Competitive Analysis of a Dispatch Policy for a Dynamic Multi-Period Routing Problem

Enrico Angelelli
Department of Quantitative Methods
University of Brescia

Martin W.P. Savelsbergh
School of Industrial and Systems Engineering
Georgia Institute of Technology

M. Grazia Speranza
Department of Quantitative Methods
University of Brescia

Abstract

We analyze a simple and natural on-line algorithm (dispatch policy) for a dynamic multi-period uncapacitated routing problem, in which at the beginning of each time period a set of orders arrive that have to be served either in that time period or in the next. The objective of the problem is to minimize the average routing cost per time period. We show that the competitive ratio of this on-line algorithm for instances with customers on the non-negative real line is $\frac{3}{2}$.

1 Introduction

The importance of dynamic vehicle routing problems has been pointed out long ago (see [7]). Several techniques can be used to model and analyze dynamic routing problems. We use competitive analysis, following a number of other authors (e.g., [2], [3], [4], [5] and [6]). Competitive analysis studies the worst-case performance of an on-line algorithm with respect to an optimal off-line algorithm, i.e., an algorithm that has access, in advance, to the information that is normally revealed dynamically over time (see [8]).

Let $z(A, C)$ denote the value of the solution produced by algorithm A on instance C, and let $z^*(C)$ denote the value of the solution produced by an optimal off-line algorithm on instance C. The *competitive ratio* r_A of Algorithm A is defined as

$$r_A = \max_C \frac{z(A, C)}{z^*(C)}.$$

An on-line algorithm A is said to be *optimal* if no other algorithm A' has a competitive ratio $r_{A'} < r_A$.

In Angelelli et al. [1] a dynamic multi-period uncapacitated routing problem (DMPRP) was introduced in which at the beginning of each time period, a set of orders arrives each of which has to be served either in that time period or in the next. Thus, in each time period there are customers which have to be served and customers whose service may be postponed. Once it has been decided which customers to serve, an optimal route visiting these customers is constructed and executed. The objective is to minimize the average cost per time period. Three simple and natural on-line algorithms were analyzed. Algorithm $IMMEDIATE$ always serves customers as soon as possible, Algorithm $DELAY$ always serves customers as late as possible, and Algorithm $SMART$ decides in each time period whether to serve customers immediately or whether to delay service on the basis of the cost of serving the set of customers that have to be served that time period and the cost of serving all the customers. Both $IMMEDIATE$ and $DELAY$ were shown to have a competitive ratio of 2. When the planning horizon is restricted to two time periods, $SMART$ was shown to be optimal with a competitive ratio of $\sqrt{2}$ for instances with customers located on the real line and was shown to have a competitive ratio of $\frac{3}{2}$ for instances with customers located in the Euclidean plane. Furthermore, it was established that $SMART$ is no longer optimal when the number of time periods is greater than two and that its competitive ratio increases. It was unclear, even for instances with customers located on the real line, whether the competitive ratio of $SMART$ strictly dominates those of $IMMEDIATE$ and $DELAY$ when the number of time periods becomes large.

In this paper, we settle that question by showing that the competitive ratio of $SMART$ for instances with customers located on the real line is $\frac{3}{2}$ for any number of time periods. Since it is known [1] that the competitive ratio of any on-line algorithm cannot be less than $\sqrt{2}$ (≈ 1.41), this result shows that $SMART$ is a simple and effective dispatch policy.

