Improved upper bounds in the moving sofa problem

Yoav Kallus* Dan Romik†

June 15, 2017

Abstract

The moving sofa problem, posed by L. Moser in 1966, asks for the planar shape of maximal area that can move around a right-angled corner in a hallway of unit width. It is known that a maximal area shape exists, and that its area is at least $2.2195\ldots$—the area of an explicit construction found by Gerver in 1992—and at most $2\sqrt{2} \approx 2.82$, with the lower bound being conjectured as the true value. We prove a new and improved upper bound of 2.37. The method involves a computer-assisted proof scheme that can be used to rigorously derive further improved upper bounds that converge to the correct value.

1 Introduction

The **moving sofa problem** is a well-known unsolved problem in geometry, first posed by Leo Moser in 1966 [3, 10]. It asks:

> *What is the planar shape of maximal area that can be moved around a right-angled corner in a hallway of unit width?*

We refer to a connected planar shape that can be moved around a corner in a hallway as described in the problem as a **moving sofa shape**, or simply a **moving sofa**. It is known [5] that a moving sofa of maximal area exists. The shape of largest area currently known is an explicit construction

*Santa Fe Institute, 1399 Hyde Park Road, Santa Fe, NM 87501, USA.

†Department of Mathematics, University of California, Davis, One Shields Ave, Davis, CA 95616, USA. Email: `romik@math.ucdavis.edu`

Key words: moving sofa problem, geometric optimization, branch-and-bound, computer-assisted proof, experimental mathematics.

2010 Mathematics Subject Classification: 49Q10.

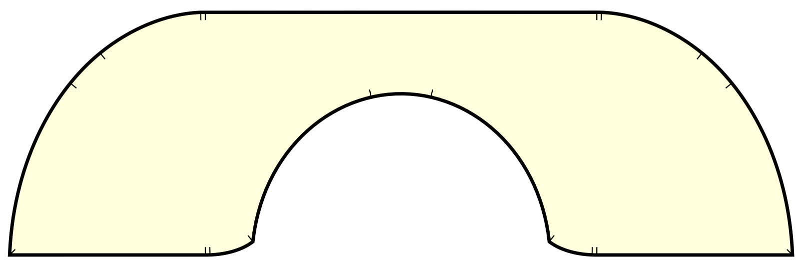

Figure 1: Gerver's sofa, conjectured to be the solution to the moving sofa problem. Its boundary is made up of 18 curves, each given by a separate analytic formula; the tick marks show the points of transition between different analytic pieces of the boundary.

found by Joseph Gerver in 1992 [5] (see also [12] for a recent perspective on Gerver's results), known as **Gerver's sofa** and shown in Figure 1. Its area is **Gerver's constant**

$$\mu_{\mathrm{G}} = 2.21953166\ldots,$$

an exotic mathematical constant that is defined in terms of a certain system of transcendental equations but which does not seem to be expressible in closed form. Gerver conjectured that μ_{G} is the largest possible area for a moving sofa, a possibility supported heuristically by the local-optimality considerations from which his shape was derived.

Gerver's construction provides a lower bound on the maximal area of a moving sofa. In the opposite direction, it was proved by Hammersley [8] in 1968 that a moving sofa cannot have an area larger than $2\sqrt{2} \approx 2.82$. It is helpful to reformulate these results by denoting

$$\mu_{\mathrm{MS}} = \max\Big\{ \mathrm{area}(S) \,:\, S \text{ is a moving sofa shape} \Big\},$$

the so-called **moving sofa constant**.[1] The above-mentioned results then translate to the statement that

$$\mu_{\mathrm{G}} \le \mu_{\mathrm{MS}} \le 2\sqrt{2}.$$

[1] See Finch's book [4, Sec. 8.12]; note that Finch refers to Gerver's constant μ_{G} as the "moving sofa constant," but this terminology currently seems unwarranted in the absence of a proof that the two constants are equal.

The main goal of this paper is to derive improved upper bounds for μ_{MS}. We prove the following explicit improvement to Hammersley's upper bound from 1968.

