DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Relatório Técnico

RT-MAC-9907

A COMBINATORIAL OPTIMIZATION TECHNIQUE FOR THE SEQUENTIAL DECOMPOSITION OF EROSIONS AND DILATIONS

Ronaldo Fumio Hashimoto, Junior Barrera e
Carlos Eduardo Ferreira

Maio de 1999

A Combinatorial Optimization Technique for the Sequential Decomposition of Erosions and Dilations

RONALDO FUMIO HASHIMOTO, JUNIOR BARRERA, CARLOS EDUARDO FERREIRA

[ronaldo, jb, cef]@ime.usp.br

Departamento de Ciência da Computação do IME-USP
Rua do Matão 1010, 05508-900, São Paulo, SP, Brasil.

Abstract. This paper presents a general algorithm for the automatic proof that an erosion (respectively, dilation) has a sequential decomposition or not. If the decomposition exists, an optimum decomposition is presented. The algorithm is based on a branch and bound search, with pruning strategies and bounds based on algebraic and geometrical properties deduced formally. This technique generalizes classical results as Zhuang and Haralick, Xu, and Park and Chin, with equivalent or improved performance. Finally, theoretical analysis of the proposed algorithm and experimental results are presented.

Keywords: structuring element, decomposition, Minkowski addition, erosion, dilation.

1. Introduction

The problem of designing automatically morphological [2] operators may be approached by a process composed of two main steps: *i* - learning of the target operator by the estimation of the structuring elements that characterize it in a standard representation form (e.g., the *sup-representation*, that is formed by the union of sup-generating operators or, the *inf-representation*, that is formed by the intersection of inf-generating operators); *ii* - transformation of the standard form morphological operator learned into other equivalent representations that permit more efficient implementations.

In this paper, we will study some aspects of step *ii* of this process. The choice of good representation structures depends on the architecture of the machine in which the morphological operator should be implemented. In general, sequential decompositions are more adequate for conventional machines, while the hybrid (sequential-parallel) ones are more adequate for parallel architectures.

We find in the literature some works that study the problem of transformation of decomposition structures. Barrera and Salas [5] presented a method for computing the sup-representation of any translation invariant (t.i.) set operator from any morphological representation of it. Barrera and Hashimoto [4] showed how the sup-representation of t.i. set operators can be transformed into the union of compositions of sup-generating operators with dilations.

However, the general problem of transforming the sup or inf-representation of t.i. set operators into sequential or hybrid representations is extremely hard and practically not studied. In fact, some aspects of this problem have been studied for the families of erosions and dilations.

The speed up achieved by representing erosions and dilations by sequential decompositions, in conventional machines, was quantitatively studied by Maragos [13, p. 77], who showed examples where the time complexity of the algorithms that implement erosions and dilations went from quadratic, in the sup or inf-representation, to linear, in the sequential decomposition.

Theoretically, the sequential decomposition of erosions and dilations can be viewed equivalently by two optics. On one hand, considering the sup-representation (respectively, inf-representation), an erosion (respectively, dilation) is the simplest operator that can be represented, since the sup-representation (respectively, inf-representation) of an erosion (respectively, dilation) is the erosion (respectively, dilation) itself [11, p. 86, Theorem 4.15]. On the other hand, the composition of erosions (respectively, dilations) is equivalent to the erosion (respectively, dilation) by the accumulate Minkowski addition of the structuring elements that characterize the erosions (respectively, dilations) [20, p. 47]. Therefore, the problem of transforming the sup-representation of an erosion (respectively, inf-representation of a dilation) in a sequential decomposition is equivalent to find a decomposition in terms of Minkowski additions for the structuring element that characterizes it.

Several researchers [24, 18, 22, 12, 14, 15, 23, 3, 8, 1] have studied the problem of decomposition of a structuring element as a sequence of Minkowski additions of smaller subsets and proposed different algorithms to generate it. Zhuang and Haralick [24] presented a tree-search algorithm for decomposition of an arbitrarily structuring element, where all elements in the decomposition have the prescribed fixed number k points. Xu [22] developed an algorithm for the decomposition of convex structuring elements in terms of subsets of the *elementary square* (i.e., the 3×3 square centered at the *origin*). Park and Chin [15] developed an extension of Xu's algorithm for the decomposition of *simply connected* (i.e., an 8-connected structuring element that contains no holes) structuring elements, where all elements in the decomposition are also simply connected.

We should remark that not all structuring elements have sequential decompositions by Minkowski additions [22]. Furthermore, it is not known an efficient algorithm for determining the existence of such decompositions for an arbitrary structuring element.

Here, we present a method for the generation of decompositions of any arbitrary structuring element as sequences of Minkowski additions of subsets of the elementary square. If there exist such decompositions, the method gives the one that uses the minimum number of subsets, otherwise, it proves that the structuring element is not decomposable.

As Zhuang and Haralick's work [24], the fundamental idea of the method proposed is the application of Combinatorial Optimization techniques, under algebraic constraints. The formulation adopted is a branch and bound search in a tree that represents the space of all possible sequences of subsets of the elementary square. In the search we look for valid solutions, pruning impossible ones. The efficiency of the pruning, that is supported by algebraic properties of Minkowski addition, is essential for the feasibility of the method. If no valid solution is found, then the structuring element has no decomposition.

Following this introduction, Section 2 presents the mathematical foundations of the paper. Section 3 presents the proposed branch and bound decomposition algorithm. Section 4 compares the proposed algorithm with other known algorithms [22, 15, 24]. Section 5 presents some experimental results. Finally, Section 6 presents some conclusions and future steps of this research.

2. Mathematical Foundations

This section gives the mathematical foundations necessary for presenting our decomposition algorithm. Subsection 2.1 states the problem of sequential decomposition of subsets of $\mathbb{Z}^2$ in terms of Minkowski additions. Subsection 2.2 recalls the formulation of Combinatorial Optimization problems by the branch and bound approach. Subsection 2.3 presents the main data structure used in our algorithm: the decomposition tree. Subsection 2.4 presents three strategies to prune nodes of a decomposition tree. Subsection 2.5 states a lower bound for the length of a decomposition.

2.1. Problem Statement

In this section, we present some definitions and properties in order to state the problem studied.

A finite subset of $\mathbb{Z}^2$ is called a *structuring element* (SE). We consider just non empty SE's.

For any $X \subseteq \mathbb{Z}^2$ and $y \in \mathbb{Z}^2$, X_y denotes the *translation* of X by y, that is, $X_y = \{x \in \mathbb{Z}^2 : x - y \in X\}$.

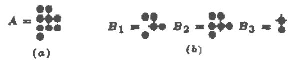

Fig. 1. (*a*) A SE *A*. (*b*) Invariant Sequence of *A*.

Let X and Y be SE's. The *Minkowski addition* and *subtraction* of X and Y are the subsets given, respectively, by $X \oplus Y = \cup\{X_y : y \in Y\}$ and $X \ominus Y = \cap\{X_{-y} : y \in Y\}$. .

Equivalently, the *Minkowski addition* of X and Y can be written as $X \oplus Y = \{x + y \in \mathbb{Z}^2 : x \in X, y \in Y\}$ [11, p. 81, Eq. 4.20].

Three important properties of Minkowski addition are commutativity (i.e., $X \oplus Y = Y \oplus X$) [11, p. 81], associativity (i.e., $(X \oplus Y) \oplus Z = Y \oplus (X \oplus Z)$) [11, p. 82, Eq. 4.29] and the translation effect (i.e., $X \oplus \{y\} = X_y$) [11, p. 82, Eq. 4.24].

We take the point $o = (0, 0)$ as the *origin* of $\mathbb{Z}^2$. We call the 3×3 square centered at the origin the *elementary square.*

The *dilation* and the *erosion* by the *structuring element* A are the mappings in the powerset of $\mathbb{Z}^2$ given, respectively, by, for any $X \subseteq \mathbb{Z}^2$, $\delta_A(X) = X \oplus A$ and $\epsilon_A(X) = X \ominus A$.

A property of dilations and erosions is their sequential decomposability [20, p. 47].

Proposition 1. *Let $A, B_1, B_2, \cdots, B_n$ be SE's. $\delta_A = \delta_{B_1}\delta_{B_2} \cdots \delta_{B_n}$ and $\epsilon_A = \epsilon_{B_1}\epsilon_{B_2} \cdots \epsilon_{B_n}$ if and only if $A = B_1 \oplus B_2 \oplus \cdots \oplus B_n$.*

Given a SE A, a *sequence of subsets of* A is the succession of subsets of A in a fixed order. For example, if $B_1, B_2, B_3, B_4, B_5, B_6, B_7$ are distinct subsets of A, then $[B_7, B_1, B_1, B_1, B_2, B_2, B_3, B_1, B_4, B_5, B_2, B_6]$ is a sequence of subsets of A. We consider just finite sequences.

Let $\mathsf{R}=[R_1, R_2, \cdots, R_m]$ and $\mathsf{S}=[S_1, S_2, \cdots, S_n]$ be sequences of subsets of a given SE A. We say R is a *subsequence* of S if and only if, for any $R_j \in \mathsf{R}$, there exists $S_{\pi(j)} \in \mathsf{S}$, such that $R_j = S_{\pi(j)}$ (i.e., $\mathsf{R}=[S_{\pi(1)}, S_{\pi(2)}, \cdots, S_{\pi(m)}]$), where $\pi(j)$ is a index in $\{1, 2, 3, \cdots, n\}$ and $\pi(1) < \pi(2) < \pi(3) < \cdots < \pi(m)$. For example, if $B_1, B_2, B_3, B_4, B_5, B_6$ are distinct subsets of A, then $[B_1, B_1, B_4, B_2]$ is a subsequence of $[B_6, B_1, B_1, B_1, B_2, B_2, B_3, B_1, B_4, B_5, B_2]$, but $[B_1, B_1, B_4, B_1]$ is not.

A SE A is said to have a *sequential decomposition* (or A is said to be *decomposable*) if there exists a sequence $[B_1, B_2, \cdots, B_n]$ of subsets of the elementary square such that $A = B_1 \oplus B_2 \oplus \cdots \oplus B_n$. The sequence $[B_1, B_2, \cdots, B_n]$ is called a *decomposition sequence* of A.

A decomposition sequence of a SE can be decomposed into two subsequences: *shape* and *translation.* The shape subsequence represents the shape of the SE and it is formed by the subsets in the sequence that have at least two points. The translation subsequence defines the position of the SE in the integer plane and it is formed by the unitary subsets in the sequence. The shape subsequence $[B_1, B_2, \cdots, B_k]$ is called the *shape decomposition* (or simply, *decomposition*) of A and the number k is the *length* of this decomposition of A.

Let A and B be SE's. We say B is an *invariant* of A if and only if $A = (A \ominus B) \oplus B$. For example, the subsets B_1, B_2, B_3, presented in Figure 1*b*, are invariants of the SE A presented in Figure 1*a*.

Propositions 2 and 3 give some properties of invariants of a given SE. The first one was stated by Serra [20, p. 53] and the second one by Zhuang and Haralick [24, Proposition 5].

Proposition 2. *Let A and X be SE's. Then, X is invariant of A if and only if there exists a SE Y such that $A = Y \oplus X$.*

Proposition 3. *Let A, X, Y be SE's. If $A = X \oplus Y$, then X and Y are both invariants of A.*

The following corollary is an immediate consequence of Proposition 3.

Corollary 1. *Let A be a SE. If the sequence $[B_1, B_2, \cdots, B_k]$ is a shape decomposition of A, then each B_i is an invariant of A.*

Proof: Since $[B_1, B_2, \cdots, B_k]$ is a shape decomposition of A, then there exists $h \in \mathbb{Z}^2$ such that $A = (B_1 \oplus B_2 \oplus \cdots \oplus B_k)_h$ or $A = (B_1 \oplus B_2 \oplus \cdots \oplus B_k \oplus \{h\}$. Hence, by commutativity and associativity of the Minkowski addition and Proposition 3, each B_i $(i = 1, 2, \cdots, k)$ is an invariant of A. □

Let X be a SE and let n be a positive integer. The succession of $n - 1$ Minkowski additions $((X \oplus X) \oplus \cdots \oplus X)$ is denoted nX. This notation is extrapolated for $n = 0$ by stating $0B = \{(0, 0)\}$.

Let A and X be SE's such that X is an invariant of A. The *multiplicity* of X with respect to A is the greatest positive integer n such that nX is an invariant of A. For example, the multiplicity of the subsets B_1, B_2 and B_3, presented in Figure 1*b*, with respect to A, presented in Figure 1*a*, is 1, since, for any $i \in \{1, 2, 3\}$, $2B_i$ is not an invariant of A. Note that unitary sets have infinity multiplicity.

Let us state an equivalence relation on a generic collection $\mathcal{C}$ of subsets of $\mathbb{Z}^2$. Let X and Y be two elements of $\mathcal{C}$. We say X and Y are *equivalent under translation* if and only if one can be built by a translation of the other, that is, $X \equiv Y$ if and only if there exists $h \in \mathbb{Z}^2$ such that $X_h = Y$.

