Konrad-Zuse-Zentrum
für Informationstechnik Berlin

Takustraße 7
D-14195 Berlin-Dahlem
Germany

BENJAMIN HILLER
SVEN O. KRUMKE[1]
SLEMAN SALIBA[1]
ANDREAS TUCHSCHERER

Randomized Online Algorithms for Dynamic Multi-Period Routing Problems

[1]University of Kaiserslautern, Department of Mathematics, P. O. Box 3049, Paul-Ehrlich-Str. 14, 67653 Kaiserslautern, Germany. {krumke,saliba}@mathematik.uni-kl.de

ZIB-Report 09-03 (Februar 2009)

Randomized Online Algorithms for Dynamic Multi-Period Routing Problems

Benjamin Hiller† Sven O. Krumke* Sleman Saliba* Andreas Tuchscherer†

Abstract

The *Dynamic Multi-Period Routing Problem* (DMPRP) introduced by Angelelli et al. [1] gives a model for a two-stage online-offline routing problem. At the beginning of each time period a set of customers becomes known. The customers need to be served either in the current time period or in the following. Postponed customers have to be served in the next time period. The decision whether to postpone a customer has to be done online.

At the end of each time period, an optimal tour for the customers assigned to this period has to be computed and this computation can be done offline. The objective of the problem is to minimize the distance traveled over all planning periods assuming optimal routes for the customers selected in each period.

We provide the first randomized online algorithms for the DMPRP which beat the known lower bounds for deterministic algorithms. For the special case of two planning periods we provide lower bounds on the competitive ratio of any randomized online algorithm against the oblivious adversary. We identify a randomized algorithm that achieves the optimal competitive ratio of $\frac{1+\sqrt{2}}{2}$ for two time periods on the real line. For three time periods, we give a randomized algorithm that is strictly better than any deterministic algorithm.

1 Introduction

Online optimization deals with optimization problems for which the information get known over time, i. e., the optimization problems are *online* or *dynamic*. Classical approaches for online optimization model this as a sequence of requests, where the decision maker has to make an irrevocable decision on how to serve a request before the next request becomes known.

In some practical applications the situation is different: Only some part of the decision is irrevocable, whereas some aspects can still be chosen to optimize overall performance [6]. An example is a company which provides technical service to customers who can require service at a call center. The company makes an appointment with the customer on which day a technician will arrive, but not the exact time of day. This flexibility allows the company to come up with good routes for its technicians. However, the day of service may not be altered any more and is thus an irrevocable decision.

Another example is a transportation company providing a delivery service to its customers. Due to some service level agreements, delivery has to be within a certain time interval, say two days. Thus if a customer demands delivery, the company is free to choose whether to serve the customer the same day or the following. Once this is decided it may not be changed any more. The freedom to choose the day of service and the possibility to use optimized routes for each day enables the company to operate efficiently. Note that in both example applications it is rather important to make the irrevocable decision carefully in order to exploit the freedom left.

To study the last problem, Angelelli et al. [1, 2] introduced the *Dynamic Multi-Period Routing Problem* (DMPRP), which we will study in this paper. In this problem, a set of customers becomes known at beginning of each time period. These customers have to be served either in that time period or in the following. At the beginning of each time period decisions have to be made based on the knowledge of the location of the customers that were postponed from the last time period and the location of the customers that have just arrived. No information about future customers are known. Therefore decisions have to be made at each time period based on incomplete knowledge.

*University of Kaiserslautern, Department of Mathematics, P. O. Box 3049, Paul-Ehrlich-Str. 14, 67653 Kaiserslautern, Germany. {krumke,saliba}@mathematik.uni-kl.de

†Konrad-Zuse-Zentrum für Informationstechnik Berlin, Department Optimization, Takustr. 7, 14195 Berlin, Germany {hiller,tuchscherer}@zib.de

	lb	ub	rand-lb	rand-ub
$T=2$	$\sqrt{2}$[6, 1]	$\sqrt{2}$[1]	1.2071	1.2071
$T=3$	1.44[1]	1.473[1]	1.2071	7/5
$T\geq 2$	3/2[1]	3/2[2]	1.2071	–

(a) Results for the real line.

	lb	ub	rand-lb	rand-ub
non-splitting				
$T=2$	3/2[1]	3/2[1]	3/2	3/2
splitting				
$T=2$	3/2[1]	3/2[1]	5/4	–

(b) Results for the Euclidean plane.

Table 1: Overview on known and new results for the Dmprp. Superscripts indicate the original paper, results without notes are new in this paper.

Following Angelelli et al. [1, 2], we analyze the performance of algorithms for the Dmprp using *competitive analysis* [3, 5].

Related work Angelelli et al. [1] introduced the Dmprp, for which they proposed and studied deterministic algorithms. They considered customers on the nonnegative real line or in the Euclidean plane and showed that the simple strategies immediate and delay have a competitive ratio of 2 for an arbitrary number of periods. Moreover, they proposed a more sophisticated algorithm smart that has a competitive ratio of $\sqrt{2}$ for two time periods and customers on the nonnegative real line. In a subsequent paper [2] they showed that smart is 3/2-competitive on the nonnegative real line for an arbitrary number of periods.

The Dmprp is closely related to the Online Target Date Assignment Problem (OnlineTDAP) introduced by Heinz et al. in [6] to model applications similar to the first one of the introduction. It is a general framework for two stage online-offline problems each of which is characterized by a so called *downstream problem*, e. g., the Traveling Salesman Problem (TSP). Each request r_i has a release date $t(r_i)$ and must be assigned online and immediately to a target date in the time period $t(r_i)+1,\dots,t(r_i)+\delta$, for some $\delta \geq 1$. All requests assigned to the same date constitute an instance of the downstream problem, which is assumed to be solved to optimality offline. The goal is to assign target dates to the requests such that the total cost of the downstream problems over all dates is as small as possible. In [6] it was already proven that $\sqrt{2}$ is a lower bound on the competitive ratio for any deterministic online algorithm. Moreover, a 2-competitive algorithm was proposed. It is easy to see that the Dmprp is a restricted version of the OnlineTDAP ($\delta = 2$) with downstream problem TSP, where all requests at one date are given together.

Our results We study the first randomized online algorithms for the Dmprp (and the OnlineTDAP). We provide lower bounds for randomized algorithms and for non-splitting algorithms and propose a family of randomized algorithms for the Dmprp. On the real line, one of these algorithms is optimal for two time periods and another one is strictly better than any deterministic algorithm for three time periods. An overview on our results is given in Table 1.