2 A Dynamic Multi-Period Routing Problem

The dynamic multi-period routing problem (DMPRP) studied in this paper is defined as follows. At the beginning of time period $t \in \{0, \ldots, T\}$, where T is the horizon length, a set of customers $C_{t|t+1}$ arrives, which can be served either in time period t or in time period $t+1$. A single vehicle is available in each time period to serve customers; there is no capacity constraint. We specify an instance C by $\{C_{0|1}, C_{1|2}, \ldots, C_{T-1|T}, C_{T|T+1}\}$. The problem is to decide in each time period $t \in \{0, \ldots, T\}$ which subset of the customers in $C_{t|t+1}$ to serve in time period t and, thus, which subset of the customers in $C_{t|t+1}$ to postpone to time period $t+1$. The sequence of decisions has to be taken in such a way that the average cost per time period is minimized. The cost per time period is taken to be the distance traveled in that period, i.e., when the set of customers to be served in a time period has been decided, then the length of the optimal traveling salesman tour through these customers determines the cost.

2.1 General results

Before focusing on instances with customer locations on the non-negative part of the real line, we present a few properties that hold as long as the customer locations are in a metric space, i.e, whenever the triangle inequality on distances holds.

Property 1 *When two sets of customers can be served either in the same time period or in two different time periods, it is always preferable to serve all customers in the same time period, e.g., on a single route, and to serve no customers in the other time period.*

Property 2 *Delaying service of all customers in* $C_{0|1}$ *to time period* 1 *is never more costly than any alternative choice. Similarly, serving all customers in* $C_{T|T+1}$ *in time period* T *is never more costly than any alternative choice.*

For the remainder of the paper, we assume that the decisions suggested in Property 2 are always applied. Consequently, no customers will be served in time periods 0 and $T+1$. This implies that for an instance with horizon length T, there are only $T-1$ decisions that have to be made (in time periods $\{1, \ldots, T-1\}$).

2.1.1 Instance splitting

Next, we define an operation that splits an instance into two smaller instances. Let C be an instance with horizon length $T > 1$, let $t \in \{1, \ldots, T-1\}$ be a time period in which a decision has to be made, and let B be a subset of $C_{t|t+1}$. The smaller instances are constructed as follows: $C^1_{t,B} = \{C_{0|1},\, C_{1|2},\, \ldots,\, C_{t-1|t},\, C_{t|t+1} \setminus B\}$ and $C^2_{t,B} = \{B, C_{t+1|t+2}, \ldots, C_{T-1|T}, C_{T|T+1}\}$. For notational convenience and when it is clear from the context, we will often drop the subscripts t and B and say that instance C is split into instances C^1 and C^2. Due to Property 2, the following hold:

- The two instances C^1 and C^2 have horizon length equal to t and $T-t$, respectively. Both horizon lengths are strictly less than T.
- The two instances C^1 and C^2 require $t-1$ and $T-t-1$ decisions, respectively. This reduces the total number of decisions from $T-1$ to $T-2$.
- The set of customers $C_{t|t+1} \setminus B$ in instance C^1 will be served in time period t.
- The set of customers B in instance C^2 will be served in time period $t+1$.

Lemma 3 *Let* C *be an instance with horizon length* $T > 1$*, let* $t \in \{1, \ldots, T-1\}$*, and let* $B \subset C_{t|t+1}$ *be the set of customers postponed by Algorithm* A *in time period* t*. Then* $z(A, C) = z(A, C^1_{t,B}) + z(A, C^2_{t,B})$.

Proof. Algorithm A results in the same costs in each time period $\{1, \ldots, T\}$ whether applied to C or to C^1 and C^2. Indeed, instances C and C^1 are identical in time periods $\{0, \ldots, t-1\}$ and Algorithm A will therefore take the same decisions for both instances in time periods 1 to $t-1$. Moreover, Algorithm A serves the customers in $C_{t|t+1} \setminus B$ in time period t in both cases. For instance C, this follows from the assumptions of the lemma and for instance C^1 this is true by construction. Similarly, Algorithm A serves the customers in B in time period $t+1$ in both cases. For instance C, this follows from the assumptions of the lemma and for instance C^2 this is true by construction. Since the conditions are identical for the remaining time periods Algorithm A results in the same decisions in time periods $\{t+1, \ldots, T\}$ for instance C and C^2. ∎

Lemma 4 *Let* C *be an instance with horizon length* $T > 1$ *such that in an optimal off-line solution and in some time period* $t \in \{1, \ldots, T-1\}$ *a set* B *of customers is postponed to time period* $t+1$*. Then* $z^*(C) = z^*(C^1_{t,B}) + z^*(C^2_{t,B})$.

