Multiobjective Decomposition of Positive Integer Matrices: Application to Radiotherapy

Thibaut Lust, Jacques Teghem

Faculté Polytechnique de Mons, Laboratory of Mathematics & Operational Research
9, rue de Houdain, 7000 Mons (Belgium)
thibaut.lust@fpms.ac.be

Abstract. We consider the following problem: to decompose a positive integer matrix into a linear combination of binary matrices that respect the consecutive ones property. The positive integer matrix corresponds to fields giving the different radiation beams that a linear accelerator has to send throughout the body of a patient. Due to the inhomogeneous dose levels, leaves from a multi-leaf collimator are used between the accelerator and the body of the patient to block the radiations. The leaves positions can be represented by segments, that are binary matrices with the consecutive ones property. The aim is to find a decomposition that minimizes the irradiation time, and the setup-time to configure the multi-leaf collimator at each step of the decomposition. We propose for this NP-hard multiobjective problem a heuristic method, based on the Pareto local search method. Experimentations are carried out on different size instances and the results are reported. These first results are encouraging and are a good basis for the design of more elaborated methods.

1 Introduction

In this paper, we consider a problem dealing with the planning of an intensity modulated radiotherapy treatment (IRMT) to individual patients. The IRMT is usually composed of three phases [5]: the selection of beam angles through which radiation is delivered (geometry problem), the computation of an optimal intensity map for each selected beam angle (intensity problem) and the determination of a sequence of configurations of a multi-leaf collimator (realization problem). In this work, we only consider the realization problem, by taking into account three different objective. We present below the mathematical model of this problem.

Throughout we use the notation $[n] := \{1, 2, \cdots, n\}$ for positive integers n.

We consider a positive integer matrix A of size $m \times n$: $A = (a_{i,j})$ with $i \in [m]$ and $j \in [n]$. The matrix corresponds to fields giving the different radiation beams that a linear accelerator has to send throughout the body of a patient. The value $a_{i,j}$ of A gives the desired intensity that should be delivered to coordinate (i, j).

We have to decompose the matrix A into a set of segments. The segments correspond to the shape of a multi-leaf collimator (MLC) which is a system containing a collection of leaves that can be moved in parallel, in order to block the

radiations (inhomogeneous dose levels are administrated: certain cancer targets receive a required amount of dose while functional organs are spared). Two types of leaves are used: left leaves that move from the left to the right and right leaves that move from the right to the left.

A segment can be represented by a special binary matrix of size $m \times n$ that describes the leaves positions. These matrices have to respect the consecutive ones property (C1), which means, in short, that the ones occur consecutively in a single block in each row (since we can only block the radiations with a left or a right leave).

A segment is noted $S = (s_{i,j})$ with $i \in [m]$ and $j \in [n]$. An example of a decomposition of a matrix A into segments is shown below.

$$\mathbf{A} = \begin{pmatrix} 4\,8\,3 \\ 5\,2\,1 \\ 5\,7\,2 \end{pmatrix} = 4 \begin{pmatrix} 1\,1\,0 \\ 1\,0\,0 \\ 1\,1\,0 \end{pmatrix} + 2 \begin{pmatrix} 0\,1\,1 \\ 0\,1\,0 \\ 0\,1\,1 \end{pmatrix} + 1 \begin{pmatrix} 0\,1\,1 \\ 1\,0\,0 \\ 1\,1\,0 \end{pmatrix} + 1 \begin{pmatrix} 0\,1\,0 \\ 0\,0\,1 \\ 0\,0\,0 \end{pmatrix}$$

The positions of the left and right leaves corresponding to a segment S are given by the l_i and r_i integers defined as follows:

$$0 \leq l_i < r_i \leq n+1 \quad (i \in [m])$$

$$s_{i,j} = \begin{cases} 1 \text{ if } l_i < j < r_i & (i \in [m], j \in [n]) \\ 0 \text{ otherwise.} \end{cases}$$

We denote the set of segments by $\mathcal{S}$.

For example, for the following segment:

$$\begin{pmatrix} 1\,1\,0 \\ 1\,0\,0 \\ 0\,0\,1 \end{pmatrix}$$

we have: $l_1 = 0, r_1 = 3; l_2 = 0, r_2 = 2$ and $l_3 = 2, r_3 = 4$.

It should be noted that in this work, a line full of zero is always represented by $l = 0$ and $r = 1$.

A feasible decomposition of A is a linear sum of segments and has the following form:

$$A = \sum_{k=1}^{K} u_k S^k \text{ with } u_k \in \mathbb{N}_0, S^k \in \mathcal{S} \; \forall \, k \in [K].$$

Two criteria are generally considered to evaluate the quality of a decomposition: the total irradiation time and the setup-time.

The total irradiation time, very important from a medical point of view, is the time during which a patient is irradiated. This criterion is proportional to the sum of the coefficients (decomposition time). The setup-time is the time to configure the MLC. This criterion is proportional to the number of segments (decomposition cardinality). It is important to minimize this criterion in order to reduce the time of the session and so the comfort of the patient.