In the next section we formally define the considered problem, provide background on online optimization and give some basic structural results. In Section 3 we provide lower bounds for randomized algorithms on the real line and the Euclidean plane. In Section 4 we introduce our new algorithm rsmart and analyze its performance for two and three planning periods.

2 Problem definition and preliminaries

Let $M = (X, d)$ be a metric space where X is a set and $d\colon X \times X \to \mathbb{R}^+$ is a metric satisfying the triangle inequality. In this paper the space M is induced either by the real line or by the Euclidean plane.

An instance of the Dynamic Multi-Period Routing Problem Dmprp(M) in the metric space M consists of a sequence $\sigma = \left(C_1, C_{1|2}, C_{2|3}, \dots, C_{T-1|T}, C_T\right)$ of customer sets, where a customer is a point in the metric space. There are T planning periods $1,\dots,T$. Furthermore, we presume that a server is located at a depot $o \in X$. In each period, the server serves a set of customers by visiting the points in the metric space on a shortest tour starting and ending at the depot. The customer set of the next period is released only after the server has returned to the depot.

At the beginning of time period one, the decision maker gets to know the customer sets C_1 and $C_{1|2}$. It may choose an arbitrary subset of $C_{1|2}$ to be served together with C_1 in the first period. The remaining customers from $C_{1|2}$ have to be served in time period 2, together with a subset of customers from $C_{2|3}$. In general, the decision maker chooses a subset of $C_{t|t+1}$ in period t, which is served in that period with the unserved customers from period $t-1$. This continues up to period T, in which all remaining customers have to be served together with C_T. The decision maker does not know T in advance. Note that the customers released in time period t have to be served in time period t or $t+1$, with the exception of customers in C_1 and C_T that have to be served in time periods 1 and T, respectively.

The cost of serving a customer set corresponds to the length of an optimal tour traveled in order to visit the customers of this set. We denote this length by L_1 and L_T for customer sets C_1 and C_T, respectively, and $L_{t|t+1}$ for customer sets $C_{t|t+1}$, $t = 1, \dots, T-1$.

In this paper we only consider two planning periods, i. e., $T = 2$, unless otherwise mentioned. For both the real line and the Euclidean plane we assume that the depot is located at the origin. While the case of two or three planning periods may seem restricted, it already captures the spirit of "delay or not-to-delay" inherent in many online problems such as the ski-rental problem, the page-replication problem or the TCP-acknowledgement problem [4].

We call an online algorithm that serves $C_{t|t+1}$ entirely in period t or period $t+1$ a *non-splitting* algorithm and algorithms without this property *splitting* algorithms. All algorithms for the DMPRP we are aware of are non-splitting.

2.1 Online algorithms and competitive analysis

We evaluate the considered online algorithms using *competitive analysis*. An online algorithm ALG is called *c-competitive* if the cost of the solution produced by ALG on any input sequence is at most c times that of an optimal offline solution for the same input. *Competitive analysis* can be seen as a game between an online player choosing an online algorithm to process an input sequence and an offline adversary generating this input sequence such that the ratio between the cost of the online player and the optimal offline cost is maximized. Note that the offline adversary knows the deterministic strategy of the online player.

A *randomized online algorithm* is a probability distribution over a set of deterministic online algorithms, where instead of sticking to one deterministic algorithm the online player chooses randomly from a set of available online algorithms according to a given probability distribution. The adversary considered in this paper is known as *oblivious adversary* and knows the description of the online algorithm (including the probability distribution) but has to construct the request sequence in advance before any moves are made. A randomized online algorithm is called *c-competitive*, if the *expected* cost of the solution produced by ALG on an input sequence is at most c times that of an optimal offline solution for that input.

The infimum of all c s. t. an online algorithm is c-competitive is called the *competitive ratio* of the algorithm. In order to prove lower bounds for *any* randomized online algorithm the following result known as Yao's Principle is often useful (see e. g., [3]).

Theorem 2.1 (Yao's Principle) *Let $\{\sigma_x \mid x \in X\}$ be the set of all possible input sequences and $\{\text{ALG}_y \mid y \in Y\}$ denote the set of deterministic online algorithms for an online minimization problem. If $\bar{X}$ is a probability distribution over all input sequences and $\bar{c} \geq 1$ is a real number such that*

$$\inf_{y \in Y} \mathbb{E}_{\bar{X}}[\text{ALG}_y(\sigma_x)] \geq \bar{c}\, \mathbb{E}_{\bar{X}}[\text{OPT}(\sigma_x)],$$

then $\bar{c}$ is a lower bound on the competitive ratio of any randomized online algorithm against the oblivious adversary.

2.2 The DMPRP on the real line

Most of our results in this paper are for customers sitting on the nonnegative real line. The following result shows that the real line and the nonnegative real line are essentially equivalent for the Dynamic Multi-Period Routing Problem. The result generalizes the corresponding one in [1].

Proposition 2.2 *If there is a (randomized) c-competitive online algorithm* Alg *for* Dmprp($\mathbb{R}^+$) *then there is a (randomized) online algorithm* $\overline{\text{Alg}}$ *that is c-competitive for* Dmprp($\mathbb{R}$). *In particular, if* Alg *has the optimal competitive ratio c, then* $\overline{\text{Alg}}$ *has also competitive ratio c and is optimal, too.*

Proof. Suppose Alg is c-competitive for Dmprp($\mathbb{R}^+$). Based on Alg, define algorithm $\overline{\text{Alg}}$ for the real line as follows. In each time period t, $\overline{\text{Alg}}$ considers the customers postponed from the last period and the new customers separately for $\mathbb{R}^+$ and $\mathbb{R}^-$, using Alg to decide for each subset which customers to postpone or not. It then serves the customers that are not postponed by Alg. Obviously, we have $\overline{\text{Alg}}(\sigma) = \overline{\text{Alg}}(\sigma^+) + \overline{\text{Alg}}(\sigma^-)$.