Proof. Let S be a feasible sequence of decisions for instance C, and, with a little abuse of notation, let $z(S)$ denote the cost resulting from implementing the decision sequence S.

Now, given a pair S^1 and S^2 of feasible decision sequences for instances C^1 and C^2 (for time periods $\{1,\ldots,t-1\}$ and $\{t+1,\ldots,T-1\}$, respectively), we observe that the sequence of decisions S, resulting from concatenating decision sequence S^1, the decision to postpone customers in set B in time period t, and decision sequence S^2, is a feasible sequence of decisions for instance C, and its cost equals the sum of the cost of decision sequences S^1 and S^2. That is $z(S) = z(S^1) + z(S^2)$.

If S^1 and S^2 are the off-line optimal decision sequences for C^1 and C^2, then $z^*(C) \leq z(S) = z(S^1) + z(S^2) = z^*(C^1) + z^*(C^2)$.

On the other hand, the assumptions of the lemma state that an optimal off-line decision sequence for C postpones customers in set B in time period t. Thus, the optimal decision sequences S^1 in time periods $\{1,\ldots,t-1\}$ and S^2 in time periods $\{t+1,\ldots,T-1\}$ are feasible decision sequences for instances C^1 and C^2. The inequality $z^*(C^1) + z^*(C^2) \leq z(S^1) + z(S^2) = z^*(C)$ follows. ■

Lemma 5 *Let C be an instance with horizon length $T > 1$ such that the set of customers B postponed by Algorithm A in time period $t \in \{1,\ldots,T-1\}$ is identical to the set of customers postponed in an optimal off-line solution. Then splitting instance C at time t based on set B results in instances $C^1_{t,B}$ and $C^2_{t,B}$ with the following property:*

$$\frac{z(A,C)}{z^*(C)} \leq max\left(\frac{z(A,C^1_{t,B})}{z^*(C^1_{t,B})}, \frac{z(A,C^2_{t,B})}{z^*(C^2_{t,B})}\right).$$

Proof. A direct consequence of Lemmas 3 and 4. ■

Note that Lemma 5 implies that if C is an instance with horizon length $T > 1$ such that there exists a time period $t \in \{1,\ldots,T-1\}$ in which the set of customers postponed by Algorithm A is identical to the set of customers postponed in an optimal off-line solution, then there exists an instance $\widetilde{C}$ with horizon length $\widetilde{T} < T$ such that

$$\frac{z(A,C)}{z^*(C)} \leq \frac{z(A,\widetilde{C})}{z^*(\widetilde{C})}.$$

3 The non-negative real line

We consider here the case where customers are located on the non-negative real line ($\mathbb{R}^+$) and the starting and ending location of the vehicle route is the origin. Thus, the length of a route visiting a set of customers is twice the distance to the farthest (rightmost) customer. The length of a route visiting two sets of customers A and B equals the maximum length of the routes that visit each set of customers independently. That is, $L_{A,B} = \max(L_A, L_B)$, where L_X is the length of the route that visits all customers of the set X.

When customers are located on $\mathbb{R}^+$, a decision to serve a subset of the customers in $C_{t|t+1}$ at time t and the remaining customers at time $t+1$ is dominated by one of two decisions: serve all customers in $C_{t|t+1}$ in time period t or postpone all of them to time period $t+1$. In other words, it is never beneficial to split the set of customers $C_{t|t+1}$.

The following theorems were shown in Angelelli et al. [1].

