$$

where $\oplus$ as usual denotes direct sum, and $conv(\mathscr{W}_E)$ denotes the convex hull of $\mathscr{W}_E$. $W(S)\in\mathscr{W}_E$ is a *supported* nondominated criterion point, if $W(S)$ is on the boundary of $\mathscr{W}^{\ge}$. Otherwise $W(S)$ is an *unsupported* point. A supported point $W(S)$ is *extreme*, if $W(S)$ is an extreme point of $\mathscr{W}^{\ge}$. Otherwise $W(S)$ is a *nonextreme* point.

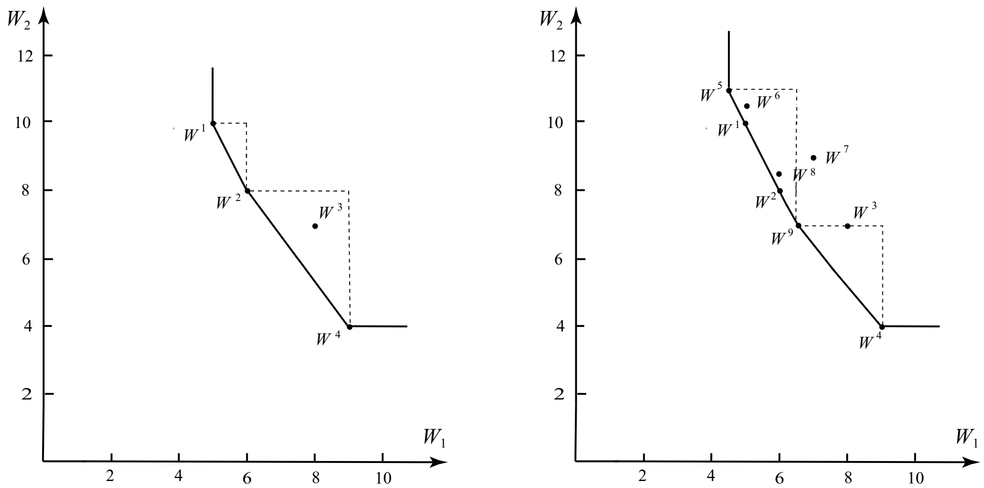
If an approximation of $\mathscr{W}_E$ is wanted, the quality can be controlled using the concepts of ε-domination and ε-approximation [15]. A point (W_1,W_2) ε-*dominates* point $(\hat{W}_1,\hat{W}_2)$ if

$$\hat{W}_1\ge(1-\varepsilon)W_1,\qquad \hat{W}_2\ge(1-\varepsilon)W_2$$

A set $\mathscr{W}_1$ is an ε-*approximation* of another nondominated set $\mathscr{W}_2$, if for each point $\hat{W}\in\mathscr{W}_2$, there exists $W\in\mathscr{W}_1$ such that W ε-dominates $\hat{W}$.

Example 1 (continued) Assume that two costs $c_i(u,v,t)$, $i=1,2$, are given for each leaving time t from node u along arc (u,v), see Table 1.

Consider the problem of finding the set of nondominated points under a priori route choice, i.e. solving (3), when both criteria are MEC. The criterion points corresponding to the four possible loopless paths in G are illustrated in Figure 2(a). In this example all four points are nondominated. The points are given by: $W^1=(5,10)$, $W^2=(6,8)$, $W^3=(8,7)$ and $W^4=(9,4)$. W^1,W^2 and W^4 are supported points all of which are extreme. Solid lines define the boundary of $W^{\ge}$; W^1 and W^4 are the *upper/left* and the *lower/right*



(a) Path-strategies. (b) Strategies.

Figure 2: Criterion spaces under a priori and time-adaptive route choice.

vertex in $W^{\geq}$, respectively. The extreme points define two triangles, shown with dashed lines, in which it may be possible to find unsupported nondominated points such as W^3.

Under time-adaptive route choice, i.e., if we consider the solution of (3) with the constraint replaced with $S \in \mathscr{S}$, we have nine possible strategies, five of which are not path-strategies; the corresponding points are: $W^5 = (4.5, 11)$, $W^6 = (5, 10.5)$, $W^7 = (7, 9)$, $W^8 = (6, 8.5)$ and $W^9 = (6.5, 7)$.

All nine points are illustrated in Figure 2(b). Five points W^1, W^2, W^4, W^5 and W^9 are supported nondominated points of which W^1 and W^2 are non-extreme. Points which do not lie inside the triangles such as W^3 and W^7 are dominated. Moreover, the two points W^6 and W^8 are dominated by W^1 and W^2, respectively. Note that a nondominated point under a priori route choice may be dominated under time-adaptive route choice, as is the case for W^3, dominated by W^9.

The four path-strategies can be seen in Figure 7 in Appendix A, while the five time-adaptive strategies not corresponding to a path are shown in Figure 8. ■

Obviously, the time-adaptive nondominated set always dominates the a priori nondominated set, however, the former set does not necessarily contain the latter. As we shall see later, the number of efficient path-strategies is in general significantly lower than the number of efficient strategies.

3 Solution method

In this section we consider the problem of finding all the efficient paths under a priori route choice between an origin node o and a destination node d, when leaving the origin at time zero. We devise a solution method based on the two-phase approach, and provide details on its implementation. We also discuss the case, where waiting at intermediate node is allowed; we argue that waiting makes the problem much harder, and we show that our method can be adapted in order to obtain an approximation of the efficient set.

The two-phase approach is a general method for solving bicriterion discrete optimization problems such as (3). As the name suggests, the two-phase method divides the search for nondominated points into two phases. In phase one, the supported extreme nondominated points are found. These extreme points define a number of triangles in which unsupported nondominated points may be found in phase two (see

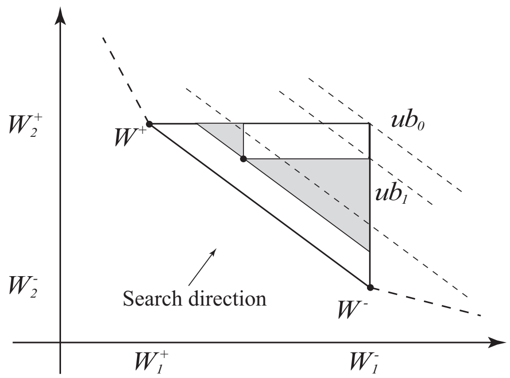


Figure 3: A triangle defined by W^+ and W^-.