Let σ be an arbitrary input sequence for the real line. Note that σ can be decomposed in the customer subsequences σ^+ (customers on the nonnegative real line) and σ^- (customers on the negative real line). Given solutions for σ^+ and σ^-, these can be combined to a solution for σ with total length at most the sum of the total lengths of σ^+ and σ^-, which implies $\text{Opt}(\sigma) \le \text{Opt}(\sigma^+) + \text{Opt}(\sigma^-)$. On the other hand, a solution for σ induces solutions for σ^+ and σ^- with the same total length, since in each time period, the server needs to return to the depot before serving customers on the other side. This implies $\text{Opt}(\sigma) \ge \text{Opt}(\sigma^+) + \text{Opt}(\sigma^-)$, so we have $\text{Opt}(\sigma) = \text{Opt}(\sigma^+) + \text{Opt}(\sigma^-)$.

From the above observations, we have

$$\frac{\overline{\text{Alg}}(\sigma)}{\text{Opt}(\sigma)} = \frac{\overline{\text{Alg}}(\sigma^+) + \overline{\text{Alg}}(\sigma^-)}{\text{Opt}(\sigma^+) + \text{Opt}(\sigma^-)} \le \max\left\{\frac{\overline{\text{Alg}}(\sigma^+)}{\text{Opt}(\sigma^+)}, \frac{\overline{\text{Alg}}(\sigma^-)}{\text{Opt}(\sigma^-)}\right\},$$

hence the algorithm $\overline{\text{Alg}}$ is $\bar{c}$-competitive, too. The second statement follows by observing that a lower bound for Dmprp($\mathbb{R}^+$) is a lower bound for Dmprp($\mathbb{R}$), too. □

Consider the Dmprp($\mathbb{R}^+$) and a customer set $C_{t|t+1}$ and let $a, b \in C_{t|t+1}$ be two different customers with $b > a$. If b is selected to be served in period t, then it does not increase the cost if a is served in period t, too. However, if b is postponed to period $t+1$, a can be served in period $t+1$ for free. Thus it is sufficient to consider only the maximum of $C_{t|t+1}$ to decide whether to serve or postpone all customers of $C_{t|t+1}$. Note that this implies that on the (nonnegative) real line, non-splitting algorithms are as powerful as splitting algorithms.

Suppose $C_{t|t+1}$ is empty. We can then add a customer at the origin to $C_{t|t+1}$ without increasing the cost. Thus we can assume that all customer sets are singleton sets. Moreover, we can assume $C_1 = \{1\}$ just by scaling all other elements appropriately. This proves the following result and justifies to consider only specially structured customer sequences when studying online algorithms for $\mathbb{R}^+$. We will use the shorter notation $\sigma = (1, c_1, \ldots, c_T)$ for them.

Proposition 2.3 *If a (randomized) online algorithm* Alg *for* $\mathbb{R}^+$ *is c-competitive for customer sequences of the form*

$$\sigma = \big(\{1\}, \{c_2\}, \ldots, \{c_T\}\big),$$

then it is c-competitive for every customer sequence.

3 Lower bounds for randomized algorithms

We derive our lower bounds already for the case of wo planning periods, i. e., $T = 2$. Therefore, there are only three different sets of customers: The sets C_1 and $C_{1|2}$ of customers are known in period 1 and the set C_2 of customers whose positions become known at the beginning of the second time period. Since set C_1 has to be served in period 1, the only decision to be made is which customers of set $C_{1|2}$ should be served immediately and which should be postponed to period 2.

In this section we give lower bounds on the competitive ratio of randomized algorithms for the nonnegative real line and the Euclidean plane. To do this, we need to consider different classes of online algorithms, depending on which customers are selected to be served in the first period. To simplify language, we will talk of algorithms, but each of those algorithms represents a class of algorithms which all do the same in the first period.

Theorem 3.1 *Any randomized online algorithm for* Dmprp($\mathbb{R}^+$) *has a competitive ratio greater or equal* $\frac{1+\sqrt{2}}{2} \approx 1.20711$ *against the oblivious adversary.*

Proof. Using Yao's Principle we can derive a lower bound for the nonnegative real line as follows. Consider an instance $\sigma = (1, a, b)$ with $a > 1$. There are two possible algorithms. The first algorithms $\textsc{Alg}_1$ serves the customers at 1 and a in period 1, whereas the second algorithm $\textsc{Alg}_2$ only serves the customer at 1 in the first period.

With probability p, b is some value greater or equal a, otherwise b is zero (meaning there is no new customer). This yields a probability distribution over the two input sequences $\sigma_1 = (1, a, b)$ (with $a \le b$) and $\sigma_2 = (1, a, 0)$.

First we will have a look at input sequence σ_1. Since $b \ge a$ it is optimal to postpone the customer at a, yielding optimal cost $\textsc{Opt}(\sigma_1) = 2 + 2b$. Algorithm $\textsc{Alg}_1$ serves the customer at a in period 1, incuring costs of $\textsc{Alg}_1(\sigma_1) = 2a + 2b$, whereas $\textsc{Alg}_2$ is optimal.

Next we consider input sequence σ_2, for which it is optimal to serve the customer at a in period 1 and to stay idle in period two, giving optimal cost $\textsc{Opt}(\sigma_2) = 2a$. Whereas $\textsc{Alg}_1$ is optimal, the cost of $\textsc{Alg}_2$ is $\textsc{Alg}_2(\sigma_2) = 2 + 2a$.

Therefore we have

$$\begin{aligned}
\mathbb{E}[\textsc{Opt}(\sigma_x)] &= p\,(2 + 2b) + (1 - p)\,2a \\
\mathbb{E}[\textsc{Alg}_1(\sigma_x)] &= p\,(2a + 2b) + (1 - p)\,2a \\
\mathbb{E}[\textsc{Alg}_2(\sigma_x)] &= p\,(2 + 2b) + (1 - p)\,(2 + 2a)
\end{aligned}$$

The maximum of the minimum cost for the two algorithms

$$\max \min \{2a + 2bp, 2 + 2a + 2bp - 2ap\}$$

in the interval $p = [0, 1]$ is attained at $p = 1/a$ yielding

$$\begin{aligned}
\mathbb{E}[\textsc{Opt}(\sigma_x)] &= \frac{2 + 2b + 2a^2 - 2a}{a}, \\
\mathbb{E}[\textsc{Alg}_1(\sigma_x)] = \mathbb{E}[\textsc{Alg}_2(\sigma_x)] &= \frac{2b + 2a^2}{a}.
\end{aligned}$$