We can formulate both objectives that is the decomposition time (DT) and the decomposition cardinality (DC) as follows:

$$\textbf{(DT)} \quad \min \left\{ \sum_{k=1}^{K} u_k \;\middle|\; A = \sum_{k=1}^{K} u_k S^k, u_k \in \mathbb{N}_0, S^k \in \mathcal{S} \; \forall \, k \in [K] \right\}$$

$$\textbf{(DC)} \quad \min \left\{ K \;\middle|\; A = \sum_{k=1}^{K} u_k S^k, u_k \in \mathbb{N}_0, S^k \in \mathcal{S}, \forall \, k \in [K] \right\}$$

Polynomial algorithms are known for the DT minimization [3, 13]. Many optimal solutions can be found for this single-objective problem.

On the other hand, the DC minimization has been proved to be NP-Hard [2, 4].

Some authors [2, 7, 11] optimize the objectives lexicographically: by first minimizing DT and then trying to reduce the DC while keeping the minimal DT. Taşkin *et al.* [14] and Wake *et al.* [15] recently considered both objectives at the same time, but by simply doing a linear sum of the objectives.

To be more realistic, we do not only consider in this paper constant times to move from one segment to the next. The variable setup-time is defined as follows:

$$(\mathbf{SU}_{\boldsymbol{var}}) \quad \min \left\{ \sum_{k=1}^{K-1} \mu(S^k, S^{k+1}) \;\middle|\; A = \sum_{k=1}^{K} u_k S^k, u_k \in \mathbb{N}_0, S^k \in \mathcal{S}, \forall \, k \in [K] \right\}$$

where μ is proportional to the time necessary to change the setup of the MLC from the configuration corresponding to S^k to the configuration corresponding to S^{k+1}. This objective is also known under the name overall leaf travel time [11].

The value μ between two segments S^k and S^{k+1} is computed as follows [13]:

$$\mu\big(S^k, S^{k+1}\big) = \max_{1 \le i \le m} \max \left\{ \left| l_i^{k+1} - l_i^k \right|, \left| r_i^{k+1} - r_i^k \right| \right\}$$

Once the segments are fixed, the minimization of the overall leaf travel time is equivalent to a search for a Hamiltonian *path* of minimal weight on the complete graph which has the segments as vertices and the weight function μ on the edges. The weight function μ has the property to be a metric [11].

This problem can be transformed to a TSP problem (Hamiltonian *cycle* of minimal weight) by adding a dummy vertex which has a distance of zero to all other vertices. However, with this transformation, the triangular inequality in Euclidean space no longer holds which makes the TSP problem a little bit harder to solve.

Since there is a positive correlation between DC and SU_{var}, the few authors that have considered SU_{var} tried to first minimize DC and then SU_{var} by generating the best sequence of segments. Kalinowski [11] uses a minimum spanning tree approximation to find the best sequence of segments, but it also possible to use an exact TSP algorithm, as done by Ehrgott *et al.* [6].

Siochi [13] considered both objectives at the same time, but through a linear sum.

In this paper, we will consider the three objectives (DT,DC,SU_{var}) simultaneously. The multiobjective formulation of the problem (P) considered is thus as follows:

$$(\mathrm{P}) \begin{cases} \min z_1(x) & = \sum_{k=1}^{K} u_k \quad (\mathbf{DT}) \\ \min z_2(x) & = K \quad (\mathbf{DC}) \\ \min z_3(x) & = \sum_{k=1}^{K-1} \max_{1 \leq i \leq m} \max \left\{ \left| l_i^{k+1} - l_i^k \right|, \left| r_i^{k+1} - r_i^k \right| \right\} \quad (\mathbf{SU}_{var}) \\ \text{s.t} & A = \sum_{k=1}^{K} u_k S^k, u_k \in \mathbb{N}_0, S^k \in \mathcal{S}, \forall\, k \in [K] \end{cases}$$

We denote by X the feasible set in the decision space, defined by $X = \left\{ x \in \{(u_k \in \mathbb{N}_0, S^k \in \mathcal{S})\}^K | A = \sum_{k=1}^{K} u_k S^k \right\}$. The feasible set in objective space is called Z and is defined by $Z = z(X) = \left\{ (z_1(x), z_2(x), z_3(x)),\ \forall\, x \in X \right\} \subset \mathbb{Z}^3$.

To our knowledge, nobody has tried to find the efficient solutions, or even a good approximation of the efficient solutions of P. Our aim is to generate a good approximation of a minimal complete set [8] of the efficient solutions of P.

2 Pareto Local Search

The Pareto local search (`PLS`) method [1, 12] is one of the simplest method for multiobjective optimization. This method is a purely local search algorithm, generalization in the multiobjective case of a basic metaheuristic: the hill-climbing method. The method does not require any objectives aggregation nor any numerical parameters, and is based on the notion of *Pareto local optimum set* [12].

