Yugoslav Journal of Operations Research
5 (1995), Number 1, 39-51

ON A CONSTRUCTION OF DIGITAL CONVEX (2S+1)-GONS OF MINIMUM DIAMETER

Snežana MATIĆ-KEKIĆ, Dragan M. ACKETA

Institute of Mathematics, University of Novi Sad
Trg Dositeja Obradovića 4, 21000 Novi Sad, Yugoslavia

Abstract: In this paper an algorithm is described for an exact construction of digital convex $(2s+1)$-gons of minimum diameter. A complete family of auxiliary so-called perfect Basic b-tuples is obtained by applying this algorithm. The required optimal $(2s+1)$-gons can be easily constructed from this family.

Keywords: Digital geometry, digital convex polygon, greedy lower bound.

1. INTRODUCTION

A *digital convex polygon* (shortly d.c.p.) is a polygon whose all vertices are points of the integer grid and all the interior angles of which are strictly smaller than π radians. The *diameter* of a d.c.p. is the minimal edge size of the enscribed digital square with edges parallel to the coordinate axes.

The following optimization problem is considered:

Given an odd natural number $2s + 1$, determine a d.c. $(2s + 1)$-gon of minimum diameter $mind(2s + 1)$.

The analogous problem for $2s$-gons was completely solved in [4]. A construction of almost optimal d.c. $(2s + 1)$-gons was given in [5]; these $(2s + 1)$-gons are almost optimal in the sense that their diameters are not greater than $1 + mind(2s + 1)$.

In this paper the last step is made for completion of these results: an algorithm is given, the results of which are used for an *exact* construction of optimal digital convex $(2s + 1)$-gons.

2. PRELIMINARIES

Let y_{min} and x_{max} respectively denote the minimal y–coordinate and the maximal x–coordinate of the considered d.c.p. P. Generally, the SE–arc (*south–east arc*) of P is the sequence of consecutive edges (V_i, V_{i+1}), $1 \le i \le k-1$, where V_i denotes a vertex (x_i, y_i) of P; $x_1 < ... < x_k = x_{max}$; $y_{min} = y_1 < ... < y_k$. In particular, if the polygon P has a *lower horizontal edge* (V_0, V_1) $(V_0 = (x_0, y_1),\ V_1 = (x_1, y_1),\ x_0 < x_1)$, then this edge is additionally considered to be the first edge of the SE–arc. The NE–arc, the NW–arc and the SW–arc of a d.c.p. are defined in the analogous way.

Given an edge $e = ((x_1, y_1), (x_2, y_2))$ of a d.c.p., the *edge slope* of e denotes the fraction:

$$\frac{|x_1 - x_2|}{|y_1 - y_2|} \ \text{ if } \ e \in \text{NE - or SW - arc}\,; \qquad \frac{|y_1 - y_2|}{|x_1 - x_2|} \ \text{ if } \ e \in \text{SE - or NW - arc}\,,$$

while *bd*–length (shortly: *bdl*) of the edge e denotes the sum $|x_1 - x_2| + |y_1 - y_2|$.

$DS(p, q)$ denotes a digital square with the property that each arc has exactly one edge with the edge–slope q/p, where p and q are relatively prime natural numbers.

If the corresponding arcs of some two d.c. polygons P_1 and P_2 have no common edge slopes, then there exists the *Minkowski sum* of P_1 and P_2, which is a uniquely determined third d.c.p. P_3 (for more details see, e.g., [6]). Each arc of P_3 includes all the edges of the corresponding arcs of P_1 and P_2, sorted so that the convexity condition is preserved. The diameter of P_3 is equal to the sum of the diameters of P_1 and P_2.

$MS(P)$ denotes the minimal (digital) square (with edges parallel to the coordinate axes) in which a d.c.p. P can be inscribed.

A "projection of an edge" of a d.c.p. P is a projection of that edge to an edge of $MS(P)$ which is not "hidden" by P (thus each "oblique" edge of P has exactly two projections).