Theorem 6 *[1] The competitive ratio of any algorithm for DMPRP with customer locations on the non-negative real line and with a planning horizon $T = 2$ is greater than or equal to $\sqrt{2}$ ($\approx$ 1.41).*

Theorem 7 *[1] The competitive ratio of any algorithm for DMPRP with customer locations on the non-negative real line and with a planning horizon $T > 2$ is greater than or equal to 1.44.*

As the competitive ratio of an algorithm involves comparisons of the value of the solution produced by the algorithm with the value of an optimal off-line solution, we next study the structure of optimal off-line solutions.

3.1 The structure of off-line optimal solutions

A solution, i.e., a sequence of dispatch decisions, can be represented by a string of $T-1$ symbols I and D, where I and D represent the decision to serve new customers immediately ($IMMEDIATE$) and the decision to postpone the service of new customers ($DELAY$). For example, the string $IDDI$ represents a solution for an instance with five time periods (decisions taken in time periods $t = 1, \dots, 4$). We characterize optimal off-line solutions in terms of such decision strings or decision sequences.

We will show that an optimal off-line solution can be represented by a string of alternating symbols I and D, i.e., a sequence of alternating decisions to dispatch immediately and to delay dispatching, with possible exceptions only in those time periods where the decision is immaterial, i.e., dispatching immediately or postponing dispatching result in the same costs, which will be represented by the symbol X. Some examples of possible optimal off-line solutions are: IXD ($T = 4$), $DIXXXD$ ($T = 7$), and $IDID$ ($T = 5$).

Let us consider any solution. A pair of two consecutive identical decisions can be replaced by a pair of alternating decisions without increasing the cost of the solution according to the following rules:

$$II \rightarrow ID, \;\; DD \rightarrow ID.$$

This simple observation allows us to transform the string representing any optimal off-line solution into a string representing an alternative optimal off-line solution in which decisions alternate between I and D, possibly with one or more Xs embedded in the string.

More specifically, we examine the string representing an optimal solution from left to right, examine pairs of consecutive decisions, i.e., (d_t, d_{t+1}) for times t and $t+1$, and replace them if necessary. Details are provided in Algorithm 1.

Algorithm 1 Replace Consecutive Identical Decisions

```
for t = 1, 2, ..., T − 2 do
  if (d_t, d_{t+1}) = II then
    (d_t, d_{t+1}) ← ID
  end if
  if (d_t, d_{t+1})=DD then
    (d_t, d_{t+1}) ← XD
  end if
end for
```

Observe that the last decision before a sequence of consecutive Xs is an I and the first decision after a sequence of consecutive Xs is a D.

This leads to the following property:

Property 8 *There always exists an optimal off-line solution that can be described as a string of alternating Is and Ds, possibly with one or more Xs embedded in the string. The solution remains optimal if any X is substituted with either an I or a D.*

For the remainder of the paper, we will assume that optimal off-line solutions have such a structure.

4 Analysis of Algorithm $SMART$

Let L_t be the length of the route that visits the customers that have to be served in time period t and $L_{t,t|t+1}$ the length of the tour that visits the customers that have to be served in time period t together with all the newly arrived customers.

The dispatching rule $IMMEDIATE$ serves all the customers of $C_{t|t+1}$ at time t, while the dispatching rule $DELAY$ postpones all the customers of $C_{t|t+1}$ to time $t+1$. Algorithm $SMART$, introduced by Angelelli et al. [1], decides in each time period t whether to immediately serve or to postpone the customers in $C_{t|t+1}$ by comparing $L_{t,t|t+1}$ with L_t.

Algorithm $SMART(p)$: If $L_t > 0$ and $L_{t,t|t+1} \leq pL_t$, then apply $IMMEDIATE$, else apply $DELAY$.

The quality of Algorithm $SMART$ depends on the parameter p. The following was shown in Angelelli et al. [1].