Consequently we have for $y = 1, 2$

$$\frac{\mathbb{E}[\textsc{Alg}_y(\sigma_x)]}{\mathbb{E}[\textsc{Opt}(\sigma_x)]} = \frac{a^2 + b}{a^2 - a + b + 1}$$

with maximal value at $b = a$. Since

$$\max_{a>1} \frac{\mathbb{E}[\textsc{Alg}_y(\sigma_x)]}{\mathbb{E}[\textsc{Opt}(\sigma_x)]} = \max_{a>1} \frac{a^2 + a}{a^2 + 1} = \frac{1 + \sqrt{2}}{2} \quad \text{for } a = 1 + \sqrt{2}$$

by Yao's principle the lower bound for any randomized online algorithm for $\textsc{Dmprp}(\mathbb{R}^+)$ is greater than or equal to $(1 + \sqrt{2})/2$. Due to Proposition 2.2 this lower bounds extends to the entire real line. □

Finally we prove a lower bound on randomized online algorithms for $\textsc{Dmprp}(\mathbb{R}^2)$ with two time periods. This lower bound construction is a randomized version of the lower bound of 3/2 given by Angelelli et al. [1] for deterministic online algorithms.

Theorem 3.2 *Any randomized online algorithm for* $\textsc{Dmprp}(\mathbb{R}^2)$ *has a competitive ratio of at least 5/4 against the oblivious adversary. Moreover, any non-splitting (randomized) online algorithm has a competitive ratio not better than 3/2.*

Proof. We consider an instance in which the depot is at the origin and the customers are located on three different line segments starting from the depot. The line segments endpoints A, B and C are situated on a circle with center O and radius 1/2. The distance between endpoint A and endpoints B and C is ϵ (see Figure 1).

To construct the customer sets, we distribute customers equidistantly on a line segment. If L is a line segment, $\langle L \rangle_n$ denotes the set of n points distributed equidistantly across L, with one point at each end of L. The customer set C_1 required to be served in time period $t = 1$ is given by $C_1 = \langle OA \rangle_n$, the set $C_{1|2}$ of customers that the algorithm needs to decide upon is $C_{1|2} = \langle OB \rangle_n \cup \langle OC \rangle_n$. At the

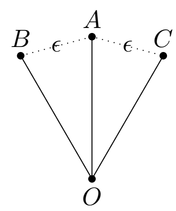

Figure 1: Instance construction for the lower bound in the Euclidean plane.

beginning of time period two, the customer set C_2 becomes known, which is $\langle OA\rangle_n \cup \langle OB\rangle_n$ with probability p and $\langle OA\rangle_n \cup \langle OC\rangle_n$ with probability $1-p$. Therefore we have a probability distribution X over the two possible request input sequences $\sigma_1 = (\langle OA\rangle_n, \langle OB\rangle_n \cup \langle OC\rangle_n, \langle OA\rangle_n \cup \langle OB\rangle_n)$ and $\sigma_2 = (\langle OA\rangle_n, \langle OB\rangle_n \cup \langle OC\rangle_n, \langle OA\rangle_n \cup \langle OC\rangle_n)$.

For input sequence σ_1 the optimal strategy is to serve the customers in C_1 and those customers of $C_{1|2}$ that lie on line segment OC in time period one, since the remaining n customers of $C_{1|2}$ can be served together with the customers of C_2, which are on line segments OA and OB. This leads to the optimal cost of $2+2\epsilon$. By symmetry, the optimal tour length for σ_2 is $2+2\epsilon$, too.

Note that depending on ϵ, if n is large enough it is always favorable to serve all customers of one line segment before switching to another line segment. It is therefore sufficient to consider the following four algorithms. The first algorithm ALG_1 serves only customers of set C_1 in the first time period and postpones all others, whereas the second algorithm ALG_2 serves all customers of the two sets C_1 and $C_{1|2}$ in time period one. Additionally, a third algorithm ALG_3 serves only the customers on line segment OB of the set $C_{1|2}$ in time period one and postpones all others. Analogously, algorithm ALG_4 serves the customers on line segment OC in the first time period and postpones those in the segment OB.

We observe that in this scenario ALG_1 and ALG_2 can never be optimal, since whatever happens at the beginning of time period two, both algorithms are forced to serve customers on all three line segments either in time period 1 (ALG_2) or in time period 2 (ALG_1). Thus they achieve cost $3+2\epsilon$, which implies a competitive ratio of at least $3/2$, establishing the second claim. Therefore we concentrate on the analysis of the remaining two algorithms.

Consider input sequence σ_1. Algorithm ALG_3 serves customers on line segment OA and OB in the first time period leaving customers on line segment OC to be served in time period two. Therefore this algorithm forces the server to traverse all three line segments in order to reach all customers in period two yielding $\text{ALG}_3(\sigma_1) = (1+\epsilon) + (2+\epsilon) = 3+2\epsilon$. On the other hand, the strategy of ALG_4 is optimal for this input sequence. By symmetry, the results for input sequence σ_2 are the same, with ALG_3 and ALG_4 interchanged.

Putting these results together we have

$$\begin{aligned}
\mathbb{E}_X[\text{OPT}(\sigma_x)] &= p(2+2\epsilon) + (1-p)(2+2\epsilon) = 2+2\epsilon \\
\mathbb{E}_X[\text{ALG}_1(\sigma_x)] &= p(3+\epsilon) + (1-p)(3+\epsilon) = 3+\epsilon \\
\mathbb{E}_X[\text{ALG}_2(\sigma_x)] &= p(3+2\epsilon) + (1-p)(3+2\epsilon) = 3+2\epsilon \\
\mathbb{E}_X[\text{ALG}_3(\sigma_x)] &= p(3+2\epsilon) + (1-p)(2+2\epsilon) = 2+p+2\epsilon \\
\mathbb{E}_X[\text{ALG}_4(\sigma_x)] &= p(2+2\epsilon) + (1-p)(3+2\epsilon) = 3-p+2\epsilon
\end{aligned}$$

For ϵ going to 0 we get a ratio of at least

$$\min\left\{\frac{3}{2}, \frac{3}{2}, \frac{2+p}{2}, \frac{3-p}{2}\right\} \geq \frac{5}{4}$$

for $p = 1/2$. Therefore, any randomized online algorithm has a competitive ratio greater than or equal to $5/4$ by Yao's Principle. □

4 Randomized algorithms

In this section we present and analyze randomized online algorithms for the DMPRP. As mentioned in previous sections, in the special case of two planning periods there is only a decision to be made for the customer set $C_{1|2}$ at the beginning of the first time period.

