Pubblicazioni del Centro De Giorgi
Proceedings of the Interdisciplinary Workshop on Mathematical Methods in Biomedical Imaging and Intensity-Modulated Radiation Therapy (IMRT), Pisa, Italy, October 2007.

Algorithms for Satisfying Dose-Volume Constraints in Intensity-Modulated Radiation Therapy

Wei Chen, Gabor T. Herman, and Yair Censor

ABSTRACT. In intensity-modulated radiation therapy (IMRT) we need to deliver a sufficient dose to target volumes (e.g., cancerous tumors) to destroy them, but at the same time we have to be careful that we do not destroy sensitive essential organs. These dual requirements can be expressed by a system of linear inequalities, in which the unknowns are the intensities to be delivered in the beamlets of the IMRT device (assuming that the dose delivered to any point in the body depends linearly on the unknowns).

It is often the case that the system of inequalities that results from an ideal plan (one in which all target locations get a dose sufficient for destruction and yet all locations within an organ at risk receive an absolutely safe dose) cannot be satisfied. In such a case, it is reasonable to relax the conditions so that a specified percent of the volume of an organ at risk may receive a dose in excess of what is absolutely safe, but still not more dose than a specified higher threshold for safety. Finding a solution to a problem that involves such dose-volume constraints is inherently more complex than finding a solution for a feasible ideal plan.

In this paper we discuss two approaches for solving problems with dose-volume constraints: one involves linear programming and the other is an adaptation of a projection method for solving feasible systems of linear inequalities. The two approaches are experimentally compared according to their ability to find a solution when there is one and according to their computational speed in case both of them succeed to find a solution.

Key words and phrases. Intensity-modulated radiation therapy, linear programming, algebraic reconstruction technique, projection method, dose-volume constraint.

The first and second authors are supported by NIH Grant HL070472.

The third author is supported by Grant 2003275 of the United States-Israel Binational Science Foundation (BSF), by NIH Grant HL070472 and by Grant 522/04 of the Israel Science Foundation (ISF) at the Center for Computational Mathematics and Scientific Computation (CCMSC) in the University of Haifa.

1. Introduction

Intensity-modulated radiation therapy (IMRT) is now widely used as a medical technique to treat cancer by radiation. The multileaf collimator was invented to generate radiation intensity patterns, with many degrees of freedom, to achieve a dose distribution in the patient's body, in which the target cells receive a sufficient dose to get killed but critical structures are spared by receiving a relatively safe dose.

The process is based on a full discretization of the volume of the patient's body. In practice this is done in 3D (three dimensions), but for the purpose of the algorithm comparison in this paper we use only 2D, which does not make an essential change to the mathematical nature of the problem. Because of the 3D nature of the practical problem, we consider the region into which radiation is delivered as divided into voxels (volume elements) rather than pixels (picture elements).

Assume that the number of radiation beamlets used is I and the number of voxels is J. The total radiation dose delivered to voxel j ($1 \le j \le J$) is determined by the unknown intensity x_i ($1 \le i \le I$) of each beamlet i and a_i^j, the dose delivered to voxel j by the ith radiation beamlet with unit intensity. Formally, it is $\sum_{i=1}^{I} a_i^j x_i$. (Here it is assumed that the dose delivered to any voxel in the body depends linearly on the intensities of the radiation beamlets.)

Initially, the desired doses may be expressed by inequalities, giving a lower bound l_j ($1 \le j \le J$) to the dose of voxels in planning target volumes (PTVs) and an upper bound u_j ($1 \le j \le J$) to the dose of voxels in organs at risk (OARs). An $x = (x_1, x_2, \ldots, x_I)^T$, with all components nonnegative, for which the total doses $\sum_{i=1}^{I} a_i^j x_i$ satisfy the linear inequality constraints is a solution to the IMRT inverse problem. However, the competing desires to destroy PTVs and not to harm OARs are likely to cause the initially given constraints to be inconsistent (the solution set of the IMRT problem is empty). A methodology in the presence of such infeasibility is provided by dose-volume constraints (DVCs) [**1**], which allow the lower bound (or upper bound) on a portion of the PTV (or OAR) to be lowered (or raised) by a certain amount. For example, the constraints for the OAR that consists of voxels whose indices are from the set B may be changed from

$$\sum_{i=1}^{I} a_i^j x_i \le u, \text{ for } j \in B, \tag{1.1}$$

to

$$\sum_{i=1}^{I} a_i^j x_i \leq (1+\beta)u, \text{ for } j \in B, \tag{1.2}$$

and

$$|\{j \in B | \sum_{i=1}^{I} a_i^j x_i > u\}| \leq \alpha |B|, \tag{1.3}$$

where $|B|$ stands for the cardinality (number of elements) of the set B. This amounts to allowing a specific portion (α) of the original inequalities to be violated, but only up to a specific fraction (β). If α and β are small enough, then delivering such doses to the voxels of this OAR should allow it to keep performing its function.