2.1. A BOUND, A CONSTRUCTION AND TOLERANCES

A theoretical lower bound for diameter of a d.c. n–gon can be derived from the following observations:

Let $Minsum(n)$ denote the minimal possible sum of *bd*–lengths of n digital edges which might be included into a d.c.p. P. We are going to make the notion of $Minsum(n)$ more precise:

Since the number of summands is fixed, the minimization requires the summands to be as small as possible. Such a choice of summands is naturally performed by the following "greedy" algorithm: choose as many summands equal to 1 as possible, then proceed with summands equal to 2 and so on. All these summands are of the form $(p + q)$, where q/p $(q = 0, 1, ..., p = 1, 2, ...)$ is an edge slope. The following two rules must be obeyed by the edge slopes q/p: the numbers p and q are relatively prime; each q/p can be used at most four times (at most once in each one of the four arcs of P) – that is, it has at most four associated summands $(p + q)$ in $Minsum(n)$.

A family $\{ P(t) \mid t = 1, 2, \dots \}$ of optimal d.c. $4s$-gons was introduced in [7] (see also [8]). Each arc of the polygon $P(t)$ contains all the possible edge slopes q/p satisfying $p + q \leq t$. The number of vertices and the diameter of the polygon $P(t)$ are denoted by $v(t)$ and $d(t)$ respectively.

One can derive ([1]) that the functions $v(t)$ and $d(t)$ can be expressed in terms of the Euler function ϕ ($\phi(i)$ denotes the number of integers between 1 and i which are relatively prime with i; e.g., $\phi(1) = 1$, $\phi(3) = \phi(4) = 2$, $\phi(5) = 4$) as follows:

$$v(t) = 4 \cdot \sum_{i=1}^{t} \phi(i) \qquad d(t) = \sum_{i=1}^{t} i \cdot \phi(i)$$

Let $n \in (v(t-1), v(t))$.

The diameter of a d.c. n-gon P cannot be smaller than one fourth of the perimeter of $MS(P)$. On the other hand, $Minsum(n)$ is a lower bound for this perimeter. Consequently, a *greedy lower bound* $gdlb(n)$ for diameter of a d.c. n-gon can be expressed as:

$$gdlb(n) = \left\lceil \frac{Minsum(n)}{4} \right\rceil = d(t-1) + \left\lceil \frac{(n - v(t-1)) \cdot t}{4} \right\rceil$$

A d.c. n-gon for n odd is called *perfect* if its diameter is equal to $1 + gdlb(n)$ for $t \bmod 4 = 0$ and $gdlb(n)$ otherwise. Namely, it was shown in [5] that there are no d.c. n-gons with n odd, $t \bmod 4 = 0$, and diameter equal to $gdlb(n)$.

Our construction of perfect d.c. n-gons is based on the key concept of perfect Basic b-tuples.

A *Basic b-tuple B* is defined as a collection of b edges partitioned w.r.t. the arcs which satisfies that each edge slope of B is used in at most three arcs. Note that B can be used as a summand of a Minkowski sum and that $MS(B)$ is well-defined. *Initial* $4s$-gons associated to B are the Minkowski sums of s arbitrary different 4-gons of the form $DS(p, q)$, which satisfy the following conditions: $p + q \leq t$; the edge slope q/p is not used in B; all the edge slopes q'/p' which are not used in B and which satisfy $q' + p' < q + p$ – are used in the corresponding Initial $4s$-gon.

A Basic b-tuple B is called *perfect* if it can be used for the construction of a perfect d.c. n-gon. The construction of perfect Basic b-tuples is the goal of the algorithm described in Section 3.

Let k_i, for $i = 1, 2, 3, 4$ denote the difference between the diameter of a Basic b-tuple B and the sum of projections of edges of B onto the north, west, south and east edge of $MS(B)$ respectively.

A perfect d.c. n-gon P is constructed from a perfect Basic b-tuple B and a corresponding Initial $4s$-gon I ($b + 4s = n$) by applying in turn the following two steps:

1. Construction of the Minkowski sum T of B and I.
2. Replacement of edges of T with edge slope $0/1$ ("flat" edges) in the i-th arc ($i = 1, 2, 3, 4$ for NW-, SW-, SE- and NE-arc respectively) by edges with edge slopes $0/(k_i + 1)$.

