

Largest Small n -polygons: Numerical Optimum Estimates for $n \geq 6$



János D. Pintér

Abstract The diameter of a convex planar polygon is defined as the maximum of the distances measured between all of its vertex pairs. $LSP(n)$, the *largest small polygon* with n vertices, is the polygon of unit diameter that has maximal area $A(n)$. It has been known for almost a century that for all odd values $n \geq 3$, $LSP(n)$ is the regular n -polygon. Perhaps surprisingly, this statement is not valid for even values of n . Finding the polygon $LSP(n)$ and $A(n)$ for even $n \geq 6$ has been a long-standing “puzzle” that can be considered as a class of global optimization problems. We present numerical solution estimates for all even values $6 \leq n \leq 80$, using the AMPL model development environment with the LGO global–local solver engine option. Based on these results, we also present a regression model-based estimate of the optimal area sequence $\{A(n)\}$.

Keywords Largest small polygons · Global optimization model · Numerical optimization by AMPL-LGO and other solvers · Illustrative results and comparisons · Regression model

1 Introduction

The diameter of a convex planar polygon is defined as the maximum of the distances measured between all of its vertex pairs. In other words, the diameter of the polygon is the length of its longest diagonal. The *largest small polygon* with n vertices is the polygon of unit diameter that has maximal area. For a given integer $n \geq 3$, we will refer to this polygon as $LSP(n)$ with corresponding area $A(n)$. To illustrate, see Fig. 1 that depicts the largest small hexagon $LSP(6)$; in this case, all polygon diagonals are of unit length.

For unambiguity, we will consider all $LSP(n)$ instances with a fixed position corresponding to appropriate modifications of Fig. 1 for even values $n \geq 6$. Specifically, we

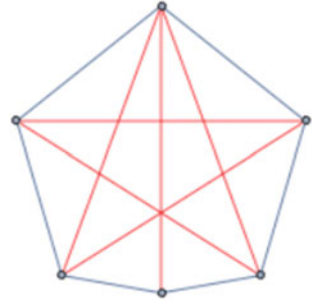
J. D. Pintér (✉)

Department of Management Science and Information Systems, Rutgers University,
100 Rockafeller Rd, Piscataway, NJ 08854, USA

e-mail: jpinter@business.rutgers.edu

URL: <https://www.business.rutgers.edu/faculty/janos-pinter>

Fig. 1 Largest small hexagon $LSP(6)$



define a planar Cartesian coordinate system in which the $LSP(n)$ polygons are positioned and express the “height” of each vertex by its coordinate on the vertical axis. Following a standard assumption, each even n -polygon considered here is symmetrical with respect to its diagonal that connects its “lowest” vertex (which is placed at the origin) with its “highest” vertex.

Reinhardt [27] proved that for all *odd* values $n \geq 3$, $LSP(n)$ is the regular n -polygon. Perhaps surprisingly, this statement is not valid for *even* values of n . For $n = 4$ (tetragon), the square with diameter 1 has maximum area $A(4) = 0.5$, but infinitely many other tetragons with diameter 1 have the same area. The case $n = 6$ (hexagon) was analyzed and solved by Graham [11]; the case $n = 8$ (octagon) was solved by Audet et al. [2]. More recently, Henrion and Messine [14] found the largest small polygons for $n = 10$ (decagon) and $n = 12$ (dodecagon) and for $n \leq 16$ presented rigorous bounds for the optimum value. We refer to these studies and cited works therein for theoretical background and for further details regarding the analysis of the problem class $\{LSP(n)\}$. We will also review the results obtained by using general-purpose nonlinear optimization software, as reported in [4] and [5]. In addition to the publications cited in our study, we refer to the topical webpages [29–31] for concise discussions with further references.

In this work, we follow a numerical global optimization approach, in order to find $LSP(n)$ configurations and corresponding estimated values $A(n)$. Following this introduction, one of the standard optimization model forms is reviewed in Sect. 2. Earlier alternative solution approaches and best-known results are reviewed in Sect. 3. The AMPL model development environment [1] and the AMPL-LGO solver option [22] are briefly discussed in Sect. 4, followed by AMPL-LGO results compared to results by other researchers and using also other solvers (Sect. 5). A regression model based on our numerical results is presented in Sect. 6, together with corresponding optimum estimates $\{A(n)\}$ for $n \geq 6$. Conclusions are presented in Sect. 7, followed by references.

2 A Standard Optimization Model for Finding $LSP(n)$

Our objective is to find numerically optimized $LSP(n)$ configurations with $n \geq 6$ vertices, n being an input parameter of the general model. The model formulation presented here is cited from Bondarenko et al. [4] who refer to Gay's model [8], discussed also in [9]. The AMPL model `pgon.mod` [8] refers to a GAMS model developed by Francisco J. Prieto (noting that a more accurate reference to Prieto's original work is unknown to this author). The corresponding GAMS model library item `polygon.gms` [7] refers to [8, 11] and the benchmarking study [5].

Following the model formulations referred to above, we consider polar coordinates to describe $LSP(n)$, assuming that vertex i is positioned at polar radius r_i and at angle θ_i . For unambiguity, we assume that the polygon vertices $i = 1, \dots, n-1$ are arranged (indexed) according to increasing angles θ_i . Placing the last vertex position at the origin, we have $r_n = 0, \theta_n = \pi$. Please refer to Fig. 1 for the hexagon instance $LSP(6)$ that corresponds to this standardized position.