Mathematically speaking, any method that can solve the problem with constraints of type (1.1) can be used to solve the problem with constraints of type (1.2) and (1.3). One by one we can select subsets C of B of size not greater than $\alpha|B|$, and use inequalities of type (1.2) for $j \in C$ and of type (1.1) for $j \in B \backslash C$ (the elements of B that are not in C). However, the number of ways of selecting C can become enormous and so this approach is not useful in practice.

A practical algorithm using linear programming, without resorting to mixed integer programming (MIP), see, e.g., Langer et al. [**4**], for solving the IMRT problem with dose-volume constraints (however one that is not guaranteed to find a solution, even when there is one) was proposed by Censor et al. [**3**]. In this paper we compare that algorithm with a new approach that is based on a fast sequential projection method called ART3+ [**7**]. This new method is also not guaranteed to find a solution, even if there is one.

The two algorithms are compared on a set of experiments based on two anthropomorphic phantoms. Both the ability of finding a solution when there is one and the computation time when both methods find a solution are reported.

We describe the algorithms in Section 2. The experiments and their outcomes are presented in Section 3 and a discussion of the results is given in Section 4.

2. Algorithms

It is mathematically simpler to consider each PTV and each OAR simply as a subvolume. Suppose that there are S such subvolumes and B_s (for $s = 1, 2, \ldots, S$) is the index set of the voxels in subvolume s. We assume that, for some $1 \leq \overline{s} \leq S$, $B_{\overline{s}}$ is an OAR and that it is the only subvolume for which we have dose-volume constraints. We assume

throughout this paper that the discretization of the body into voxels and the discretization of the external radiation beams into beamlets, are done ahead of time. Also, all index sets of voxels that describe the subvolumes have been identified and are given to us. In a real-world situation, these numbers would be given to a dose-calculation program, which would then calculate for us the a_i^j. Here we calculate them as explained in Section 3 below. Upper and lower bounds on permitted and required doses in subvolumes are prescribed by the radiation oncologist, thus known to us, and so are the numbers α and β for the DVC.

Under these assumptions the IMRT inverse problem requires finding an $x = (x_1, x_2, \ldots, x_I)^T$ such that

$$l_{\overline{s}} \leq \sum_{i=1}^{I} a_i^j x_i \leq (1+\beta) u_{\overline{s}}, \text{ for } j \in B_{\overline{s}}, \tag{2.1}$$

$$|\{j \in B_{\overline{s}} | \sum_{i=1}^{I} a_i^j x_i > u_{\overline{s}}\}| \leq \alpha |B_{\overline{s}}|, \tag{2.2}$$

$$l_s \leq \sum_{i=1}^{I} a_i^j x_i \leq u_s, \text{ for all } j \in B_s, \text{ for } s \neq \overline{s}, 1 \leq s \leq S, \tag{2.3}$$

$$0 \leq x_i \leq u, \text{ for } i = 1, 2, \ldots, I. \tag{2.4}$$

2.1. Linear Programming. The linear programming (LP) method of [**3**] introduces, for each inequality j of the OAR $B_{\overline{s}}$, an auxiliary variable t_j that controls the amount by which the right-hand side of (2.1) goes above its initially prescribed upper bound $u_{\overline{s}}$. The LP task is formulated as follows:

$$\text{minimize } \sum_{j \in B_{\overline{s}}} t_j, \tag{2.5}$$

$$\text{subject to } l_{\overline{s}} \leq \sum_{i=1}^{I} a_i^j x_i \leq t_j u_{\overline{s}}, \text{ for } j \in B_{\overline{s}}, \tag{2.6}$$

$$1 \leq t_j \leq 1+\beta, \text{ for } j \in B_{\overline{s}}, \tag{2.7}$$

$$\sum_{j \in B_{\overline{s}}} t_j \leq (1+\alpha\beta)|B_{\overline{s}}|, \tag{2.8}$$