Theorem 1 (New area upper bound in the moving sofa problem)**.** *We have the bound*

$$\mu_{MS} \leq 2.37. \tag{1}$$

More importantly than the specific bound 2.37, our approach to proving Theorem 1 involves the design of a computer-assisted proof scheme that can be used to rigorously derive even sharper upper bounds; in fact, our algorithm can produce a sequence of rigorously-certified bounds that converge to the true value μ_{MS} (see Theorems 5 and 8 below). An implementation of the scheme we coded in `C++` using exact rational arithmetic certifies 2.37 as a valid upper bound after running for 480 hours on one core of a 2.3 GHz Intel Xeon E5-2630 processor. Weaker bounds that are still stronger than Hammersley's bound can be proved in much less time—for example, a bound of 2.7 can be proved using less than one minute of processing time.

Our proof scheme is based on the observation that the moving sofa problem, which is an optimization problem in an infinite-dimensional space of shapes, can be relaxed in many ways to arrive at a family of finite-dimensional optimization problems in certain spaces of polygonal shapes. These finite-dimensional optimization problems are amenable to attack using a computer search.

Another of our results establishes new restrictions on a moving sofa shape of largest area. Gerver [5] proved that such a largest area shape must undergo rotation by an angle of at least $\pi/3$ as it moves around the corner, and does not need to rotate by an angle greater than $\pi/2$. As explained in the next section, Gerver's argument actually proves a slightly stronger result with $\pi/3$ replaced by the angle $\beta_0 = \sec^{-1}(\mu_{\mathrm{G}}) \approx 63.22°$. We will prove the following improved bound on the angle of rotation of a moving sofa of maximal area.

Theorem 2 (New rotation lower bound in the moving sofa problem)**.** *Any moving sofa shape of largest area must undergo rotation by an angle of at least* $\sin^{-1}(84/85) \approx 81.203°$ *as it moves around the corner.*

There is no reason to expect this bound to be sharp; in fact, it is natural to conjecture that any largest area moving sofa shape must undergo rotation by an angle of $\pi/2$. As with the case of the bound (1), our techniques make it possible in principle to produce further improvements to the rotation lower bound, albeit at a growing cost in computational resources.

The paper is arranged as follows. Section 2 below defines the family of finite-dimensional optimization problems majorizing the moving sofa problem and develops the necessary theoretical ideas that set the ground for the computer-assisted proof scheme. In Section 3 we build on these results and introduce the main algorithm for deriving and certifying improved bounds, then prove its correctness. Section 4 discusses specific numerical examples illustrating the use of the algorithm, leading to a proof of Theorems 1 and 2. The Appendix describes `SofaBounds`, a software implementation we developed as a companion software application to this paper [9].

Acknowledgements. Yoav Kallus was supported by an Omidyar Fellowship at the Santa Fe Institute. Dan Romik thanks Greg Kuperberg for a key suggestion that was the seed from which Proposition 4 eventually grew, and John Sullivan, Joel Hass, Jesús De Loera, Maria Trnkova and Jamie Haddock for helpful discussions.

2 A family of geometric optimization problems

In this section we define a family of discrete-geometric optimization problems that we will show are in a sense approximate versions of the moving sofa problem for polygonal regions. Specifically, for each member of the family, the goal of the optimization problem will be to maximize the area of the intersection of translates of a certain finite set of polygonal regions in $\mathbb{R}^2$. It is worth noting that such optimization problems have been considered more generally in the computational geometry literature; see, e.g., [7, 11].

We start with a few definitions. Set

$$\begin{aligned} H &= \mathbb{R} \times [0,1], \\ V &= [0,1] \times \mathbb{R}, \\ L_{\text{horiz}} &= (-\infty, 1] \times [0,1], \\ L_{\text{vert}} &= [0,1] \times (-\infty, 1], \\ L_0 &= L_{\text{horiz}} \cup L_{\text{vert}}. \end{aligned}$$