2.1 Model Formulation

Maximize total area of the n -polygon:

$$\max A(n) = 1/2 \sum_{i=1, \dots, n-1} r_i r_{i+1} \sin(\theta_{i+1} - \theta_i). \quad (1)$$

Bounds for pairwise distance between vertices i and j :

$$\begin{aligned} r_i^2 + r_j^2 - 2r_i r_j \cos(\theta_i - \theta_j) &\leq 1, \\ \text{for } 1 \leq i \leq n-2, i+1 \leq j \leq n-1. \end{aligned} \quad (2)$$

Vertex angle ordering relations:

$$\theta_{i+1} - \theta_i \geq 0, \text{ for } 1 \leq i \leq n-2. \quad (3)$$

Decision variable bounds, including the two fixed settings:

$$0 \leq \theta_i \leq \pi \text{ and } 0 \leq r_i \leq 1, \text{ for } 1 \leq i \leq n-1; r_n = 0, \theta_n = \pi. \quad (4)$$

2.2 Numerical Challenges

Difficulties can be expected to arise, due to the nonconvex objective function (1) and the nonconvex constraints (2). The number of these nonlinear constraints increases *quadratically* as a function of n . For example, the $LSP(80)$ model instance has 158 decision variables (since r_n and θ_n are fixed) with corresponding bound constraints; and it has 3241 constraints of which 3161 are nonconvex (adding to the 78 linear constraints (3) the two fixed value constraints from (4)). As conjectured by other researchers and numerically supported also by the present study, while the standardized $LSP(n)$ model instances have a unique globally optimal solution, the number of local optima increases with n . Many of the local optima are close in quality to the (unknown or only approximately known) global optimum. These features make the $\{LSP(n)\}$ problem-class numerically challenging, similarly to many other object configuration design problems arising e.g. in computational physics, chemistry and biology.

3 Related Earlier Studies

3.1 Analytical Approaches

Following Graham [11]—who combines geometric insight with results based on [33]—finding $LSP(6)$ requires the *exact* solution of a 10th order irreducible polynomial equation. More specifically, the area $A(6)$ of $LSP(6)$ can be found as the second-largest real root r of the equation

$$11993 - 78488r + 144464r^2 + 1232r^3 - 221360r^4 + 146496r^5 \\ + 21056r^6 - 30848r^7 - 3008r^8 + 8192r^9 + 4096r^{10} = 0.$$

Audet et al. [2], Henrion and Messine [14] follow a different approach: in their studies finding $LSP(n)$ requires the *exact* solution of a corresponding nonconvex quadratic programming problem with quadratic constraints, combined with geometric analysis. These solution strategies, based on a different model from the one cited in Sect. 2, also brings the $LSP(n)$ problem-class into the realm of global optimization.

In [14] it is conjectured that $LSP(n)$ for all even values $n \geq 4$ has a symmetry axis, as indicated by Fig. 1 for $LSP(6)$. This conjecture was proven by Reinhardt [27] for $n = 4$, and by Yuan [34] for $n = 6$. As noted by Henrion and Messine, Graham used this conjecture to find $LSP(6)$; the $LSP(8)$ configuration found in [2] also supports the conjecture. The software packages SeDuMi [28], VSDP [16], and GloptiPoly [12, 13] are used in [14] to solve $LSP(n)$ instances for $6 \leq n \leq 16$. Henrion

Table 1 Numerical results based on analytical approaches

n	$LSP(n)$ area $A(n)$	References
4	0.5	Reinhardt [27]
6	0.674981	Graham [11] ^a
8	0.726867	Audet et al. [2] ^a
8	$0.72686845 \leq A(8) \leq 0.72686849$	Henrion and Messine [14] ^b
10	$0.74913721 \leq A(10) \leq 0.74913736$	Henrion and Messine [14]
12	$0.76072986 \leq A(12) \leq 0.76072997$	Henrion and Messine [14]
14	$0.76753100 \leq A(14) \leq 0.76893595$	Henrion and Messine [14]
16	$0.77185969 \leq A(16) \leq 0.77279135$	Henrion and Messine [14]

^aThe results given with lower (6-decimal digit) precision are cited from [2, 11, 29]

^bNotice the slight numerical discrepancy between the results of [2, 14] for the case $n = 8$

and Messine also discuss the current computational limitations of this approach, as runtimes increase rapidly from seconds to tens of minutes in their numerical tests.

Table 1 summarizes *all* currently known validated numerical results, including also the bounds reported in [14].

3.2 Numerical Solution Approaches

The COPS technical report [4] by Bondarenko et al. presents comparative numerical results for several $LSP(n)$ instances as shown in Table 2. These results were obtained by using the *local* nonlinear optimization software packages DONLP2, LANCELOT, LOQO, MINOS, and SNOPT (as of September 1998) linked to the AMPL modeling environment.