Figure 2). For a description of a generic two-phase method see Pedersen, Nielsen, and Andersen [13].

Both phases make use of a parametric function $\gamma : (\mathscr{W}, \mathbb{R}_+) \to \mathbb{R}_+$ which defines the *parametric cost* of a path-strategy $S \in \mathscr{S}_{\mathscr{P}}$:

$$\gamma(W(S), \lambda) = W_1(S)\lambda + W_2(S). \tag{4}$$

It is well-known that given $\lambda > 0$ the path-strategy S with minimum parametric cost $\gamma(W(S), \lambda)$ corresponds to a supported nondominated point and hence is efficient. As a result all supported extreme nondominated points can be found in phase one by solving (4) for different values of λ, see [3].

The extreme points found in phase one define a number of triangles in which unsupported nondominated points may exist. Phase two searches each triangle using an algorithm for ranking path-strategies with respect to the parametric weight (4), where λ is a function of the slope of the line joining the two points $W^+ = (W_1^+, W_2^+)$ and $W^- = (W_1^-, W_2^-)$ defining the triangle (see Figure 3). The search stops, when the parametric cost reaches an upper bound, which is initially set to $ub_0 = W_1^- \lambda + W_2^+$. During the search, the upper bound is dynamically updated (decreased), when new nondominated points are found. Searching in a triangle can be seen as a "sweep line" method, as shown in Figure 3; note that the upper bound is updated to ub_1 when the new nondominated point is found inside the triangle, since the area where further nondominated points can be located (shaded in the picture) decreases.

It must be kept in mind that in both phases we have to solve a sequence of difficult problems, since a priori routing even for the single criterion case is NP-hard. In order to solve these problems we adopted the algorithm for ranking paths in STD networks (*procedure* K-BPS) recently devised by Nielsen et al. [12]. However, the effectiveness of this approach is quite different for expectation and min-max criteria.

3.1 Two expectation criteria

Consider (3) and assume that we are minimizing expected cost for both criteria. Let

$$\begin{aligned} c_\lambda(u,v,t) &= c_1(u,v,t)\lambda + c_2(u,v,t), && (u,v) \in A,\ t \in L(u,v) \\ g_\lambda(t) &= g_1(t)\lambda + g_2(t), && t \in L(d) \end{aligned}$$

The following theorem has been proved in Nielsen et al. [11].

Theorem 2 *Let $W_\lambda(S)$ denote the weight of a strategy S using costs $c_\lambda(u,v,t)$ and $g_\lambda(t)$ under the MEC criterion. For every $\lambda > 0$ and for every S, we have that $W_\lambda(S) = \gamma(W(S), \lambda)$.*

Clearly, the result holds for path-strategies in particular. As a consequence, we can rank paths with respect to the parametric cost $\gamma(W(S), \lambda)$ by applying procedure *K-BPS* with the costs $c_\lambda(u,v,t)$ and $g_\lambda(t)$. This procedure is also used in phase one, stopping as soon as the best parametric cost path is found. Theorem 2 also holds, when minimizing expected travel time instead of cost since due to Theorem 1 we have that the MET problem can be formulated as a MEC problem.

3.2 Two min-max criteria

Consider the case where we are minimizing maximum cost for both criteria. Unfortunately, we do not have a result similar to Theorem 2. However, a lower bound on the parametric cost of a strategy can be determined, Nielsen et al. [11].

Theorem 3 *Let $W_\lambda(S)$ denote the weight of a strategy S using costs $c_\lambda(u,v,t)$ and $g_\lambda(t)$ under the MMC criterion. For every $\lambda > 0$ and for every S, we have that $W_\lambda(S) \leq \gamma(W(S),\lambda)$.*

Due to Theorem 3 we can generate path-strategies in non-decreasing order of $W_\lambda(S)$ by applying procedure *K-BPS* with costs $c_\lambda(u,v,t)$ and $g_\lambda(t)$. For each generated path-strategy S we can calculate $W_1(S)$ and $W_2(S)$ and hence $\gamma(W(S),\lambda)$. Note that $\gamma(W(S),\lambda)$ provide us with an upper bound on the minimal parametric cost. As a result, in phase one we can stop procedure *K-BPS* as soon as $W_\lambda(S)$ reaches the parametric cost of the best path-strategy generated so far. In phase two a triangle is searched until the lower bound $W_\lambda(S)$ reaches the current upper bound.

Observe that in phase one, since we rank according to $W_\lambda(S)$ instead of $\gamma(W(S),\lambda)$, procedure *K-BPS* may generate many paths that actually fall inside the triangle defined by a certain λ. In order to take advantage of this fact, we devised a *hybrid* algorithm, where the two phases are combined. More precisely, when a new triangle is identified in the first phase, we immediately search inside the triangle by letting procedure *K-BPS* continue until the lower bound $W_\lambda(S)$ reaches the upper bound. In practice, the hybrid approach avoids repeating the same computations performed in phase one to find the minimum parametric cost path. For this reason, we adopt the hybrid algorithm in our computational tests.

3.3 STD networks with waiting allowed

In this section we will briefly discuss the case in which *waiting* at the nodes is allowed. The subject will not be discussed in great detail here, but a thorough treatment of the subject is available in [9].

From a theoretical point of view, waiting at intermediate nodes should not be considered within an a priori route choice model, as it is an inherently time-adaptive behavior. Indeed, while travelling along a path, at any given time in a node, a traveller has to choose, whether to wait or proceed to the next node in the path. Clearly, the decision cannot be "waiting" at each time, since a traveller cannot wait indefinitely at intermediate nodes. However, it may be interesting to consider waiting as a limited form of time-adaptive behavior, thus defining an intermediate model between a priori and time-adaptive routing.

A key observation here is that an *o-d* path P in G defines several strategies, actually, an exponential number in the length of P. That is, we may have many path-strategies corresponding to the same path, distinguished from each other only by the use of waiting. As a consequence, different nondominated points may correspond to the same path in G. From a theoretical point of view, the two-phase method described above remains valid since we rank path-strategies. However, due to the fact that a particular path P in G may define a huge number of path-strategies, we may face difficulties in phase two, when searching a triangle using ranking. A lot more path-strategies may have to be ranked before reaching the upper bound of the triangle, resulting in much larger CPU times.