Theorem 9 *[1]*

- *For $T = 2$, $r_{SMART(1+\sqrt{2})} = \sqrt{2}$ and the algorithm is optimal.*
- *For $T = 2$, $r_{SMART(2)} = \frac{3}{2}$.*
- *For $T = 3$, $r_{SMART(2)} = \frac{3}{2}$ and $r_{SMART(2)} \leq r_{SMART(p)}$ for all other values of p.*

Here, we will extend these results to planning horizons of arbitrary length.

Main Result *The competitive ratio of Algorithm $SMART(2)$ for instances with customers on the non-negative real line and a horizon length $T > 3$ is $\frac{3}{2}$.*

Let us call a decision sequence the *opposite* of another decision sequence when it takes a different decision in every time period. Next, we define irreducible instances, which turn out to be fundamental in the proof of our main result.

Definition 10 *An instance is called* irreducible *when the decision sequence of $SMART$ alternates between I and D, ends with I, and is the opposite of the optimal off-line decision sequence. An instance is called* reducible *otherwise.*

The remainder of the paper focuses on the proof of the main result. For the benefit of the readers, we first provide a sketch of the proof. Consider an instance C with horizon length T. The proof will consider three different cases:

- The horizon length is short: $T \leq 3$. The bound $\frac{z(SMART,C)}{z^*(C)} \leq \frac{3}{2}$ is already guaranteed by Theorem 9.
- The instance is irreducible. The bound $\frac{z(SMART,C)}{z^*(C)} \leq \frac{3}{2}$ will be shown directly.
- In all other cases a "reduction" technique is applied to obtain an instance $\widetilde{C}$ with horizon length $\widetilde{T} < T$ and such that $\frac{z(SMART,C)}{z^*(C)} \leq \frac{z(SMART,\widetilde{C})}{z^*(\widetilde{C})}$. By repeatedly applying reductions, we obtain an instance $\widehat{C}$ with horizon length $\widehat{T} < T$, with $\frac{z(SMART,C)}{z^*(C)} \leq \frac{z(SMART,\widehat{C})}{z^*(\widehat{C})}$, and either $\widehat{T} \leq 3$ or $\widehat{C}$ irreducible.

Lemma 11 *Let C be an instance with horizon length $T > 3$ for which $SMART$ produces a decision sequence that is the opposite of an optimal off-line decision sequence, that alternates between I and D, and ends with D. Then an instance $C^{'}$ with horizon length $T^{'}$ exists such that (i) $T^{'} < T$, (ii) the last decision of $SMART$ is I, and (iii) $\frac{z(SMART,C)}{z^*(C)} \leq \frac{z(SMART,C^{'})}{z^*(C^{'})}$.*

Proof. Under the lemma's assumptions, we have

$$z(SMART, C) = \ldots + max(L_{T-3|T-2}, L_{T-2|T-1}) + max(L_{T-1|T}, L_{T|T+1})$$

while the optimum value is

$$z^*(C) = \ldots + max(L_{T-2|T-1}, L_{T-1|T}) + L_{T|T+1}.$$

Let $C^{'}$ be the instance obtained from C by dropping the set of customers $C_{T|T+1}$, i.e., $C' = \{C_{0|1}, C_{1|2}, \ldots, C_{T-1|T}\}$. We observe the following:

- The horizon length of $C^{'}$ is $T - 1$.
- Since instances C and $C^{'}$ are identical up to time period $T-2$, $SMART$ takes the same decisions up to time period $T-2$, i.e., $SMART$ alternates between I and D and its last decision is I.
- The total cost resulting from the application of $SMART$ to instance C' is

  $$z(SMART, C^{'}) = \ldots + max(L_{T-3|T-2}, L_{T-2|T-1}) + L_{T-1|T}$$

  and thus

  $$z(SMART, C) \leq z(SMART, C^{'}) + L_{T|T+1}.$$

The decision sequence $S^{'} = \ldots ID$ obtained from the optimal off-line decision sequence S^* for C by dropping the decision I in time period $T-1$ is a feasible and suboptimal decision sequence for $C^{'}$. Therefore, we have

$$z^*(C) = z(S^{'}) + L_{T|T+1} \geq z^*(C^{'}) + L_{T|T+1}.$$

In conclusion,

$$\frac{z(SMART, C)}{z^*(C)} \leq \frac{z(SMART, C^{'}) + L_{T|T+1}}{z^*(C') + L_{T|T+1}} \leq \frac{z(SMART, C^{'})}{z^*(C')}.$$