Since the equivalence under translation is an equivalence relation (i.e., reflexive, symmetric and transitive), the set of their equivalence classes (i.e., the sets composed exactly by all the equivalent elements in $\mathcal{C}$) constitutes a partition of $\mathcal{C}$.

We denote by $\mathbf{P}(\mathcal{C})$ the set of all the equivalence classes (under translation) on $\mathcal{C}$. We denote by $\mathbf{E}(\mathcal{C})$ a set composed by exactly one element of each equivalence class in $\mathbf{P}(\mathcal{C})$, that is, $\mathbf{E}(\mathcal{C})$ is a set such that $|\mathbf{E}(\mathcal{C})| = |\mathbf{P}(\mathcal{C})|$.

The set of all subsets of the elementary square that have at least two points is denoted $\mathcal{Q} = \{B \subseteq \{-1, 0, 1\}^2 : |B| \geq 2\}$.

Given a SE A, the set of all elements of $\mathbf{E}(\mathcal{Q})$ that are invariants of A is denoted $\mathcal{B}(A) = \{B \in \mathbf{E}(\mathcal{Q}) : B \text{ is an invariant of } A\}$. For example, the set $\mathcal{B}(A)$ for the SE A presented in Figure 1*a* is $\mathcal{B}(A) = \{B_1, B_2, B_3\}$, where B_1, B_2 and B_3 are the sets presented in Figure 1*b*.

Proposition 4. *Let A be a SE and $X \in \mathcal{B}(A)$. If n is the multiplicity of X with respect to A, then any decomposition sequence of A contains at most n elements equal to X.*

Proof: Suppose that there exists a decomposition sequence of A that contains $m > n$ elements equal to X, that is, $A = mX \oplus B_1 \oplus B_2 \oplus \cdots \oplus B_k$. By Proposition 3, mX is an invariant of A that contradicts the definition of multiplicity. □

Let X be a SE and n be a non-negative integer. If $n \neq 0$, then the sequence formed by the succession of n subsets X is denoted by $\mathsf{Seq}[X, n]$, that is, $\mathsf{Seq}[X, n] = [X, X, \cdots, X]$. If $n = 0$, $\mathsf{Seq}[X, 0]$ denotes the empty sequence.

Let A be a SE. Let $B_1, B_2, \cdots, B_k$ be all elements of $\mathcal{B}(A)$ in a fixed order and n_i be the multiplicity of B_i with respect to A $(i = 1, \cdots, k)$. The *invariant sequence* of A is the sequence $\mathsf{InvSeq}[A] = \mathsf{Seq}[B_1, n_1] \cdots \mathsf{Seq}[B_k, n_k]$. For example, the sequence $[B_1, B_2, B_3]$ (of subsets presented in Figure 1*b*) is the invariant sequence of the SE A presented in Figure 1*a*.

The following proposition is an immediate consequence of Corollary 1 and Proposition 4.

Proposition 5. *If A is a SE, then A has a sequential decomposition if and only if there exists a subsequence of $\mathsf{InvSeq}[A]$ that is a shape decomposition of A.*

2.2. Combinatorial Optimization Techniques

Given a SE A, it is not possible to enumerate, in a reasonable time, all subsequences of $\mathsf{InvSeq}[A]$. This makes impossible any attempt to solve the problem of finding a decomposition of minimum length by explicit enumeration. We can view this problem as a Combinatorial Optimization problem. The aim of

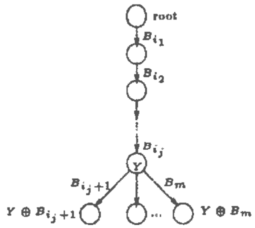

Fig. 2. A node Y in the decomposition tree.

optimization problems is to maximize or minimize a given function over a certain domain. Combinatorial Optimization problems are characterized by the case in which this domain is finite and its elements can be "easily" generated. The difficulty to solve such problems usually stays in the cardinality of the domain.

A strategy to partially enumerate the solutions of Combinatorial Optimization problem is known as the *Branch and Bound Method* [17, p. 40]. The idea is to partition the set of possible solutions into some small sets, originating independent subproblems. When one solves all these subproblems, the best solution found is the optimal solution of the original problem. This partition can be viewed as the construction of a rooted tree, whose nodes correspond to partial solutions, and the root node to the original problem. The search in the tree can be improved if some lower and upper bounds on the value of the optimal solution are known. For example, if it is known a feasible solution whose value is say 25, and we are in a node with lower bound 28, this branch can be pruned, since the best possible solution in this branch cannot be better than a known solution. The art of the strategy is to find good bounds in order to avoid visiting all (or too many) possibilities.

There are many different ways to visit the nodes of a tree. In the algorithm presented in Section 3, we use the depth first search [17, p. 39], that is described in the following. In the beginning all nodes are marked as "unvisited". In the first iteration, some arbitrary node is selected (if the tree has a root, this is the selected node) and marked as "visited". Its neighbors are marked "reachable", and pushed into a stack. In an arbitrary iteration, if the stack is empty, the search stops. Otherwise, the top element is taken out and marked as "visited". The still "unvisited" neighbors of this node are marked as "reachable", pushed into the stack, and a new iteration begins.

2.3. The Decomposition Tree

Given a SE A, we define a labeled tree that represents the space of all possible subsequences of $\mathsf{InvSeq}[A]$. Let $[B_1, B_2, \cdots, B_m]$ be the invariant sequence of A, i.e., $\mathsf{InvSeq}[A] = [B_1, B_2, \cdots, B_m]$. The ***decomposition tree*** of A, denoted $\mathcal{T}(A)$, is a labeled tree such that:

(1) All *nodes* are labeled by a subset $Y = B_{i_1} \oplus B_{i_2} \oplus \cdots \oplus B_{i_j}$, where $[B_{i_1}, B_{i_2}, \cdots, B_{i_j}]$ is a subsequence of $\mathsf{InvSeq}[A]$.
(2) The label of the root is the unitary set that contains the origin and it is denoted by $\{o\}$;
(3) The labels of the direct descendants of a node whose label is $Y = B_{i_1} \oplus B_{i_2} \oplus \cdots \oplus B_{i_j}$ are $Y \oplus B_{i_j+1}$, $Y \oplus B_{i_j+2}, \cdots, Y \oplus B_m$ (see Figure 2).
(4) The edge that joins a node whose label is Y and its direct descendant whose label is $Y \oplus B_k$ is labeled B_k. (see Figure 2).

We often use node Y meaning node whose label is Y.

By construction of the decomposition tree, it is not difficult to see that if the invariant sequence of a given SE A has m elements, then $\mathcal{T}(A)$ has 2^m nodes. See Figure 3 for an example of $\mathcal{T}(A)$.

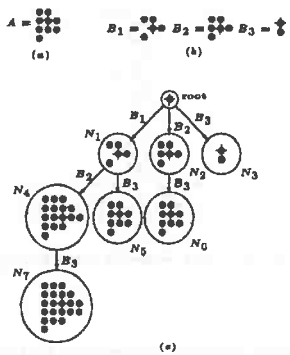

Fig. 3. (*a*) A SE A. (*b*) Invariant Sequence of A. (*c*) Decomposition Tree of A.

Fig. 4. (*a*) A SE A. (*b*) The convex hull of A.

Let A be a SE and let X and Y be two nodes of $\mathcal{T}(A)$ such that X is a descendant of Y. We denote $\mathsf{Path}_A[Y,X]$ the sequence of the labels of the edges of $\mathcal{T}(A)$ that are on the path from Y to X. It is not difficult to see that if $\mathsf{Path}_A[Y,X] = [C_1, C_2, \cdots, C_j]$, then $X = Y \oplus C_1 \oplus \cdots \oplus C_j$. For example, in Figure 3c, $\mathsf{Path}_A[N_1, N_7] = [B_2, B_3]$ and $N_7 = N_1 \oplus B_2 \oplus B_3$.

According to this notation, $\mathsf{Path}_A[\{o\}, Y]$ denotes the sequence of labels of the edges of $\mathcal{T}(A)$ that are on the path from the root to the node Y, which in turns forms a subsequence of $\mathsf{InvSeq}[A]$ such that the Minkowski addition of all elements of this subsequence is equal to the node Y. For example, in Figure 3c, $\mathsf{Path}_A[\{o\}, N_6] = [B_2, B_3]$ is a subsequence of $\mathsf{InvSeq}[A]$ and $N_6 = B_2 \oplus B_3$. So, if the SE A is equal to a translation of the node Y (that is, $Y \equiv A$), then A has a shape decomposition given by the sequence $\mathsf{Path}_A[\{o\}, Y]$.

The *level* of a node Y in the decomposition tree is the length of the path from the root to Y and it is denoted by *level*(Y). Given a SE A, our objective is to find a node of label $Y \equiv A$ in $\mathcal{T}(A)$ with minimum level. If there is no node $Y \equiv A$, then the SE is not decomposable.

Let A be a SE. Given a node Y of $\mathcal{T}(A)$, we say Y is a *feasible node* if and only if there exists a node X, descendant of Y, such that $X \equiv A$. For example, in Figure 3c, the node N_1 is a feasible node, while the node N_4 is an unfeasible one.

The art of the strategy presented in Section 3 is to visit the nodes of the decomposition tree, looking for feasible nodes and pruning unfeasible ones.

2.4. Necessary conditions for feasible nodes

In this subsection, we state some necessary conditions for the existence of feasible nodes in the decomposition tree of a given SE. For that, we analyze some measures taken on the SE.

The convex hull $C(A)$ of a subset $A \subseteq \mathbb{Z}^2$ is the intersection of all half planes that contain A. We suppose that the subsets are represented in a square grid and consider just the half planes with slopes 0, 45, 90 and 135 degrees to build the convex hull (see Figure 4 for an example).

A subset $A \subseteq \mathbb{Z}^2$ is said *convex* if and only if $A = C(A)$. Note that, in particular, for a half plane H, the convex hull $C(H) = H$.

The Propositions 6 to 11 give some properties of the convex hull.

Proposition 6. *Let $A, B \subseteq \mathbb{Z}^2$. Then,*

(i) $A \subseteq C(A)$,

(ii) $A \subseteq B \Rightarrow C(A) \subseteq C(B)$,

(iii) $(C(A))_h = C(A_h)$, for any $h \in \mathbb{Z}^2$.

Proof: Let $\mathcal{A}$ and $\mathcal{B}$ be the sets of all the half planes that contain, respectively, A and B.

Since $C(A)$ is the intersection of all half planes that contain A, then, by a property of intersection, $C(A) \supseteq A$. This proves *i*.

In order to prove *ii*, let $X \in \mathcal{B}$. Since $X \supseteq B \supseteq A$, then $X \in \mathcal{A}$. Thus, $\mathcal{B} \subseteq \mathcal{A}$. Hence,

$$\begin{aligned}
&(\bigcap_{Y \in \mathcal{B}} Y) \cap (\bigcap_{X \in \mathcal{A}} X) = \bigcap_{X \in \mathcal{A}} X \Leftrightarrow \\
&C(B) \cap C(A) = C(A) \Leftrightarrow \\
&C(A) \subseteq C(B).
\end{aligned}$$

Finally, to prove *iii*, let $h \in \mathbb{Z}^2$ and let $\mathcal{H}$ be the set all the half planes that contain A_h. Thus,

$$\begin{aligned}
H \in \mathcal{H} &\Leftrightarrow H \supseteq A_h \\
&\Leftrightarrow H_{-h} \supseteq A \\
&\Leftrightarrow H_{-h} \in \mathcal{A}
\end{aligned}$$

Hence,

$$\begin{aligned}
(C(A))_h &= (\bigcap_{H \in \mathcal{A}} H)_h \\
&= (\bigcap_{H \in \mathcal{A}} H_h) \\
&= (\bigcap_{H_{-h} \in \mathcal{A}} H) \\
&= (\bigcap_{H \in \mathcal{H}} H) \\
&= C(A_h).
\end{aligned}$$

□

Proposition 7. *If $A, B \subseteq \mathbb{Z}^2$, then*

$$(i)\ C(A) \cup C(B) \subseteq C(A \cup B),$$
$$(ii)\ C(A) \cap C(B) \supseteq C(A \cap B).$$

Proof: Let us prove *i*. Clearly, $A \subseteq A \cup B$ and $B \subseteq A \cup B$. Thus, by Proposition 6–*ii*, $C(A) \subseteq C(A \cup B)$ and $C(B) \subseteq C(A \cup B)$. Hence, by a property of union, $C(A) \cup C(B) \subseteq C(A \cup B)$. Property *ii* follows by dual arguments. □

Proposition 8. *If $A \subseteq \mathbb{Z}^2$, then $C(C(A)) = C(A)$.*

Proof: By Proposition 6, properties *i* and *ii*, it is easy to see that $C(A) \subseteq C(C(A))$. It remains to be proved that $C(C(A)) \subseteq C(A)$. Let $\mathcal{A}$ be the set all the half planes that contain A. Hence,

$$\begin{aligned} C(C(A)) &= C(\bigcap_{X \in \mathcal{A}} X) \\ &\subseteq \bigcap_{X \in \mathcal{A}} C(X) && \text{(by Proposition 7)} \\ &= \bigcap_{X \in \mathcal{A}} X && \text{(since } C(X) = X, \text{ for any } X \in \mathcal{A}) \\ &= C(A) \end{aligned}$$

Therefore, $C(C(A)) = C(A)$.