One way to deal with this problem is to consider only one path-strategy for each path in G, when ranking path-strategies in a triangle. In particular, we take a path-strategy yielding minimum parametric cost with respect to the value λ corresponding to the triangle. This approach guarantees that no substantial loss in computational performance will be incurred, but provides us with an approximation of the true nondominated set. Furthermore, we may not always be able to give an estimate of the quality of the approximation found; we shall try to evaluate this quality empirically in our computational experience.

4 Computational results

We implemented the algorithm in C++ and tested it on a 1 GHz PIII computer with 1GB RAM using a Linux Red Hat operating system. The source code has been compiled with the GNU C++ compiler with optimize option -O. The main goals of the computational experiments can be summarized as follows:

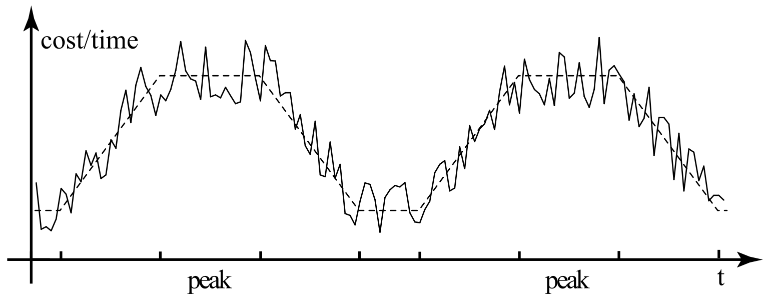


Figure 4: Peak effect and random perturbation for an arc.

1. Validating the performance of the algorithm on reasonably hard test instances.
2. Evaluating the algorithm and the solution set under different criteria and different correlation structure between the two criteria.
3. Comparing the nondominated points found under a priori and time-adaptive route choice.

4.1 Time-Expanded Generator with Peaks (TEGP)

The STD network test instances are generated using the newly developed generator TEGP[2] (*Time-Expanded Generator with Peaks*). The generator includes several features inspired by typical aspects of road networks, such as congestion effects, waiting, random perturbations.

An underlying grid graph G of *base* b and *height* h is assumed, and we search optimal routes from the bottom-right corner node (origin o) to the upper left corner node (destination d). The choice of grids is motivated by their resemblance to road networks, and by the fact that they are usually considered computationally harder. Indeed, each origin-destination path has at least $b+h-2$ arcs and the number of such paths grows exponentially with the size of G. Thus paths are not too short, as may happen in random graphs, and there is a large number of potentially efficient solutions.

The generator considers *cyclic* time periods. In each cyclic period there are some *peak periods* (e.g. rush hours). Each peak consists of three parts; a *transient* part, where the mean travel time (traffic) increases, a *pure peak* part, where it stays the same, and a transient part, where it decreases again. This is illustrated in Figure 4 for a specific arc (u,v). Here two peaks are considered and the dotted line is the mean travel time. Given the mean travel time $\mu_{uv}(t)$ for arc (u,v) at leaving time t the travel time distribution is generated such that:

1. the interval of possible travel times is $\{\lfloor \mu_{uv}(t) - 0.25\mu_{uv}(t) \rfloor, \ldots, \lceil \mu_{uv}(t) + 0.25\mu_{uv}(t) \rceil\}$;
2. the probabilities give a rough discrete approximation of a normal distribution with mean $\mu_{uv}(t)$ and standard deviation $0.25\mu_{uv}(t)$.

This setting gives travel times with higher mean and standard deviation in peaks.

The time horizon consists of one or more cyclic periods, with peaks placed at the same time in each cycle. Note that the STD network does not have a final deterministic steady state, as often assumed in the literature (see e.g. [8]), but a stochastic and time-adaptive behaviour in the whole time-horizon.

Costs are generated taking three components into account: a off-peak cost, the peak effect and a random perturbation. The *random perturbation* is used to introduce small variations, not intercepted by the peak effect, e.g. special information about the cost at exactly that leaving time. Hence, the cost follows a slightly different pattern compared to mean travel time; this pattern is shown by the solid line in Figure 4. If waiting is allowed, waiting costs are generated using an off-peak component and a random perturbation. For more details on the TEGP generator, see Nielsen [10].

[2]The problem generator and the test instances used in this paper are downloadable from the following web-page `http://www.research.relund.dk/`.

Class	1	2	3	4
Grid size	5×8	10×10	5×8	10×10
H	144	288	144	288
I_T	[3,20]	[3,20]	[3,20]	[3,20]
I_C	[1,2200]	[1,2200]	[1,2200]	[1,2200]
I_W	-	-	[1,550]	[1,550]
κ	6	6	6	6
Waiting	no	no	yes	yes
n	2254	14877	2263	14886
m_h	7383	53220	7405	53241
m_a	80	196	2262	14885

Table 2: Test classes.

4.2 Test instances

Four classes of STD networks are considered with two different grid sizes, namely 5×8 and 10×10. In the first two classes, corresponding to the two different grid sizes, waiting is not allowed. In the first class the time-horizon corresponds to one cycle with 144 time instances, e.g., 12 hours divided into 5 minute intervals, whereas in the second class the time-horizon corresponds to 288 time instances (2 cycles). Note in general the time-horizon depends on the size of the network. Here the time-horizon is chosen so that it should be possible to travel along all paths with length $b+h$. Classes three and four are similar to the first two classes with the exception that waiting is allowed. A cycle has two peaks, each with a total length of 5 hours with each part of the peak lasting 1 hour and 40 minutes. The first peak starts after half an hour ($t=6$).

The interval of possible off-peak mean travel times is $[lb_t, ub_t] = [4,8]$, i.e. an off-peak mean travel time between 20 and 40 minutes. The mean travel time increases by 100% in the pure peak part. Similarly, the interval of possible off-peak costs is $[lb_C, ub_C] = [1,1000]$ and the cost $c_i(u,v,t)$ increases by 100% in the pure peak part. In the case of waiting, waiting costs I_W are generated randomly between 1 and 500. The random perturbation increases or decreases a cost value by 10% at most.

The parameters defining our set of instances are summarized in Table 2. Here, I_T is the range of possible travel times (for all arcs and departure times); similar, I_C and I_W denote the range of possible travel and waiting costs, respectively. κ denotes the average size of the travel time distributions. The parameters n, m_h and m_a are related to the time-expanded hypergraph representation of the STD used as the underlying data structure in our algorithms. In particular, they denote the number of nodes, hyperarcs and arcs in the hypergraph, respectively. The values reported in the table are an average over all the instances in the same class generated in our tests, and refer to the reduced hypergraph obtained at the end of a *preprocessing phase*, that deletes each node and hyperarc that cannot belong to a strategy; see [9] for details.