The pseudo-code of the `PLS` method is given by the algorithm 1.

The method starts with a population P composed of potentially efficient solutions given by the initial population P_0, which is an input parameter of the method. Then, all the neighbors p' of each solution p of P are generated. If a neighbor p' is not weakly dominated by the current solution p, we try to add the solution p' to the approximation $\widehat{X_E}$ of the efficient set, which is updated with the procedure `AddSolution`. This procedure is not described in this work but simply consists of updating an approximation $\widehat{X_E}$ of the efficient set when a new solution p is added to $\widehat{X_E}$. This procedure has four parameters, the set $\widehat{X_E}$ to actualize, the new solution p, its evaluation $z(p)$ and an optional boolean variable called *Added* that returns $True$ if the new solution has been added and $False$ otherwise. If the solution p' has been added to $\widehat{X_E}$, the boolean variable *Added* is true and the solution p' is added to an auxiliary population P_a, which is updated also with the procedure `AddSolution`. Once all the neighbors of each solution of P have been generated, the algorithm starts again, with P equal to P_a, until $P = \emptyset$. The auxiliary population is used such that the neighborhood of each solution of the population P is explored, even if some solutions of P become dominated

Algorithm 1 PLS

Parameters ↓: An initial population P_0
Parameters ↑: An approximation $\widehat{X_E}$ of the efficient set

--| Initialization of $\widehat{X_E}$ and a population P with the initial population P_0
$\widehat{X_E} \leftarrow P_0$
$P \leftarrow P_0$
--| Initialization of an auxiliary population P_a
$P_a \leftarrow \emptyset$
while $P \neq \emptyset$ **do**
  --| Generation of all the neighbors p' of each solution $p \in P$
  for all $p \in P$ **do**
    for all $p' \in \mathcal{N}(p)$ **do**
      if $z(p) \not\leq z(p')$ **then**
        AddSolution($\widehat{X_E}$ ↕, p' ↓, $z(p')$ ↓, *Added* ↑)
        if *Added* = *true* **then**
          AddSolution(P_a ↕, p' ↓, $z(p')$ ↓)
  --| P is composed of the new potentially efficient solutions
  $P \leftarrow P_a$
  --| Reinitialization of P_a
  $P_a \leftarrow \emptyset$

following the addition of a new solution to P. Thus, sometimes, neighbors are generated from a dominated solution.

3 Adaptation of PLS to the Multiobjective Decomposition Problem

3.1 Initial Population

Two initial solutions of good quality are generated and added to the initial population. The first solution is a good approximation of lexmin(DT,DC,SU_{var}) and the second one is a good approximation of lexmin(DT,SU_{var},DC). In both cases, we first minimize DT since polynomial algorithms are known for this problem. We can remark that a solution corresponding to lexmin(DT,SU_{var}) is also a solution corresponding to lexmin(DT,SU_{var},DC), since once SU_{var} is minimized, the DC value is set and can not be modified.

To approximate lexmin(DT,DC,SU_{var}), we first approximate lexmin(DT,DC) with the heuristic of Engel [7] which is efficient for this problem. We then apply a TSP heuristic to reduce the SU_{var} value of the solution. We use a very efficient heuristic for the TSP: the Lin and Kernighan heuristic implemented by Helsgaun (LKH) [10].

To approximate lexmin(DT,SU_{var}), we will adapt the heuristic of Engel since SU_{var} is correlated to the DC objective. The algorithm developed by Engel is a deterministic construction algorithm, that allows to find an optimal solution for

the DT objective with a low DC value. He tackles this problem as follows: he removes different well-selected combinations of couples (u_k, S^k) from the current matrix until $A_{t+1} = 0$, with $A_{t+1} = A_t - uS$, where t represent the index of the step of the construction algorithm. He starts the algorithm with $A_0 = A$. A move consists thus of removing from the current matrix a segment multiplied by a certain coefficient.

At each step of the construction heuristic, the maximum coefficient (u_{max}) that can be used while ensuring that the optimal objective DT can be achieved is considered.

Engel has developed a theory to compute u_{max}. The coefficient u_{max} can be easily obtained in $O(mn^2)$. Using u_{max} allows to find very good results for the lexicographic problem (DT,DC) [6, 7, 11].

But once u_{max} has been defined, we also have to define which segment to use among all the segments that respect u_{max}. Kalinoswki [11] has developed a rule which gives slightly better results that the initial rule of Engel. The rule is as follows. If two zero columns are added to A, that is let:

$$a_{i,0} = a_{i,n+1} = 0 \quad \forall\, i \in [m]$$

we can associate to A its difference matrix D of dimension $m \times (n+1)$:

$$d_{i,j} = a_{i,j} - a_{i,j-1} \quad \forall\, i \in [m], j \in [n+1]$$

Now, we put

$$q(A) = \big|\{(i,j) \in [m] \times [n] : d_{i,j} \neq 0\}\big|,$$

and in the method of Kalinowski, we choose a segment S so that $q(A - uS)$ is minimized.