■

Lemma 12 *Let C be an irreducible instance with horizon length $T > 3$. Then the ratio $\frac{z(SMART,C)}{z^*(C)}$ is bounded from above by $\frac{3}{2}$.*

Proof. If the bound does not hold and an irreducible instance C with horizon length $T > 3$ exists such that $z(SMART, C) \geq (\frac{3}{2} + \varepsilon)z^*(C)$ for some $\varepsilon > 0$, then such an instance can be found as a feasible solution to a system of linear inequalities in continuous and binary variables. We start with the following set of linear inequalities:

$$\begin{aligned}
&z(SMART) \geq \ (\frac{3}{2} + \varepsilon)z^* \\
&z(SMART) = \ ... + \max(L_{T-4|T-3}, L_{T-3|T-2}) + \max(L_{T-2|T-1}, L_{T-1|T}) + L_{T|T+1} \\
&z^* = \ ... + \max(L_{T-5|T-4}, L_{T-4|T-3}) + \max(L_{T-3|T-2}, L_{T-2|T-1}) + \max(L_{T-1|T}, L_{T|T+1}) \\
&z^* \geq 0, \ z(SMART) \geq 0, \ L_{t-1|t} \geq \ 0 \text{ for } t = 1, 2, ..., T+1.
\end{aligned}$$

Next, we add inequalities to ensure that we identify an instance in which $SMART$ applies $IMMEDIATE$ in even (odd) time periods if $T-1$ is even (odd):

$$\begin{aligned}
&L_{2k|2k+1} \leq pL_{2k-1|2k} \text{ for } k = 1, ..., (T-1)/2, \text{ if } T-1 \text{ is even;} \\
&L_{2k-1|2k} \leq pL_{2k-2|2k-1} \text{ for } k = 1, ..., T/2, \text{ if } T-1 \text{ is odd.}
\end{aligned}$$

Finally, we linearize the expressions $\max(L_{t-1|t}, L_{t|t+1})$ by introducing a large constant M, continuous variables M_t ($M_t \equiv \max(L_{t-1|t}, L_{t|t+1})$) and binary variables x_t for $t = 1, ..., T$, and the following inequalities:

$$\begin{aligned}
M_t &\geq L_{t-1|t} \\
M_t &\geq L_{t|t+1} \\
M_t &\leq L_{t-1|t} + M \cdot (1 - x_t) \\
M_t &\leq L_{t|t+1} + M \cdot x_t \\
M_t &\geq 0 \\
x_t &\in \{0, 1\}.
\end{aligned}$$

Proving the lemma is equivalent to proving that the system of linear inequalities in continuous and binary variables defined above has no feasible solution for any positive ε and any number of time periods T.

We prove that the system of linear inequalities in continuous and binary variables defined above has no feasible solution for any positive ε and any number of time periods T by showing that the system of linear inequalities in continuous variables that results when the values of the binary variables (i.e., the x variables) are known has no feasible solution *independent* of the choice of values of the binary variables. Recall Farkas' Lemma:

Farkas' Lemma *Either the system* $Ax = b, x \geq 0$ *is non-empty or (exclusively) the system* $vA \geq 0$ *and* $vb < 0$ *is non-empty.*

Assume T is odd. Let x_t, $t = 1, ..., T$, represent an *arbitrary* choice indicating how the maximum M_t is achieved. Consider the following set of linear inequalities.