We report results on three ways of generating costs, namely *C/C* neg_{cor}, *C/C* no_{cor} and *T/C*. In *C/C* neg_{cor} the costs are assumed to be negatively correlated. This is a typical situation in hazardous material transportation, where travel cost and risk/exposure are conflicting. In this case, the off-peak costs for a specific arc are generated so that if one belongs to the first half of the interval $[lb_C, ub_C]$ then the other belongs to the second half. In $C/C\ no_{cor}$ there is no correlation between the two costs, which are generated independently. Finally, for *T/C* the first cost corresponds to travel time (treated as a cost, according to Theorem 1) and the second cost is generated as for *C/C* no_{cor}. In all combinations the penalty costs are assumed to be zero, except when the cost corresponds to time, in which case the penalty costs are defined according to Theorem 1.

4.3 Performance measures/statistics

In this section performance measures/statistics used to evaluate the algorithm are described. For each class, combination of criteria and cost correlation type, the measures are average (or maximum) over five

Class		$\lvert\mathscr{W}_{SE}\rvert$	CPU$_{SE}$	RI_1	RI_2	$\triangle$	$\lvert\mathscr{W}_{\triangle}\rvert$ ave	$\lvert\mathscr{W}_{\triangle}\rvert$ max	CPU$_{\triangle}$ ave	CPU$_{\triangle}$ max
1	T/C	5	0.54	48	94	4	1	4	0.16	0.33
1	C/C no_{cor}	4	0.57	98	103	3	1	4	0.20	0.40
1	C/C neg_{cor}	8	1.45	201	324	7	3	11	0.73	2.22
2	T/C	6	8.90	36	136	5	2	8	3.79	22.91
2	C/C no_{cor}	8	21.45	140	92	7	3	17	5.76	47.57
2	C/C neg_{cor}	11	25.04	280	365	10	6	26	12.08	43.05

Table 3: Results expectation criteria.

independent instances obtained using a different seed. The first group of statistics refers to phase one; the abbreviation used in the tables is given in parentheses.

Extreme supported size ($\lvert\mathscr{W}_{SE}\rvert$): The number of supported extreme nondominated points

CPU time (CPU_{SE}): For expectation criteria, the total CPU time for phase one in seconds; not reported for min-max criteria, where the hybrid algorithm is used.

Number of triangles ($\triangle$): Number of triangles defined by supported extreme points.

Relative increase (RI_j): The relative increase from the upper/left point W^{ul} to the lower/right point W^{lr} for the j'th criteria defined as $RI_1 = (W_1^{lr} - W_1^{ul})/W_1^{ul}$ and $RI_2 = (W_2^{ul} - W_2^{lr})/W_2^{lr}$. Reported in percent.

In the second group we report statistics for each triangle searched in phase two.

CPU time ($CPU_{\triangle}$): The CPU time for searching a triangle, in seconds. The average and maximum over all the triangles searched are reported.

Points in the triangle ($\lvert\mathscr{W}_{\triangle}\rvert$): The number of nondominated points in the triangle not including the two points defining the triangle. Average and maximum results are reported.

Upper bound on epsilon (ε_{ub}): For expectation criteria, an indication of the quality of the approximation obtained when waiting is permitted. An upper bound ε_{ub} is computed, for each triangle, with the following property: the approximated nondominated set found for a triangle is an ε-approximation of the true set with $\varepsilon \leq \varepsilon_{ub}$. See Nielsen [9, Th. 5.4.5] for details. Average and maximum results over all triangles are reported in percent.

4.4 Results - expectation criteria

First, we report on the results obtained, when both criteria represent expectation and waiting at the nodes in G is not allowed. The results are reported in Table 3. In phase one all extreme supported nondominated points can be determined in a reasonable amount of time. The same holds for phase two (which is the most time-consuming phase). That is, we can find the nondominated set for all the test instances considered.

Comparing the different criteria, the results for phase one and two reveal that minimizing expected travel time and cost (*T/C*) is in general easier than when considering cost/risk criteria. If we compare the two different types of correlation for cost (*C/C*) the results indicate that negatively correlated costs produce more extreme nondominated points. Moreover, these points define a larger gap between the upper/left and lower/right points (RI_j columns), i.e. we have to search a larger area of the criterion space. As a result we have more triangles to search, and it takes longer time to search each one of them. This fact was also observed under time-adaptive route choice, Nielsen et al. [11] and is a general feature for discrete bicriterion optimization problems, see e.g. Pedersen et al. [13]. Figure 5 gives an example of the effect of negative correlation among costs.

Class		$\lvert\mathscr{W}_{SE}\rvert$	$\triangle$	$\lvert\mathscr{W}_{\triangle}\rvert$ ave	$\lvert\mathscr{W}_{\triangle}\rvert$ max	$\text{CPU}_{\triangle}$ ave	$\text{CPU}_{\triangle}$ max
1	C/C no_{cor}	4	3	1	4	0.32	0.68
1	C/C neg_{cor}	7	6	4	14	2.01	5.65
2	C/C no_{cor}	8	7	2	12	17.91	80.12
2	C/C neg_{cor}	10	9	7	59	100.42	439.72

Table 4: Results min-max criteria.

Class		$\lvert\mathscr{W}_{SE}\rvert$	CPU_{SE}	RI_1	RI_2	$\triangle$	$\lvert\mathscr{W}_{\triangle}\rvert$ ave	$\lvert\mathscr{W}_{\triangle}\rvert$ max	$\text{CPU}_{\triangle}$ ave	$\text{CPU}_{\triangle}$ max	ε_{ub} ave	ε_{ub} max
3	T/C	35	6.01	84	100	34	0	10	0.12	0.48	0.5	7.3
3	C/C no_{cor}	41	6.74	424	121	40	0	6	0.13	0.60	0.9	9.1
3	C/C neg_{cor}	32	5.70	201	349	31	1	11	0.42	2.86	1.3	8.1
4	T/C	73	96.10	71	155	60	2	26	3.43	43.05	0.1	4.6
4	C/C no_{cor}	113	265.71	756	126	112	1	18	1.79	58.43	0.2	5.6
4	C/C neg_{cor}	114	259.11	532	541	113	1	26	2.63	56.94	0.3	6.6

Table 5: Results expectation criteria (waiting allowed).