This method gives very good results for lexmin(DT,DC), but there is no theoretical evidence for that.

For lexmin(DT,SU_{var}), we keep the principle of the construction algorithm of Engel by removing well-selected combinations of couples (u_k, S^k) from the current matrix until $A(t+1) = 0$.

As in the Engel algorithm, it is worthwhile to take the maximal coefficient that allows to keep the minimal DT value, since the SU_{var} objective is linked to the DC objective. For the definition of the segment that corresponds to u_{max}, we try three new rules. These three new rules are presented below.

For each line of the new segment to define, we have to choose between different intervals. If we sort out the intervals as follows: $\{(0,1), (0,2), \cdots, (0,n+1), (1,3), \cdots, (1,n+1), \cdots, (n-2,n), (n-2,n+1), (n-1,n+1)\}$, the first rule is to take the first feasible interval and the second rule is to take the last feasible interval. For these two rules, we expect to always move the leaves from the left to the right or to the right from the left to minimize the maximal distance between two consecutive segments. The third rule is to take the first feasible interval which is the closest to the preceding interval of the same line: the aim is to minimize the maximal distance between two consecutive segments. The first interval defined with this rule is the same than the one selected with the first rule. The results of the comparison between the different rules will be given at section 4.

3.2 Neighborhood

The neighborhood is the most important element of the PLS method. On the other hand, it is not trivial to produce neighbors from a feasible solution for the problem P. Removing one segment of the current solution is not enough. Removing two segments from the current solution requires to determine how to select both segments and how to recombine these two segments to produce a new feasible solution, ideally of better quality. Moreover, it will be difficult with this kind of technique to find neighbors with different DC values. Modifying the sequence of segments can improve the SU_{var} objective, but we can always apply a TSP algorithm at the end of the decomposition to improve this objective.

The neighborhood developed in this work is thus a bit complex. It works as follows:

1. Selection of a segment S that belongs to the current decomposition.
2. We modify a line i of S in the following way:

$$\begin{aligned} l_i &= l_i + (-1 \text{ or } 0 \text{ or } 1) \\ r_i &= r_i + (-1 \text{ or } 0 \text{ or } 1) \end{aligned}$$

3. We put S at the first place of the new decomposition.
4. We eventually modify the coefficient of this segment.
5. We construct a neighbor by adding the segments in the order of the current decomposition. If a segment is not feasible, we skip it. We adapt the coefficient of the segments added, which is equal to the minimum between the current coefficient of the segment and the maximal feasible coefficient.
6. The matrix that remains after adding all the possible segments is decomposed with the heuristic of Engel.
7. Once a decomposition is obtained, we optimize the SU_{var} objective by using a simple and fast TSP heuristic: the first improvement local search based on the 2-edges exchange moves [9].

This neighborhood requires the definition of many elements, but we will see that it is possible to explore many possibilities.

To illustrate the neighborhood, we show its functioning on the following example:

$$\mathbf{A} = \begin{pmatrix} 8 & 5 & 6 \\ 5 & 3 & 6 \end{pmatrix}$$

Ehrgott *et al.* [6] showed that for this example, DT and SU_{var} are contradictory, as well as DT and DC.

Let us consider that we start the neighborhood from the following solution, which minimizes the DT objective:

$$\mathbf{A} = \begin{pmatrix} 8 & 5 & 6 \\ 5 & 3 & 6 \end{pmatrix} = 3 \begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} + 1 \begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix} + 3 \begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} + 2 \begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$

The DT value of this solution is optimal and equal to 9, DC is equal to 4 and the SU_{var} value is equal to (2+2+2)=6.

We apply the neighborhood to this solution:

1. We select the first segment:
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$
2. We select the second line. For this line, $l = 2$ and $r = 4$. We modify this line by putting $l = 1$, $r = 3$. We obtain this segment:
$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix}$$
3. We put this segment at the first place of the new decomposition.
4. The current coefficient of this segment is equal to 3. As for this segment, the maximal feasible coefficient that we can put is 3, we keep the coefficient equal to 3. The remaining matrix is:
$$\begin{pmatrix} 5 & 5 & 6 \\ 5 & 0 & 6 \end{pmatrix}$$
5. The first segment that we can consider is:
$$\begin{pmatrix} 0 & 0 & 1 \\ 0 & 1 & 1 \end{pmatrix}$$
but we can not add it.
The second segment is:
$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix}$$
We add it, with a coefficient equal to 3, that is its current coefficient. The remaining matrix is:
$$\begin{pmatrix} 2 & 2 & 3 \\ 2 & 0 & 6 \end{pmatrix}$$
The last segment in the initial decomposition is:
$$\begin{pmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \end{pmatrix}$$
but we can not add it.
6. The decomposition of the remaining matrix with the heuristic of Engel gives:
$$5\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} + 2\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} + 1\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$
The decomposition obtained is thus:
$$3\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{pmatrix} + 3\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} + 5\begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \end{pmatrix} + 2\begin{pmatrix} 1 & 1 & 1 \\ 1 & 0 & 0 \end{pmatrix} + 1\begin{pmatrix} 0 & 0 & 1 \\ 0 & 0 & 1 \end{pmatrix}$$
But as we can see, it is possible to combine the second segment with the fourth one, to obtain this decomposition:

$$3\begin{pmatrix}1&0&0\\0&1&0\end{pmatrix}+5\begin{pmatrix}1&1&1\\1&0&0\end{pmatrix}+5\begin{pmatrix}0&0&0\\0&0&1\end{pmatrix}+1\begin{pmatrix}0&0&1\\0&0&1\end{pmatrix}$$

Therefore, each time we try to add a segment to a current decomposition, we check if we can combine this segment with other segments of the decomposition, in order to reduce the DC and SU_{var} values of the decomposition (this algorithm is not described in this work because of limited space).
The DT value of this solution is equal to 14, DC is equal to 4 and SU_{var} is equal to (2+3+3)=8.

7. The matrix of distances between the segments is as follows:

$$\begin{bmatrix} & S_1 & S_2 & S_3 & S_4 \\ S_1 & 0 & 2 & 1 & 2 \\ S_2 & 2 & 0 & 3 & 2 \\ S_3 & 1 & 3 & 0 & 3 \\ S_4 & 2 & 2 & 3 & 0 \end{bmatrix}$$

We see that by doing a 2-edges exchange move ($S_3 + S_1 + S_2 + S_4$), we can improve the SU_{var} objective by 3 units. We obtain the following neighbor:

$$5\begin{pmatrix}0&0&0\\0&0&1\end{pmatrix}+3\begin{pmatrix}1&0&0\\0&1&0\end{pmatrix}+5\begin{pmatrix}1&1&1\\1&0&0\end{pmatrix}+1\begin{pmatrix}0&0&1\\0&0&1\end{pmatrix}$$

The evaluation vector z of this solution is thus equal to (14,4,5).

We have thus obtained a new potentially efficient solution, and we can again apply the neighborhood from this solution:

1. We select the third segment:
$$\begin{pmatrix}1&1&1\\1&0&0\end{pmatrix}$$
2. We select the first line. For this line, $l = 0$ and $r = 4$. We modify this line by putting $r = 3$. We obtain this segment:
$$\begin{pmatrix}1&1&0\\1&0&0\end{pmatrix}$$
3. We put this segment at the first place of the new decomposition.
4. The current coefficient of this segment is equal to 5. As for this segment, the maximal feasible coefficient that we can put is 5, we keep the coefficient equal to 5. The remaining matrix is:
$$\begin{pmatrix}3&0&6\\0&3&6\end{pmatrix}$$
5. The first segment that we can consider is:
$$\begin{pmatrix}0&0&0\\0&0&1\end{pmatrix}$$

We add it, with a coefficient equal to 5, that is its current coefficient. The remaining matrix is:

$$\begin{pmatrix} 3\,0\,6 \\ 0\,3\,1 \end{pmatrix}$$

The second segment is:

$$\begin{pmatrix} 1\,0\,0 \\ 0\,1\,0 \end{pmatrix}$$

We add it, with a coefficient equal to 3, that is its current coefficient. The remaining matrix is:

$$\begin{pmatrix} 0\,0\,6 \\ 0\,0\,1 \end{pmatrix}$$

The last segment in the initial decomposition is:

$$\begin{pmatrix} 0\,0\,1 \\ 0\,0\,1 \end{pmatrix}$$

We add it, with a coefficient equal to 1, that is its current coefficient.

6. The remaining matrix is:

$$\begin{pmatrix} 0\,0\,5 \\ 0\,0\,0 \end{pmatrix}$$

The decomposition of this matrix with the heuristic of Engel gives:

$$5\begin{pmatrix} 0\,0\,1 \\ 0\,0\,0 \end{pmatrix}$$

The decomposition obtained is thus:

$$5\begin{pmatrix} 1\,1\,0 \\ 1\,0\,0 \end{pmatrix} + 5\begin{pmatrix} 0\,0\,0 \\ 0\,0\,1 \end{pmatrix} + 3\begin{pmatrix} 1\,0\,0 \\ 0\,1\,0 \end{pmatrix} + 1\begin{pmatrix} 0\,0\,1 \\ 0\,0\,1 \end{pmatrix} + 5\begin{pmatrix} 0\,0\,1 \\ 0\,0\,0 \end{pmatrix}$$

But as we can see, it is possible to combine the second segment with the last one, and then the result of this combination with the fourth segment to obtain this decomposition:

$$5\begin{pmatrix} 1\,1\,0 \\ 1\,0\,0 \end{pmatrix} + 3\begin{pmatrix} 1\,0\,0 \\ 0\,1\,0 \end{pmatrix} + 6\begin{pmatrix} 0\,0\,1 \\ 0\,0\,1 \end{pmatrix}$$

with a SU_{var} value equal to (1+2)=3.