Let a perfect Basic b-tuple B be devoted to the constructions of perfect n-gons satisfying $(n-b) \bmod 4 = 0$ and $n \in (v(t-1), v(t))$. We say that B leaves a *gap* if B cannot be used for constructions of perfect d.c. n-gons with some of the considered values of n.

Let a Basic b-tuple B be used for the construction of a d.c. n-gon P. The *used tolerance* (shortly: *UT*) of B is equal to the difference of the sum of *bd*-lengths of edges of P and $Minsum(n)$.

Assume now that both B and P are perfect. Then the *allowed tolerance* (shortly: *AT*) of B is equal to the difference of the perimeter of $MS(P)$ and $Minsum(n)$. It is obvious that $AT \geq UT$ and that $AT - UT = k_1 + k_2 + k_3 + k_4$.

It turns out that AT depends merely on $n' = n \bmod 4$ and $t' = t \bmod 4$; its values are given by Table 1:

Table 1.

(n', t')	(1, 0)	(1, 1)	(1, 2)	(1, 3)	(3, 0)	(3, 1)	(3, 2)	(3, 3)
AT	4	3	2	1	4	1	2	3

3. ALGORITHM

There are four (hierarchically nested) levels of search for perfect Basic b-tuples: Cases determined by combinations of used *bd*-lengths, *bd*-lengths used within a Case, edge slopes of a given *bd*-length and arcs in which a given edge slope is used. For example, if Case and *bd*-length are fixed, then, when looking for an edge slope of that *bd*-length, all the possibilities are tried, and for each one of them all the possibilities for the arcs are examined. The preparatory stages of the algorithm include Case level – generation of all the Cases (Section 3.1), as well as the preparation of the edge slope level – generation of all the edge slopes, which might be used in a Case (Section 3.2).

3.1. LIST OF CASES

When looking for perfect Basic b-tuples, we make the complete *List_of_Cases* for a choice of *bd*-lengths of their edges. The diameter and the used tolerance of these Basic b-tuples are determined by the Case and denoted as *Diameter*(*Case*) and *UT*(*Case*) respectively.

List_of_Cases is determined by hand, depending on $n \bmod 4$ in $\{1, 3\}$. The partition into Cases is an application of the divide-and-conquer approach to the search: a huge amount of unusable combinations is eliminated.

Each Case requires a fixed number of edges with a given *bdl*. Moreover, the number of edge slopes with that *bdl* is sometimes (see mode 1 in Section 3.4) also fixed, as well as the number of edges (arcs) with the corresponding edge slope.

As an example, we give *List_of_Cases* (Table 2) for choice of *bd*–lengths of edges of perfect Basic $(4s + 1)$–tuples, which can be used for $n = v(t-1) + 1$. Each Case is written in the form of sum of the used *bd*–lengths. A summand of the form $q * p[t-i]$ means that the corresponding perfect Basic b–tuple should use q distinct edge slopes with $bdl = t - i$, so that each one of these edge slopes is used in exactly p arcs. If either of the integers q and p is equal to 1, then it is omitted. A summand of the form qt or $q(t+i)$ means that the perfect Basic b–tuple should use q edge slopes with $bdl = t$, respectively with $bdl = t + i$ (these edge slopes, when used in distinct arcs, need not be distinct). Each sum in the list is followed by *UT*(*Case*). The additional denotation ! (w) means that a perfect Basic b–tuple with such a choice of *bd*–lengths has been effectively constructed (would leave no gaps).

Table 2.