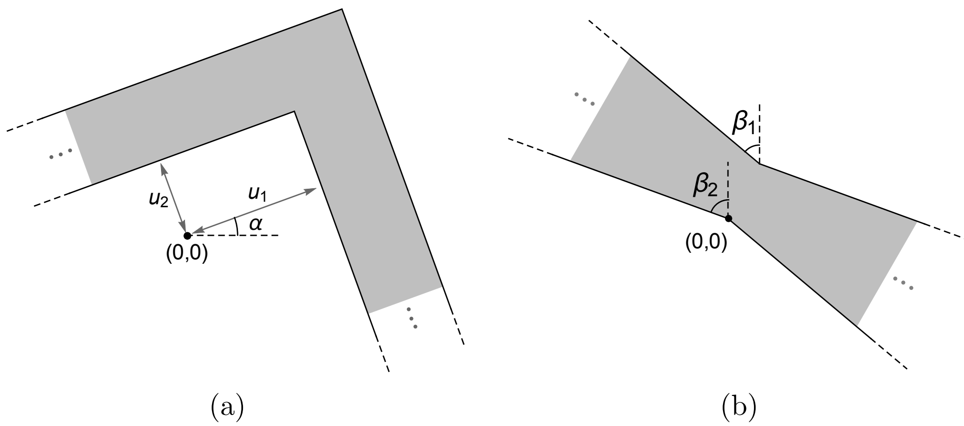

Figure 2: a) The rotated and translated L-shaped corridor $L_\alpha(u_1, u_2)$. (b) The "butterfly set" $B(\beta_1, \beta_2)$.

For an angle $\alpha \in [0, \pi/2]$ and a vector $\mathbf{u} = (u_1, u_2) \in \mathbb{R}^2$, denote

$$\begin{aligned} L_\alpha(\mathbf{u}) = \Big\{(x,y) \in \mathbb{R}^2 \,:\, u_1 \leq x\cos\alpha + y\sin\alpha \leq u_1 + 1 \\ \text{and} \quad -x\sin\alpha + y\cos\alpha \leq u_2 + 1\Big\} \\ \cup \Big\{(x,y) \in \mathbb{R}^2 \,:\, x\cos\alpha + y\sin\alpha \leq u_1 + 1 \\ \text{and } u_2 \leq -x\sin\alpha + y\cos\alpha \leq u_2 + 1\Big\}, \end{aligned}$$

For angles β_1, β_2, denote

$$\begin{aligned} B(\beta_1, \beta_2) = \Big\{(x,y) \in \mathbb{R}^2 \,:\, 0 \leq x\cos\beta_1 + y\sin\beta_1 \\ \text{and} \quad x\cos\beta_2 + y\sin\beta_2 \leq 1\Big\} \\ \cup \Big\{(x,y) \in \mathbb{R}^2 \,:\, x\cos\beta_1 + y\sin\beta_1 \leq 1 \\ \text{and} \quad 0 \leq x\cos\beta_2 + y\sin\beta_2\Big\}, \end{aligned}$$

Geometrically, $L_\alpha(\mathbf{u})$ is the L-shaped hallway L_0 translated by the vector $\mathbf{u}$ and then rotated around the origin by an angle of α; and $B(\beta_1, \beta_2)$, which we nickname a "butterfly set," is a set that contains a rotation of the vertical strip V around the origin by an angle β for all $\beta \in [\beta_1, \beta_2]$. See Fig. 2.

Next, let λ denote the area measure on $\mathbb{R}^2$ and let $\lambda^*(X)$ denote the maximal area of any connected component of $X \subset \mathbb{R}^2$. Given a vector $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$ of angles $0 < \alpha_1 < \ldots < \alpha_k < \pi/2$ and two additional angles $\beta_1, \beta_2 \in (\alpha_k, \pi/2]$ with $\beta_1 \le \beta_2$, define

$$g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(\mathbf{u}_1, \ldots, \mathbf{u}_k) = \lambda^* \left(H \cap \bigcap_{j=1}^{k} L_{\alpha_j}(\mathbf{u}_j) \cap B(\beta_1, \beta_2) \right) \quad (\mathbf{u}_1, \ldots, \mathbf{u}_k \in \mathbb{R}^2), \tag{2}$$

$$G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2} = \sup \left\{ g_{\boldsymbol{\alpha}}(\mathbf{u}_1, \ldots, \mathbf{u}_k) \,:\, \mathbf{u}_1, \ldots, \mathbf{u}_k \in \mathbb{R}^2 \right\}. \tag{3}$$

An important special case is $G_{\boldsymbol{\alpha}}^{\pi/2,\pi/2}$, which we denote simply as $G_{\boldsymbol{\alpha}}$. Note that $B(\beta_1, \pi/2) \cap H = H$, so in that case the inclusion of $B(\beta_1, \beta_2)$ in (2) is superfluous.