7. The matrix of distances between the segments is as follows:

$$\begin{bmatrix} & S_1 & S_2 & S_3 \\ S_1 & 0 & 1 & 2 \\ S_2 & 1 & 0 & 2 \\ S_3 & 2 & 2 & 0 \end{bmatrix}$$

We see that it is impossible to improve the SU_{var} objective by doing 2-edges exchange moves.

The DT value of this solution is equal to 14, and the DC and SU_{var} values are equal to 3 (which are the optimal values [6]).

Therefore, by applying two times the neighborhood from a solution that minimizes the DT objective, we have found a solution that minimizes the DC and SU_{var} values.

It should be noted that we have found one more potential non-dominated point for this problem: the point (10,4,4).

For this small problem, we have thus found three potential non-dominated points: (9,4,6), (10,4,4) and (14,3,3).

3.3 Final optimization step

For each potentially efficient solution found at the end of the `PLS` method, we apply the LKH heuristic, to eventually improve the SU_{var} value of the solutions.

4 Results

4.1 Rules for the Heuristic of Engel

We experiment here the different rules for the selection of the intervals in the heuristic of Engel. The rule "Min" is to take the first feasible interval which is the closest to the preceding interval, the rule "First" is to take the first feasible interval, the rule "Last" is to take the last feasible interval and the rule "Kali" is the rule developed by Kalinowski (see section 3.1).

To compare the four rules, we use the same instances of Engel, that is, matrix 15x15 with randomly generated elements (uniformly distributed) between zero and the parameter L. For each value of L we make the average on 1000 different matrices for three values: the DC objective, the SU_{var} objective and the SU_{var} optimized objective which is the value of SU_{var} obtained after optimization of the sequence of segments with the LKH heuristic.

The results are given at table 4.1, for L going from 3 to 16.

We remark that for the DC objective, the "Kali" rule is better than the others. But if we want to minimize the SU_{var} objective, we remark that the "Last" rule allows to obtain better results for the SU_{var} optimized objective. If we do not consider the optimization step, the rule "First" is the best for $L = 3$ and $L = 4$, the "Last" rule is the best for L going from 5 to 13 and the rule "Min" is the best for L going from 14 to 16.

Therefore, the "Last" rule allows to considerably reduces the SU_{var} objective comparing to the rule of Kalinowski, even if this rule gives higher DC value on average. The running time of the heuristic of Engel with each of these rules is negligible.

4.2 Pareto Local Search

The initial population of `PLS` is updated with two solutions: the first one is a good approximation of lexmin(DT,DC,SU_{var}) (obtained with the "Kali" rule and LKH) and the second one is a good approximation of lexmin(DT,SU_{var},DC) (obtained with the "Last" rule and LKH). If there is one solution that dominates the

Table 1. DC, SU_{var}, SU_{var} optimized values obtained by the different rules. A=15x15.

	DC				SU_{var}				SU_{var} optimized			
L	Min	First	Last	Kali	Min	First	Last	Kali	Min	First	Last	Kali
3	10.32	10.32	9.93	**9.72**	79.69	**69.57**	70.38	113.68	77.43	63.05	**61.23**	103.14
4	11.73	11.74	11.33	**10.94**	94.22	**83.74**	84.10	127.63	91.72	74.72	**71.63**	114.44
5	12.69	12.69	12.29	**11.76**	104.81	96.55	**95.11**	137.43	101.82	84.85	**80.34**	122.28
6	13.56	13.56	13.14	**12.50**	112.35	107.29	**105.53**	146.62	108.98	93.19	**88.62**	129.35
7	14.27	14.26	13.85	**13.12**	119.49	117.02	**114.22**	153.55	115.75	101.14	**95.83**	134.90
8	14.91	14.90	14.48	**13.71**	126.53	126.12	**122.94**	160.69	122.39	108.12	**102.30**	140.40
9	15.52	15.46	15.09	**14.20**	133.32	133.65	**129.75**	166.67	128.38	114.32	**107.66**	144.95
10	16.04	15.98	15.59	**14.69**	139.10	141.08	**137.05**	172.46	134.04	120.17	**113.93**	149.60
11	16.46	16.37	16.01	**15.06**	144.16	147.09	**142.57**	177.08	138.56	125.22	**118.31**	152.92
12	16.92	16.81	16.49	**15.46**	149.41	153.17	**148.31**	182.08	143.47	130.06	**122.71**	156.71
13	17.30	17.18	16.87	**15.81**	153.56	158.54	**153.42**	186.16	147.34	134.47	**126.85**	160.10
14	17.62	17.53	17.21	**16.13**	**157.19**	162.91	157.96	189.64	150.70	138.10	**130.82**	162.75
15	18.01	17.91	17.55	**16.52**	**162.00**	168.27	163.02	194.47	154.91	142.46	**135.04**	166.35
16	18.29	18.16	17.89	**16.79**	**166.22**	172.34	167.41	197.64	158.84	146.25	**138.53**	169.18

other, the population will be composed of only one solution: the non-dominated one.