Table 2 summarizes the *best* numerical solution—obtained by at least one of the above listed solvers—cited from the COPS report. The term *best* refers to the solution

Table 2 Numerical results obtained by using local nonlinear optimization software [4, 5]. Best results attained by one of DONLP2, LANCELOT, LOQO, MINOS, SNOPT

n	$LSP(n)$ area $A(n)$
6	0.6749814429
10	0.7491373458
20	0.7768587560
25	0.779740 ^a
50	0.7840161480
75	0.784769 ^a
100	0.7850565708

^aThe numerical results given with 6 decimal digit precision are cited from [5]

Table 3 Numerical results reported by Mossinghoff [17]

n	$LSP(n)$ area $A(n)$
6	0.6749814429
8	0.7268684828
10	0.7491373459
12	0.7607298734
14	0.7675310111
16	0.7718613220
18	0.7747881651
20	0.7768587560

which has the highest objective function value, while meeting all model constraints with at least 10^{-8} precision. For completeness, we also added results for $n = 25, 50, 75, 100$ from the subsequent benchmarking study [5] in which LANCELOT, LOQO, MINOS, and SNOPT were tested: again, we cite only the *best* results.

Mossinghoff [17] also studied the $\{LSP(n)\}$ problem-class. He describes an approach to search for polygons with an even number of sides n and fixed diameter d (here $d = 1$), aiming at the largest possible area. The construction is based on optimizing a parameterized polygon model, leading to an apparently difficult numerical problem to handle. For arbitrary even $n \geq 6$, Mossinghoff's construction leads to a polygon, denoted by Q_n which provably has a larger area than that of the regular polygon P_n . *Mathematica*, by Wolfram Research [32], has been used by Mossinghoff to find Q_n for $6 \leq n \leq 20$: see Table 3. The required calculations are far from trivial: consult [17] for further details and alternatives. To the author's knowledge, this approach has not been applied to find Q_n for $n > 20$. Arguably—and similarly to work performed by other researchers—this is due to the rapidly increasing difficulties to carry out the calculations required.

3.3 The Asymptotic Behaviour of $A(n)$

According to the optimization model presented in Sect. 2, the numerical solution of $LSP(n)$ instances requires the handling of a nonlinear programming problem with $O(n^2)$ nonconvex constraints, and a nonconvex objective function. In spite of the implied difficulty, $LSP(n)$ problems are thought not to become dramatically more difficult to handle as n increases—at least in terms of finding reasonable numerical optimum estimates. This opinion is based on the generally postulated structural similarity and symmetry of the sequence of $\{LSP(n)\}$ configurations. As noted e.g., in [4], the optimal $LSP(n)$ configurations approach the circle of unit diameter as $n \rightarrow \infty$ consult [17] for related asymptotic results. Consequently,

$$A(\infty) = \lim_{n \rightarrow \infty} A(n) = \pi/4 \sim 0.7853981634.$$

Based on this conjectured structure, in [4, 5] “a polygon with almost equal sides” is used as the initial solution guess in all numerical tests. This approach is implemented in both the AMPL and GAMS model codes referred to earlier. Such solution strategy illustrates the point that, in optimization models good insight and resulting initial solution guess can become a valuable step towards finding credible numerical solutions efficiently. This heuristic approach, however, does not guarantee provable global optimality in many nonconvex models, including the $\{LSP(n)\}$ model-class discussed here. In spite of such “hand-crafted” initial solutions, the high-quality local nonlinear solvers mentioned above often failed to find solutions, the solutions returned were typically somewhat different, and in a number of cases evidently suboptimal. For further details regarding this point, consult [4, 5], and the results presented later on in Tables 4 and 5.

4 Solving LSP Problems Numerically by AMPL-LGO

4.1 Solution Approach

In this study, we follow a numerical optimization approach, keeping in mind also the cautionary notes presented above. Specifically, using the LGO global–local optimization solver engine linked to the AMPL modeling environment, we present *numerical optimum estimates* for all even values $6 \leq n \leq 80$. Our results, presented in Sect. 5, are in close agreement with the best results reported in Tables 1, 2 and 3—when comparable results are available. For comparison, we also report results obtained by using the currently available alternative solver options MINOS [18], SNOPT [10], and IPOPT [15] linked to AMPL.

4.2 The AMPL Model Development Environment

AMPL is a powerful modeling language that facilitates the formulation of optimization models. AMPL enables model development in a natural, concise, and scalable manner. AMPL also supports model analysis, the usage of different data sets for the same model, the seamless invocation of various solver engine options to handle optimization models, together with report generation and many other useful features not discussed here. AMPL has been extensively documented elsewhere: we refer to the AMPL book [6], and to the resources available at the AMPL website [1].

4.3 LGO Solver Suite for Nonlinear Optimization

Nonlinear optimization models frequently have multiple—local and global—optima: the objective of global optimization is to find the best possible solution under such circumstances. LGO is an integrated global–local solver suite for constrained nonlinear optimization. The model-class addressed by LGO is concisely defined by the vector of decision variables $x \in R^n$; the explicit, finite n -vector variable bounds l and u ; the continuous objective function $f(x)$; and the (possibly absent) m -vector of continuous constraint functions $g(x)$. Applying these notations, LGO is aimed at numerically solving models of the general form.

$$\min f(x) \text{ subject to } x \in D := \{l \leq x \leq u, g(x) \leq 0\}. \quad (5)$$

In (5) all vector inequalities are interpreted component-wise: l , x , u , are n -component vectors and 0 denotes the m -component zero vector. Formally more general optimization models that include also $=$ and constraint relations and/or explicit lower and upper bounds on the constraint function values can be directly deduced to the model form (5). If D is non-empty, then the stated key analytical assumptions guarantee that the optimal solution set X^* of the model is non-empty; however, finding X^* could still remain a formidable analytical and/or numerical challenge. Clearly, the $\{LSP(n)\}$ problem-class is encompassed by the optimization model form (5).