□

Proposition 9. *If $A, B \subseteq \mathbb{Z}^2$, then $C(A) \oplus B \subseteq C(A \oplus B)$.*

Proof:

$$\begin{aligned} C(A) \oplus B &= \bigcup_{b \in B} (C(A))_b \\ &= \bigcup_{b \in B} C(A_b) && \text{(By Proposition 6–}iii) \\ &\subseteq C(\bigcup_{b \in B} A_b) && \text{(By Proposition 7)} \\ &= C(A \oplus B). \end{aligned}$$

□

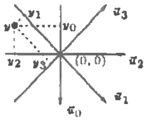

Fig. 5. A point y and the axis $\vec{u}_0$, $\vec{u}_1$, $\vec{u}_2$ and $\vec{u}_3$.

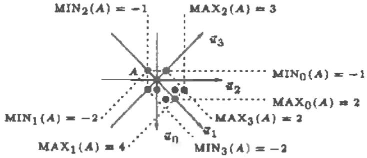

Fig. 6. A SE A with the axis $\vec{u}_0$, $\vec{u}_1$, $\vec{u}_2$ and $\vec{u}_3$.

Proposition 10. *If $A, B \subseteq \mathbb{Z}^2$, then $C(A \oplus C(B)) \subseteq C(A \oplus B)$.*

Proof:
By Proposition 9, $A \oplus C(B) = C(B) \oplus A \subseteq C(A \oplus B)$. By Proposition 6–*ii* and Proposition 8, $C(A \oplus C(B)) \subseteq C(C(A \oplus B)) = C(A \oplus B)$. □

Proposition 11. *If $A, B \subseteq \mathbb{Z}^2$, then $C(C(A) \oplus C(B)) = C(A \oplus B)$.*

Proof: First, let us prove that $C(A \oplus B) \subseteq C(C(A) \oplus C(B))$. By Proposition 6–*i*, $A \subseteq C(A)$ and $B \subseteq C(B)$. Then, $A \oplus B \subseteq C(A) \oplus B$ [11, p. 82, Eq. 4.30] and $C(A) \oplus B \subseteq C(A) \oplus C(B)$ [11, p. 82, Eq. 4.26]. Hence, $A \oplus B \subseteq C(A) \oplus C(B)$ and by Proposition 6–*ii*, $C(A \oplus B) \subseteq C(C(A) \oplus C(B))$.

Finally, we prove that $C(C(A) \oplus C(B)) \subseteq C(A \oplus B)$. By Proposition 9, $C(A) \oplus C(B) \subseteq C(A \oplus C(B))$.

Thus, $C(C(A) \oplus C(B)) \subseteq C(C(A \oplus C(B)))$
(By Proposition 6–*ii*)

$= C(A \oplus C(B))$
(By Proposition 8)

$\subseteq C(A \oplus B)$
(By Proposition 10)

Hence,

$$C(C(A) \oplus C(B)) = C(A \oplus B).$$

□

Let $\vec{u}_0$, $\vec{u}_1$, $\vec{u}_2$ and $\vec{u}_3$ be the Cartesian axis that intersect the origin and have slopes, respectively, −90, −45, 0 and 45 degrees (see Figure 5). For a given point $x \in \mathbb{Z}^2$, let $l_0(x)$, $l_1(x)$, $l_2(x)$, $l_3(x)$ be the orthogonal projections of x at the Cartesian axis $\vec{u}_0$, $\vec{u}_1$, $\vec{u}_2$ and $\vec{u}_3$, respectively. Observe that these projections are integer numbers at the directions −90 and 0 degrees and real numbers (proportional to $\frac{\sqrt{2}}{2}$) at the directions −45 and 45 degrees. Given a point $x \in \mathbb{Z}^2$, we denote by x_0, x_1, x_2 and x_3 the *normalized orthogonal projections* of the point x at the Cartesian axis $\vec{u}_0$, $\vec{u}_1$, $\vec{u}_2$ and $\vec{u}_3$ given, respectively, by $x_0 = l_0(x)$, $x_1 = l_1(x) \cdot \sqrt{2}$, $x_2 = l_2(x)$ and $x_3 = l_3(x) \cdot \sqrt{2}$. For example, the normalized projections of the point $y = (-5, 2) \in \mathbb{Z}^2$ presented in Figure 5 are $y_0 = -2$, $y_1 = -7$, $y_2 = -5$ and $y_3 = -3$.

Let A be a SE. For $i = 0, 1, 2, 3$, let $\text{MAX}_i(A)$ and $\text{MIN}_i(A)$ be, respectively, the maximum and the minimum normalized orthogonal projection at the Cartesian axis $\vec{u}_i$ of the points in A, that is,

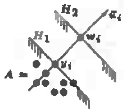

Fig. 7. Illustration for demonstration of Proposition 13.

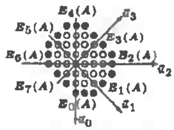

Fig. 8. The eight edges of a convex SE A.

$\mathrm{MAX}_i(A) = \max\{x_i : x \in A\}$ and $\mathrm{MIN}_i(A) = \min\{x_i : x \in A\}$. For example, the maximum and the minimum normalized orthogonal projections of the set A presented in Figure 6 are, respectively, $\mathrm{MAX}_0(A) = 2$, $\mathrm{MAX}_1(A) = 4$, $\mathrm{MAX}_2(A) = 3$, $\mathrm{MAX}_3(A) = 2$ and $\mathrm{MIN}_0(A) = -1$, $\mathrm{MIN}_1(A) = -2$, $\mathrm{MIN}_2(A) = -1$, $\mathrm{MIN}_3(A) = -2$.

Propositions 12 and 13 give some properties of $\mathrm{MAX}_i(A)$ and $\mathrm{MIN}_i(A)$ of a given SE A.

Proposition 12. *If A and B are SE's, then, for any $i \in \{0,1,2,3\}$,* $\mathrm{MAX}_i(A \oplus B) = \mathrm{MAX}_i(A) + \mathrm{MAX}_i(B)$ *and* $\mathrm{MIN}_i(A \oplus B) = \mathrm{MIN}_i(A) + \mathrm{MIN}_i(B)$.

Proof: $\mathrm{MAX}_i(A \oplus B) = \max\{x_i : x \in A \oplus B\} = \max\{a_i + b_i : a \in A, b \in B\} = \max\{a_i : a \in A\} + \max\{b_i : b \in B\} = \mathrm{MAX}_i(A) + \mathrm{MAX}_i(B)$. In the same way, $\mathrm{MIN}_i(A \oplus B) = \mathrm{MIN}_i(A) + \mathrm{MIN}_i(B)$. □

Proposition 13. *If A is a SE, then, for any $i \in \{0,1,2,3\}$,*

$$\mathrm{MAX}_i(C(A)) = \mathrm{MAX}_i(A) \text{ and } \mathrm{MIN}_i(C(A)) = \mathrm{MIN}_i(A).$$

Proof: For any $i = 0,1,2,3$, by definition of $\mathrm{MAX}_i(A)$, there exists $y \in A$ such that $y_i = \mathrm{MAX}_i(A)$. On one hand, $\mathrm{MAX}_i(A) \leq \mathrm{MAX}_i(C(A))$, since, by Proposition 6–i, $A \subseteq C(A)$. On other hand, $\mathrm{MAX}_i(A) \geq \mathrm{MAX}_i(C(A))$. In fact, suppose that there exists $w \in C(A)$ such that $\mathrm{MAX}_i(A) < w_i$. Let $H_1 = \{x \in \mathbb{Z}^2 : x_i \leq \mathrm{MAX}_i(A)\}$ and $H_2 = \{x \in \mathbb{Z}^2 : x_i \leq w_i\}$. Clearly, H_1 and H_2 are half planes that contain A and $w \notin H_1$ (see Figure 7). Since $C(A)$ is the intersection of all half planes that contain A, then, $w \notin C(A)$. But it contradicts the hypothesis that $w \in C(A)$. The proof for $\mathrm{MIN}_i(C(A)) = \mathrm{MIN}_i(A)$ can be done in a similar way. □

Let A be a SE. We define the eight *edges* of A, denoted by $E_0(A), \cdots, E_7(A)$, in the following way. For $i = 0,1,2,3$, $E_i(A)$ and $E_{i+4}(A)$ are the sets containing all points of $C(A)$ that have, respectively, the same maximum and minimum normalized orthogonal projection at axis $\vec{u}_i$, i.e., $E_i(A) = \{x \in C(A) : x_i = \mathrm{MAX}_i(A)\}$ and $E_{i+4}(A) = \{x \in C(A) : x_i = \mathrm{MIN}_i(A)\}$ (see Figure 8 for an example). Note that the axis u_i is perpendicular to edges $E_i(A)$ and $E_{i+4}(A)$ (see Figure 8 for an example). By construction of $E_i(A)$, $i = 0,1,\cdots,7$, and by Propositions 8 and 13, it is clear that $E_i(A) = E_i(C(A)) \subseteq C(A)$.

Given a SE A, the *next edge* of $E_i(A)$ is $E_{i+1}(A)$, if $i < 7$, or $E_0(A)$, if $i = 7$. The *last edge* of $E_i(A)$ is $E_{i-1}(A)$, if $i > 0$, or $E_7(A)$, if $i = 0$. For example, the next edges of $E_1(A)$ and $E_7(A)$ are, respectively, $E_2(A)$ and $E_0(A)$; the last edges of $E_5(A)$ and $E_0(A)$ are, respectively, $E_4(A)$ and $E_7(A)$. For simplicity of notation, we denote $E_{i-1}(A)$ and $E_{i+1}(A)$, respectively, the last and the next edges of $E_i(A)$.

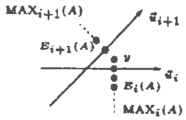

Fig. 9. Edges $E_i(A)$ and $E_{i+1}(A)$.

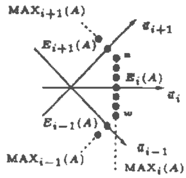

Fig. 10. Extremities of edge $E_i(A)$.

Given a SE A, Proposition 14 states a property of edges $E_i(A)$ and $E_{i+1}(A)$.

Proposition 14. *If A is a SE, then, for any $i \in \{0, 1, \cdots, 7\}$, $|E_i(A) \cap E_{i+1}(A)| = 1$.*

Proof:
We suppose that $i = 1$ or 2 (the other cases for $i = 0, 3, 4, 5, 6, 7$ can be proved in a similar way).
Consider the coordinate system formed by the Cartesian axis $\vec{u}_i$ and $\vec{u}_{i+1}$ (see Figure 9, for an example). In this coordinate system any point $x \in \mathbb{Z}^2$ can be uniquely represented by the ordered pair (x_i, x_{i+1}).
Clearly, $E_i(A) \cup E_{i+1}(A)$ is an 8−connected subset of $C(A)$ (see Figure 9). Thus, by definitions of $E_i(A)$ and $E_{i+1}(A)$, there is a point $y \in E_i(A) \cup E_{i+1}(A)$ such that $y_i = \text{MAX}_i(A)$ and $y_{i+1} = \text{MAX}_{i+1}(A)$ (see Figure 9). So, also by definitions of $E_i(A)$ and $E_{i+1}(A)$, $y \in E_i(A)$ and $y \in E_{i+1}(A)$. It remains to show that this point is unique. Suppose there exist two points $y, z \in E_i(A) \cap E_{i+1}(A)$. In this case, $y_i = z_i$ (since $y, z \in E_i(A)$) and $y_{i+1} = z_{i+1}$ (since $y, z \in E_{i+1}(A)$). Thus, $y = (y_i, y_{i+1}) = (z_i, z_{i+1}) = z$ and therefore $|E_i(A) \cap E_{i+1}(A)| = 1$. □

Given a SE A, by definition, for any $i \in \{0, 1, \cdots, 7\}$, $E_i(A)$ is a line formed by a segment of consecutive points of $C(A)$ at 0, 45, 90 or 135 degrees. Thus, each $E_i(A)$ contains at most two points that we call *extremities* of $E_i(A)$. More formally, the two extremities of $E_i(A)$ are the points $x, y \in E_i(A)$ such that $x \in E_{i+1}(A)$ and $y \in E_{i-1}(A)$. For example, in Figure 10, the points x and w are extremities of $E_i(A)$.
Given two SE's A and B, the following proposition gives an important property of edges of A and B.