4.5 Results - min-max criteria

Assume that both criteria are min-max and waiting is not allowed. We only consider the situation, where we are minimizing maximum cost for both criteria, i.e., (MMC,MMC), since we believe that the time/cost case is not very interesting in practice. Indeed, for a decision maker interested in path-strategies with bounded maximum travel time, finding the minimum cost path-strategy for different settings of the time-horizon may be a much simpler and more efficient approach.

The results for the hybrid algorithm are presented in Table 4. All instances can be solved. Compared to expectation criteria the total number of nondominated points is about the same in average. However, the CPU time spent is considerably higher as the lower bound used for ranking is not very tight. If comparing the different criteria the same results hold as under expectation criteria.

4.6 Results - waiting allowed

Even though waiting at the nodes in G is essentially a time-adaptive behavior we tested our algorithms on classes 3 and 4. Here the topology structure of the STD networks is the same as in class 1 and 2 except that waiting is allowed at intermediate nodes.

The results for expectation criteria are presented in Table 5. In the first phase all extreme nondominated points can be found. The number of extreme points is much higher compared to when no waiting is allowed (see Table 3). This may be due to the fact that many extreme points correspond to the same path, that is, differ from each other only in what concerns waiting. Furthermore, the gap between the upper/left and lower/right points is much larger in the waiting case (RI_j columns), i.e. we have to search a larger area of the criterion space. However, since many extreme points exist a decision maker may be satisfied with the extreme points offered by phase one, which would make phase two superfluous.

In the second phase we computed an approximation of the nondominated set by choosing only one path-strategy for each path in G, when a triangle is searched. In order to evaluate the quality of the approximation we computed the upper bound ε_{ub} described earlier. In general, acceptable approximations are found, however, in a few large triangles poor values of ε_{ub} are obtained. In general a large value of ε_{ub} does not necessarily mean that we have found a poor approximation of the nondominated set, but may be due to the fact that the true set lies deep inside the triangle.

The results for min-max criteria are presented in Table 6 and do not seem to differ substantially from

Class		$\triangle$	$\lvert\mathscr{W}_\triangle\rvert$ ave	$\lvert\mathscr{W}_\triangle\rvert$ max	$\text{CPU}_\triangle$ ave	$\text{CPU}_\triangle$ max	ε_{ub} ave	ε_{ub} max
3	C/C no_{cor}	6	1	7	0.32	1.04	-	-
3	C/C neg_{cor}	9	3	16	2.27	8.52	-	-
4	C/C no_{cor}	10	2	12	14.92	76.02	-	-
4	C/C neg_{cor}	15	5	35	69.70	385.93	-	-

Table 6: Results min-max criteria (waiting allowed).

the no waiting classes. This is probably because the maximum possible cost of a strategy is always an integer value when using integer costs. Hence there will not be so many different nondominated points, even if many path-strategies may correspond to the same point.

4.7 Comparison to the time-adaptive case

Comparing the results for the a-priori case to previous results for the time-adaptive case [9, 11] allows us to point out interesting differences. Here we restrict ourselves to expectation criteria, since for min-max criteria the approximation found in [11] is usually rather week.

First of all, recall that the set of nondominated points has been found for all the instances considered. This is in deep contrast to time-adaptive route choice, where not even an ε-approximation with $\varepsilon = 1\%$ could be found for the same set of instances [11]. This result may be viewed as surprising, since finding the best strategy in the single criterion case is easy (can be done in linear time) while finding the best path-strategy is NP-hard. A reasonable explanation of this apparent paradox is that the solution space is much more dense in the time-adaptive case, that is, the total number of path-strategies is much lower than the total number of strategies. Therefore the ranking procedure used in the second phase does not have to rank as many solutions.

In order to get a deeper insight in this issue, we made plots comparing the nondominated set for the a priori case with an approximation of the nondominated set for the time-adaptive case, obtained using the algorithms from [11]. Figure 5 shows two instances on a 5×8 grid with uncorrelated costs (left) and negatively correlated costs (right).

First, as noted above, negatively correlated costs produce more nondominated points, spread in a wider area; this situation arises for both a priori and adaptive routing. Second, in some cases the a priori nondominated set may contain points close to the time-adaptive nondominated set. Hence solutions found when a priori routing must be adopted, due e.g. to outside regulations, may still be as good as those found without this regulation. However, in other cases costs might be substantially higher, see e.g. the left plot in Figure 5.

Finally, in general for our instances there are large variations in the values of ε for which the a priori nondominated set turns out to ε-dominate the time-adaptive nondominated set. On average, we have $\varepsilon = 0.1$, but the minimum ε value found was 0.03 and the maximum 0.25.

5 Conclusions

In this paper we have considered bicriterion a priori route choice in stochastic time-dependent networks. A new algorithm for solving this problem was presented. It is based on the two-phase approach, and exploits a recently developed algorithm for ranking path-strategies. Furthermore, if waiting in the nodes in G is allowed, our algorithm can compute an approximation of the nondominated set.

Numerical results were obtained on reasonably hard test instances, considering both min-max and expectation criteria. The reported results are encouraging. If no waiting is allowed ("true" a priori route choice) we were able to solve all instances completely. This is in contrast to time-adaptive route choice, and is primarily due to the fact that the total number of path-strategies is much lower than the total number of strategies. For the waiting case, we obtained a reasonable approximation of the true nondominated set.