Without going into further details regarding LGO, we mention that the foundations of the LGO software development project are discussed by Pintér [19]; further implementation aspects are discussed e.g., in [20, 21]. Here we utilize the LGO solver option available for use with AMPL [22]; the current stand-alone LGO implementation is documented in [23]. In addition to these references, the studies [24, 25] present numerical results using LGO to solve a range of nonlinear optimization problems, from relatively simple standard test problems to well-known challenges. LGO also has been applied to handle a broad range of business, engineering and scientific optimization problems.

5 Numerical Results and Comparisons

5.1 AMPL-LGO Results

In our numerical tests reported here, the AMPL code implementation `pgon.mod` was used. All test runs were conducted on a several years old laptop PC with Intel Core i5-3337-U CPU @ 1.80 GHz (x-64 processor), 16 Gb RAM, running under the Windows 10 (64-bit) operating system.

The results of a *single, completely reproducible* run-sequence are summarized in Table 4 for all even values $6 \leq n \leq 80$, with a single default setting of all solver options,

using LGO in global search mode. In several (seemingly more difficult) cases, we received somewhat better numerical results in additional tests, at the expense of longer runtimes: however, for consistency, we did not include those results here. Let us also mention that LGO in its local search mode often found optimum estimates that are numerically identical or close to the global search-based solution, at a fraction of the runtimes reported below. (Recall the related comment from Sect. 3.3.)

The results summarized in Table 4 are directly cited from the AMPL-LGO solver output; the corresponding $LSP(n)$ configurations can be optionally reported in the AMPL command window, and/or written to a result text file. To avoid reporting such excessive details, the optimized configurations found are not presented here. An illustrative collection of detailed results has been kept for documentation and archival purposes, and all results can be reproduced in a few minutes. The reported precision of our numerical results is set to 10 digits after the decimal point. Arguably, this is a bit of “overkill”, but it is in line with the required constraint satisfaction precision as shown below. The results reported also support an in-depth comparison with the results cited earlier, as well as with the results obtained using alternative AMPL solvers (noting that in some cases the differences between the optimum estimates found are rather small).

Our numerical results for $n \leq 20$ are in fairly close agreement with the *best* results displayed in Tables 1, 2 and 3. In several cases, we found somewhat better *conjectured* optimum estimates compared to the earlier results in Tables 1 and 2; and our results up to $n = 20$ are in close agreement (up to 8 decimal digits) with those reported by Mossinghoff, see Table 3. The runtimes appear to scale rather well for $6 \leq n \leq 80$, mostly (but not always) increasing with n . The entire sequence of the 38 optimization runs reported here took a little over 10 min.

Although AMPL-LGO seems to perform reasonably well in comparison to the other solvers tested by us or by others (as reported above), its numerical limitations start to show around $n = 64$ when used in a pre-set default mode. The results presented in Table 4 for $n = 64, 74$, and 78 are clearly suboptimal, while all other $A(n)$ values are monotonically increasing with n , as expected. Instead of “tweaking” the LGO option parameters—e.g., by increasing the pre-set global search effort limit (which was actually reached in several cases reported above, for some of the larger n values), or increasing the runtime limit (set to 5 min for each run, but never reached)—here we very simply use linear interpolation to “adjust” the clearly suboptimal results based on the bracketing values in Table 4. For example, $A(64)$ is estimated on the basis of the results obtained for $A(62)$ and $A(66)$. Applying such simple interpolation leads to the following estimated values: $A(64) \sim 0.7845510976$, $A(74) \sim 0.7847519869$, $A(78) \sim 0.7847919330$.

The reason to produce these simple estimates is to use them in Sect. 6, to develop a regression model for the entire sequence $A(n)$.

Table 4 AMPL-LGO numerical results

n	$LSP(n)$ area $A(n)$	Runtime (seconds)	Maximum constraint violation
6	0.6749814433	0.55	2.21e-09 (i.e., $2.21 \cdot 10^{-9}$, etc.)
8	0.7268684830	0.70	6.47e-09
10	0.7491373457	0.95	2.96e-10
12	0.7607298709	1.30	8.9e-10
14	0.7675310106	1.69	3.91e-09
16	0.7718613224	2.55	4.09e-09
18	0.7747881650	2.63	9.78e-09
20	0.7768587506	3.02	2.23e-09
22	0.7783773308	3.95	9.08e-09
24	0.7795240461	5.22	7.73e-09
26	0.7804111201	5.34	6.34e-09
28	0.7811114192	6.05	9.83e-09
30	0.7816739255	6.98	3.67e-09
32	0.7818946320	5.72	6.29e-10
34	0.7823103007	7.61	9.03e-09
36	0.7826513767	9.50	9.75e-09
38	0.7829526627	9.34	5.08e-09
40	0.7832011589	9.55	8.47e-11
42	0.7834135187	12.06	4.62e-09
44	0.7835966860	13.22	1.42e-09
46	0.7837554636	16.88	3.43e-09
48	0.7838942710	17.95	8.31e-09
50	0.7840161496	16.53	9.99e-09
52	0.7841233641	20.61	8.78e-09
54	0.7842192995	21.38	9.18e-09
56	0.7843044654	23.91	3.87e-09
58	0.7843807534	22.95	8.43e-09
60	0.7844492943	27.97	9.79e-09
62	0.7845111362	21.22	8.93e-09
64	0.7834620877	30.48	9.82e-09
66	0.7845910589	34.17	1.19e-09
68	0.7846139029	35.84	9.00e-09
70	0.7846403575	22.33	6.45e-09
72	0.7847454020	42.75	7.34e-09