Proposition 15. *If A and B are SE's, then, for any $i \in \{0, 1, \cdots, 7\}$, $E_i(A \oplus B) = E_i(A) \oplus E_i(B)$.*

Proof: We suppose that $i = 1$ or 2 (the other cases for $i = 0, 3, 4, 5, 6, 7$ can be proved in a similar way).
First, we prove that $E_i(A) \oplus E_i(B) \subseteq E_i(A \oplus B)$.
By definition of $E_i(A \oplus B)$, $x \in E_i(A \oplus B)$ if and only if $x \in C(A \oplus B)$ and $x_i = \text{MAX}_i(A \oplus B)$. So, in order to prove that $E_i(A) \oplus E_i(B) \subseteq E_i(A \oplus B)$, we have to show that, if $x \in E_i(A) \oplus E_i(B)$, then $x \in C(A \oplus B)$ and $x_i = \text{MAX}_i(A \oplus B) = \text{MAX}_i(A) + \text{MAX}_i(B)$ (by Proposition 12).
Since $E_i(A) \subseteq C(A)$ and $E_i(B) \subseteq C(B)$, then $E_i(A) \oplus E_i(B) \subseteq E_i(A) \oplus C(B) \subseteq C(A) \oplus C(B)$ [11, p. 82, Eq. 4.26]. Thus, by Proposition 6−i and 11, $E_i(A) \oplus E_i(B) \subseteq C(A \oplus B)$. It remains to show that $x_i = \text{MAX}_i(A) + \text{MAX}_i(B)$. By defintion of Minkowski addition, if $x \in E_i(A) \oplus E_i(B)$, then there

exist $y \in E_i(A)$ and $z \in E_i(B)$ such that $x = y + z$. By definition of $E_i(A)$, $y \in E_i(A)$ if and only if $y \in C(A)$ and $y_i = \mathrm{MAX}_i(A)$. Similarly, $z \in E_i(B)$ if and only if $z \in C(B)$ and $z_i = \mathrm{MAX}_i(B)$. Thus, since $x = y + z$, then $x_i = y_i + z_i = \mathrm{MAX}_i(A) + \mathrm{MAX}_i(B)$. Hence, $E_i(A) \oplus E_i(B) \subseteq E_i(A \oplus B)$.

Now, we prove that $E_i(A \oplus B) \subseteq E_i(A) \oplus E_i(B)$.

Since, by definition, $E_i(A)$ and $E_i(B)$ are lines formed by segments of consecutive points of $C(A)$ and $C(B)$, respectively, and the Cartesian axis u_i is perpendicular to $E_i(A)$ and $E_i(B)$, then clearly, by definition of Minkowski addition, $E_i(A) \oplus E_i(B)$ is a line formed by a segment of consecutive points of $C(A) \oplus C(B)$ and the Cartesian axis u_i is also perpendicular to $E_i(A) \oplus E_i(B)$. In addition, by definition, $E_i(A \oplus B)$ is a line formed by a segment of consecutive points of $C(A \oplus B)$ and the Cartesian axis u_i is perpendicular to $E_i(A \oplus B)$. So, if the extremities of $E_i(A \oplus B)$ belong to $E_i(A) \oplus E_i(B)$, then obviously $E_i(A \oplus B) \subseteq E_i(A) \oplus E_i(B)$.

Let t and z be extremities of $E_i(A \oplus B)$ such that $\{t\} = E_{i-1}(A \oplus B) \cap E_i(A \oplus B)$ and $\{z\} = E_i(A \oplus B) \cap E_{i+1}(A \oplus B)$. We will show that $z, t \in E_i(A) \oplus E_i(B)$. For that, let x and y be extremities of, respectively, $E_i(A)$ and $E_i(B)$ such that $\{x\} = E_i(A) \cap E_{i+1}(A)$ and $\{y\} = E_i(B) \cap E_{i+1}(B)$.

In the coordinate system formed by the Cartesian axis $\vec{u}_i$ and $\vec{u}_{i+1}$, we have $x = (x_i, x_{i+1})$, $y = (y_i, y_{i+1})$ and $z = (z_i, z_{i+1})$.

Since $x \in E_i(A)$, $y \in E_i(B)$ and $z \in E_i(A \oplus B)$, then, respectively, $x_i = \mathrm{MAX}_i(A)$, $y_i = \mathrm{MAX}_i(B)$ and $z_i = \mathrm{MAX}_i(A \oplus B)$. Thus, by Proposition 12, $z_i = \mathrm{MAX}_i(A \oplus B) = \mathrm{MAX}_i(A) + \mathrm{MAX}_i(B) = x_i + y_i$. Analogously, $z_{i+1} = x_{i+1} + y_{i+1}$. Hence, $z = (z_i, z_{i+1}) = (x_i, x_{i+1}) + (y_i, y_{i+1}) = x + y$.

In a similar way, if r, s are extremities of, respectively, $E_i(A)$ and $E_i(B)$ such that $\{r\} = E_{i-1}(A) \cap E_i(A)$ and $\{s\} = E_{i-1}(B) \cap E_i(B)$, then $t = r + s$.

Thus, since $E_i(A) \oplus E_i(B) = \{u + v : u \in E_i(A), v \in E_i(B)\}$, then $z = x + y$ and $t = r + s$ belong to $E_i(A) \oplus E_i(B)$.

□

Given a SE A, the *length of an edge* $E_i(A)$ is defined as $|E_i(A)| - 1$. The following proposition is an immediate consequence of Proposition 15.

Proposition 16. *If A and B are SE's, then, for any $i \in \{0, 1, \cdots, 7\}$, $|E_i(A \oplus B)| = |E_i(A)| + |E_i(B)| - 1$.*

Proof: Since $E_i(A)$ and $E_i(B)$ are lines formed by segments of consecutive points of $C(A)$ and $C(B)$, respectively, and the Cartesian axis $\vec{u}_i$ is perpendicular to both $E_i(A)$ and $E_i(B)$, then, by definition of Minkowski addition, clearly, the cardinality of $E_i(A) \oplus E_i(B)$ is equal to $|E_i(A)| + |E_i(B)| - 1$.

Therefore, $|E_i(A \oplus B)| = |E_i(A)| + |E_i(B)| - 1$, since by Proposition 15, $E_i(A \oplus B) = E_i(A) \oplus E_i(B)$. □

The *vector projection* of a given SE A is the vector $\nu(A) \in \mathbb{Z}^8$ such that its coordinates are the lengths of the edges of A (see Figures 11a, 11b and 11d for an example). More formally, the vector projection of the SE A is $\nu(A) = (\nu_0(A), \nu_1(A), \cdots, \nu_7(A))$, where $\nu_i(A)$ is the length of $E_i(A)$, that is, $\nu_i(A) = |E_i(A)| - 1$. Kanungo and Haralick [12] studied some properties and decomposition for convex SE's that are 4−connected (that they called *restricted domains*) and they used a boundary enconding scheme (called *B−coded*) that is very similar to the vector projection defined above.

Note that the vector projection is independent of translation, that is, $\nu(A) = \nu(A_h)$, for any $h \in \mathbb{Z}^2$. Note also that, for any SE X, $\nu(X) = \nu(C(X))$, since, by defitinion, $E_i(X) = E_i(C(X))$.

It is known that the chain code [6][7, p. 484] describes completely the shape of a convex SE. Without loss of generality, we assume that the chain code starts at 0 degree direction and runs counterclockwise. Thus, the chain code of a convex SE is represented by the sequence of numbers between 0 and 7: $0^{n_0}1^{n_1}\cdots 7^{n_7}$, where i^{n_i} ($n_i \geq 0$) is the string defined in the following way: i repeats n_i times if $n_i > 0$ or it is an empty string if $n_i = 0$. For example, the chain code of the SE presented in Figure 11b is shown in Figure 11c.

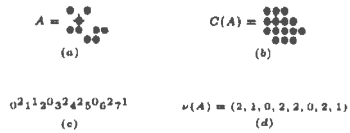

Fig. 11. (*a*) A SE A. (*b*) The convex hull of A. (*c*) The chain code of $C(A)$. (*d*) The vector projection of A.

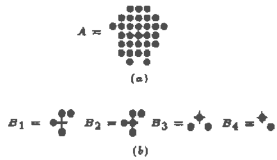

Fig. 12. (*a*) A SE A. (*b*) Invariant Sequence of A.

We can easily see that the vector projection of a given SE A and the chain code of $C(A)$ are equivalent definitions (see Figures 11*c* and 11*d* for an example).

An $O(n)$ time algorithm for computing the chain code of a convex SE A (or equivalently, the vector projection of A) can be found in [16, p. 143], where n is the number of points of A.

The next proposition gives an important property of vector projection. The same result for convex SE's that are 4–connected can be found in [12].

Proposition 17. *Let A, X and Y be SE's. If $A \equiv X \oplus Y$, then $\nu(A) = \nu(X) + \nu(Y)$.*

Proof: Since the vector projection is independent of translation, then for any $i \in \{0, 1, \cdots, 7\}$,

$$\begin{aligned}
\nu_i(A) &= \nu_i(X \oplus Y) \\
&= |E_i(X \oplus Y)| - 1 \\
&= (|E_i(X)| + |E_i(Y)| - 1) - 1 \\
&\qquad\qquad \text{(by Proposition 16)} \\
&= (|E_i(X)| - 1) + (|E_i(Y)| - 1) \\
&= \nu_i(X) + \nu_i(Y).
\end{aligned}$$

Therefore, $\nu(A) = \nu(X) + \nu(Y)$.

□

The following proposition gives a necessary condition for the existence of a decomposition for a given SE.

Proposition 18. *Let A be a SE. Let Z be the SE obtained by Minkowski addition of all subsets in the sequence* InvSeq$[A]$*. If there exists $i \in \{0, \cdots, 7\}$ such that $\nu_i(Z) < \nu_i(A)$, then A has no decomposition.*

Proof: Suppose that A has a decomposition. By Proposition 5, there exists a subsequence of InvSeq$[A]$, say $[B_1, B_2, \cdots, B_k]$, that is a shape decomposition of A. Hence, there exists $h \in \mathbb{Z}^2$ such that $A = (B_1 \oplus B_2 \oplus \cdots \oplus B_k)_h$. So, by Proposition 17, $\nu(A) = \nu(B_1) + \nu(B_2) + \cdots + \nu(B_k)$. Thus, $\nu_j(A) = \nu_j(B_1) + \nu_j(B_2) + \cdots + \nu_j(B_k) \leq \nu_j(Z)$, for any $j \in \{0, 1, \cdots, 7\}$, since Z is the Minkowski addition of all subsets in the sequence InvSeq$[A]$. But it contradicts the hypothesis that there exists $i \in \{0, 1, \cdots, 7\}$ such that $\nu_i(A) > \nu_i(Z)$. Therefore, A has no decomposition. □

For example, Figure 12*a* presents an undecomposable SE A. Let Z be the SE obtained by Minkowski addition of all subsets in the sequence $\mathsf{InvSeq}[A]$ (presented in Figure 12*b*), that is, $Z = B_1 \oplus B_2 \oplus B_3 \oplus B_4$. The vector projections of A and Z are, respectively, $\nu(A) = (2,2,2,2,2,2,2,2)$ and $\nu(Z) = (2,2,2,2,2,3,0,3)$. Thus, by Proposition 18, A has no decomposition, since $\nu_6(Z) < \nu_6(A)$.

The next proposition, a consequence of Proposition 17, states a necessary condition for feasible nodes, and, therefore, it gives a strategy to prune some unfeasible nodes of the decomposition tree of a given SE.

Proposition 19. *Let A be a SE. Let Y be a node in $\mathcal{T}(A)$. If there exists $i \in \{0,\cdots,7\}$ such that $\nu_i(Y) > \nu_i(A)$, then Y is not a feasible node.*

Proof: Suppose that Y is a feasible node. Thus, there exists a descendant of Y, say X, such that $X \equiv A$. Let $W \subseteq \mathbb{Z}^2$ be the set obtained by the Minkowski addition of all subsets in $\mathsf{Path}_A[Y,X]$. So, $X = Y \oplus W \equiv A$, and thus, there exists $h \in \mathbb{Z}^2$ such that $A = (Y \oplus W)_h$. Hence, by Proposition 17, $\nu(A) = \nu(Y) + \nu(W)$. Thus, for any $j \in \{0,1,\cdots,7\}$, $\nu_j(A) = \nu_j(Y) + \nu_j(W)$, and $\nu_j(A) \geq \nu_j(Y)$. But it contradicts the hypothesis that there exists $i \in \{0,1,\cdots,7\}$ such that $\nu_i(A) < \nu_i(Y)$. Therefore, Y is not a feasible node. □

For example, in Figure 3c, the vector projections of N_4 and A are, respectively, $\nu(N_4) = (2,2,0,2,2,0,4,0)$ and $\nu(A) = (1,1,1,1,1,0,3,0)$. Thus, by Proposition 19, N_4 is not a feasible node, since $\nu_0(N_4) > \nu_0(A)$.

The following proposition, an immediate consequence of Proposition 3, is another pruning strategy.