$3[t-4] + 2t$	4	$2 * 3[t-1] + 2t + (t+1)$	3
$3[t-3] + 3[t-1] + 3t$	4	$2 * 3[t-1] + t + 2(t+1)$	4 w
$3[t-3] + 2t$	3	$2 * 3[t-1] + 2t + (t+2)$	4
$3[t-3] + t + (t+1)$	4 w	$3[t-1] + 2[t-1] + 4t$	3 !
$2 * 3[t-2] + 3t$	4	$3[t-1] + 2[t-1] + 3t + (t+1)$	4
$3[t-2] + 2 * 3[t-1] + 4t$	4	$3[t-1] + [t-1] + 5t$	4
$3[t-2] + 2 * 3[t-1]$	4 w	$3[t-1] + [t-1] + t$	4
$3[t-2] + 3[t-1] + 3t$	3	$2 * 2[t-1] + 5t$	4
$3[t-2] + 3[t-1] + 2t + (t+1)$	4	$2 * 2[t-1] + t$	4
$3[t-2] + 2[t-1] + 4t$	4	$3[t-1] + 2t$	1
$3[t-2] + 2[t-1]$	4 w	$3[t-1] + t + (t+1)$	2 ! w
$3[t-2] + 2t$	2	$3[t-1] + 2(t+1)$	3 w
$3[t-2] + t + (t+1)$	3 w	$3[t-1] + t + (t+2)$	3 w
$3[t-2] + 2(t+1)$	4 w	$3[t-1] + (t+1) + (t+2)$	4 w
$3[t-2] + t + (t+2)$	4 w	$3[t-1] + t + (t+3)$	4 w
$2[t-2] + 3t$	4	$2[t-1] + 3t$	2
$4 * 3[t-1] + 5t$	4	$2[t-1] + 2t + (t+1)$	3 !
$4 * 3[t-1] + t$	4	$2[t-1] + t + 2(t+1)$	4 w
$3 * 3[t-1] + 4t$	3	$2[t-1] + 2t + (t+2)$	4
$3 * 3[t-1]$	3 w	$[t-1] + 4t$	3
$3 * 3[t-1] + 3t + (t+1)$	4	$[t-1] + 3t + (t+1)$	4
$2 * 3[t-1] + 3t$	2 !		

3.2. CANDIDATES FOR EDGE SLOPES

Given $n \in (v(t-1), v(t))$, a family $F(bdl)$ of candidates for edge slopes of a perfect Basic b–tuple is generated for each $bdl \in [\,t-AT, t+AT\,]$. Given a bdl of the form $4k + u$ ($k = 0, 1, \ldots$, $u = 0, 1, 2, 3$), the candidates in $F(bdl)$ are chosen to be of the bilinear form $(i \cdot k + j) / ((4-i) \cdot k + (u-j))$, so that the denominator and numerator are relatively prime. In particular, in the case $bdl = 4k + 2$ we distinguish the subcases $bdl = 8k + 2$ and $bdl = 8k + 6$. The definition of perfect d.c. n-gons motivates the partitioning w.r.t $bdl \bmod 4$.

Table 3 is obtained by quoting merely one of each two mutually reciprocal fractions of the family $F(bdl)$.

Note that for each odd bdl there exists a subfamily of $F(bdl)$ of the form $2^s / (bdl - 1 - (2^s - 1))$ for $s = 0, 1, \dots, [\log_2(bdl)]$, while for each bdl satisfying $bdl \bmod 4 = 2$ there exists a subfamily of the form $((bdl/2) - 1 - (2^s - 1)) / ((bdl/2) - 1 + (2^s + 1))$, for $s = 1, 2, \dots, [\log_2(bdl/2)]$.

Table 3.

bdl	candidates
$4k$	$\frac{1}{4k+1}, \frac{2k-1}{2k+1}$
$4k+1$	$\frac{k}{3k+1}, \frac{2k}{2k+1}, \frac{2^s}{4k-(2^s-1)}, \quad s = 0, 1, \dots, \lfloor \log_2(4k+1) \rfloor$
$4k+2$	$\frac{1}{4k+1}, \frac{2k-(2^s-1)}{2k+(2^s+1)}, \quad s = 1, 2, \dots, \lfloor \log_2(2k+1) \rfloor$
$4k+3$	$\frac{k+1}{3k+2}, \frac{2k+1}{2k+2}, \frac{2^s}{4k-(2^s-3)}, \quad s = 0, 1, \dots, \lfloor \log_2(4k+3) \rfloor$
$8k+2$	$\frac{1}{8k+1}, \frac{2k+1}{6k+1}, \frac{4k-(2^s-1)}{4k+(2^s+1)}, \quad s = 1, 2, \dots, \lfloor \log_2(4k+1) \rfloor$
$8k+6$	$\frac{1}{8k+5}, \frac{2k+1}{6k+5}, \frac{4k-(2^s-3)}{4k+(2^s+3)}, \quad s = 1, 2, \dots, \lfloor \log_2(4k+3) \rfloor$