$$l_s \leq \sum_{i=1}^{I} a_i^j x_i \leq u_s, \text{ for all } j \in B_s, \text{ for } s \neq \overline{s}, 1 \leq s \leq S, \tag{2.9}$$

$$0 \leq x_i \leq u, \text{ for } i = 1, 2, \ldots, I. \tag{2.10}$$

The solution x of this LP problem will satisfy (2.1) (because it satisfies (2.6) and (2.7)), (2.3) and (2.4) (because these are the same as (2.9) and (2.10)). Given (2.6) and (2.7), (2.8) can be derived from (2.1) and (2.2) (by letting at most a fraction α of the t_j to be $1 + \beta$ and the remaining t_j to be 1). The justification of the optimization goal (2.5) is that it pushes the solution set of the linear constraints (2.6)-(2.10) towards the solution set of (2.1)-(2.4).

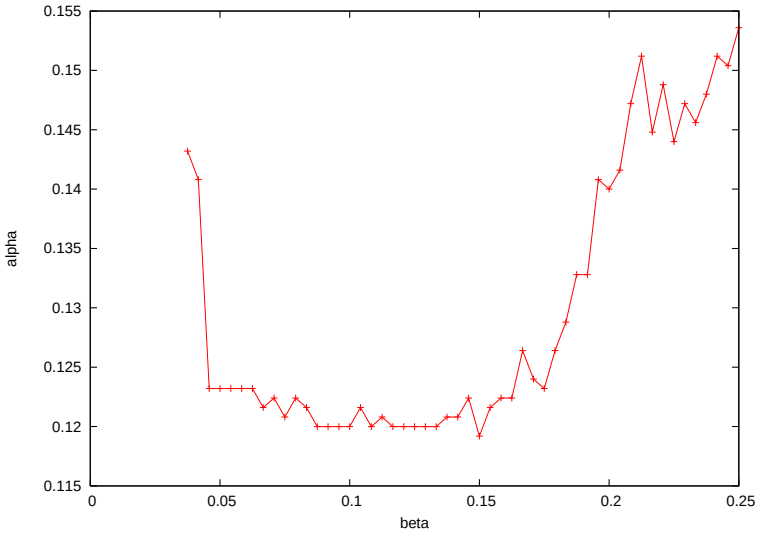
In spite of this, there is still no guarantee that x satisfies (2.2); checking the solution x against (2.2) is thus required in the algorithm. However, the experiments show that, most of the time, solving the LP task gives us a solution that satisfies also (2.2). It turns out in our experiments that the optimization is not necessary most of the time and the LP problem can be solved much more efficiently without the optimization goal (2.5). In the experiments, our strategy is to solve the LP task first without the optimization goal and check if the solution satisfies (2.2); if not, then we solve the LP task again with the optimization goal and check again.

Our experiments use the COIN-OR Linear Program (CLP) Solver [**5**] to solve the LP task.

2.2. ART3+ Algorithm. This algorithm is designed to find a common point in the intersection of hyperslabs, determined by interval inequalities of the form (2.9) and (2.10). It may start anywhere, usually at the origin (the zero vector). In each iteration, the algorithm picks one interval inequality and moves the current point into the hyperslab. In the ART3 algorithm [**6**] the hyperslabs are picked in a repetitive cyclic order. If a point is within the hyperslab, then it is not moved. If the current point is outside the hyperslab but sufficiently near it, then the next point is obtained by reflection in the nearest face of the hyperslab. Otherwise, the current point is projected onto the center hyperplane of the hyperslab. It is proved in [**6**] that, if the solution set is full-dimensional (i.e., has a nonempty interior), then ART3 finds a point in it within a finite number of steps. The newly developed ART3+ algorithm [**7**] differs from ART3 by using a more sophisticated ordering of the hyperslabs. It retains the finite convergence property of ART3 and it usually finds a solution at a lesser computational cost.