Proposition 20. *Let A be a SE. Let Y be a node in $\mathcal{T}(A)$. If Y is not an invariant of A, then Y is not a feasible node.*

Proof: Suppose that Y is a feasible node. Then, there exists a descendant of Y, say X, such that $X \equiv A$. Let $W \subseteq \mathbb{Z}^2$ be the set obtained by the Minkowski addition of all subsets in $\mathsf{Path}_A[Y,X]$. So, $X = Y \oplus W \equiv A$, and thus, there exists $z \in \mathbb{Z}^2$ such that $A = (Y \oplus W) \oplus \{z\} = Y \oplus (W \oplus \{z\})$. Hence, by Proposition 3, Y is an invariant of A. But it contradicts the hypothesis that the node Y is not an invariant of A. Therefore, Y is not a feasible node. □

The following definitions are necessary in order to get another pruning strategy (given by Proposition 22).

Let A be a SE. Let Y be a node of $\mathcal{T}(A)$. We denote $\mathsf{Direct}_A[Y]$ the sequence formed by the labels of the edges that join Y and its direct descendants, that is, if Y is the root, $\mathsf{Direct}_A[Y] = \mathsf{InvSeq}[A]$; if $Y = B_{i_1} \oplus \cdots \oplus B_{i_j}$, then $\mathsf{Direct}_A[Y] = [B_{i_j+1}, B_{i_j+2}, \cdots, B_m]$ (see Figure 2). For example, in Figure 3c, $\mathsf{Direct}_A[N_1] = [B_2, B_3]$. If X is a descendant of Y, then, by construction of $\mathcal{T}(A)$, the sequence $\mathsf{Path}_A[Y,X]$ is a subsequence of the sequence $\mathsf{Direct}_A[Y]$. For example, in Figure 3c, $\mathsf{Path}_A[N_1, N_5] = [B_3]$ is a subsequence of $\mathsf{Direct}_A[N_1] = [B_2, B_3]$.

Let A be a SE. Let Y be a node of $\mathcal{T}(A)$. We define the subsequence $\mathsf{Possible}_A[Y]$ of $\mathsf{Direct}_A[Y]$ in the following way. Let $D_1, D_2, \cdots, D_k$ be all distinct elements of $\mathsf{Direct}_A[Y]$ such that, for any $i \in \{1,2,\cdots,k-1\}$, D_i appears before D_{i+1} in $\mathsf{Direct}_A[Y]$. Let m_i be the number of occurrences of D_i in $\mathsf{Direct}_A[Y]$. Clearly, $\mathsf{Direct}_A[Y] = \mathsf{Seq}[D_1, m_1] \cdots \mathsf{Seq}[D_k, m_k]$. Let n_i be the greatest non-negative integer such that $(Y \oplus n_i D_i)$ is invariant of A. Let $d_i = \min\{n_i, m_i\}$. The *possible sequence* is $\mathsf{Possible}_A[Y] = \mathsf{Seq}[D_1, d_1] \cdots \mathsf{Seq}[D_k, d_k]$. For example, in Figure 3c, $\mathsf{Possible}_A[N_1] = B_3$, since $\mathsf{Direct}_A[N_1] = B_2, B_3$, $N_1 \oplus B_2 = N_4$ is not an invariant of A and $N_1 \oplus B_3 = N_5$ is an invariant of A.

In the next proposition, we have an interesting property for possible sequences. As a consequence of this property, we can get a new pruning strategy that is given by Proposition 22.

Proposition 21. *Let A be a SE. Let X, Y be two nodes of $\mathcal{T}(A)$ such that X is a descendant of Y. If $X \equiv A$, then $\mathsf{Path}_A[Y,X]$ is a subsequence of $\mathsf{Possible}_A[Y]$.*

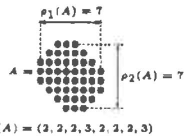

Fig. 13. Orthogonal projection of A.

Proof: Let $S_1, S_2, \cdots, S_k$ be all distinct elements of the sequence of $\mathsf{Path}_A[Y, X]$, such that, for any $i \in \{1, 2, 3, \cdots, k-1\}$, S_i appears before S_{i+1} in $\mathsf{Path}_A[Y, X]$. If s_i is the number of occurrences of S_i in $\mathsf{Path}_A[Y, X]$, then $\mathsf{Path}_A[Y, X] = \mathsf{Seq}[S_1, s_1] \cdots \mathsf{Seq}[S_k, s_k]$. Thus, $X = Y \oplus s_1 S_1 \oplus s_2 S_2 \oplus \cdots \oplus s_k S_k$. In order to prove that $\mathsf{Path}_A[Y, X]$ is a subsequence of $\mathsf{Possible}_A[Y]$, we have to show that:

(1) each $S_i \in \mathsf{Possible}_A[Y]$;

(2) if d_i is the number of occurrences of S_i in the sequence $\mathsf{Possible}_A[Y]$, then $s_i \leq d_i$;

(3) for any $i \in \{1, \cdots, k-1\}$, S_i appears before S_{i+1} in $\mathsf{Possible}_A[Y]$.

Since $X = Y \oplus s_1 S_1 \oplus \cdots \oplus s_k S_k \equiv A$, there exists $z \in \mathbb{Z}^2$ such that $A = Y \oplus s_1 S_1 \oplus \cdots \oplus s_k S_k \oplus \{z\}$. So, by Proposition 3, $(Y \oplus s_i S_i)$ is invariant of A and, by definition of $\mathsf{Possible}_A[Y]$, $S_i \in \mathsf{Possible}_A[Y]$. This proves (1).

Let n_i be the greatest positive integer such that $(Y \oplus n_i S_i)$ is invariant of A. Let m_i be the number of occurrences of S_i in $\mathsf{Direct}_A[Y]$. By construction of the sequence $\mathsf{Possible}_A[Y]$, the number of occurrences of S_i in $\mathsf{Possible}_A[Y]$ is $d_i = \min\{n_i, m_i\}$. If $d_i = n_i$, then $s_i \leq n_i$, since $Y \oplus s_i S_i$ is invariant of A. If $d_i = m_i$, then $s_i \leq m_i$, since $\mathsf{Path}_A[Y, X]$ is a subsequence of $\mathsf{Direct}_A[Y]$. This proves (2).

Suppose that there exists $j \in \{1, 2, \cdots, k-1\}$ such that S_{j+1} appears before that S_j in $\mathsf{Possible}_A[Y]$. Since the sequence $\mathsf{Possible}_A[Y]$ is a subsequence of the sequence $\mathsf{Direct}_A[Y]$, then S_{j+1} appears before that S_j in $\mathsf{Direct}_A[Y]$. Since $\mathsf{Path}_A[Y, X]$ is a subsequence of $\mathsf{Direct}_A[Y]$, then S_{j+1} appears before that S_j in the sequence $\mathsf{Path}_A[Y, X]$. But it is a contradiction, since, by construction of $\mathsf{Path}_A[Y, X]$, S_j appears before that S_{j+1} in the sequence $\mathsf{Path}_A[Y, X]$. Therefore, for any $i \in \{1, 2, \cdots, k-1\}$, S_i appears before S_{i+1} in the sequence $\mathsf{Possible}_A[Y]$. This proves (3). □

The following proposition gives a new pruning strategy.

Proposition 22. *Let A be a SE. Let Y be a node in $\mathcal{T}(A)$. Let $Z \subseteq \mathbb{Z}^2$ be the set obtained by Minkowski addition of all subsets in the sequence $\mathsf{Possible}_A[Y]$. If there exists $i \in \{0, \cdots, 7\}$ such that $\nu_i(Z) < \nu_i(A) - \nu_i(Y)$, then Y is not a feasible node.*

Proof: Suppose that Y is a feasible node. Thus, there exists a descendant of Y, say X, such that $X \equiv A$. Let $W \subseteq \mathbb{Z}^2$ be the set obtained by Minkowski addition of all subsets in $\mathsf{Path}_A[Y, X]$. So, $X = Y \oplus W \equiv A$, and thus, there exists $h \in \mathbb{Z}^2$ such that $A = (Y \oplus W)_h$. Hence, by Proposition 17, we get $\nu(A) = \nu(Y) + \nu(W)$. Since, by Proposition 21, the sequence $\mathsf{Path}_A[Y, X]$ is a subsequence of $\mathsf{Possible}_A[Y]$, then, for any $j \in \{0, \cdots, 7\}$, $\nu_j(Z) \geq \nu_j(W) = \nu_j(A) - \nu_j(Y)$. But, it is a contradiction, because we assume that there exists $i \in \{0, \cdots, 7\}$ such that $\nu_i(Z) < \nu_i(A) - \nu_i(Y)$. Therefore, Y is not a feasible node. □

With the pruning strategies given by Propositions 19, 20 and 22 we can avoid some (not all) unfeasible nodes of the decomposition tree.

In the next subsection, we present a lower bound for the length of shape decompositions. If a shape decomposition of a given SE A is found, we can check if it is an optimum solution verifying if the lower bound is equal to the length of the solution found.

2.5. Lower Bound

In order to state a lower bound for the length of shape decompositions of a given SE, we define a new measure taken on SE's.

The *orthogonal projection* of a SE A is the vector $\rho(A) \in \mathbb{Z}^2$ such that, $\rho_1(A) = \nu_3(A) + \nu_4(A) + \nu_5(A)$ and $\rho_2(A) = \nu_1(A) + \nu_2(A) + \nu_3(A)$. See Figure 13 for an example. In other words, the coordinates of the orthogonal projection of a SE A are the lengths of the edges of the smallest rectangle that contains A.

Since the vector projection is independent of translation, we have that, for any $h \in \mathbb{Z}^2$, $\rho(A) = \rho(A_h)$. Since, for any SE X, $\nu(X) = \nu(C(X))$, we have that $\rho(X) = \rho(C(X))$.

The next result is an immediate consequence of Proposition 17.

Proposition 23. *Let A, X and Y be SE's. If $A \equiv X \oplus Y$, then $\rho(A) = \rho(X) + \rho(Y)$.*

The following proposition gives a lower bound for the length of shape decompositions of a SE by Minkowski additions.

Proposition 24. *Let A be a SE. If A has a decomposition, then a shape decomposition of A contains at least $lower(A) = \lceil \max\{\rho_1(A), \rho_2(A)\}/2 \rceil$ elements.*

Proof: Let $[B_1, B_2, \cdots, B_m]$ be a shape decomposition of the SE A. Then, $A \equiv B_1 \oplus B_2 \oplus \cdots \oplus B_m$, and thus, there exists $h \in \mathbb{Z}^2$ such that $A = (B_1 \oplus B_2 \oplus \cdots \oplus B_m)_h$. Let S_i be the 3×3 square that contains B_i, for $i = 1, 2, \cdots, m$. Clearly, $A \subseteq (S_1 \oplus S_2 \oplus \cdots \oplus S_m)_h$, and the orthogonal projection of $(S_1 \oplus S_2 \oplus \cdots \oplus S_m)_h$ is $\rho((mS)_h) = \rho(mS) = (2m, 2m)$, where S is the elementary square.

Hence, since $A \subseteq (S_1 \oplus S_2 \oplus \cdots \oplus S_m)_h$, then $\rho_1(A) \leq 2m$ and $\rho_2(A) \leq 2m$. Thus, $\max\{\rho_1(A), \rho_2(A)\} \leq 2m$ and, therefore, $m \geq \lceil \max\{\rho_1(A), \rho_2(A)\}/2 \rceil$. □

Notice that, given a SE A, the length of the optimum solution of A must be greater or equal than the lower bound stated by Proposition 24. Besides, it is not the only lower bound that can be computed. Others can be determined using sophisticated combinatorial and optimization techniques (relaxation, primal-dual, etc...). In this work, we just consider the lower bound fixed by Proposition 24.

3. Search of Optimum Decomposition

In this section, we present an algorithm for finding an optimum shape decomposition of a given SE A.

The following proposition characterizes a node Y in the decomposition tree of a given SE A such that $Y \equiv A$.

Proposition 25. *Let A be a SE and let Y be a node of $\mathcal{T}(A)$. Then, $Y \equiv A$ if and only if Y is an invariant of A and $\nu(Y) = \nu(A)$.*

Proof: ($\Rightarrow$) If $Y \equiv A$, then, there exists $h \in \mathbb{Z}^2$ such that $Y = A_h = A \oplus \{h\}$. Hence, by Proposition 3, Y is invariant of A. Since $\nu(A_h) = \nu(A)$, then, $\nu(Y) = \nu(A)$.

($\Leftarrow$) Since Y is invariant of A, by Proposition 2, there exists a SE X such that $A = X \oplus Y$. By Proposition 17, $\nu(A) = \nu(X) + \nu(Y)$. Since $\nu(A) = \nu(Y)$, then $\nu(X) = 0$ and therefore $|X| = 1$. Let $h \in \mathbb{Z}^2$ such that $X = \{h\}$. In this case, the Minkowski addition $X \oplus Y = A$ is a translation of the set Y by h. Hence, $Y \equiv A$. □

Let A be a SE. When a node Y of $\mathcal{T}(A)$ such that Y is an invariant of A and $\nu(Y) = \nu(A)$ is found, then, by Proposition 25, $Y \equiv A$, and therefore, the sequence $\text{Path}_A[\{o\}, Y]$ is a shape decomposition of A. If *level*(Y) is equal to *lower*(A) (the lower bound fixed by Proposition 24), then $\text{Path}_A[\{o\}, Y]$ is an optimum solution. Otherwise, the optimum solution contains at most *level*(Y) elements. So, we get an upper bound for decomposition of A. Note that this upper bound can change dynamically. We denote *c.u.b.* the current upper bound.