(continued)

Table 4 (continued)

n	$LSP(n)$ area $A(n)$	Runtime (seconds)	Maximum constraint violation
74	0.7845564840	26.25	3.54e-09
76	0.7847585719	49.19	8.95e-09
78	0.7845160579	49.47	9.64e-09
80	0.7848252941	51.45	7.25e-09

5.2 An Illustrative Comparison with Results Obtained by Several AMPL Solvers

For a somewhat more comprehensive picture, we also generated a set of comparative results using several currently available AMPL solvers, namely: MINOS, SNOPT, IPOPT, and LGO. All solvers are used with their default settings. We did not include all even values $6 \leq n \leq 80$, only a representative subset (starting from $n = 30$, we increased n by 10), since—based on the results obtained—the solver performance tendencies seem rather clear. Table 5 summarizes these numerical results; the LGO results are directly imported from Table 4.

In several cases, MINOS and SNOPT issued interim (runtime) warning messages—while in most runs they report optimal solutions on return—but all runs were properly terminated with the results shown in Table 5. All solvers return close, but slightly different results for $n = 6$ and $n = 8$. The first clearly notable difference appears at $n = 10$, where IPOPT returns a somewhat inferior result compared to the other three solvers.

The numerical limitations of *all* tested solvers become more apparent as n increases. In several cases, MINOS returns clearly inferior results, and except for small values of n , it produces inferior results that are a few percent below the best solution returned considering all solvers. IPOPT consistently returns somewhat inferior results, but still within a few percent of the best solution returned by at least one of the solvers. SNOPT works well for the considered range of n values, in several cases returning slightly better objective function value estimates than LGO. In the other cases, LGO returns best results, with relatively little difference between LGO and SNOPT results. Let us point out that the constraint satisfaction levels attained by these solvers are a bit different, depending also on the $LSP(n)$ instance solved: therefore, it would be inappropriate to draw far-reaching conclusions based on rather small differences in the reported objective function values.

The LSP model-class clearly poses a challenge to the high-quality solvers tested here. Arguably, the same conclusion remains valid for other solver engines which could not be considered in the present study. To support this statement, consult also [4, 5] for the numerical results included for LSP models.

Table 5 Comparative numerical results obtained by using several AMPL solvers $nLSP(n)$ area $A(n)$

	MINOS	SNOPT	IPOPT	LGO
6	0.6749814429	0.6749814429	0.6749814308	0.6749814433
8	0.7268684828	0.7268684827	0.7268684678	0.7268684830
10	0.7491373459	0.7491373459	0.7371215901	0.7491373457
12	0.7607298734	0.7607298734	0.7542668597	0.7607298709
14	0.7521931121	0.7675310112	0.7675309793	0.7675310106
16	0.7625954979	0.7718613220	0.7696844715	0.7718613224
18	0.7554106917	0.7747881651	0.7491373424	0.7747881650
20	0.7649920891	0.7768587560	0.7732071277	0.7768587506
22	0.7640946468	0.7783773301	0.7607298336	0.7783773308
24	0.7640946468	0.7795240452	0.7548403603	0.7795240461
26	0.7636943870	0.7804111199	0.7523851367	0.7804111201
28	0.7641232665	0.7807502582	0.7523851373	0.7811114192
30	0.4738428148	0.7813775853	0.7491373081	0.7816739255
40	0.7740433310	0.7832011593	0.7268684622	0.7832011589
50	0.5591307889	0.7820205034	0.7197409051	0.7840161496
60	0.7403488333	0.7827992931	0.6749814462	0.7844492943
70	0.7605166660	0.7846838685	0.7268685003	0.7846403575
80	0.5070738413	0.7848417622	0.7197409068	0.7848252941

6 Regression Model Development

Based on the AMPL-LGO numerical results, we present a simple nonlinear regression model that enables the estimation of the optimal area $A(n)$, for all *even* values of $n \geq 6$. Obviously, the same type of regression model could be used to estimate $A(n)$ also for *odd* values. However, based on the optimality of regular n -polygons [27], one could *exactly* compute $A(n)$ for all odd values of n .