Since ART3+ requires the solution set to be full-dimensional, we use it in the following manner to search a solution of (2.1)-(2.4). We generate a sequence of real numbers $\beta[k] \in (0, \beta]$, for $k = 0, 1, \ldots$. We then use ART3+ to attempt to solve

FIGURE 1. A typical curve of $\alpha[k]$ plotted against $\beta[k]$



$$l_{\overline{s}} \leq \sum_{i=1}^{I} a_i^j x_i \leq (1+\beta[k])u_{\overline{s}}, \text{ for } j \in B_{\overline{s}}, \tag{2.11}$$

$$l_s \leq \sum_{i=1}^{I} a_i^j x_i \leq u_s, \text{ for all } j \in B_s, \text{ for } s \neq \overline{s}, 1 \leq s \leq S, \tag{2.12}$$

$$0 \leq x_i \leq u, \text{ for } i = 1, 2, \ldots, I. \tag{2.13}$$

If ART3+ does not terminate within a pre-specified number of steps, then we set $\alpha[k] = +\infty$. Otherwise we calculate and set

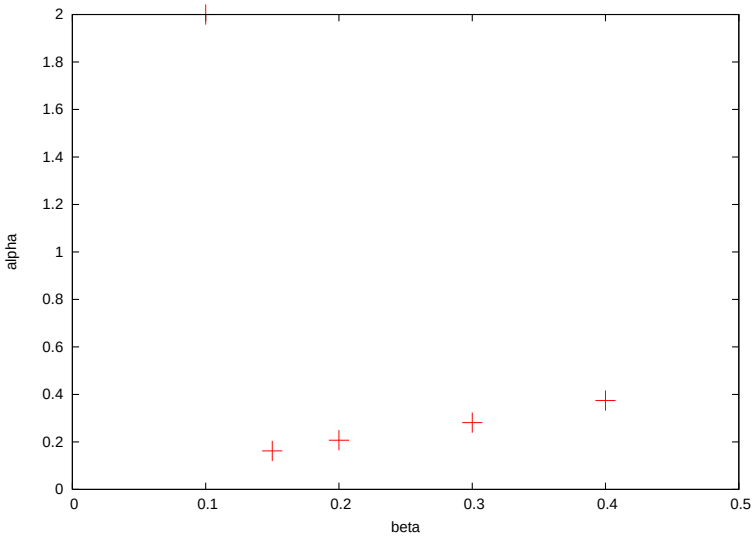
$$\alpha[k] = |\{j \in B_{\overline{s}} | u_{\overline{s}} < \sum_{i=1}^{I} a_i^j x_i \leq (1+\beta[k])u_{\overline{s}}\}|/|B_{\overline{s}}|. \tag{2.14}$$

If $\alpha[k] \leq \alpha$ then the solution of (2.11)-(2.13) is also a solution of (2.1)-(2.4) and we are done. Otherwise we repeat the process for $\beta[k+1]$.

This strategy does not guarantee that we find a solution to (2.1)-(2.4), even if there is one. However, the curve of $\alpha[k]$ plotted against $\beta[k]$ is typically "$\cup$-shaped" (see Figure 1) and hopefully the minimal $\alpha[k]$ we can find is smaller than the required α.

In the experiments, we use for ART3+ a simple recursive search strategy. The starting search range is set to be $[0, \beta]$. We then divide the search

FIGURE 2. Searching for solution with $\alpha = 0.2, \beta = 0.4$. For $\beta[k] = 0.1$, ART3+ failed to converge. The search order for $\beta[k]$ is 0.4, 0.2, 0.1, 0.3, 0.15.



range into half, pick as $\beta[k]$ that value of β that gives smallest $\alpha[k]$ among the middle point and the two endpoints, change the search range to the neighborhood of that $\beta[k]$ with half of the previous length. An example of the pairs $(\alpha[k], \beta[k])$ generated by this strategy is illustrated in Figure 2.

Another trick we use is that, for a given $\beta[k]$, once the algorithm have been running for a much longer time than a typical run, we save the current result and go to the next $\beta[k]$. We may go back to continue this saved run, if we do not get a good solution for the other $\beta[k]$ we tried.

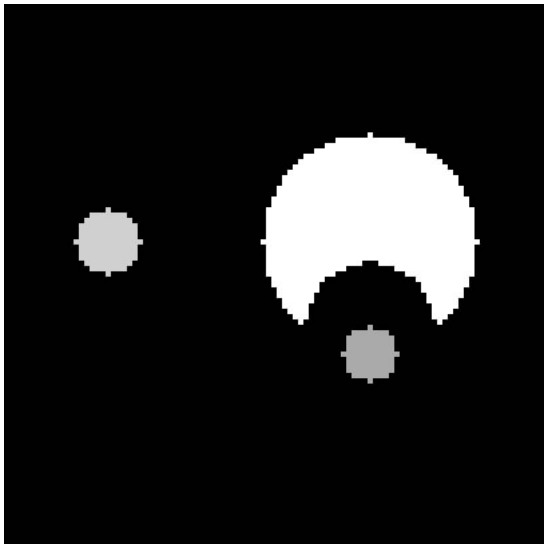
We implemented the ART3+ algorithm within the programming system SNARK05 [**2**].