In [1] three deterministic online algorithms are described:

> **Algorithm** IMMEDIATE
> Visit all pending customers as soon as possible.
>
> **Algorithm** DELAY
> Delay all customers that can be postponed and only serve those that have to be served.
>
> **Algorithm** SMART$_q$
> Let L_{must} be the length of the optimal tour visiting only customers that have to be served, L_{all} the length of the optimal tour visiting all pending customers and $q \geq 1$ a real number. If $L_{\text{all}} \leq qL_{\text{must}}$, then apply IMMEDIATE otherwise apply DELAY.

In this section we will show that randomizing the decision to serve customers of set $C_{1|2}$ or to postpone them improves upon any deterministic online algorithm. A first simple idea is to choose randomly between IMMEDIATE and DELAY in each period. It turns out that using any fixed probability $p \in [0, 1]$ results in an competitive ratio of at least $3/2$. This ratio is still better than IMMEDIATE and DELAY, which have competitive ratios of 2 [1], but is only as good as SMART.

A more sophisticated strategy is to take the overhead of serving all pending customers into account. Let C_{must} be the set of customers that have to be visited in the current time period and C_{dec} the set of customers that can either be visited in the current or in the following time period. Moreover, let L_{must} be the length of the optimal tour for C_{must} and L_{all} be the length of an optimal tour for C_{dec} and C_{must}. If the ratio between L_{must} and L_{all} is small, the probability of visiting all pending customers should be higher than if the ratio is big, since a small ratio means that there is little overhead in serving the customers together. Let α be this ratio, i. e.,

$$\alpha := \frac{L_{\text{all}}}{L_{\text{must}}} \geq 1.$$

The algorithm RSMART$_f$ uses a function $f\colon \mathbb{R}^+ \to [0, 1]$ to determine the probability for choosing between IMMEDIATE and DELAY depending on α. We assume that $f(1) = 1$, since $L_{\text{all}} = L_{\text{must}}$ implies that it is optimal to serve all pending requests at once.

> **Algorithm** RSMART$_f$
> If $L_{\text{must}} = 0$ then do nothing in this period. Otherwise, let $\alpha = L_{\text{all}}/L_{\text{must}}$. With probability $p = f(\alpha)$, serve all pending customers, i. e., $C_{\text{must}} \cup C_{\text{dec}}$, and with probability $1 - p$ serve only customers of set C_{must}.

4.1 Randomized algorithms for the real line

The next result analyzes the performance of RSMART$_f$ for $T = 2$ and a constant function f. This allows us to calculate an optimal choice for $f(\alpha)$, showing that this approach actually leads to an optimal randomized online algorithm hitting the lower bound of Section 3.

Proposition 4.1 *Consider an instance of* DMPRP($\mathbb{R}^+$) *with two planning periods and let* $\alpha := L_{\text{all}}/L_{\text{must}}$. *Then, for any real number* $p \in [0, 1]$, *the algorithm* RSMART$_f$ *with* $f(\alpha) = p$ *has a competitive ratio of* c *with*

$$c = \max\left\{1 + \frac{p\,(\alpha - 1)}{\alpha + 1}, 1 + \frac{1 - p}{\alpha}\right\}. \tag{1}$$

Proof. By Proposition 2.3 we can assume that the instance is of the form $\sigma = (1, a, b)$. Since $L_{\text{must}} = 1$ and $L_{\text{all}} = \max\{1, a\}$ we have $\alpha = L_{\text{all}}$. If $a \leq 1$, RSMART$_f$ serves the customer at a immediately, which is optimal. We can thus assume that $a = \alpha \geq 1$.

Case 1: $b \in [0, 1]$

The optimal cost OPT$\,(\sigma)$ for serving this input sequence is $2a + 2b$, the randomized online algorithm on the other hand serves the customer at a in the first time period with probability p and delays it with probability $1-p$ implying expected costs of $\mathbb{E}[\text{RSMART}_f(\sigma)] = p\,(2a+2b) + (1-p)\,(2+2a) = 2a+2pb-2p+2$. The ratio

$$\frac{\mathbb{E}[\text{RSMART}_f(\sigma)]}{\text{OPT}(\sigma)} = \frac{a+pb-p+1}{a+b}$$

is a decreasing function in b. Choosing $b=0$, therefore, maximizes the ratio.

Case 2: $b \in [1,a]$

Since the customer at b can be served without additional cost if the customer at a is delayed, the optimal cost is $\text{OPT}\,(\sigma) = 2+2a$. The expected cost of our randomized online algorithm is again $\mathbb{E}[\text{RSMART}_f(\sigma)] = 2a+2pb-2p+2$. This time the ratio

$$\frac{\mathbb{E}[\text{RSMART}_f(\sigma)]}{\text{OPT}(\sigma)} = \frac{a+pb-p+1}{a+1}$$

is increasing for b yielding a maximal ratio for $b=a$.

Case 3: $b \geq a$:

In this scenario it is optimal to delay the customer at a, since it can be served without additional cost while traveling to b implying optimal costs of $\text{OPT}\,(\sigma) = 2+2b$. The randomized online algorithm, however, serves a with probability p yielding expected costs of $\mathbb{E}[\text{RSMART}_f(\sigma)] = p\,(2a+2b) + (1-p)\,(2+2b) = 2ap+2b-2p+2$. The resulting ratio

$$\frac{\mathbb{E}[\text{RSMART}_f(\sigma)]}{\text{OPT}(\sigma)} = \frac{ap+b-p+1}{b+1}$$

shrinks for growing b, therefore the smallest possible value for b, namely $b=a$, maximizes this ratio.

Putting these results together and using $a=\alpha$ we get the result that the competitive ratio of algorithm $\text{RSMART}_f(\sigma)$ for $\text{DMPRP}(\mathbb{R}^+)$ is given by (1). □

We will now show that fixing the probability parameter to a real number $p \in [0,1]$ gives a 2-competitive algorithm. The best ratio $3/2$ is attained for $p=1/2$. Furthermore, we observe that the worst competitive ratio of 2 is obtained for p fixed at 0 or 1, for which the algorithm coincides with DELAY and IMMEDIATE, respectively. However, by choosing f appropriately, RSMART_f is an optimal randomized online algorithm.