Given that $A(n)$ is a monotonically increasing function of n , and $A(\infty) = \pi/4$, the following model form is conjectured for even values $n \geq 6$:

$$A(n) = \pi/4 - c_1/n - c_2/n^2 - c_3/n^3. \tag{6}$$

In (6), the parameters c_1, c_2, c_3 are expected to be positive. To calibrate this model, we use the estimated $A(n) 6 \leq n \leq 80$ values found by AMPL-LGO, substituting the three apparently suboptimal calculated $A(n)$ values by their interpolated approximation as discussed earlier.

The regression model parameters were determined using the NonlinearModelFit function of *Mathematica* [32]. This leads to the following model, rounding the coefficients to five digits after the decimal point (to reflect the expected model

Table 6 Estimated $A(n)$ values based on the regression model (7) vs. best known results

n	6	8	10	20	30	40
$A(n)$ est	0.674983	0.726829	0.749185	0.776816	0.781572	0.783209
Best res	0.674981	0.726868	0.749137	0.776859	0.781674	0.783201
n	50	60	70	80	90	100
$A(n)$ est	0.783965	0.784378	0.784629	0.784794	0.784908	0.784991
Best res	0.784016	0.784449	0.784684	0.784842	0.784946	0.785028
n	200	300	400	500	1000	2000
$A(n)$ est	0.785270	0.785329	0.785352	0.785364	0.785384	0.785392
Best res	0.785316	0.785309	0.785356	???	???	???

accuracy):

$$A(n) \sim \pi/4 - 0.01098/n - 2.91512/n^2 - 5.96369/n^3. \tag{7}$$

Applying this regression model, in Table 6 we present an illustrative set of $A(n)$ estimates vs. the best known numerical results from Tables 1, 2, 3, 4 and 5. All values are rounded to 6-digit precision after the decimal point.

All estimated values shown in Table 6 are in reasonable agreement with the best numerical results presented in Tables 1, 2, 3, 4 and 5 whenever such values are available. The relative difference between the calculated best values and estimated values is less than 10^{-4} in all examples included in Table 5, except for $n = 30$, where the relative difference approximately equals $1.3 \cdot 10^{-4}$.

The calculated optima shown for $n = 90, 100, 200, 300, 400$ were found using SNOPT. (Due to the current pre-set model size limitations of AMPL-LGO, it could not be used for doing these calculations; and the solvers MINOS and IPOPT were deemed unsuitable due to their inferior performance experienced for smaller n values). The SNOPT runtimes—which were below one second in most cases for $n < 100$ (reaching about 5 s for $n = 100$) – increased rapidly when n was sequentially set to 200, 300, and 400. SNOPT was running for more than an hour on the computer mentioned earlier to solve the $n = 400$ model instance. The entries ??? for $n = 500, 1000, 2000$ indicate that we did not attempt to calculate these, since none of the solvers used in this study seemed capable to return numerical results within an acceptable timeframe.

Let us point out that while the $LSP(100)$ model-instance has “only” 198 decision variables and 5049 constraints (omitting the prefixed values), the $LSP(400)$ model-instance has 798 variables, and the number of constraints is 80199. It seems clear that solving $LSP(n)$ models directly, for *arbitrarily large even* values n , is beyond the capability of current (and perhaps also of future) numerical optimization tools. This aspect makes the regression model based estimation approach a simple viable alternative.

Since the data used to develop the regression model (2) are likely to be at least slightly suboptimal, one can expect that—generally speaking—the estimated $A(n)$ values could be also suboptimal. This tendency can be observed in Table 6, but one can see also some exceptions, indicating regression model error and/or optimization inaccuracy.

Let us also note that—for the purpose of developing a regression model—the exact values $\{A(n)\}$ for odd n could also be used. However, this approach would be based on “mixing” numerically *exact* and *estimated* optima, and hence would give less indication of the quality of the numerical solutions found for even values of n . For this reason, we used only the results obtained in the present study, and we did not attempt to find adjusted optimum estimates based on further information.

To support a more complete comparative analysis, first and second order regression models (with $1/n$ as their input argument) were also calculated, but the third order model (6) clearly resulted in a superior fit to the entire data set used. Obviously, within reason, higher order models (or perhaps other model types) could give even more precise fit to the data, but—considering also the inherent data inaccuracies—the third order model (6) already gives a fairly good fit. Figure 2 displays the model function curve defined by (7) together with the adjusted data set (represented by dots) that includes the interpolated data.

Figure 3 displays the regression model residuals. With a few exceptions, the residual errors are less than $1 \cdot 10^{-4}$; the absolute value of the singularly largest estimated error is around $8 \cdot 10^{-4}$. All estimated error values are fairly small, compared to the approximate range $[0.674981, 0.784825]$ of the observed data.

One can observe that most of the residuals seem to follow an interesting cyclical pattern that—in the author’s opinion—seems more due to the inherent structure of the $LSP(n)$ problem-class than to numerical fluctuations and other “noise” induced by the computational environment.

Fig. 2 The nonlinear regression model (7), vs. the adjusted data set $A(n)$ (dots) for $6 \leq n \leq 80$

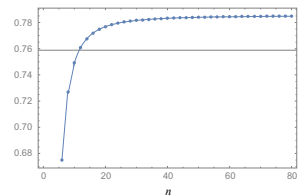
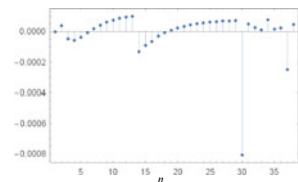


Fig. 3 Residuals (see dots) in the regression model (7) of $A(n)$, for $3 \leq n \leq 40$



7 Concluding Remarks

In this study, we address the problem of finding numerically the sequence of largest small n -polygons $LSP(n)$ with unit diameter and maximal area $A(n)$. Finding $LSP(n)$ and $A(n)$ for even values of $n \geq 6$ has been a long-standing challenge, leading to an interesting class of nonlinear optimization problems with different formulations by a number of researchers.