For the neighborhood, we adopt the following choices:

- We try all the possible segments for the segment that we put at the beginning of the new decomposition.
- Either we do not modify the segment or we modify it by trying all the possibilities of modification. We modify each line of the segment separately, by considering all the feasible possibilities for each line (equal to maximum 8).
- We try all the feasible coefficients for the segment that we put at the beginning of the decomposition.
- The remaining matrix is decomposed with the heuristic of Engel ("Last" rule).

If the number of segments of the current decomposition is equal to K and the maximal value of the matrix equal to L, a crude bound for the number of neighbors generated is equal to $KL(8m + 1)$.

To experiment the method, we use the same instances than in section 4.1. As no state-of-the-art results are known for this multiobjective problem, we use very specific indicators to measure the quality of the approximations obtained.

At table 4.2, we report the number of potential non-dominated points found (NNDP), the number of phases of the `PLS` method before convergence and the running time in seconds (on a Intel Core 2 Duo T7500 2.2 GHz processor). We indicate for each indicator the minimum, maximum and mean value found. For each value of L we make the average on 20 different matrices.

We remark that:

- The number of potential non-dominated points is not high: between 1 and 10, with a mean value between 1 and 4. The correlation between the objectives seems thus high for these instances.
- The mean number of phases is included between 3 and 7, what means that the neighborhood is efficient since at each phase improvements are realized. Please remind that if there is no new potential efficient solution generated during a phase, the PLS method stops since a Pareto local optimum set has been found.
- The mean running time is acceptable, between 1 and 41 seconds. But for some instances, the running time can be higher, until 99 seconds for an instance with $L = 15$.

To evaluate the quality of the results, we evaluate the improvement of DC by comparing the best value obtained with `PLS` to the value obtained by the heuristic of Engel with the "Kali" rule. We evaluate the improvement of SU_{var} by comparing the best SU_{var} value obtained with `PLS` to the SU_{var} value obtained with the heuristic of Engel with the "Kali" rule and LKH and to the value obtained with the heuristic of Engel with the "Last" rule and LKH. Two cases are distinguished: initially we evaluate the improvements made if we keep the optimal value for DT, and secondly, we have no restrictions on the DT objective.

Results are shown at table 4.2 (mean values). We see that the improvements of the DC value are very small. Indeed, the heuristic of Engel with the "Kali" rule is known to give near-optimal results for lexmin(DT,DC) on random instances [11]. On the other hand, the improvements of the SU_{var} values are remarkable. Comparing to the values obtained by the heuristic of Engel with the "Kali" rule and LKH, we obtain improvements from 22 % to 44 %. Comparing

Table 2. Indicators `PLS`(1)

	NNDP			Number of phases			Time(s)		
L	Mean	Min	Max	Mean	Min	Max	Mean	Min	Max
3	1.35	1	2	3.55	1	8	1.39	0.28	3.78
4	1.80	1	3	5.35	2	9	3.79	0.84	7.90
5	1.80	1	3	5.50	2	11	5.43	1.14	13.00
6	2.00	1	3	5.40	2	10	7.34	1.51	17.36
7	2.15	1	4	6.10	1	11	10.10	1.22	24.49
8	2.45	1	4	6.55	1	12	12.85	1.26	29.28
9	2.60	1	6	5.90	2	10	15.45	2.95	35.48
10	3.15	2	5	6.35	2	11	18.60	5.02	49.82
11	2.75	1	5	6.30	2	11	21.42	4.29	46.24
12	2.95	1	5	5.95	2	11	22.40	4.34	49.20
13	2.95	1	6	5.75	1	11	28.63	2.42	93.13
14	2.95	1	5	6.10	2	14	30.28	4.60	72.08
15	3.50	1	10	6.95	2	18	40.77	5.82	98.94
16	2.75	1	5	5.20	2	12	28.08	5.56	83.57

to the heuristic of Engel with the "Last" rule and LKH (which is one of the initial solution of PLS), we obtain improvements from 5 % to 12 %.

We also see that allowing to deteriorate the DT value is only interesting for some instances. Both objectives DC and SU_{var} seem thus very correlated with the DT objective.