Theorem 4.2 *1. For any fixed $p \in [0,1]$, algorithm RSMART_f with $f(\alpha)=p$ is 2-competitive for $\text{DMPRP}(\mathbb{R}^+)$. The best competitive ratio of $3/2$ is attained at $p=1/2$.*

2. For $f(\alpha) = \frac{\alpha+1}{\alpha^2+1}$, algorithm RSMART_f is $\left(\frac{1+\sqrt{2}}{2}\right)$-competitive for $\text{DMPRP}(\mathbb{R}^+)$.

Proof. We know from Proposition 4.1 that the competitive ratio of RSMART_f with $f(\alpha)=p$ is is given by (1). The first ratio in (1) is an increasing function in p with value 1 for $a=1$. For a going to infinity the function tends to $1+p$. On the other hand, the second function $1+(1-p)\,/a$ is decreasing in p with value $2-p$ for $a=1$ and tending to 1 as a goes to infinity. Therefore fixing p, RSMART_p has a competitive ratio of $\max\{1+p, 2-p\}$ which is maximal for $p=0$ or $p=1$ with value 2 and minimal for $p=1/2$ with value $3/2$.

We now want to determine the value of the probability parameter $p=f(\alpha)$ that minimizes the maximum of the two ratios. Since the first ratio is an increasing function in p and the second one a decreasing function in p the maximum is minimal at the intersection of the two functions yielding $p=f(\alpha)=\frac{\alpha+1}{\alpha^2+1}$. It is easy to check that $\alpha = 1+\sqrt{2}$ is a maximizer of f, giving a competitive ratio of $\frac{1+\sqrt{2}}{2}$. □

The function f identified in Theorem 4.2 is quite involved. It is interesting to see how good RSMART_f can be with simpler functions f. We therefore study functions of the type

$$f(\alpha) = \begin{cases} 1 & 1 \leq \alpha \leq \alpha_1, \\ p & \alpha_1 < \alpha \leq \alpha_2, \\ 0 & \alpha_2 < \alpha \end{cases} \tag{2}$$

for $1 \leq \alpha_1 \leq \alpha_2$ and $p \in [0,1]$. Note that this family generalizes the idea for the SMART_q algorithms.

Theorem 4.3 *Let f be of type* (2). RSMART$_f$ *has a competitive ratio of*

$$\left\{\begin{array}{r} 4/3 \\ 1.2808 \end{array}\right\} \text{ for parameters } \left\{\begin{array}{c} \alpha_1 = 2, \alpha_2 = \infty, p = 1/3 \\ \alpha_1 = 1.7808, \alpha_2 = 2\alpha_1, p = 1/2 \end{array}\right\}.$$

Proof. It is again sufficient to consider sequences $\sigma = (1, a, b)$ with $a \geq 1$. We analyse the same cases as in the proof of Proposition 4.1, but now for an arbitrary function f as in (2).

Case 1: $b \in [0, 1]$

It is optimal to serve the customer at a immediately, giving $\textsc{Opt}(\sigma) = 2a + 2b$. RSMART is nonoptimal only if $a = \alpha > \alpha_1$, with ratio

$$\frac{\mathbb{E}[\textsc{rsmart}_f(\sigma)]}{\textsc{Opt}(\sigma)} = \begin{cases} \frac{1+a-p+pb}{a+b} & \alpha_1 < a \leq \alpha_2, \\ \frac{1+a}{a+b} & \alpha_2 < a. \end{cases}$$

Both expressions are decreasing in b, hence the competitive ratio is bounded by $\max\{\frac{1+a-p}{a}, \frac{1+a}{a}\}$.

Case 2: $b \in [1, a]$

In this and the following case it is optimal to postpone the customer at a, i. e., $\textsc{Opt}(\sigma) = 2 + 2a$. The ratio of RSMART in the nonoptimal cases is

$$\frac{\mathbb{E}[\textsc{rsmart}_f(\sigma)]}{\textsc{Opt}(\sigma)} = \begin{cases} \frac{a+b}{1+a} & 1 \leq a \leq \alpha_1, \\ \frac{1+a-p+pb}{1+a} & \alpha_1 < a \leq \alpha_2, \end{cases}$$

which is bounded by $\max\{\frac{2a}{1+a}, \frac{(1+p)a+1-p}{1+a}\}$, since the expressions increase in b. Note that the second term is increasing in a with limit $1 + p$.

Case 3: $b \geq a$

We have $\textsc{Opt}(\sigma) = 2 + 2b$. The competitive ratio of RSMART in the nonoptimal cases is

$$\frac{\mathbb{E}[\textsc{rsmart}_f(\sigma)]}{\textsc{Opt}(\sigma)} = \begin{cases} \frac{a+b}{1+b} & 1 \leq a \leq \alpha_1, \\ \frac{1+b-p+pa}{1+a} & \alpha_1 < a \leq \alpha_2. \end{cases}$$

This time the ratio decreases with b and we get the same bound as in the preceding case.

Assuming $\alpha_2 = \infty$, we can find optimal parameters α_1 and p by minimizing

$$\max\{\tfrac{1+\alpha_1-p}{\alpha_1}, \tfrac{2\alpha_1}{1+\alpha_1}, 1 + p\}.$$

Since the first term is decreasing and the second increasing in α_1, we can determine a point $\alpha_1 = \alpha_1(p)$ where they are equal and then solve for p, yielding $p = 1/3$ and $\alpha_1 = 2$ and a competitive ratio of $4/3$.

Similarly, the optimal parameters for $\alpha_2 < \infty$ have to minimize

$$\max\{\tfrac{1+\alpha_1-p}{\alpha_1}, \tfrac{2\alpha_1}{1+\alpha_1}, \tfrac{1+\alpha_2}{\alpha_2}, \tfrac{(1+p)\alpha_2+1-p}{1+\alpha_2}\}.$$

The mimizer can be found by determining $\alpha_1 = \alpha_1(p)$ and $\alpha_2 = \alpha_2(p)$ from the first and second pair of terms, respectively. We get $\alpha_1 = \frac{1}{4}(3 + \sqrt{17})$, $\alpha_2 = 2\alpha_1$, $p = 1/2$ and a competitive ratio of $\frac{5+\sqrt{17}}{3+\sqrt{17}} \leq 1.2808$. □

Remark It is easy to see that RSMART$_f$ with f as in Theorem 4.2 has *worst-case* competitive ratio 2, since it choses to serve $C_{1|2}$ immediately with a positive probability no matter how large α is. Furthermore, for f of type (2), the *worst-case* competitive ratio of RSMART$_f$ is

$$\max\{\tfrac{\alpha_1+1}{\alpha_1}, \tfrac{2\alpha_2}{\alpha_2+1}\}.$$

Thus RSMART$_f$ with $\alpha_1 = \frac{1}{4}(3+\sqrt{17})$, $\alpha_2 = 2\alpha_1$, $p = 1/2$ has a competitive ratio of $\frac{1}{2}(\sqrt{17}-1) \approx 1.56$ in the worst case. Thus RSMART$_f$ is only slightly worse than SMART$_q$, but significantly better than IMMEDIATE or DELAY even in the worst case.