The structural properties of this problem, and of similar optimization challenges—e.g., atomic structure models, potential energy models, regular object packings, and other problems in which the goal is to find the best configuration of identical objects—often support the proposition of “credible” initial solutions and solution guesses. However, finding the true global solution typically remains difficult, as the cited earlier studies and our present work illustrate.

Using the AMPL modeling environment with the LGO solver option, we present global search based numerical solutions for all even values $6 \leq n \leq 80$, in a matter of minutes. Our results are comparable to (and in a number of cases are somewhat better than) the best results obtained earlier by other authors and by other solver software options, before our results were produced. Based on the results obtained, we also propose a regression model that enables the simple estimation of the optimal area sequence $\{A(n)\}$, for arbitrary integer values of n .

Upon revising the manuscript of this work, it was brought to our attention by a helpful reviewer that a recently posted (September 2020) study [3] presents *numerical optimum estimates* which are somewhat better for $n \geq 32$ than the *numerical optimum estimates* presented in our work. To illustrate, [3] reports for $n = 32$ the estimate 0.7821325276 *versus* our estimate 0.7818946320: the approximate ratio of these values is 0.9996958. To produce the results reported in [3] for a selection of even values $n \leq 128$, a different global optimization model was used; MATLAB and CVX serve as the modeling environment, and the MOSEK Optimization Suite was used (with a default precision setting which, without delving into further details, seems to be similar to the numerical feasibility tolerance used in our study).

In response to the above, we point out that [3] cites our numerical results obtained in 2018 (and left unchanged for the present study). We never claimed more than producing credible numerical optimum estimates using off-the-shelf optimization software which returns results in seconds or minutes for the model instances discussed here.

Let us add that in a recently completed study [26] we present numerical results for an illustrative sequence of even values of n , up to $n \leq 1000$. Our results are in close agreement with (or surpass) the best results reported in all earlier studies known to us, including the results presented in [3]. For completeness, we also calculated numerically optimized results for a selection of odd values of n , up to $n \leq 999$. In this study, we used a tighter model formulation; to handle this model, *Mathematica* was used with the IPOPT solver option. Following up by corresponding regression models (similarly to our present work), we present *numerical solution estimates* for the entire LSP model-class.

The motto of the benchmarking studies [4, 5] is, arguably, somewhat provocative and funny, but the message is worth quoting: “*COPS: Keeping optimization software honest.*” In line with this message, let us conclude with some honest and pragmatic advice, not driven by unconditional “*software developer’s pride*”. Facing the vast universe of nonlinear optimization problems, it is advisable to refrain from confident blanket statements regarding the superiority of any particular solver software over others. Instead, it is good practice to use a repertoire of appropriate model versions and solver options whenever possible, especially since it may not be obvious a priori which model type or solver engine will work best for a novel or unusually hard optimization challenge.

Acknowledgements The author thanks David M. Gay (AMPL Optimization, Inc.) for his contributions to the AMPL-LGO solver link development, and for many useful discussions. Thanks are due also to AMPL Optimization and Wolfram Research for their long-standing support of the author’s research.

References

1. AMPL Optimization, AMPL. www.ampl.com, 2020.
2. C. Audet, P. Hansen, F. Messine, and J. Xiong, The Largest Small Octagon, *Journal of Combinatorial Theory, Series A* 98 (2002), pp. 46–59.
3. Bingane, Largest Small Polygons: A Sequential Convex Optimization Approach. Available at <https://arxiv.org/pdf/2009.07893.pdf>. (Manuscript dated September 18, 2020.)
4. A.S. Bondarenko, D.M. Bortz, and J.J. Moré, *COPS: Large-Scale Nonlinearly Constrained Optimization Problems*, Technical Report ANL/MCS-TM-237, Argonne National Laboratory, Argonne, IL, 1998.
5. E.D. Dolan, and J.J. Moré, *Benchmarking Optimization Software with COPS*, Technical Report ANL/MCS-TM-246, Argonne National Laboratory, Argonne, IL, 2000.
6. R. Fourer, D.M. Gay, and B.W. Kernighan, *AMPL: A Modeling Language for Mathematical Programming* (2nd Edition), Brooks/Cole-Thomson Learning, Pacific Grove, CA, 2003.
7. GAMS Development Corporation, *Model libraries; polygon.gms: Largest small polygon COPS 2.0 #1*, https://www.gams.com/latest/gamslib_ml/libhtml/gamslib_polygon.html; 2018.
8. D.M. Gay, *pgon.mod*, <https://netlib.org/ampl/models/pgon.mod>; 1998.
9. D.M. Gay, The AMPL Modeling Language: An Aid to Formulating and Solving Optimization Problems, in M. Al-Baali, L. Grandinetti, and A. Purnama, Eds., *Numerical Analysis and Optimization: NAO-III* (Muscat, Oman, January 2014), pp. 95–116. Springer International Publishing, Switzerland, 2015.
10. P.E. Gill, W. Murray, and M.A. Saunders, *User’s Guide for SNOPT Version 7: Software for Large-Scale Nonlinear Programming*, Systems Optimization Laboratory, Department of Management Science and Engineering, Stanford University, Stanford, CA, 2006.
11. R.L. Graham, The Largest Small Hexagon, *Journal of Combinatorial Theory, Series A* 18 (1975) pp. 165–170.
12. D. Henrion, and J.B. Lasserre, GloptiPoly: Global Optimization over Polynomials with Matlab and SeDuMi. *ACM Transactions on Mathematical Software* 29 (2003) 2, pp. 165–194.
13. D. Henrion, J.B. Lasserre, and Loeferberg, J., GloptiPoly 3: Moments, Optimization and Semidefinite Programming, *Optimization Methods and Software* 24 (2009), 4–5, pp. 761–779.
14. D. Henrion, and F. Messine, Finding Largest Small Polygons with GloptiPoly. Available at <https://arxiv.org/pdf/1103.4456.pdf>. (Current manuscript version dated June 17, 2018.)