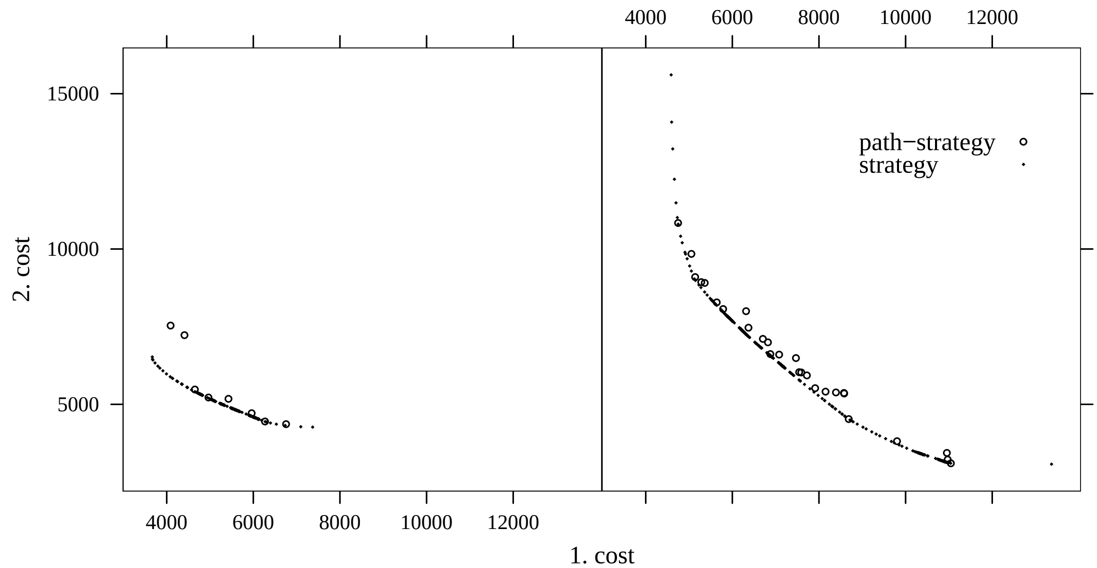


Figure 5: The nondominated sets for an uncorrelated (left) and negatively correlated (right) test instance. Both criteria are minimizing expected cost.

Due to the effectiveness of our algorithm, we were able to compare the nondominated set under a priori route choice to the approximations of the nondominated set obtained under time-adaptive route choice. Based on the set of instances considered here, we may conclude that nondominated points under a priori routing often are equal or close to nondominated points found under time-adaptive routing, but time-adaptive routing may result in much better solutions in some cases.

References

[1] R.K. Ahuja, T.L. Magnanti, and J.B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice Hall, 1993.

[2] T.S. Chang, L.K. Nozick, and M.A. Turnquist. Multiobjective path finding in stochastic dynamic networks, with application to routing hazardous materials shipments. *Transportation Science*, 39(3): 383–399, August 2005. doi:10.1287/trsc.1040.0094.

[3] J. Cohen. *Multiobjective Programming and Planning*. Academic Press, New York, 1978.

[4] N. Deo and C. Pang. Shortest-path algorithms: Taxonomy and annotation. *Networks*, 14:275–323, 1984.

[5] G. Gallo, G. Longo, S. Pallottino, and S. Nguyen. Directed hypergraphs and applications. *Discrete Applied Mathematics*, 42(2-3):177–201, 1993. doi:10.1016/0166-218X(93)90045-P.

[6] M. Garey and D. Johnson. *Computers and Intractability. A Guide of the Theory of NP-Completeness*. W.H. Freeman, 1979.

[7] R.W. Hall. The fastest path through a network with random time-dependent travel times. *Transportation Science*, 20(3):182–188, 1986.

[8] E.D. Miller-Hooks and H.S. Mahmassani. Optimal routing of hazardous materials in stochastic, time-varying transportation networks. *Transportation Research Record*, 1645:143–151, 1998.

[9] L.R. Nielsen. *Route Choice in Stochastic Time-Dependent Networks*. PhD thesis, Department of Operations Research, University of Aarhus, 2004. URL `http://www.imf.au.dk/publs?id=499`.

[10] L.R. Nielsen. TEGP - Time-Expanded Generator with Peaks, February 2006. URL `http://www.research.relund.dk`. Reference manual.

[11] L.R. Nielsen, K.A. Andersen, and D. Pretolani. Bicriterion shortest hyperpaths in random time-dependent networks. *IMA Journal of Management Mathematics*, 14(3):271–303, 2003.

[12] L.R. Nielsen, D. Pretolani, and K.A. Andersen. K shortest paths in stochastic time-dependent networks. Working Paper WP-L-2004-05, Department of Accounting, Finance and Logistics, Aarhus School of Business, 2004. URL `http://www.asb.dk/`.

[13] C.R. Pedersen, L.R. Nielsen, and K.A. Andersen. On the bicriterion multi modal assignment problem. Working Paper WP-2005-3, Department of Operations Research, University of Aarhus, December 2005. URL `http://www.imf.au.dk/publs?id=586`.

[14] D. Pretolani. A directed hypergraph model for random time-dependent shortest paths. *European Journal of Operational Research*, 123(2):315–324, 2000. doi:10.1016/S0377-2217(99)00259-3.

[15] A. Warburton. Approximation of pareto optima in multiple-objective, shortest-path problems. *Operations Research*, 35(1):70–79, 1987.

A A hypergraph model for STD networks

As shown in Pretolani [14] a *time-expanded hypergraph* $\mathscr{H} = (\mathscr{V}, \mathscr{E})$ can be used to model an STD network. We shall introduce the directed hypergraph model by means of Example 1, but first a few definitions are needed.

A *directed hypergraph* is a pair $\mathscr{H} = (\mathscr{V}, \mathscr{E})$, where $\mathscr{V} = (v_1, ..., v_n)$ is the set of *nodes*, and $\mathscr{E} = (e_1, ..., e_m)$ is the set of *hyperarcs*. A hyperarc $e \in \mathscr{E}$ is a pair $e = (T(e), h(e))$, where $T(e) \subset \mathscr{V}$ denotes the set of *tail* nodes and $h(e) \in \mathscr{V} \setminus T(e)$ denotes the *head* node. Note that a hyperarc has exactly one node in the head, and possibly several nodes in the tail. A hypergraph $\tilde{\mathscr{H}} = (\tilde{\mathscr{V}}, \tilde{\mathscr{E}})$ is a *subhypergraph* of $\mathscr{H} = (\mathscr{V}, \mathscr{E})$, if $\tilde{\mathscr{V}} \subseteq \mathscr{V}$ and $\tilde{\mathscr{E}} \subseteq \mathscr{E}$. A subhypergraph is *proper* if at least one of the inclusions is strict.

Definition 4 An *s-t* hyperpath $\pi = (\mathscr{V}_\pi, \mathscr{E}_\pi)$ from source s to terminal t, is a subhypergraph of $\mathscr{H}$ satisfying that, if $t = s$, then $\mathscr{E}_\pi = \varnothing$; otherwise the $q \geq 1$ hyperarcs in $\mathscr{E}_\pi$ can be ordered in a sequence $(e_1, ..., e_q)$ such that

1. $t = h(e_q)$.
2. $T(e_i) \subseteq \{s\} \cup \{h(e_1), ..., h(e_{i-1})\}, \quad \forall e_i \in \mathscr{E}_\pi$.
3. No proper subhypergraph of π is an *s-t* hyperpath.