$$(1.5 + \epsilon)z^* - z(SMART) \leq 0 \quad (u_1)$$

$$z(SMART) - L_{0|1} - \sum_{\{t=1,...,T|t \text{ even}\}} M_t - L_{T|T+1} = 0 \quad (u_2)$$

$$z^* - \sum_{\{t=1,...,T|t \text{ odd}\}} M_t = 0 \quad (u_3)$$

$$z^* = 1 \quad (u_4)$$

$$2L_{0|1} - L_{1|2} \leq 0 \quad (v_1)$$

$$-2L_{t-1|t} + L_{t|t+1} \leq 0 \qquad t = 1, ..., T, \ t \text{ even} \quad (v_t)$$

$$L_{t-1|t} - M_t \leq 0 \qquad t = 1, ..., T \quad (w_{t,1})$$

$$L_{t|t+1} - M_t \leq 0 \qquad t = 1, ..., T-1 \quad (w_{t,2})$$

$$M_t - L_{t-1|t} \leq (1 - x_t) \qquad t = 1, ..., T \quad (w_{t,3})$$

$$M_t - L_{t|t+1} \leq x_t \qquad t = 1, ..., T-1, \quad (w_{t,4})$$

where in parentheses we show the name of the dual variable we associate with each inequality. Observe that imposing $z^* = 1$ has the effect of scaling the solution and that, because of this scaling, the large constant M used when defining the maximum can be taken equal to 1. By introducing slack variables, we can obtain a formulation in the form required by the variant of Farkas' Lemma stated above. The effect of introducing a slack variable in an inequality is that the dual variable associated with that inequality has to be nonnegative.

The corresponding dual set of inequalities is

$$-u_1 + u_2 \geq 0 \quad (z(SMART))$$

$$(1.5 + \epsilon)u_1 + u_3 + u_4 \geq 0 \quad (z^*)$$

$$-u_2 + 2v_1 + w_{1,1} - w_{1,3} \geq 0 \quad (L_{0|1})$$

$$-v_1 - 2v_2 + w_{1,2} - w_{1,4} + w_{2,1} - w_{2,3} \geq 0 \quad (L_{1|2})$$

$$v_t + w_{t,2} - w_{t,4} + w_{t+1,1} - w_{t+1,3} \geq 0 \qquad t = 2, ..., T-1, \ t \text{ even} \quad (L_{t|t+1})$$

$$-2v_{t+1} + w_{t,2} - w_{t,4} + w_{t+1,1} - w_{t+1,3} \geq 0 \qquad t = 2, ..., T-1, \ t \text{ odd} \quad (L_{t|t+1})$$

$$-u_2 + w_{T,2} - w_{T,4} \geq 0 \quad (L_{T|T+1})$$

$$-u_3 - w_{t,1} - w_{t,2} + w_{t,3} + w_{t,4} \geq 0 \qquad t = 1, ..., T, \ t \text{ odd} \quad (M_t)$$

$$-u_2 - w_{t,1} - w_{t,2} + w_{t,3} + w_{t,4} \geq 0 \qquad t = 1, ..., T, \ t \text{ even} \quad (M_t)$$

$$u_4 + \sum_{t=1,\ldots,T|x_t=0} w_{t,3} + \sum_{t=1,\ldots,T|x_t=1} w_{t,4} < 0$$

$$u_1 \geq 0, v_t \geq 0, t = 1, ..., T, w_{t,k} \geq 0, t = 1, ..., T, k = 1, 2, 3, 4.$$

Note that the dual set of inequalities is independent of the values of x_t except for the *less-than* inequality.

By substitution, it can easily be verified that the dual system is non-empty for any binary x-vector as the following solution is feasible

$$u_1 = u_2 = 1.0,\ u_3 = -1.5,\ u_4 = -\delta\ (\delta \leq \epsilon)$$

$$t = 1 : v_1 = 0.5,\ w_{1,2} = 1.5$$

$$t = 3, ..., T, odd : w_{t,1} = 0.5,\ w_{t,2} = 1.0$$

$$t = 1, ..., T, even : w_{t,3} = 1.0 \text{ if } x_t = 1$$

$$t = 1, ..., T, even : v_t = 0.5,\ w_{t,4} = 1.0 \text{ if } x_t = 0.$$

The case T is even is shown analogously. ■

We our now ready to prove our main result.