3.3. SKETCH OF THE ALGORITHM

The shell of the algorithm for the construction of a complete family of perfect Basic b-tuples, which can be used for construction of perfect digital convex n-gons for each n odd – has the following outlook in PseudoPascal:

```
BEGIN (* main *)
    Generate F(bdl), bdl ∈ { 4k – 4, ... ,4k + 4 } ∪ { 8k, ... ,8k + 8 };
        (* these bdl-s are sufficient for all the Cases *)
    FOR (n mod 4) in {1, 3} DO BEGIN
        Generate the List_of_Cases;
            (* for bd-lengths of edges of perfect Basic b-tuple *)
        FOR (t mod 4) in {0, 1, 2, 3} DO BEGIN
```

```
        Calculate AT(* Table 1 *);
        Determine interval [ t - AT, t + AT ] for bdl;
        Found := FALSE;
        REPEAT
          Take next Case from the List_of_Cases;
          IF UT(Case) ≤ AT THEN BEGIN
            Calculate Diameter(Case);
            Initialize Basic 0-tuple;
            Augment(No_slope, No_arc(s), 0) END
        UNTIL Found or (List_of_Cases is exhausted)
END (* for t *) END (* for n *) END. (* main *)
```

We also sketch the recursive procedure **Augment**, which searches for perfect Basic b-tuples by backtracking. This procedure incorporates the last three levels of the search; in particular, the *bdl* level is treated by **Jump**, while the WHILE loop searches through the edge slope level and the arc level at the same time.

Each call of **Augment** in the main program corresponds to an attempt (determined by Case) for construction of a perfect Basic b-tuple, while each successful recursive call inserts one or more edges with the same edge slope into the current Basic c-tuple ($c < b$).

```
PROCEDURE Augment(Last_slope, Last_arc(s), Last_diameter);
  BEGIN
    IF Completed THEN BEGIN
      Print perfect Basic b-tuple;
      Found := TRUE END
    ELSE BEGIN
      IF the current Basic c-tuple has the sufficient number
      of edges with bdl = bdl(Last_slope) THEN BEGIN (* Jump *)
        bdl := the next bd-length required by the Case;
        New_slope := the first candidate of F(bdl);
        New_arc(s) := the first possible;
        F(bdl)_exhausted := FALSE; END (* Jump *)
      ELSE BEGIN (* an attempt for regular advancing *)
        bdl := the bd-length of Last_slope;
        IF Last_arc(s) = last possible THEN
          IF Last_slope is the last candidate in F(bdl) THEN
            F(bdl)_exhausted := TRUE
          ELSE BEGIN
            New_slope := the next candidate in F(bdl);
            New_arc(s) := the first possible END
        ELSE New_arc(s) := the next possible END;
```

```
WHILE NOT F(bdl)_exhausted DO BEGIN
    Insert(New_slope, New_arc(s), New_diameter);
    IF Feasible(Augmented_tuple) THEN
        Augment(New_slope, New_arc(s), New_diameter);
    Delete(New_slope, New_arc(s), Last_diameter);
    IF New_slope is the last candidate in F(bdl) AND
    New_arc(s) = last possible THEN
        F(bdl)_exhausted := TRUE;
    ELSE IF New_arc(s) = last possible THEN BEGIN
        New_slope := the next candidate in F(bdl);
        New_arc(s) := the first possible  END
    ELSE New_arc(s) := the next possible
END(* while *)  END(* if not Completed *)  END(* Augment *);
```

3.4. SOME FURTHER EXPLANATIONS ON AUGMENT

We elaborate some details within the procedure Augment:

Completed is the Boolean variable which becomes true when a perfect Basic b-tuple (where b is the number required by Case) is constructed. In that moment the Boolean variable *Found* becomes true and breaks the REPEAT loop in the main program.

Feasible(*Augmented_tuple*) is the Boolean function which is true iff *Augmented_tuple* satisfies the following conditions:

- its diameter is not greater than *Diameter*(*Case*);
- no edge slope is used in more than three arcs;
- if the numbers of edges in NW-, SW-, SE- and NE-arc are denoted by nw, sw, se, ne respectively, then $nw \geq sw$, $nw \geq se$, $sw \geq ne$ (this condition corresponds to avoidance of symmetry and shortens the search within unsuccessful branches of the backtracking tree).