The problem of computing $G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$ is an optimization problem in $\mathbb{R}^{2k}$. The following lemma shows that the optimization can be performed on a compact subset of $\mathbb{R}^{2k}$ instead.

Lemma 3. *There exists a box $\Omega_{\boldsymbol{\alpha}}^{\beta_1,\beta_2} = [a_1, b_1] \times \ldots \times [a_{2k}, b_{2k}] \subset \mathbb{R}^{2k}$, with the values of a_i, b_i being explicitly computable functions of $\boldsymbol{\alpha}, \beta_1, \beta_2$, such that*

$$G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2} = \max \left\{ g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(\mathbf{u}) \,:\, \mathbf{u} \in \Omega_{\boldsymbol{\alpha}}^{\beta_1,\beta_2} \right\}. \tag{4}$$

Proof. We will show that any value of $g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(\mathbf{u})$ attained for some $\mathbf{u} \in \mathbb{R}^2$ is matched by a value attained inside a sufficiently large box. This will establish that $g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(\mathbf{u})$ is bounded from above; the fact that it attains its maximum follows immediately, since $g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(\mathbf{u})$ is easily seen to be a continuous function.

Start by observing that for every interval $[x_1, x_2]$ and $0 < \alpha < \pi/2$, there are intervals I and J such that if $(u, v) \in \mathbb{R}^2 \setminus I \times J$, the set $\big([x_1, x_2] \times [0, 1]\big) \cap L_\alpha(u, v)$ is either empty or is identical to $\big([x_1, x_2] \times [0, 1]\big) \cap L_\alpha(u', v')$ for some $(u', v') \in I \times J$. Indeed, it can be checked that this is correct with the choices

$$I = [x_1 \cos\alpha - 1, x_2 \cos\alpha + \sin\alpha],$$
$$J = [-x_2 \sin\alpha - 1, -x_1 \sin\alpha + \cos\alpha].$$

We now divide the analysis into two cases. First, if $\beta_2 < \pi/2$, then $H \cap B(\beta_1, \beta_2) \subseteq [-\tan\beta_2, \sec\beta_2] \times [0, 1]$. Therefore, if we define $\Omega_{\boldsymbol{\alpha}}^{\beta_1,\beta_2} = I_1 \times J_1 \times I_2 \times J_2 \times \cdots \times I_k \times J_k$, where for each $1 \le i \le k$, I_i and J_i are intervals I, J as described in the above observation as applied to the angle $\alpha = \alpha_i$, then $g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(u_1, \ldots, u_{2k})$ is guaranteed to attain its maximum on $\Omega_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$, since any value attained outside $\Omega_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$ is either zero or matched by a value attained inside $\Omega_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$.