Condition 3 implies that, for each $u \in \mathscr{V}_\pi \setminus \{s\}$, there exists a unique hyperarc $e \in \mathscr{E}_\pi$, such that $h(e) = u$. We denote hyperarc e as the *predecessor* of u in π. An immediate consequence of this is that a hyperpath π can be described by a *predecessor function* $p : \mathscr{V}_\pi \to \mathscr{E}_\pi$; for each $u \in \mathscr{V}_\pi$. $p(u)$ is the unique hyperarc in π which has node u as the head.

Example 1 (continued) The time expanded hypergraph $\mathscr{H} = (\mathscr{V}, \mathscr{E})$ for the STD network given in Example 1 is shown in Figure 6. It represents the relationships between leaving time and arrival time. The set $\mathscr{V}$ contains one node u_t for each pair (u,t), $t \in L(u)$ and a source node s. For each $(u,v) \in A$ and $t \in L(u,v)$ a hyperarc $e_{uv}(t) = (\{v_{t_i} : t_i \in I(u,v,t)\}, u_t)$ is defined, i.e. each column in Table 1 defines a hyperarc in $\mathscr{H}$. Assigned to each hyperarc $e_{uv}(t)$ are the corresponding costs $c_i(u,v,t)$, $i = 1,2$ given in Table 1. Finally, a dummy arc $e_d(t) = (\{s\}, d_t)$ is defined for each $t \in L(d)$ with zero costs. ■

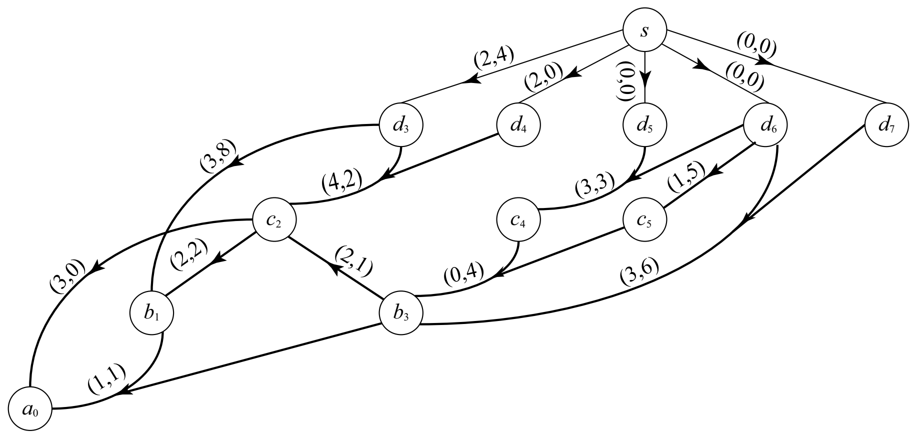


Figure 6: The time-expanded hypergraph $\mathscr{H}$.

It is not hard to recognize that there is a one to one correspondence between a strategy and a predecessor function p on the time-expanded hypergraph $\mathscr{H}$, i.e. choosing $p(u_t) = e_{uv}(t)$ is equivalent to choosing $S(u,t) = (u,v)$. In particular, an $(o,0)$ strategy can be represented by a hyperpath from node s to node o_0 in the time-expanded hypergraph $\mathscr{H}$, and vice-versa.

Property 1 *There is a one-to-one correspondence between* $(o,0)$ *strategies and* s-o_0 *hyperpaths in* $\mathscr{H}$.

Pretolani [14] showed that the weight $W(S)$ of a strategy S under MEC, MEC, MMC or MMT corresponds to the weight of the corresponding s-o_0 hyperpath in $\mathscr{H}$ using suitable hyperarc weights. Therefore, the best strategy can be found by finding the minimum weight hyperpath, i.e., by solving a *shortest hyperpath* problem in $\mathscr{H}$. Efficient procedures for finding shortest hyperpaths are defined in Gallo, Longo, Pallottino, and Nguyen [5].

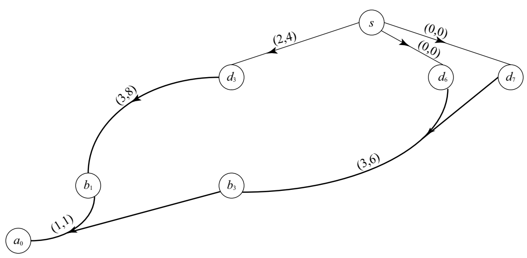
Example 1 (continued) Due to Proposition 1 strategies can be illustrated using hyperpaths in the time-expanded hypergraph $\mathscr{H}$. All the possible strategies are illustrated in figures 7 and 8 together with the subgraph of G that may be travelled using the specific strategy.

Under a priori route choice only strategies corresponding to a loopless path are allowed, i.e. we consider the solution of (3). The set of possible path-strategies are shown in Figure 7.

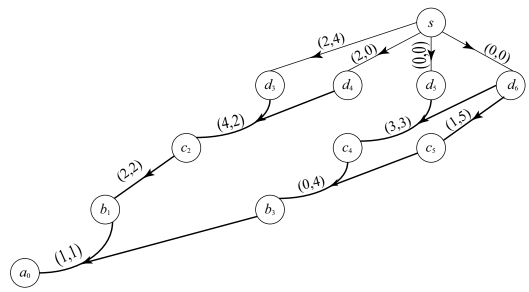
Under time-adaptive route choice strategies deviating from a path are also allowed, i.e. we consider the solution of (3) with the constraint replaced with $S \in \mathscr{S}$ instead. The set of possible strategies not corresponding to a path is shown in Figure 8.

Recall that the cost of the strategies described in figures 7 and 8 are given in Section 2, assuming that both criteria are MEC with costs as given in Table 1. For more details on how to calculate the costs of a strategy using its hyperpath representation see Nielsen [9].