Theorem 13 *The competitive ratio of Algorithm $SMART(2)$ for instances with customers on the non-negative real line and a horizon length T is $\frac{3}{2}$.*

Proof. From the third claim of Theorem 9, we know that $r_{SMART(2)} \geq \frac{3}{2}$. Thus, in order to show that the competitive ratio of $SMART(2)$ is $\frac{3}{2}$, we need to show that $r_{SMART(2)}$ is bounded from above by $\frac{3}{2}$.

Let C be an instance with horizon length T.

- If $T \leq 3$, then the bound follows from Theorem 9.
- If $T > 3$ and C is irreducible, then the bound is shown in Lemma 12.
- If $T > 3$ and C is reducible, then by repeatedly splitting the instance, we can bound the ratio $\frac{z(SMART,C)}{z^*(C)}$ from above by the ratio $\frac{z(SMART,C')}{z^*(C')}$, where instance C' has a horizon length less than or equal to 3 or is irreducible. To be more precise, consider the following cases:
  - The decision sequence produced by $SMART$ alternates between I and D but is not the opposite of an optimal off-line decision sequence. In this case, we apply Lemma 5.
  - The decision sequence produced by $SMART$ alternates between I and D, ends with D, and is the opposite of an optimal off-line decision sequence. In this case, we apply Lemma 11.
  - The decision sequence produced by $SMART$ does not alternate between I and D. In this case, we consider an optimal off-line decision sequence. If the sequence contains Xs, we replace them with the corresponding decisions taken by $SMART$. Next, we apply Lemma 5 repeatedly until we reach a situation in which the decision sequence produced by $SMART$ alternates between I and D and is the opposite of the optimal off-line decision sequence. Then, if the decision sequence produced by $SMART$ ends with D, we can apply Lemma 11. Otherwise, we have an irreducible instance and apply Lemma 12.

■

Open Questions

- Is the competitive ratio of Algorithm *SMART* when applied to instances with customers in the Euclidean plane bounded from above by a value that is strictly less than 2?
- Consider a variant of the problem in which the customer locations are on the non-negative real line and are known to be drawn randomly from a uniform distribution between 0 and 1. What is the optimal dispatch policy?

References

[1] E. Angelelli, M.W.P. Savelsbergh, M.G. Speranza. Competitive Analysis for Dynamic Multi-Period Uncapacitated Routing Problems. *Networks.* To appear.

[2] G. Ausiello, E. Feuerstein, S. Leonardi, L. Stougie, M. Talamo. Algorithms for the on-line travelling salesman. *Algorithmica 29*, 560-581, 2001.

[3] M. Blom, S.O. Krumke, W.E. de Paepe, L. Stougie. The online TSP against fair adversaries. *INFORMS Journal on Computing 13*, 138-148, 2001.

[4] S. Irani, X. Lu, A.C. Regan. On-line algorithms for the dynamic traveling repair problem. *Journal of Scheduling 7*, 243-258, 2004.

[5] P. Jaillet, M. Wagner. Online routing problems: Value of advanced information and improved competitive ratios. *MIT Working paper*, 2005.

[6] S.O. Krumke, W.E. de Paepe, D. Poensgen, L. Stougie. News from the online traveling repairman. *Theoretical Computer Science 295*, 279-294, 2003.

[7] H. Psaraftis. Dynamic vehicle routing problems. In *Vehicle Routing: Methods and Studies*, B. Golden and A. Assad (eds.), North Holland, Amsterdam, 223-248, 1988.

[8] D. Sleator, R.E. Tarjan. Amortized efficiency of list update and paging rules. *Communications of ACM 28*, 202-208, 1985.