A crucial task of the procedure is an adequate pruning of the decomposition tree when its nodes are being visited. Proposition 18 gives a necessary condition for the existence of a decomposition for A. Thus, we should begin the search just if this condition is satisfied. Proposition 24 fixes the lower bound for a decomposition of A, while Propositions 19, 20 and 22 guarantees important pruning, since they detect unfeasible nodes.

Under this context, after setting *c.u.b.* with infinite, the search dynamics goes on. In the first iteration, the root is selected to be visited. In an arbitrary iteration, when the node Y is being visited:

(*i*) – *Verify if this node can be pruned.*

There are four pruning strategies that have to be checked:

(*i.a*) *Pruning by Upper Bound.*

$level(Y) \geq c.u.b.$

(*i.b*) *Pruning by Projection.*

There exists i such that

$\nu_i(Y) > \nu_i(A)$

(*i.c*) *Pruning by Invariance.*

Y is not invariant of A.

(*i.d*) *Pruning by Possible Sequence.*

Let $Z \subseteq \mathbb{Z}^2$ be the set obtained by Minkowski addition of all subsets in the sequence $\mathsf{Possible}_A[Y]$.

If there exists i such that

$\nu_i(Z) < \nu_i(A) - \nu_i(Y)$

If one of these conditions is satisfied, then the node Y is pruned and a new iteration begins.

(*ii*) – *Verify if this node is a solution.*

Y is a solution if the following two conditions are satisfied:

- $level(Y) \geq lower(A)$ and
- $\nu(Y) = \nu(A)$.

In the case of these two conditions are satisfied, Y is an invariant of A (otherwise, Y would be pruned in step (*i.c*)) and, since $\nu(Y) = \nu(A)$, by Proposition 25, $Y \equiv A$, and then, $\mathsf{Path}_A[\{o\}, Y]$ is a shape decomposition of A.

(*ii.a*) – *This node is a solution.*

There are two possibilities:

- if $level(Y) = lower(A)$
  $\Rightarrow$ The search stops.
- if $level(Y) > lower(A)$
  $\Rightarrow c.u.b. \leftarrow level(Y)$
  and a new iteration begins.

(*ii.b*) – *This node is not a solution.*

The algorithm begins a new iteration.

In Figure 14, we show a simple example of the algorithm running for finding a shape decomposition of the SE A presented in Figure 14*a*. The invariant sequence of A is presented in Figure 14*b*. The vector projection, the orthogonal projection and the lower bound of A are presented in Figure 14*c*. The root is selected to be visited in Figure 14*d*. The node N_1 is being visited in Figure 14*e*. The node N_4 is being visited in Figure 14*f*. The node N_4 is pruned (pruning by Projection, see Proposition 19) in Figure 14*g*. The node N_5 is being visited in Figure 14*h*. Since $N_5 \equiv A$, then $\mathsf{Path}_A[\{o\}, N_5] = [B_1, B_3]$ is a decomposition of A. Besides, since $level(N_5) = 2 = lower(A)$, then $\mathsf{Path}_A[\{o\}, N_5]$ is an optimum

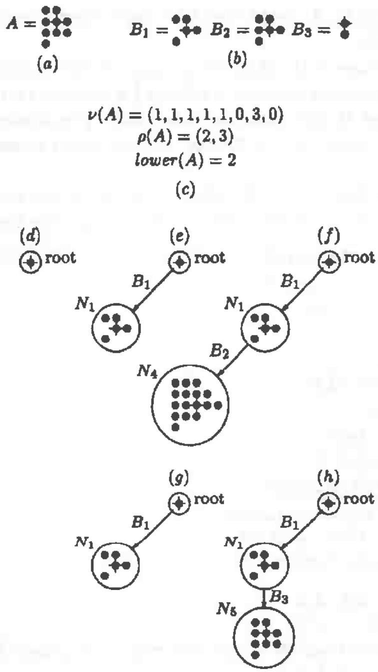

Fig. 14. An example showing the algorithm running.

solution and the search stops. In this example, only the pruning by projection was detected. The other pruning strategies are detected in a similar way.

Proposition 26. *Let A be a SE. If m is the number of elements in the invariant sequence of A, then $m = O(n)$, where n is the sum of all coordinates of the vector projection $\nu(A)$.*

Proof: Clearly, the multiplicity of a given SE with respect to A is at most $\max\{\rho_1(A), \rho_2(A)\} = O(n)$ and the number of all possible subsets of the elementary square is 2^9. Thus, the number of elements in InvSeq$[A]$ is at most $2^9 \cdot \max\{\rho_1(A), \rho_2(A)\} = O(n)$. Therefore, $m = O(n)$. □

Since the decomposition tree of a given SE A contains 2^m nodes, where m is the number of elements in the invariant sequence of A, and by Proposition 26, the time complexity of our algorithm, in the worst case, is $O(2^n)$, where n is the sum of all coordinates of $\nu(A)$.

4. Comparison with some known Algorithms

In this section, we compare the algorithm presented in Section 3 with some known algorithms. Subsection 4.1 compares with Zhuang and Haralick's algorithm [24]. Subsection 4.2 compares with Xu's algorithm [22]. Subsection 4.3 compares with Park and Chin's algorithm [15].

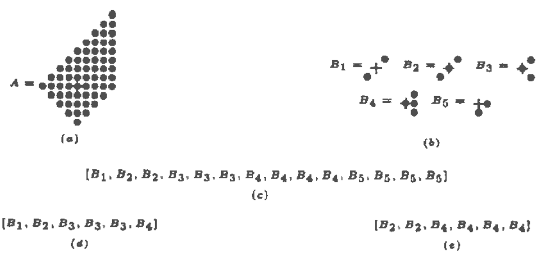

Fig. 15. (*a*) A SE A. (*b*) The subsets of the elementary square that are in $\mathcal{B}(A)$. (*c*) The invariant sequence of A according to the order chosen to construct it. (*d*) Output of our algorithm. (*e*) Output of Xu's algorithm.

4.1. Zhuang and Haralick's Algorithm

Zhuang and Haralick [24] presented an algorithm for finding the optimum decomposition of an arbitrarily SE, where all elements in the decomposition have the prescribed fixed number k points. Their algorithm perfoms the breadth-first search in a tree (that they called *tree search*) and the essence of the solution technique is divided into two parts: (i) the recognition that SE's participating in the decomposition must have points which are the differences between points of the given SE and (ii) the reduction of the search space using pruning by the invariance strategy that they called *forward checking*.

The breadth-first search has advantages and disadvantages over the depth-first search. The main advantage it that the first solution found is always the optimum one. The principal disadvantage is that all nodes in the current level have to be kept in memory. So, we decided to use the depth-first search because it requires less memory and, in our experiments (see Section 5), we observed that the distance between the optimum and the first solution found usually is small.

The problem decomposition considered in our work (stated in Subsection 2.1) is a special case of the problem studied by Zhuang and Haralick (with $k = 2, 3, \cdots, 9$ and each SE in the decomposition being a subset of the elementary square). In this particular case, our algorithm can reduce more the search space, since it uses two more pruning strategies (prunings by projection and possible sequence).

4.2. Xu's Algorithm

Xu [22] developed an algorithm for finding the optimum decomposition of convex SE's in terms of subsets of the elementary square, where all SE's in the decomposition are also convex.

In order to use Xu's algorithm for decomposing a convex SE A, it is necessary to compute the chain code of A. The time complexity of the algorithm given in [16, p. 143] for determining the chain code is linear with respect to the number of points in the whole SE A, or equivalently, the time complexity is $O(n^2)$, where n is the sum of all coordinates of the vector projection $\nu(A)$. In this subsection, we show that, if the input of our algorithm is a convex SE, then its time complexity is $O(n^4)$ and its output can contain non-convex SE's (in this sense, it is more general than Xu's algorithm).

Depending on the order chosen to construct the invariant sequence, different heuristic search procedures arise. We have sorted the elements of the invariant sequence in decreasing order, according to the sum of the coordinates of the orthogonal projections of each subset in the invariant sequence, and, at the same time, in increasing order, according to the number of points of each subset in the invariant sequence. For example, Figure 15*c* presents the invariant sequence of the SE A (presented in Figure 15*a*) according to the order chosen to construct it. In this figure, observe that $\rho_0(B_1) + \rho_1(B_1) = \rho_0(B_2) + \rho_1(B_2) > \rho_0(B_3) + \rho_1(B_3)$ and B_1 contains less points than B_2.

According to this sorting, the algorithm prefers to choose non-convex SE's rather than convex ones for the shape decomposition. Thus, as the time complexity of algorithms that implement erosions and dilations depends on the number of points in the SE, our algorithm has an advantage over Xu's algorithm, since all elements in the output of Xu's algorithm are convex subsets of the elementary square [22]. For an example, in Figures 15*d* and 15*e* are presented, respectively, the output of our and Xu's algorithm. In this particular example, the difference is just four points, but for bigger SE's the difference can be considerable.

Given a SE A and a node Y of $\mathcal{T}(A)$, let B be the first element in $\text{Direct}_A[Y]$ such that $Y \oplus B$ is an invariant of A. We define the node $Y \oplus B$ as *the leftmost invariant direct descendant of* Y. We define *the leftmost node sequence* of the decomposition tree $\mathcal{T}(A)$ as the sequence $[Y_0, Y_1, Y_2, \cdots, Y_k]$ formed by the nodes of $\mathcal{T}(A)$ such that Y_0 is the root (i.e., the unitary set that contains the origin) and, for $i = 1, 2, \ldots, k$, Y_i is the leftmost invariant direct descendant of Y_{i-1}.

Given a convex SE A, if the invariant sequence of A is built in the manner described above, then the following proposition, proved in [10], gives an important result in order to prove that the output of the algorithm is an optimum shape decomposition of A.

Proposition 27. *Let A be a convex SE. If the sequence $[Y_0, Y_1, Y_2, \cdots, Y_k]$ is the maximal leftmost node sequence of $\mathcal{T}(A)$, then $lower(Y_k) = k$ and $Y_k \equiv A$.*

The next proposition is an immediate consequence of Proposition 27.

Proposition 28. *Let A be a convex SE. If the sequence $[Y_0, Y_1, Y_2, \cdots, Y_k]$ is the maximal leftmost node sequence of $\mathcal{T}(A)$, then $\text{Path}_A[\{o\}, Y_k]$ is an optimum shape decomposition of A.*

Proof: By Proposition 27, $lower(Y_k) = k$ and $Y_k \equiv A$. So, $\nu(Y_k) = \nu(A)$, and, consequently, $k = lower(Y_k) = lower(A)$. Since $\text{Path}_A[\{o\}, Y_k]$ contains exactly k elements and $k = lower(A)$, then, clearly, $\text{Path}_A[\{o\}, Y_k]$ is the optimum shape decomposition of A. □

Given a convex SE A, when a maximal leftmost node sequence of $\mathcal{T}(A)$, say $[Y_0, Y_1, Y_2, \cdots, Y_k]$, is found, then, by Proposition 28, $\text{Path}_A[\{o\}, Y_k]$ is the optimum shape decomposition of A. It remains to show that the time complexity to find the first maximal leftmost node sequence is $O(n^4)$, where n is the sum of all coordinates of the vector projection $\nu(A)$.

Let A and Y be SE's such that Y is a node of $\mathcal{T}(A)$. Let $\text{Path}_A[\{o\}, Y] = [B_1, B_2, \cdots, B_k]$. In order to verify if Y is an invariant of A, we have to check if $A = (A \ominus Y) \oplus Y = (\cdots((A \ominus B_1) \ominus B_2 \ominus \cdots \ominus B_k) \oplus B_1) \oplus B_2 \oplus \cdots \oplus B_k)$. The time complexity for computing $A \ominus B_i$ or $A \oplus B_i$ is linear with respect to the number of points in A, since B_i contains at most 9 points. If n is the sum of all coordinates of the vector projection $\nu(A)$, then the time complexity for computing $A \ominus B_i$ or $A \oplus B_i$ is $O(n^2)$. So, the overall complexity for verifying if a node Y at level k is an invariant of A is $O(k \cdot n^2)$.

Since the algorithm presented in Section 3 uses the depth first search, then, the first maximal sequence of nodes visited by our algorithm is the maximal leftmost node sequence $[Y_0, Y_1, Y_2, \cdots, Y_k]$ of $\mathcal{T}(A)$. By Proposition 28, $k = lower(A)$. Hence the time taken for finding the maximal leftmost node sequence of $\mathcal{T}(A)$ is $O(1 \cdot n^2) + O(2 \cdot n^2) + \cdots + O(lower(A) \cdot n^2)$. Therefore, the time complexity of the algorithm for finding an optimum decomposition of a convex SE A is $O(lower(A)^2 \cdot n^2)$, that is, $O(n^4)$, since $lower(A) = O(n)$.