15. IPOPT, *Ipopt, a Library for Large-scale Nonlinear Optimization*; <https://projects.coin-or.org/Ipopt>. The authors of Ipopt are listed at <https://projects.coin-or.org/Ipopt/browser/trunk/Ipopt/AUTHORS>; 2018.
16. C. Jansson, VSDP: A Matlab Software Package for Verified Semidefinite Programming, *Nonlinear Theory and its Applications* (2006) pp. 327–330.
17. M.J. Mossinghoff, Isodiametric Problems for Polygons. *Discrete and Computational Geometry*, 36 (2006) 2, pp. 363–379.
18. B.A. Murtagh, and M.A. Saunders, *MINOS 5.5 User's Guide. Technical Report SOL 83–20R*. Systems Optimization Laboratory, Department of Management Science and Engineering, Stanford University, Stanford, CA, 1983; revised version 1998.
19. J.D. Pintér, *Global Optimization in Action*. Kluwer Academic Publishers, Dordrecht, 1996.
20. J.D. Pintér, Global Optimization: Software, Test Problems, and Applications, in Pardalos, P.M. and Romeijn, H.E., Eds. *Handbook of Global Optimization, Volume 2*, pp. 515–569. Kluwer Academic Publishers, Dordrecht, 2002.
21. J.D. Pintér, Software Development for Global Optimization, in Pardalos, P.M. and T. F. Coleman, Eds. *Global Optimization: Methods and Applications*, pp. 183–204. Fields Institute Communications Volume 55. Published by the American Mathematical Society, Providence, RI, 2009.
22. J.D. Pintér, *AMPL-LGO User's Guide*, AMPL Optimization, Inc.
23. J.D. Pintér, *LGO Solver Suite for Global and Local Nonlinear Optimization: User's Guide*, PCS, Inc.
24. J.D. Pintér, How Difficult is Nonlinear Optimization? A Practical Solver Tuning Approach, with Illustrative Results, *Annals of Operations Research* 265 (2018), pp. 119–141.
25. J.D. Pintér, F.J. Kampas, and I. Castillo, Globally Optimized Packings of Non-uniform Size Spheres in \mathbb{R}^d : A Computational Study, *Optimization Letters* 12 (2018) 3, pp. 585–613.
26. J.D. Pintér, F.J. Kampas, and I. Castillo, Finding the Sequence of Largest Small n -Polygons by Numerical Optimization. https://www.optimization-online.org/DB_FILE/2020/11/8112.pdf.
27. K. Reinhardt, Extremale Polygone gegebenen Durchmessers, *Jahresbericht der Deutschen Mathematiker-Vereinigung* 31(1922), pp. 251–270.
28. J.F. Sturm, Using SeDuMi 1.02, A Matlab Toolbox for Optimization over Symmetric Cones, *Optimization Methods and Software* 11 (1999) (1–4), 625–653.
29. E.W. Weisstein, “Biggest Little Polygon.” From MathWorld – A Wolfram Web Resource. <https://mathworld.wolfram.com/BiggestLittlePolygon.html>. (Retrieved on June 28, 2020)
30. E.W. Weisstein, “Graham's Biggest Little Hexagon.” From MathWorld—A Wolfram Web Resource. <https://mathworld.wolfram.com/GrahamsBiggestLittleHexagon.html>. (Retrieved on June 28, 2020)
31. E.W. Weisstein, “Regular Polygon.” From MathWorld—A Wolfram Web Resource. <https://mathworld.wolfram.com/RegularPolygon.html>. (Retrieved on June 28, 2020)
32. Wolfram Research, *Mathematica (Version 11.0)*, Wolfram Research, Inc., Champaign, IL, 2018.
33. D.R. Woodall, Thrackles and Deadlock, in D.J.A. Welsh, Ed. *Combinatorial Mathematics and Its Applications*, pp. 335–347, Academic Press, New York, 1971.
34. B. Yuan, *The Largest Small Hexagon*, M.Sc. Thesis, Department of Mathematics, National University of Singapore, 2004.