During the search for edges of a perfect Basic b-tuple, we distinguish two modes:

mode 1 ($bdl \leq t$): given an edge slope p/q with $p + q < t$, all the arcs of B in which that edge slope will be used (at most three of them) are chosen at once (the edge slope is treated as a whole);

mode 2 ($bdl \geq t$): each edge (= the ordered pair of an edge slope and an arc) is chosen independently of the others.

Note that both modes may be used with $bdl = t$ (this bdl is preferable, since tolerance is not used). During the backtracking, the modes can be alternatively used several times. Determination of *New_slope*, *New_arc*(s), as well as the performance of the procedures **Insert** and **Delete** – are mode-dependent.

If Case requires the edges with edge slope = *New_slope* to be inserted into j arcs ($j \in \{1,2,3\}$ in mode 1 and $j = 1$ in mode 2), then *New_arc(s)* is chosen as the lexicographically next combination of four arcs, without repetitions, of order j. The attributes "the first possible", "the next possible" and "the last possible" are in accordance with this lexicographical order.

The Boolean variable *F(bdl)_exhausted* becomes true if there are no new possibilities for *New_slope* and *New_arc(s)*, within the given *bdl*. Note that each pass through the WHILE loop corresponds to one *bdl*.

The procedure **Insert** effectively inserts the *edge(s)* with edge slope = *New_slope* into the *arc(s)* determined by the combination *New_arc(s)*. In this way the *Augmented_tuple* is produced and its diameter (called *New_diameter*) is determined. If the insertion implies that the *Augmented_tuple* is not **Feasible** or the recursive call of the procedure **Augment** terminates with failure, the reverse procedure **Delete** is activated; it returns the *Augmented_tuple* and its diameter into the previous state (before the insertion).

4. PERFECT BASIC b–TUPLES

In this section we present the main result of the paper, which is obtained by the algorithm of Section 3: a complete collection of perfect Basic b–tuples, for constructions of perfect d.c. n–gons for each odd n. The Basic b–tuples in the collection are partitioned w.r.t. ten cases, depending on $n \bmod 4$ and $t \bmod 4$; in particular, two cases are used for $t = 4k + 2 : t = 8k + 2$ and $t = 8k + 6$.

The data for each perfect Basic b–tuple B are listed. The first part of a list consists of the denotations of the form $\frac{q}{p}$ (*List_of_arcs*), where $\frac{q}{p}$ is an edge slope (written in the bilinear form) used in those arcs of B, which are mentioned in *List_of_arcs* (1, 2, 3, 4 denotes NW–, SW–, SE– and NE–arc respectively).

The second part of a list contains: the number b; the lower bounds for k and n, to which B is applicable; the diameter d of B, which is equal to $mind(n) - gdlb(n-b)$; the values g of gaps, which are left by B ($+i$ stands for the gap $v(t-1)+i$, while $-i$ stands for the gap $v(t)-i$).

Case 1. $n \bmod 4 = 1, \quad t = 4k$

$$\frac{3k-1}{k}(13) \quad \frac{2k-1}{2k+1}(14) \quad \frac{4k}{1}(2),$$

$$b=5, \quad k \geq 1, \quad n \geq 17, \quad d=5k$$

Case 2. $n \bmod 4 = 1, \quad t = 4k + 1$

$$\frac{3k+1}{k}(13) \quad \frac{2k+1}{2k}(23) \quad \frac{1}{4k+1}(1),$$

$b=5, \quad k \ge 1, \quad n \ge 29, \quad d=5k+2, \quad g=+1$

$$\frac{2k+1}{2k-1}(123) \quad \frac{1}{4k-1}(12) \quad \frac{k}{3k+1}(1) \quad \frac{4k}{1}(2) \quad \frac{3k+1}{k}(3) \quad \frac{2}{4k-1}(4),$$