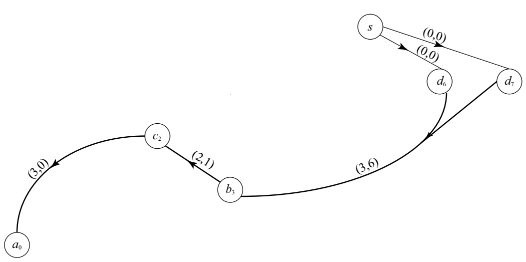
■



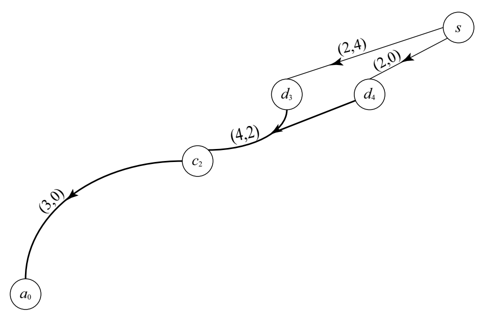
(a) A priori strategy corresponding to W^1.



(b) A priori strategy corresponding to W^2.

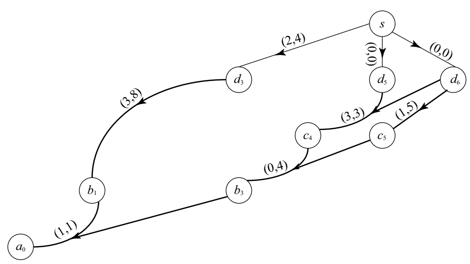


(c) A priori strategy corresponding to W^3.

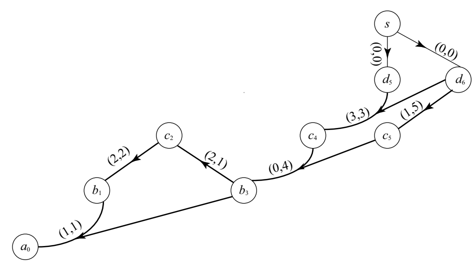


(d) A priori strategy corresponding to W^4.

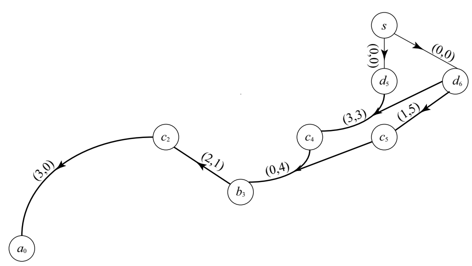
Figure 7: Possible strategies under a priori route choice (path-strategies).



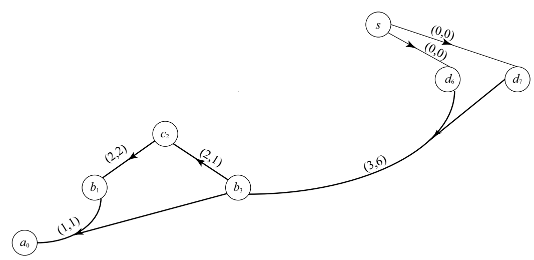
(a) Strategy corresponding to W^5.



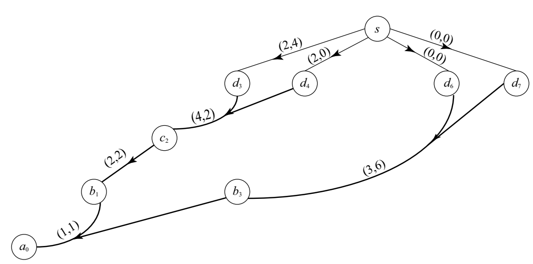
(b) Strategy corresponding to W^6.



(c) Strategy corresponding to W^7.



(d) Strategy corresponding to W^8.



(e) Strategy corresponding to W^9.

Figure 8: Time-adaptive strategies not corresponding to a path.

Working Papers from Logistics/SCM Research Group

L-2006-10 Lars Relund Nielsen, Kim Allan Andersen & Daniele Pretolani: Bicriterion a priori route choice in stochastic time-dependent networks.

L-2006-09 Christian Larsen & Gudrun P. Kiesmüller: Developing a closed-form cost expression for an *(R,s,nQ)* policy where the demand process is compound generalized Erlang.

L-2006-08 Eduardo Uchoa, Ricardo Fukasawa, Jens Lysgaard, Artur Pessoa, Marcus Poggi de Aragão, Diogo Andrade: Robust Branch-Cut-and-Price for the Capacitated Minimum Spanning Tree Problem over a Large Extended Formulation.

L-2006-07 Geir Brønmo, Bjørn Nygreen & Jens Lysgaard: Column generation approaches to ship scheduling with flexible cargo sizes.

L-2006-06 Adam N. Letchford, Jens Lysgaard & Richard W. Eglese: A Branch-and-Cut Algorithm for the Capacitated Open Vehicle Routing Problem.

L-2006-05 Ole Mortensen & Olga W. Lemoine: Business integration between manu-facturing and transport-logistics firms.

L-2006-04 Christian H. Christiansen & Jens Lysgaard: A column generation approach to the capacitated vehicle routing problem with stochastic demands.

L-2006-03 Christian Larsen: Computation of order and volume fill rates for a base stock inventory control system with heterogeneous demand to investigate which customer class gets the best service.

L-2006-02 Søren Glud Johansen & Anders Thorstenson: Note: Optimal base-stock pol-icy for the inventory system with periodic review, backorders and sequential lead times.

L-2006-01 Christian Larsen & Anders Thorstenson: A comparison between the order and the volume fill rates for a base-stock inventory control system under a compound renewal demand process.

L-2005-02 Michael M. Sørensen: Polyhedral computations for the simple graph partitioning problem.

L-2005-01 Ole Mortensen: Transportkoncepter og IT-støtte: et undersøgelsesoplæg og nogle foreløbige resultater.

L-2004-05 Lars Relund Nielsen, Daniele Pretolani & Kim Allan Andersen: *K* shortest paths in stochastic time-dependent networks.

L-2004-04 Lars Relund Nielsen, Daniele Pretolani & Kim Allan Andersen: Finding the *K* shortest hyperpaths using reoptimization.

L-2004-03 Søren Glud Johansen & Anders Thorstenson: The (r,q) policy for the lost-sales inventory system when more than one order may be outstanding.

L-2004-02 Erland Hejn Nielsen: Streams of events and performance of queuing systems: The basic anatomy of arrival/departure processes, when focus is set on autocorrelation.

L-2004-01 Jens Lysgaard: Reachability cuts for the vehicle routing problem with time windows.

Handelshøjskolen i Århus

Aarhus School of Business

ISBN 87-7882-154-1

Department of Business Studies

Aarhus School of Business
Fuglesangs Allé 4
DK-8210 Aarhus V - Denmark

Tel. +45 89 48 66 88
Fax +45 86 15 01 88

www.asb.dk