The following result shows that RSMART$_f$ with a suitable f as in (2) is strictly better than any deterministic algorithm for $T = 3$, too.

decision sequence	cost
ID	$a + \max\{b, c\}$
DI	$1 + \max\{a, b\} + c$
DD	$1 + a + \max\{b, c\}$

(a) Decision sequences and their cost.

case	sequence	optimal cost
$b \le c$	ID	$a + c$
$b > c, \min\{a, b\} \le c + 1$	ID	$a + b$
$b > c, \min\{a, b\} \ge c + 1$	DI	$1 + \max\{a, b\} + c$

(b) Optimal cost.

Table 2: Overview on costs of the possible decision sequences.

Theorem 4.4 *Consider the* DMPRP($\mathbb{R}^+$) *with* $T = 3$. RSMART$_f$ *is* $7/5$*-competitive for* f *of type* (2) *with parameters* $\alpha_1 = 2$, $\alpha_2 = \infty$, *and* $p = 2/5$.

Proof. We consider an arbitrary sequence $\sigma = (1, a, b, c)$ with $a \ge 1$. At periods 1 and 2, RSMART can either serve the new customers immediately ("I") or delay ("D") them. Thus there are four possible decision sequences, namely II, ID, DI, and DD. The first decision sequence does not occur by construction of RSMART. A result in [2] implies that either ID or DI is optimal. Table 2 gives an overview on the decision sequences and their cost. We distinguish three cases given by the optimum solution.

Case 1: $b \le c$

If $a \le 2$, RSMART$_f$ is optimal. For $a > 2$, $b/a \le 2$, the decision sequence is DI or DD with ratio

$$\frac{\frac{2}{5}(a+c) + \frac{3}{5}(1 + \max\{a, b\} + c)}{a + c} \le \frac{2}{5} + \frac{3}{5}\frac{1 + \max\{a, \min\{2a, c\}\} + c}{a + c}.$$

If the minium is $2a$, i. e., $c \ge 2a \ge 2$, the last fraction can be bounded as

$$\le \frac{1 + 2a + c}{a + c} \le \frac{1 + 2c}{\frac{3}{2}c} \le \frac{5}{3}$$

and if $c \le 2a$ by

$$\le \frac{1 + \max\{a, c\} + c}{a + c} \le \frac{1 + 4a}{3a} \le \frac{5}{3}.$$

In both cases the ratio is at most $7/5$.

For $a > 2$, $b/a > 2$, the ratio is

$$\frac{\frac{2}{5}(a+c) + \frac{6}{25}(1 + b + c) + \frac{9}{25}(1 + a + c)}{a + c} \le \frac{\frac{19}{25}a + \frac{3}{5} + \frac{31}{25}c}{a + c} \le \frac{\frac{31}{25}a + \frac{31}{25}c}{a + c} \le \frac{31}{25} < \frac{7}{5}.$$

Case 2: $b > c$, $\min\{a, b\} \le c + 1$

If $a \le 2$, RSMART$_f$ is optimal again. For $a > 2$, $b/a \le 2$ the ratio is

$$\frac{\frac{2}{5}(a+c) + \frac{3}{5}(1 + \max\{a, b\} + c)}{a + b} \le \frac{2}{5} + \frac{3}{5}\frac{1 + \max\{a, b\} + b}{a + b}.$$

The last fraction is at most $3/2$, since if the maximum is b, it is increasing in b, but $b \le 2a$ and $a > 2$. Otherwise the fraction is decreasing in $(a + b)$ and is thus maximal for $a = 2$ and $b = 0$. The overall ratio is therefore at most $13/10$.

In case of $a > 2$, $b/a > 2$ we obtain

$$\frac{\frac{2}{5}(a+b) + \frac{6}{25}(1 + b + c) + \frac{9}{25}(1 + a + b)}{a + b} \le \frac{\frac{19}{25}a + \frac{3}{5} + b + \frac{6}{25}c}{a + b} \le \frac{\frac{31}{25}a + \frac{31}{25}b}{a + b} \le \frac{31}{25}.$$

Case 3: $b > c$, $\min\{a, b\} \geq c + 1$
If $a \leq 2$ we have the ratio

$$\frac{a+b}{1+\max\{a,b\}+c} \leq \frac{2a}{1+a} \leq \frac{4}{3} < \frac{7}{5}.$$

For $a > 2$, $b/a \leq 2$ we get

$$\frac{\frac{2}{5}(a+b)+\frac{3}{5}(1+\max\{a,b\}+c)}{1+\max\{a,b\}+c} \leq \frac{3}{5} + \frac{2}{5}\frac{2\max\{a,b\}}{\max\{a,b\}} \leq \frac{7}{5}.$$

Finally, in case of $a > 2$, $b/a > 2$ we obtain

$$\begin{aligned}\frac{\frac{2}{5}(a+b)+\frac{6}{25}(1+b+c)+\frac{9}{25}(1+a+b)}{1+b+c} &\leq \frac{6}{25} + \frac{\frac{19}{25}a+\frac{19}{25}b+\frac{9}{25}}{1+b} \\ &\leq \frac{6}{25} + \frac{\frac{57}{25}a+\frac{9}{25}}{1+2a} \leq \frac{6}{25} + \frac{57}{50} = \frac{69}{50} < \frac{7}{5}. \quad \square\end{aligned}$$

4.2 Randomized algorithms for the Euclidean plane

We know from Theorem 3.2 that RSMART cannot be better than 3/2-competitive, since it is a non-splitting algorithm. The following result shows that there is a version of RSMART that actually attains this competitive ratio.