$b=9, \quad k \ge 1, \quad n \ge 25, \quad d=9k+1, \quad g=-3,-7,-11$

Case 3. $n \bmod 4 = 1, \quad t = 8k + 2$

$$\frac{4k+1}{4k}(124) \quad \frac{6k+1}{2k+1}(2) \quad \frac{2k+1}{6k+2}(4),$$

$b=5, \quad k \ge 0, \quad n \ge 5, \quad d=10k+2$

Case 4. $n \bmod 4 = 1, \quad t = 8k + 6$

$$\frac{4k+5}{4k+1}(123) \quad \frac{4k+1}{4k+5}(4) \quad \frac{4k-1}{4k+7}(123) \quad \frac{6k+5}{2k+2}(3) \quad \frac{2k+2}{6k+5}(1),$$

$b=9, \quad k \ge 1, \quad n \ge 241, \quad d=18k+14, \quad g=+1,+5$

$$\frac{4k+2}{4k+3}(134) \quad \frac{8k+4}{1}(124) \quad \frac{6k+5}{2k+1}(1) \quad \frac{2k+1}{6k+5}(3) \quad \frac{1}{8k+5}(4),$$

$b=9, \quad k \ge 0, \quad n \ge 41, \quad d=18k+12, \quad g=-3,-7$

Case 5. $n \bmod 4 = 1, \quad t = 4k + 3$

$$\frac{3k+2}{k+1}(13) \quad \frac{2k+2}{2k+1}(23) \quad \frac{1}{4k+3}(1),$$

$b=5, \quad k \ge 1, \quad n \ge 51, \quad d=5k+4, \quad g=+1$

Case 6. $n \bmod 4 = 3, \quad t = 4k$

$$\frac{3k+1}{k}(1) \quad \frac{k}{3k+1}(3) \quad \frac{2k+1}{2k}(4),$$

$b=3, \quad k \ge 1, \quad n \ge 19, \quad d=3k+1$

Case 7. $n \bmod 4 = 3, \quad t = 4k + 1$

$$\frac{3k+1}{k}(1) \quad \frac{k}{3k+1}(3) \quad \frac{2k+1}{2k}(4),$$

$b=3, \quad k \ge 1, \quad n \ge 27, \quad d=3k+1, \quad g=-1,-5$

$$\frac{k}{3k+1}(24) \quad \frac{2k}{2k+1}(13) \quad \frac{2k+1}{2k}(23) \quad \frac{1}{4k+1}(1),$$

$b=7, \quad k \ge 1, \quad n \ge 35, \quad d=7k+2, \quad g=+3,-1$

Case 8. $n \bmod 4 = 3, \quad t = 8k + 2$

$$\frac{8k+1}{1}(234) \quad \frac{6k+1}{2k+1}(124) \quad \frac{2k+1}{6k+1}(234) \quad \frac{4k+2}{4k+1}(4) \quad \frac{1}{8k+2}(2),$$

$b = 11, \quad k \geq 1, \quad n \geq 123, \quad d = 22k+6, \quad g = +3, +7$

$$\frac{8k+1}{1}(2) \quad \frac{6k+1}{2k+1}(4) \quad \frac{2k+1}{6k+1}(2) \quad \frac{4k+1}{4k}(124) \quad \frac{1}{8k+2}(4),$$

$b = 7, \quad k \geq 1, \quad n \geq 115, \quad d = 14k+3, \quad g = -1, -5$

Case 9. $n \bmod 4 = 3, \quad t = 8k + 6$

$$\frac{6k+5}{2k+1}(1) \quad \frac{2k+2}{6k+5}(3) \quad \frac{4k+4}{4k+3}(4),$$

$b = 3, \quad k \geq 0, \quad n \geq 43, \quad d = 6k+5$

Case 10. $n \bmod 4 = 3, \quad t = 4k + 3$

$$\frac{3k+2}{k+1}(1) \quad \frac{k+1}{3k+2}(3) \quad \frac{2k+2}{2k+1}(4),$$

$b = 3, \quad k \geq 0, \quad n \geq 11, \quad d = 3k+3, \quad g = -1, -5$

$$\frac{3k+2}{k+1}(24) \quad \frac{2k+2}{2k+1}(14) \quad \frac{1}{4k+3}(234),$$

$b = 7, \quad k \geq 1, \quad n \geq 56, \quad d = 7k+6, \quad g = +3,$

In some of the Cases (2.,4.,7.,8. and 10.) two different perfect Basic b-tuples are used, in order to leave as few gaps as possible.