3. Experiments

In the experiments, we use two two-dimensional anthropomorphic phantoms provided by the Radiological Physics Center (RPC) [**8**]. One is the head and neck phantom (Figure 3), the other is the prostate phantom (Figure 4).

For our experiments, the square region into which the phantoms are embedded is subdivided into 101×101 voxels of size 16 mm^2, i.e., $J = 10,201$.

FIGURE 3. The head and neck phantom (s=1: bright moon-shaped region, a PTV; s=2: small disk on the left, a PTV; s=3: small disk on the right, an OAR; s=4: rest of the voxels within the body, normal tissue.)



We always use the same five beam directions: $0°$, $72°$, $144°$, $216°$, $288°$. For each direction, there are 103 beamlets of width 4 mm, with the center of the center beamlet going through the center of the square region. Therefore the total number of beamlets is $I = 515$. The intensity of the radiation beamlets x_i $(1 \leq i \leq I)$ is nonnegative and has an upper bound 100, i.e., $u = 100$ in (2.4).

For $1 \leq i \leq I$ and $1 \leq j \leq J$, $a_i^j = 1$ if the center of ith voxel is within the jth beamlet and is zero otherwise. (This is not a realistic model for IMRT, but it is reasonable enough for an evaluation of the relative performance of the two algorithms.)

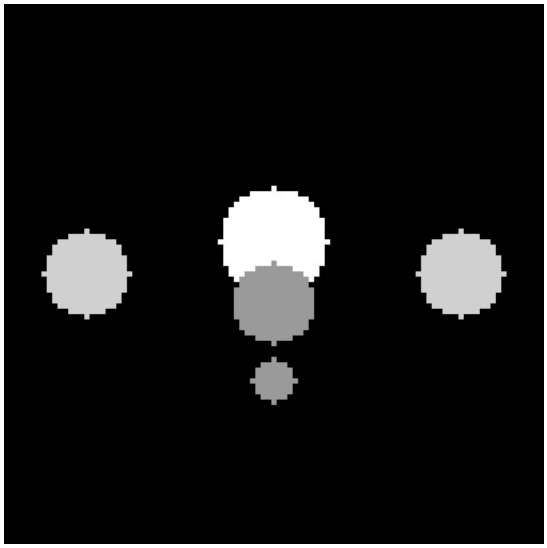
3.1. Head and neck. In Figure 3, on the right the bright moon-shaped region is the primary PTV. The small disk under the primary PTV is the only OAR. The left disk is the secondary PTV. The rest of the voxels are considered to be in normal tissue that form the fourth subvolume. The l_s, u_s, α and β are presented in Table 1.

3.2. Prostate. In Figure 4, the bright moon-shaped region is an OAR (the bladder), under it is a PTV (the prostate), and under that is another OAR (the rectum). The two symmetric disks on the left and right are both

TABLE 1. The prescription for the subvolumes in the head and neck phantom.

s	l_s	u_s	(α, β)
1	66	127.5	
2	54	127.5	
3=$\overline{s}$	0	20	(0.200, 0.400)
4	0	73.6	

FIGURE 4. The prostate phantom (s=1: center disk, a PTV; s=2: bright moon-shaped region, an OAR; s=3: bottom disk, an OAR; s=4: side disks, two OARs; s=5: rest of the voxels within the body, normal tissue.)



OARs (the femoral heads). The l_s, u_s, α and β are presented in Table 2. There are eight tasks to be run.