Second, if $\beta_2 = \pi/2$, then $H \cap B(\beta_1, \beta_2) = H$. The optimization objective function $g_{\alpha}^{\beta_1,\beta_2}(\mathbf{u}_1, \ldots, \mathbf{u}_k)$ is invariant to translating all the rotated L-shaped hallways horizontally by the same amount (which corresponds to translating each variable $\mathbf{u}_j$ in the direction of the vector $(\cos\alpha_j, -\sin\alpha_j)$). Therefore, fixing an arbitrary $1 \le j \le k$, any value of $g_{\alpha}^{\beta_1,\beta_2}(\mathbf{u}_1, \ldots, \mathbf{u}_k)$ attained on $\mathbb{R}^{2k}$ is also attained at some point satisfying $\mathbf{u}_j = (0, u_{j,2})$. Furthermore, we can constrain $u_{j,2}$ as follows: first, when $u_{j,2} < -\tan\alpha_j - 1$, then $L_{\alpha_j}(0, u_{j,2}) \cap H$ is empty. Second, when $u_{j,2} > \sec\alpha_j$, then $L_{\alpha_j}(0, u_{j,2}) \cap H$ is the union of two disconnected components, one of which is a translation of $\mathrm{Rot}_{\alpha_j}(H) \cap H$ and the other is a translation of $\mathrm{Rot}_{\alpha_j}(V) \cap H$. Since the largest connected component of $H \cap \bigcap_{j=1}^{k} L_{\alpha_j}(\mathbf{u}_j)$ is contained in one of these two rhombuses, and since the translation of the rhombus does not affect the maximum attained area, we see that any objective value attained with $\mathbf{u}_j = (0, u_{j,2})$, where $u_{j,2} > \sec\alpha_j$, can also be attained with $\mathbf{u}_j = (0, \sec\alpha_j)$. So, we may restrict $\mathbf{u}_j \in I_j \times J_j$, where $I_j = \{0\}$ and $J_j = [-\tan\alpha_j - 1, \sec\alpha_j]$. Finally, when $\mathbf{u}_j \in I_j \times J_j$, we have $H \cap L_{\alpha_j}(\mathbf{u}) \subseteq [\csc\alpha_j, \sec\alpha_j] \times [0, 1]$, so we can repeat a procedure similar to the one used in the case $\beta = \pi/2$ above to construct intervals I_i and J_i for all $i \neq j$ to ensure that (4) is satisfied. $\square$

We now wish to show that the function $G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$ relates to the problem of finding upper bounds in the moving sofa problem. The idea is as follows. Consider a sofa shape S that moves around the corner while rotating continuously and monotonically (in a clockwise direction, in our coordinate system) between the angles 0 and $\beta \in [0, \pi/2]$. A key fact proved by Gerver [5, Th. 1] is that in the moving sofa problem it is enough to consider shapes being moved in this fashion. By changing our frame of reference to one in which the shape stays fixed and the L-shaped hallway L_0 is dragged around the shape while being rotated, we see (as discussed in [5, 12]) that S must be contained in the intersection

$$S_{\mathbf{x}} = L_{\text{horiz}} \cap \bigcap_{0 \le t \le \beta} L_t(\mathbf{x}(t)) \cap \Big(\mathbf{x}(\beta) + \mathrm{Rot}_\beta(L_{\text{vert}})\Big),$$

where $\mathbf{x} : [0, \beta] \to \mathbb{R}^2$ is a continuous path satisfying $\mathbf{x}(0) = (0, 0)$ that encodes the path by which the hallway is dragged as it is being rotated, and where $\mathrm{Rot}_\beta(L_{\mathrm{vert}})$ denotes L_{vert} rotated by an angle of β around $(0,0)$ (more generally, here and below we use the notation $\mathrm{Rot}_\beta(\cdot)$ for a rotation operator by an angle of β around the origin). We refer to such a path as a **rotation path**, or a **β-rotation path** when we wish to emphasize the dependence on β. Thus, the area of S is at most $\lambda^*(S_{\mathbf{x}})$, the maximal area of a connected component of $S_{\mathbf{x}}$, and conversely, a maximal area connected component of $S_{\mathbf{x}}$ is a valid moving sofa shape of area $\lambda^*(S_{\mathbf{x}})$. Gerver's result therefore implies that

$$\mu_{\mathrm{MS}} = \sup\Big\{\lambda^*(S_{\mathbf{x}}) \,:\, \mathbf{x} \text{ is a } \beta\text{-rotation path for some } \beta \in [0,\pi/2]\Big\}.$$