It can be shown that the gaps $g = +1$ in Case 5. and $g = -1$ in Case 7. must be left: the corresponding perfect d.c. n-gons do not exist; the diameter of an optimal d.c. n-gon is for 1 greater than the diameter required for a perfect d.c. n-gon. The same conclusion can be derived for the special value $n = 45$ in Case 4.

One can also check that the perfect d.c. n-gons for the special values $n = 13$ (Case 5), $n = 7$ (Case 8) and $n = 15$ (Case 10) cannot be constructed by using the given perfect Basic b-tuples. However, it is easy to construct these perfect d.c. n-gons directly.

5. CONCLUSION

The results of the previous section can be summarized in the form of the following theorem:

THEOREM 1. Let the number of edges of a d.c. n-gon P for some odd n belong to the interval $(v(t-1), v(t))$ for some natural number $t > 1$. Then the minimum diameter $mind(n)$ of P is equal to $gdlb(n)$ for each odd integer $n > 4$, except for the following cases in which $mind(n) = gdlb(n) + 1$ is satisfied:

1. n is odd, t is divisible by 4;
2. $n = v(t-1) + 1$, where t is of the form $4k + 3$, $k > 0$;
3. $n = v(t) - 1$, where t is of the form $4k + 1$, $k > 0$;
4. n = 45.

REMARK. It follows from the results of [4] that an analogous statement is valid for n even. The only exceptional values of n in which $mind(n) = gdlb(n) + 1$ is satisfied - are of the form:

5. $n = v(t-1) + 2$, where t is of the form $2k$, $k>1$;
6. $n = v(t) - 2$, where t is of the form $2k$, $k>1$.

Note that all the non-exceptional optimal d.c. n-gons, as well as the exceptional ones corresponding to the case 1. - are perfect. The algorithms for constructions with the cases 2.,3.,4. (respectively 5., 6.) are described in [2] and [3]. Using these algorithms, it can be shown that the optimal (either perfect or not) d.c. n-gons can be *efficiently* constructed from a family of (perfect) Basic b-tuples.

The results of this paper (exact constructions for n odd) put an end to a series of results motivated by the initial paper [7]: approximation formulae for minimum diameter of a d.c. n-gon ([1]), exact constructions for n even ([4]) and suboptimal constructions for n odd ([5]).

We suggest two related topics for future investigations: the maximal number of edges of a d.c.p. inscribed into a given rectangle and a generalization of the considered problem to the 3D-case.

REFERENCES

[1] Acketa,D., and Žunić,J., "On the maximal number of edges of digital convex polygons included into an m × m - grid", accepted for *Journal of Combin. Theory - Series A.*

[2] Acketa,D., Matić-Kekić,S., and Žunić,J., "An algorithm for a simple construction of suboptimal digital convex polygons", *Yugoslav Journal of Operations Research* 2/1 (1992) 73-81.

[3] Acketa,D., Matić-Kekić,S., and Žunić,J., "A linear algorithm for construction of optimal digital convex 2k-gons", *Yugoslav Journal of Operations Research* 3/2 (1993) 159-170.

[4] Acketa,D., and Žunić,J., "The minimal size of a square which includes a digital convex 2k-gon", *Bull. Austral. Math. Soc.* 48 (1993) 465-474.

[5] Acketa,D., and Žunić,J., "A simple construction of a digital convex n-gon with almost minimal diameter", *Inform. Sciences* 77 (3/4) March (1994) 275-291.

[6] Guardina,C.R., and Dougherty,E.R., *Morphological methods in image and signal processing*, Prentice Hall, Englewood Cliffs, New Jersey, 1987.

[7] Voss,K., and Klette,R., "On the maximal number of edges of a convex digital polygon included into a square", *Pocitace a umela inteligencia* 1/6 Dec. (1982) 549–558.

[8] Voss,K., *Discrete Images, Objects and Functions in Z^n*, Algorithms and Combinatorics 11, Springer, 1992.