3.3. Results. The experiments were conducted using an Intel Xeon 1.7MHz processor, 1G RAM workstation. The total computation times (the duration of the algorithm until it finds a solution) needed by the two algorithms for each of the experiments are shown in Table 3. In the LP

TABLE 2. Prescriptions for the subvolumes in the prostate phantom. DVCs are applied to the bladder in the upper table and to the rectum to the lower table.

s	l_s	u_s	(α, β)
1	60	127.5	
2=$\overline{s}$	0	49	(0.070, 0.347) (0.075, 0.276) (0.080, 0.225) (0.085, 0.164)
3	0	49	
4	0	49	
5	0	49	

s	l_s	$u_s(\alpha, \beta)$	
1	60	127.5	
2	0	55	
3=$\overline{s}$	0	35	(0.200, 0.286) (0.250, 0.225) (0.300, 0.164) (0.350, 0.103)
4	0	49	
5	0	49	

time column, "3.452+194.032 (No solution)" means that the time LP runs without optimization is 3.452 seconds, but the solution it returns fails to satisfy (2.2) and so we run the LP again with optimization, and the time for this run is 194.032 seconds, but the solution it gives still fails to satisfy (2.2).

TABLE 3. Timings (in seconds) for finding solutions with the two algorithms.

Algorithm	LP time (s)	ART3+ time (s)
Head and Neck	20.285+200.437	3.472
Prostate 1	3.452+194.032 (No solution)	9.324 (No solution)
Prostate 2	3.488+172.999 (No solution)	1.700
Prostate 3	3.360	2.776
Prostate 4	3.548	1.056
Prostate 5	2.296	6.532
Prostate 6	2.332	3.728
Prostate 7	2.372	2.336
Prostate 8	2.300	1.196

4. Discussion

We can see from Table 3 that generally ART3+ works a little bit faster than LP does when we do not need to run the LP again with the optimization goal. Otherwise, ART3+ is much faster. Also, there was a case for which ART3+ found a solution, but LP did not. However, more experiments need to be done to arrive at a solid conclusion. Experiments with real data need to be conducted to test whether the algorithms can be extended and will be fast enough for practical (3D) IMRT planning with multiple dose-volume constraints.

References

1. T. Bortfeld, J. Stein and K. Preiser, *Clinically relevant intensity modulation optimization using physical criteria*, in: D.D. Leavitt and G. Starkschall (Editors), The XIIth International Conference on the Use of Computers in Radiation Therapy (ICCR), Salt Lake City, Utah, USA, May 27-30, 1997, pp. 1–4.
2. B. Carvalho, W. Chen, J. Dubowy, G.T. Herman, M. Kalinowski, H.Y. Liao, L. Rodek, L. Rusk, S.W. Rowland, and E. Vardi-Gonen, *SNARK05: A programming system for the reconstruction of 2D images from 1D projections*, available on the internet, *http://www.cisdd.org/snark05/SNARK05.pdf*, 2006.
3. Y. Censor, A. Ben-Israel, Y. Xiao, J.M. Galvin, *On linear infeasibility arising in intensity-modulated radiation therapy inverse planning*, Linear Algebra Appl., accepted for publication.
4. M. Langer, S. Morill, R. Brown, O. Lee, and R. Lane, *A compasion of mixed integer programming and fast simulated annealing for optimizing beam weights in radiation therapy*, Med. Phys. **23**(1996), 957–964.

5. R. Lougee-Heimer, *The common optimization interface for operations research*, IBM J. Res. Dev. **47**(2003), 57–66. *http://www.coin-or.org/Clp*.
6. G.T. Herman, *A relaxation method for reconstructing objects from noisy X-rays*, Math. Program. **8**(1975), 1–19.
7. G.T. Herman, W. Chen, *A fast algorithm for solving a linear feasibility problem with application to intensity-modulated radiation therapy*, Linear Algebra Appl. (2006), doi:10.1016/j.laa.2006.11.009.
8. *http://rpc.mdanderson.org/RPC/services/Anthropomorphic_ Phantoms/Anth_PPP.htm*.

DEPARTMENT OF COMPUTER SCIENCE, THE GRADUATE CENTER, CITY UNIVERSITY OF NEW YORK, NEW YORK, NY 10016-4309, UNITED STATES
E-mail address: wchen@gc.cuny.edu

DEPARTMENT OF COMPUTER SCIENCE, THE GRADUATE CENTER, CITY UNIVERSITY OF NEW YORK, NEW YORK, NY 10016-4309, UNITED STATES
E-mail address: gaborherman@yahoo.com

DEPARTMENT OF MATHEMATICS, UNIVERSITY OF HAIFA, MOUNT CARMEL, HAIFA 31905, ISRAEL
E-mail address: yair@math.haifa.ac.il