Theorem 4.5 *Algorithm* RSMART$_f$ *with* $f(\alpha) = 1/2$ *is* 3/2*-competitive against the oblivious adversary for* DMPRP($\mathbb{R}^2$) *with two planning periods.*

Proof. Since we are considering only two planning periods, any solution for an instance of the DMPRP($\mathbb{R}^2$) partitions all customers into two sets, those that are visited in the first time period and those that are visited in the second. We denote by $C_{1|2_a}$ those customers of set $C_{1|2}$ visited in time period 1 in an optimal solution and by $C_{1|2_b}$ those postponed to the second time period in an optimal solution. In consistency with the introduced notation, $L_{1|2_a}$ and $L_{1|2_b}$ are defined as the optimal tour length for visiting all customers of set $C_{1|2_a}$ and $C_{1|2_b}$, respectively.

We know from the triangle inequality that it is always more favorable to serve all customers of the sets C_1 and $C_{1|2}$ together than serving customers of sets C_1 together with $C_{1|2_a}$ and customers of set $C_{1|2_b}$ on their own. In particular, we have

$$\begin{aligned} L_{1,1|2} &\leq L_{1,1|2_a} + L_{1|2_b} \leq L_{1,1|2_a} + \max\left\{L_2, L_{1|2_b}\right\}, \\ L_{1|2,2} &\leq L_{1|2,2_b} + L_{1|2_a} \leq L_{1|2,2_b} + \max\left\{L_1, L_{1|2_a}\right\}, \end{aligned}$$

and

$$\text{OPT} \geq L_{1,1|2_a} + \max\left\{L_2, L_{1|2_b}\right\} \geq L_1 + \max\left\{L_2, L_{1|2_b}\right\}.$$

Moreover, we know that since the length of the optimal tours is at least the length of the tour serving all possible requests together, we have

$$\text{OPT} \geq \max\left\{L_{1,1|2}, L_{1|2,2}\right\}.$$

Taking these observations into account yields

$$\begin{aligned} \frac{\mathbb{E}[\text{RSMART}_f(\sigma)]}{\text{OPT}(\sigma)} &= \frac{p\left(L_{1,1|2}+L_2\right)+(1-p)\left(L_1+L_{1|2,2}\right)}{L_{1,1|2_a}+L_{1|2_b,2}} \\ &\leq \frac{p\left(L_{1,1|2_a}+2\max\left\{L_2,L_{1|2_b}\right\}\right)+(1-p)\left(L_{1|2_b,2}+2\max\left\{L_1,L_{1|2_a}\right\}\right)}{L_{1,1|2_a}+L_{1|2_b,2}}. \end{aligned}$$

For $p = 1/2$ we get

$$\begin{aligned} \frac{\mathbb{E}[\text{RSMART}_f(\sigma_x)]}{\text{OPT}(\sigma)} &\leq \frac{1}{2}\frac{L_{1,1|2_a}+L_{1|2_b,2}}{L_{1,1|2_a}+L_{1|2_b,2}} + \frac{\max\left\{L_2,L_{1|2_b}\right\}+\max\left\{L_1,L_{1|2_a}\right\}}{L_{1,1|2_a}+L_{1|2_b,2}} \\ &= \frac{1}{2} + \frac{\max\left\{L_2,L_{1|2_b}\right\}+\max\left\{L_1,L_{1|2_a}\right\}}{L_{1,1|2_a}+L_{1|2_b,2}} \\ &\leq \frac{3}{2} \end{aligned}$$

for all input sequences σ. The last inequality follows from the fact that both tours L_2 and $L_{1|2_b}$ (L_1 and $L_{1|2_a}$) are by the triangle inequality smaller than $L_{1|2_b,2}$ ($L_{1,1|2_a}$). $\square$

In order to achieve a better competitive ratio than 3/2, it is necessary to split $C_{t|t+1}$. However, the criterion for splitting the customer set needs to be more involved.

Proposition 4.6 *Suppose* ALG *is a splitting (randomized) online algorithm that determines the subset* S *of* $C_{t|t+1}$ *to serve in period* t *using a probability distribution that does not depend on* $L_{t,S}$*. Then* ALG *is not better than 3/2-competitive.*

Proof. Consider $T = 2$ and the instance $\sigma = (\{1\}, \{a\}, \{b\})$ for the real line. Since $C_{1|2}$ is a singleton set, ALG is essentially a non-splitting algorithm on this instance. As its decision does not depend on a, it behaves as RSMART$_f$ with $f \equiv p$ and p is the probability for selecting $\{a\}$ to serve immediately. By Theorem 4.2, ALG is at best 3/2-competitive. $\square$

5 Conclusion

In this paper we proposed and analyzed randomized online algorithms for the DMPRP with two and three planning periods. We showed that there are functions f such that the randomized algorithm RSMART$_f$ is optimal for $T = 2$ and strictly better than any deterministic algorithm for $T = 3$ for the real line. For customers in the Euclidean plane, we established a lower bound of 3/2 on the competitive ratio of non-splitting randomized algorithms and a lower bound of 5/4 on the competitive ratio of splitting algorithms. This implies that the only way to improve upon the competitive ratio of 3/2 for known algorithms is to develop randomized splitting algorithms. A very interesting question is whether this is possible. Another promising research direction is to extend the analysis of algorithms for the real line to an arbitrary number of planning periods.

References

[1] E. Angelelli, M. W. P. Savelsbergh, and M. G. Speranza. Competitive analysis for dynamic multiperiod uncapacitated routing problems. *Networks*, 49(4):308–317, 2007.

[2] E. Angelelli, M. W. P. Savelsbergh, and M. G. Speranza. Competitive analysis of a dispatch policy for a dynamic multi-period routing problem. *Operations Research Letters*, 35(6):713–721, 2007.

[3] A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis*. Cambridge University Press, 1998.

[4] D. R. Dooley, S. A. Goldman, and S. D. Scott. TCP dynamic acknowledgement delay: Theory and practice. In *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing*, pages 389–398, 1998.

[5] A. Fiat and G. J. Woeginger, editors. *Online Algorithms: The State of the Art*, volume 1442 of *Lecture Notes in Computer Science*. Springer, 1998.

[6] S. Heinz, S. O. Krumke, N. Megow, J. Rambau, A. Tuchscherer, and T. Vredeveld. The online target date assignment problem. In Thomas Erlebach and G. Persiano, editors, *Approximation and Online Algorithms (WAOA 2005)*, volume 3879 of *Lecture Notes in Computer Science*, pages 230–243. Springer, Berlin, Germany, 2006.