Table 3. Indicators PLS(2)

	DT optimal			DT not necessary optimal		
	% DC	% SU_{var}		% DC	% SU_{var}	
L	Kali	Kali+LKH	Last+LKH	Kali	Kali+LKH	Last+LKH
3	0.00	44.35	12.49	0.45	44.52	12.77
4	0.42	40.11	8.12	0.42	40.11	8.12
5	0.83	37.90	12.20	0.83	37.90	12.20
6	0.38	33.93	7.98	0.38	33.93	7.98
7	0.00	31.18	7.74	0.00	31.24	7.83
8	0.33	33.32	7.79	0.33	33.32	7.79
9	0.00	31.06	7.97	0.00	31.23	8.17
10	0.33	28.68	6.75	0.33	28.98	7.12
11	0.31	26.11	5.61	0.31	26.47	6.06
12	0.00	25.30	6.81	0.00	25.30	6.81
13	0.31	25.36	6.11	0.31	26.25	7.19
14	0.29	24.69	5.70	0.29	24.78	5.81
15	0.29	24.44	6.90	0.29	25.76	8.53
16	0.00	22.59	6.65	0.00	22.89	7.00

5 Conclusion and Discussion

We have presented in this paper first results for the multiobjective decomposition of positive integer matrices, within the framework of the radiotherapy treatment.

More experimentations on different types of instances (size and characteristics) and on real instances will have to be carried out to obtain more information about the correlation between the objectives, and to validate the approach.

But first results obtained with PLS are encouraging since the method allows to improve state-of-the-art results. This method could be the basis for a more elaborated method, like an evolutionary multiobjective algorithm where PLS would be used as an intensification operator.

Acknowledgments

We thank Céline Engelbeen for introducing this subject and for several discussions. T. Lust thanks the "Fonds National de la Recherche Scientifique" for a research fellow grant (Aspirant FNRS).

References

1. E. Angel, E. Bampis, and L. Gourvès. A dynasearch neighborhood for the bicriteria traveling salesman problem. In X. Gandibleux, M. Sevaux, K. Sörensen, and V. T'kindt, editors, *Metaheuristics for Multiobjective Optimisation*, pages 153–176. Springer. Lecture Notes in Economics and Mathematical Systems Vol. 535, Berlin, 2004.
2. D. Baatar, H. Hamacher, M. Ehrgott, and G. Woeginger. Decomposition of integer matrices and multileaf collimator sequencing. *Discr Appl Math*, 152:6–34, 2005.
3. T. Bortfeld, A. Boyer, D. Kahler, and T. Waldron. X-ray filed compensation with multileaf collimators. *Int J Radiat Oncol Biol Phys*, 28(3):723–730, 1994.
4. R. Burkard. Open problem session. In *Conference on Combinatorial Optimization*, pages 24–29, Oberwolfach, November 2002.
5. M. Ehrgott, Ç. Güler, H. W. Hamacher, and L. Shao. Mathematical optimization in intensity modulated radiation therapy. *4OR: a Quarterly Journal of Operations Research*, 6(3):199–162, 2008.
6. M. Ehrgott, H. Hamacher, and M. Nußbaum. Decomposition of matrices and static multileaf collimators: a survey. In C. Alves, P. Pardalos, and L. Vincente, editors, *Optimization in medicine*, pages 27–48. Springer, Berlin, 2007.
7. K. Engel. A new algorithm for optimal multileaf collimator field segmentation. *Discrete Appl. Math.*, 152(1-3):35–51, 2005.
8. P. Hansen. Bicriterion path problems. *Lecture Notes in Economics and Mathematical Systems*, 177:109–127, 1979.
9. P. Hansen and N. Mladenovic. First vs. best improvement: An empirical study. *Discrete Appl. Math.*, 154:802–817, 2006.
10. K. Helsgaun. An effective implementation of the lin-kernighan traveling salesman heuristic. *European Journal of Operational Research*, 126:106–130, 2000.
11. T. Kalinowski. Realization of intensity modulated radiation fields using multileaf collimators. In *General Theory of Information Transfer and Combinatorics*, pages 1010–1055. Springer Berlin / Heidelberg, 2006.
12. L. Paquete, M. Chiarandini, and T. Stützle. Pareto Local Optimum Sets in the Biobjective Traveling Salesman Problem: An Experimental Study. In X. Gandibleux, M. Sevaux, K. Sörensen, and V. T'kindt, editors, *Metaheuristics for Multiobjective Optimisation*, pages 177–199, Berlin, 2004. Springer. Lecture Notes in Economics and Mathematical Systems Vol. 535.
13. R. A. C. Siochi. Minimizing static intensity modulation delivery time using an intensity solid paradigm. *Int J Radiat Oncol Biol Phys*, 43:671–680, 1999.
14. Z. Taşkin, J. Smith, H. Romeijn, and J. Dempsey. Optimal multileaf collimator leaf sequencing in IMRT treatment planning. Technical report, Department of Industrial and Systems Engineering, University of Florida, 2007.
15. G. M. G. H. Wake, N. Boland, and L. S. Jennings. Mixed integer programming approaches to exact minimization of total treatment time in cancer radiotherapy using multileaf collimators. *Computers and Operations Research*, 36(3):795–810, 2009.