4.3. Park and Chin's Algorithm

Park and Chin [15] developed an extension of Xu's algorithm for finding the optimal decomposition of simply connected SE's, where all elements in the decomposition are also simply connected. In this subsection, we show that there exist infinite families of simply connected SE's that have shape decompositions

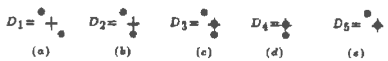

Fig. 16. (a) – (e) SE's D_1, D_2, D_3, D_4 and D_5.

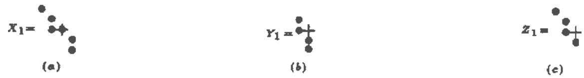

Fig. 17. (a) – (c) SE's X_1, Y_1 and Z_1.

but are not decomposable according to the Park and Chin's decomposability definition. In addition, we give some comments about the time complexity of their algorithm.

In this subsection, we consider the SE's $D_1 = \{d_1, d_2\}$, $D_2 = \{d_1, d_4\}$, $D_3 = \{d_1, d_3, d_4\}$, $D_4 = \{d_3, d_4\}$ and $D_5 = \{d_1, d_3\}$, where $d_1 = (-1, 1)$, $d_2 = (1, -1)$, $d_3 = (0, 0)$ and $d_4 = (0, -1)$. These SE's are presented in Figure 16.

For any integer $i > 0$, consider the SE's $X_i = iD_1 \oplus D_3$, $Y_i = iD_2 \oplus D_4$ and $Z_i = iD_2 \oplus D_5$. See Figure 17 for some examples of these SE's. These SE's X_i, Y_i and Z_i are simply connected and at least one element in the decomposition of X_i, Y_i and Z_i is not simply connected [9]. So, the families of simply connected SE's $\mathcal{X} = \{X_i : i > 0\}$, $\mathcal{Y} = \{Y_i : i > 0\}$ and $\mathcal{Z} = \{Z_i : i > 0\}$ are not decomposable according to the Park and Chin's decomposability definition [9].

In their work, Park and Chin [15] did not mention the time complexity of their algorithm. In a certain step of the Park and Chin's algorithm, it is necessary to find an integer solution of a linear system with a fixed number of variables [15, p. 8], but they did not show how to do it. Theoretically, for each fixed natural number n, there is a polynomial algorithm solving systems of linear inequalities in n integer variables [19, p. 256], but its implementation is not practical. In general case, integer linear systems are very hard problems [19, p. 227].

Although the time complexity of our algorithm is $O(2^m)$, where m is the number of elements in the invariant sequence, our algorithm has an advantage over Park and Chin's algorithm, since it can decompose any type of decomposable SE, including the SE's in families $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{Z}$.

5. Experimental Results

In this subsection, we present some experimental results of application of our algorithm for finding an optimum decomposition for some different types of SE's, namely: digital disks (see definition below), convex SE's (see an example in Figure 18*a*), decomposable connected SE's that contain holes (see an example in Figure 18*b*), decomposable connected SE's that contain no holes (see an example in Figure 18*c*), decomposable disconnected SE's that contain holes (see an example in Figure 18*d*) and decomposable disconnected SE's that contain no holes (see an example in Figure 18*e*). These experiments have been performed using a Sun Ultra Enterprise 3000. Processing time is measured in hours (h), minutes (m) and seconds (s).

The *digital disk* of radius $r > 0$, centered at the origin, is the SE given by $D(r) = \{(x, y) \in \mathbb{Z}^2 : x^2 + y^2 \leq r^2\}$ (see Figure 19 for some examples). Note that it is not the only definition to discrete circular SE's. A method to obtain some types of decomposable discrete circular SE's and their decomposition can be found in [21].

We divide this subsection into three parts. In the first one, we show some results for digital disks; in the second one, for convex SE's; in the third one, for decomposable connected and disconnected SE's that contain holes and no holes. In the tables, we use the following notation:

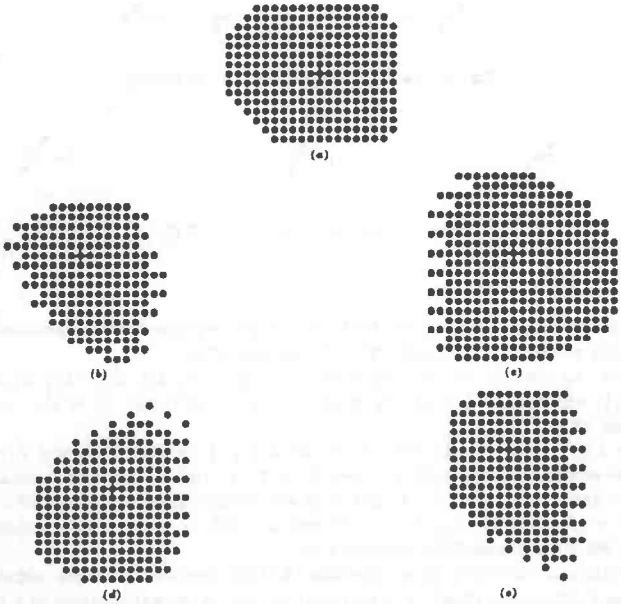

Fig. 18. (*a*) A convex SE. (*b*) A decomposable connected SE that contains a hole. (*c*) A decomposable connected SE that contains no holes. (*d*) A decomposable disconnected SE that contains some holes. (*e*) A decomposable disconnected SE that contains no holes.

NSI = number of subsets in the invariant sequence.
NFS = number of subsets in the first solution.
NOS = number of subsets in the optimum solution.
TND = time taken for detecting the non-decomposability.
TFS = time taken for detecting the first solution.
TOS = time taken for detecting the optimum solution.

5.1. *Digital Disks*

The disks $D(2)$ and $D(4)$ are decomposable, while disks $D(3)$ and $D(5)$ to $D(50)$ have no decomposition. The time taken for detecting the non-decomposability of the disks from $D(5)$ to $D(50)$ was less than 40 seconds. The time taken for detecting the first solution (that was the optimum one) of the disks $D(2)$ and $D(4)$ are presented in Table 1. Table 2 shows the time taken for detecting the non-decomposability of some disks of radius between 5 and 50.

5.2. *Convex SE's*

We have applied the procedure to find the optimum decomposition for about 250 convex SE's. As stated in Proposition 28, all the first solutions found were the optimum one. Table 3 presents the average time

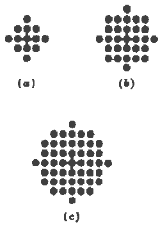

Fig. 19. (*a*) – (*c*) Disks of radius 2, 3 and 4.

Table 1. Decomposable Disks.

radius	NSI	NFS=NOS	TFS=TOS
2	6	2	0.5s
4	16	4	0.7s

for detecting the optimum solution for convex SE's that are subsets of 20×20, 40×40, 56×56, 72×72 and 88×88 square. We have observed that the time taken for finding the optimum decomposition of convex SE's is very small, even for large SE's. Table 4 presents the time taken for detecting the optimum solution of some convex SE's.

5.3. Decomposable Connected and Disconnected SE's that contain holes and no holes

In this subsection, we use the following notation for denoting the SE's:

DCH = decomposable connected SE's that contain holes.

DC = decomposable connected SE's that contain no holes.

DDH = decomposable disconnected SE's that contain holes.

DD = decomposable disconnected SE's that contain no holes.

We have applied the procedure to find the first and optimum shape decomposition for about, respectively, 400 and 200 decomposable connected and disconnected SE's that contain holes and no holes.

Table 5 presents the average time for detecting the first solution of SE's that are subsets of 20×20 and 40×40 square. In most cases, the first solutions were the optimum ones. In addition, the distance between the lower bound and the number of elements in the first solution was at most two. Therefore, the first solutions was very close to the optimum ones.

In Table 6, we present the time taken for detecting the first and the optimum solutions of some SE's that are subsets of the 20×20 square and the lower bound was not equal to the number of elements in the first solution. In this table, observe that the distance beetween the lower bound and the first solution is usually small and, in most cases, the first solution is the optimum one. Since the complexity time for detecting the optimum solution, in the worst case, is exponential, in practical applications, it may be a good heuristic to stop when the first solution is found.

Despite the good results presented in Tables 5 and 6, the time taken for finding the first solution has increased exponentially with the size of the SE. This was because the three prunings strategies used in

Table 2. Undecomposable Disks.

radius	NSI	TND
05	4	0.7s
10	22	1.0s
15	38	2.2s
20	54	11.1s
25	70	36.3s
30	70	2.5s
35	54	4.4s
40	70	4.9s
45	86	6.3s
50	102	7.7s

Table 3. Average time for detecting the optimum solution of convex SE's.

Subset of the square	TFS=TOS
20 × 20	5.5s
40 × 40	19.4s
56 × 56	47.2s
72 × 72	1m42.0s
88 × 88	3m21.0s

the algorithm were unable to avoid many unfeasible nodes in the decomposition tree. Table 7 presents the time taken for detecting the first solution of some SE's that are bigger than the SE's presented in Table 6.

Table 4. Time for detecting the optimum solution of some convex SE's.

SE A	$\rho(A)$	NSI	NFS=NOS	TFS=TOS
01	(17, 17)	448	9	5.5s
02	(17, 19)	531	10	7.7s
03	(31, 34)	844	17	23.3s
04	(32, 36)	918	18	22.0s
05	(48, 48)	1637	24	50.0s
06	(52, 55)	1619	28	57.0s
07	(65, 68)	2114	34	2m7.0s
08	(69, 68)	2217	35	2m30.0s
09	(85, 82)	2586	43	3m55.0s
10	(88, 84)	2802	44	4m32.0s

Table 5. Average time for detecting the first solution decomposable connected and disconnected SE's that contain holes and no holes.

	20 × 20	40 × 40
DCH	3.266s	13m41.0s
DC	2.876s	22.2s
DDH	1.682s	34.0s
DD	1.506s	18m27.0s

Table 6. Time for detecting the first and optimum solution of some decomposable connected and disconnected SE's that contain holes and no holes.

SE A	$\rho(A)$	NSI	lower bound	NFS	TFS	NOS	TOS
DDH	(18,16)	17	9	10	1.2s	10	3.0s
	(16,16)	17	8	9	1.0s	9	4.1s
	(18,18)	38	9	10	1.8s	10	29.4s
	(18,15)	21	9	10	1.1s	10	6.0s
	(14,18)	10	9	10	1.0s	10	1.3s
DD	(16,16)	19	8	9	1.1s	9	7.5s
	(18,17)	29	9	10	1.2s	10	43.5s
	(18,14)	36	9	10	1.5s	10	1m54.0s
	(18,16)	20	9	10	1.3s	10	7.7s
DCH	(18,16)	24	9	10	1.2s	10	11.6s
	(16,16)	41	8	9	3.5s	9	2m39.0s
	(18,18)	30	9	10	1.2s	10	57.2s
	(16,18)	19	9	10	1.0s	10	2.5s
	(16,18)	30	9	10	1.8s	10	18.0s
DC	(16,17)	60	9	10	4.1s	9	47.0s
	(14,18)	43	9	10	5.3s	10	2m41.0s
	(16,15)	31	8	9	1.6s	9	27.0s
	(18,18)	41	9	10	1.6s	10	8m20.3s

Table 7. Time for detecting the first solution of some decomposable connected and disconnected SE's that contain holes and no holes.

SE A	$\rho(A)$	NSI	TFS
DDH	(62,75)	114	67h38m56.0s
DD	(52,56)	91	5h25m30.0s
DCH	(50,54)	137	23h24m21.0s
DC	(50,58)	521	at least 201h

6. Conclusion

The change of decomposition structure of morphological operators for improving the performance of their implementation is a fundamental step in the process of automatic programming of Morphological Machines [2]. In this paper, we studied a particular aspect of this problem: the sequential decomposition of erosions (respectively, dilations).

A general algorithm for the automatic proof that an erosion (respectively, dilation) has a sequential decomposition or not was presented. The proof of existence is constructive and an optimum solution is exhibited. This algorithm is based on a branch and bound search, with prunning strategies and bounds based on algebraic and geometrical properties deduced formally.

The proposed algorithm is not efficient for all the cases, but is generalizes important classical results as Zhuang and Haralick, Xu, and Park and Chin, with equivalent or improved performances. Theoretical analysis and experimental results illustrated these facts. The combinatorial algorithm is open to improvements if new bounds or prunings are discovered.

The same kind of combinatorial algorithm could be applied, for example, to compute the sequential decomposition of alternate sequential filters from their basis. This should be the next problem to be examined in our research.

References

1. G. Anelli, A. Broggi, and G. Destri. Decomposition of Arbitrarily Shaped Binary Morphological Structuring Elements Using Genetic Algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(2):217–224, February 1998.
2. J. Barrera, E. R. Dougherty, and N. S. Tomita. Automatic Programming of Binary Morphological Machines by Design of Statistically Optimal Operators in the Context of Computational Learning Theory. *Journal of Electronic Imaging*, 6(1):54–67, January 1997.
3. J. Barrera, C. E. Ferreira, and R. F. Hashimoto. Finding Optimal Sequential Decompositions of Erosions and Dilations. In H. J. A. M. Heijmans and J. B. T. M. Roerdink, editors, *Mathematical Morphology and its Applications to Image and Signal Processing*, pages 299–306, Amsterdam, The Netherlands, June 1998. International Symposium on Mathematical Morphology, Kluwer Academic Publishers.
4. J. Barrera and R. F. Hashimoto. Compact Representation of W-Operators. In *10th Annual Symposium on Electronic Imaging*, pages 84–94, San Jose, California, January 1998. SPIE Press.
5. J. Barrera and G. P. Salas. Set Operations on Collections of Closed Intervals and their Applications to the Automatic Programming of Morphological Machines. *Journal of Eletronic Imaging*, 5(3):335–352, July 1996.
6. H. Freeman. Computer Processing of Line-Drawing Images. *Computer Surveys*, 6(1):57–97, March 1974.
7. R. C. Gonzalez and R. E. Woods. *Digital Image Processing*. Addison-Wesley Publishing Company, 1992.
8. R. F. Hashimoto. An Extension of an Algorithm for Finding Sequential Decomposition of Erosions and Dilations. In *SIBGRAPI'98 - XI Brazilian Symposium of Computer Graphic and Image Processing*, pages 443–449. IEEE Computer Society, October 1998.
9. R. F. Hashimoto and J. Barrera. A Comment on Park and Chin's Algorithm. Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence.
10. R. F. Hashimoto and J. Barrera. A Greedy Algorithm for Decomposing Convex Structuring Elements. Submitted to IEEE Transactions on Pattern Analysis and Machine Intelligence.
11. H. J. A. M. Heijmans. *Morphological Image Operators*. Academic Press, 1994.

12. T. Kanungo and R. Haralick. Vector-Space Solution for a Morphological Shape-Decomposition Problem. *Journal of Mathematical Imaging and Vision*, 2:51–82, 1992.
13. P. A. Maragos. *A Unified Theory of Translation-invariant Systems with Applications to Morphological Analysis and Coding of Images.* PhD thesis, School of Elect. Eng. - Georgia Inst. Tech., 1985.
14. H. Park and R. T. Chin. Optimal Decomposition of Convex Morphological Structuring Elements for 4-Connected Parallel Processors. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(3):304–313, March 1994.
15. H. Park and R. T. Chin. Decomposition of Arbitrarily Shaped Morphological Structuring Elements. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(1):2–15, January 1995.
16. T. Pavlidis. *Algorithms for Graphics and Image Processing.* Computer Science Press, Rockville, Maryland, 1982.
17. E. Rich and K. Knight. *Artificial Intelligence.* McGraw-Hill, USA, second edition, 1991.
18. C. H. Richardson and R. Schafer. A Lower Bound for Structuring Element Decompositions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):365–369, April 1991.
19. A. Schrijver. *Theory of Linear and Integer Programming.* Wiley-Interscience series in discrete mathematics. Jon Wiley & Sons, 1986.
20. J. Serra. *Image Analysis and Mathematical Morphology.* Academic Press, 1982.
21. M. Vanrell and J. Vitrià. Optimal 3 × 3 Decomposable Disks for Morphological Transformations. *Image and Vision Computing*, 15:845–854, 1997.
22. J. Xu. Decomposition of Convex Polygonal Morphological Structuring Elements into Neighborhood Subsets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(2):153–162, February 1991.
23. H. Yang and S. Lee. Optimal Decomposition of Morphological Structuring Elements. In *Proceedings of IEEE International Conference on Image Processing*, volume 3, pages 1–4, Lausanne-Switzerland, 1996. IEEE Computer Society.
24. X. Zhuang and R. Haralick. Morphological Structuring Element Decomposition. *Computer Vision, Graphics, and Image Processing*, 35:370–382, 1986.

RELATÓRIOS TÉCNICOS

DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Instituto de Matemática e Estatística da USP

A listagem contendo os relatórios técnicos anteriores a 1996 poderá ser consultada ou solicitada à Secretaria do Departamento, pessoalmente, por carta ou e-mail(mac@ime.usp.br).

Daniela V. Carbogim and Flávio S. Corrêa da Silva
FACTS, ANNOTATIONS, ARGUMENTS AND REASONING
RT-MAC-9601, janeiro de 1996, 22 pp.

Kunio Okuda
REDUÇÃO DE DEPENDÊNCIA PARCIAL E REDUÇÃO DE DEPENDÊNCIA GENERALIZADA
RT-MAC-9602, fevereiro de 1996, 20 pp.

Junior Barrera, Edward R. Dougherty and Nina Sumiko Tomita
AUTOMATIC PROGRAMMING OF BINARY MORPHOLOGICAL MACHINES BY DESIGN OF STATISTICALLY OPTIMAL OPERATORS IN THE CONTEXT OF COMPUTATIONAL LEARNING THEORY.
RT-MAC-9603, abril de 1996, 48 pp.

Junior Barrera e Guillermo Pablo Salas
SET OPERATIONS ON CLOSED INTERVALS AND THEIR APPLICATIONS TO THE AUTOMATIC PROGRAMMINIG OF MMACH'S
RT-MAC-9604, abril de 1995, 66 pp.

Kunio Okuda
CYCLE SHRINKING BY DEPENDENCE REDUCTION
RT-MAC-9605, maio de 1996, 25 pp.

Julio Stern, Fabio Nakano e Marcelo Lauretto
REAL: REAL ATTRIBUTE LEARNING FOR STRATEGIC MARKET OPERATION
RT-MAC-9606, agosto de 1996, 16 pp.

Markus Endler
SISTEMAS OPERACIONAIS DISTRIBUÍDOS: CONCEITOS, EXEMPLOS E TENDÊNCIAS
RT-MAC-9607, agosto de 1996, 120 pp.

Hae Yong Kim
CONSTRUÇÃO RÁPIDA E AUTOMÁTICA DE OPERADORES MORFOLÓGICOS E EFICIENTES PELA APRENDIZAGEM COMPUTACIONAL
RT-MAC-9608, outubro de 1996, 19 pp.

Marcelo Finger
NOTES ON COMPLEX COMBINATORS AND STRUCTURALLY-FREE THEOREM PROVING
RT-MAC-9609, dezembro 1996, 28 pp.

Carlos Eduardo Ferreira, Flávio Keidi Miyazawa e Yoshiko Wakabayashi (eds)
ANAIS DA I OFICINA NACIONAL EM PROBLEMAS DE CORTE E EMPACOTAMENTO
RT-MAC-9610, dezembro de 1996, 65 pp.

Carlos Eduardo Ferreira, C. C. de Souza e Yoshiko Wakabayashi
REARRANGEMENT OF DNA FRAGMENTS: A BRANCH-AND-CUT ALGORITHM
RT-MAC-9701, janeiro de 1997, 24 pp.

Marcelo Finger
NOTES ON THE LOGICAL RECONSTRUCTION OF TEMPORAL DATABASES
RT-MAC-9702, março de 1997, 36 pp.

Flávio S. Corrêa da Silva, Wamberto W. Vasconcelos e David Robertson
COOPERATION BETWEEN KNOWLEDGE BASED SYSTEMS
RT-MAC-9703, abril de 1997, 18 pp.

Junior Barrera, Gerald Jean Francis Banon, Roberto de Alencar Lotufo, Roberto Hirata Junior
MMACH: A MATHEMATICAL MORPHOLOGY TOOLBOX FOR THE KHOROS SYSTEM
RT-MAC-9704, maio de 1997, 67 pp.

Julio Michael Stern e Cibele Dunder
PORTFÓLIOS EFICIENTES INCLUINDO OPÇÕES
RT-MAC-9705, maio de 1997, 29 pp.

Junior Barrera e Ronaldo Fumio Hashimoto
COMPACT REPRESENTATION OF W-OPERATORS
RT-MAC-9706, julho de 1997, 13 pp.

Dilma M. Silva e Markus Endler
CONFIGURAÇÃO DINÂMICA DE SISTEMAS
RT-MAC-9707, agosto de 1997, 35 pp

Kenji Koyama e Routo Terada
AN AUGMENTED FAMILY OF CRYPTOGRAPHIC PARITY CIRCUITS
RT-MAC-9708, setembro de 1997, 15 pp

Routo Terada e Jorge Nakahara Jr.
LINEAR AND DIFFERENTIAL CRYPTANALYSIS OF FEAL-N WITH SWAPPING
RT-MAC-9709, setembro de 1997, 16 pp

Flávio S. Corrêa da Silva e Yara M. Michelacci
MAKING OF AN INTELLIGENT TUTORING SYSTEM (OR METHODOLOGICAL ISSUES OF ARTIFICIAL INTELLIGENCE RESEARCH BY EXAMPLE)
RT-MAC-9710, outubro de 1997, 16 pp.

Marcelo Finger
COMPUTING LIST COMBINATOR SOLUTIONS FOR STRUCTURAL EQUATIONS
RT-MAC-9711, outubro de 1997, 22 pp.

Maria Angela Gurgel and E.M.Rodrigues
THE F-FACTOR PROBLEM
RT-MAC-9712, dezembro de 1997, 22 pp.

Perry R. James, Markus Endler, Marie-Claude Gaudel
DEVELOPMENT OF AN ATOMIC-BROADCAST PROTOCOL USING LOTOS
RT-MAC-9713, dezembro de 1997, 27 pp.

Carlos Eduardo Ferreira and Marko Loparic
A BRANCH-AND-CUT ALGORITHM FOR A VEHICLE ROUTING PROBLEM WITH CAPACITY AND TIME CONSTRAINTS
RT-MAC-9714, dezembro de 1997, 20 pp.

Nami Kobayashi
A HIERARCHY FOR THE RECOGNIZABLE M-SUBSETS
RT-MAC-9715, dezembro de 1997, 47 pp.

Flávio Soares Corrêa da Silva e Daniela Vasconcelos Carbogim
A TWO-SORTED INTERPRETATION FOR ANNOTATED LOGIC
RT-MAC-9801, fevereiro de 1998, 17 pp.

Flávio Soares Corrêa da Silva, Wamberto Weber Vasconcelos, Jaume Agustí, David Robertson e Ana Cristina V. de Melo.
WHY ONTOLOGIES ARE NOT ENOUGH FOR KNOWLEDGE SHARING
RT-MAC-9802, outubro de 1998, 15 pp.

J. C.de Pina e J. Soares
ON THE INTEGER CONE OF THE BASES OF A MATROID
RT-MAC-9803, novembro de 1998, 16 pp.

K. Okuda and S.W.Song
REVISITING HAMILTONIAM DECOMPOSITION OF THE HYPERCUBE
RT-MAC-9804, dezembro 1998, 17pp.

Markus Endler
AGENTES MÓVEIS: UM TUTORIAL
RT-MAC-9805, dezembro 1998, 19pp.

Carlos Alberto de Bragança Pereira, Fabio Nakano e Julio Michael Stern
A DYNAMIC SOFTWARE CERTIFICATION AN VERIFICATION PROCEDURE
RT-MAC-9901, março 1999, 21pp.

Carlos E. Ferreira e Dilma M. Silva
BCC DA USP: UM NOVO CURSO PARA OS DESAFIOS DO NOVO MILÊNIO
RT-MAC-9902, abril 1999, 12pp.

Ronaldo Fumio Hashimoto and Junior Barrera
A SIMPLE ALGORITHM FOR DECOMPOSING CONVEX STRUCTURING ELEMENTS
RT-MAC-9903, abril 1999, 24 pp.

Jorge Euler, Maria do Carmo Noronha e Dilma Menezes da Silva
ESTUDO DE CASO: DESEMPENHO DEFICIENTE DO SISTEMA OPERACIONAL LINUX PARA CARGA MISTA DE APLICAÇÕES.
RT-MAC-9904, maio 1999, 27 pp.

Carlos Humes Junior e Paulo José da Silva e Silva
AN INEXACT CLASSICAL PROXIMAL POINT ALGORITHM VIEWED AS DESCENT METHOD IN THE OPTIMIZATION CASE
RT-MAC-9905, maio 1999, pp.

Carlos Humes Junior
STRICT CONVEX REGULARIZATIONS, PROXIMAL POINTS AND AUGMENTED LAGRANGIANS
RT-MAC-9906, maio 1999, pp.

Ronaldo Fumio Hashimoto, Junior Barrera, Carlos Eduardo Ferreira
A COMBINATORIAL OPTIMIZATION TECHNIQUE FOR THE SEQUENTIAL DECOMPOSITION OF EROSIONS AND DILATIONS
RT-MAC-9907, maio 1999, 30 pp.