It is also convenient to define

$$\mu_*(\beta) = \sup\Big\{\lambda^*(S_{\mathbf{x}}) \,:\, \mathbf{x} \text{ is a } \beta\text{-rotation path}\Big\} \qquad (0 \le \beta \le \pi/2).$$

so that we have the relation

$$\mu_{\mathrm{MS}} = \sup_{0<\beta\le\pi/2} \mu_*(\beta). \tag{5}$$

Moreover, as Gerver pointed out in his paper, $\mu_*(\beta)$ is bounded from above by the area of the intersection of the horizontal strip H and its rotation by an angle β, which is equal to $\sec(\beta)$. Since $\mu_{\mathrm{MS}} \ge \mu_{\mathrm{G}}$, and $\sec(\beta) \ge \mu_{\mathrm{G}}$ if and only if $\beta \in [\beta_0, \pi/2]$, where we define $\beta_0 = \sec^{-1}(\mu_{\mathrm{G}}) \approx 63.22°$, we see that in fact

$$\mu_{\mathrm{MS}} = \sup_{\beta_0\le\beta\le\pi/2} \mu_*(\beta), \tag{6}$$

and furthermore, $\mu_{\mathrm{MS}} > \mu_*(\beta)$ for any $0 < \beta < \beta_0$, i.e., any moving sofa of maximal area has to rotate by an angle of at least β_0. (Gerver applied this argument to claim a slightly weaker version of this result in which the value of β_0 is taken as $\pi/3 = \sec^{-1}(2)$; see [5, p. 271]). Note that it has not been proved, but seems natural to conjecture, that $\mu_{\mathrm{MS}} = \mu_*(\pi/2)$—an assertion that would follow from Gerver's conjecture that the shape he discovered is the moving sofa shape of largest area, but may well be true even if Gerver's conjecture is false.

The relationship between our family of finite-dimensional optimization problems and the moving sofa problem is made apparent by the following result.

Proposition 4. *(i) For any $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$ and β with $0 < \alpha_1 < \ldots < \alpha_k \le \beta \le \pi/2$, we have*

$$\mu_*(\beta) \le G_{\boldsymbol{\alpha}}. \tag{7}$$

(ii) For any $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$ with $0 < \alpha_1 < \ldots < \alpha_k \leq \beta_0$, we have

$$\mu_{MS} \leq G_{\boldsymbol{\alpha}}. \tag{8}$$

(iii) For any $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$ and β_1, β_2 with $0 < \alpha_1 < \ldots < \alpha_k \leq \beta_1 < \beta_2 \leq \pi/2$, we have

$$\sup_{\beta \in [\beta_1, \beta_2]} \mu_*(\beta) \leq G_{\boldsymbol{\alpha}}^{\beta_1, \beta_2}. \tag{9}$$

Proof. Start by noting that, under the assumption that $0 < \alpha_1 < \ldots < \alpha_k \leq \beta \leq \pi/2$, if $\mathbf{x}$ is a β-rotation path then the values $\mathbf{x}(\alpha_1), \ldots, \mathbf{x}(\alpha_k)$ may potentially range over an arbitrary k-tuple of vectors in $\mathbb{R}^2$. It then follows that

$$\begin{aligned}
\mu_*(\beta) &= \sup \Big\{ \lambda^*(S_{\mathbf{x}}) \,:\, \mathbf{x} \text{ is a } \beta\text{-rotation path} \Big\} \\
&= \sup \left\{ \lambda^* \left(L_{\text{horiz}} \cap \bigcap_{0 \leq t \leq \beta} L_t(\mathbf{x}(t)) \cap \Big(\mathbf{x}(\beta) + \text{Rot}_\beta(L_{\text{vert}}) \Big) \right) \,: \right. \\
&\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad \left. \mathbf{x} \text{ is a } \beta\text{-rotation path} \vphantom{\bigcap_{0}} \right\} \\
&\leq \sup \left\{ \lambda^* \left(H \cap \bigcap_{j=1}^{k} L_{\alpha_j}(\mathbf{x}(\alpha_j)) \right) \,:\, \mathbf{x} \text{ is a } \beta\text{-rotation path} \right\} \\
&= \sup \left\{ \lambda^* \left(H \cap \bigcap_{j=1}^{k} L_{\alpha_j}(\mathbf{x}_j) \right) \,:\, \mathbf{x}_1, \ldots, \mathbf{x}_k \in \mathbb{R}^2 \right\} = G_{\boldsymbol{\alpha}}.
\end{aligned}$$

This proves claim (i) of the Proposition. If one further assumes that $\alpha_k \leq \beta_0$, (8) also follows immediately using (6), proving claim (ii).

The proof of claim (iii) follows a variant of the same argument used above; first, note that we may assume that $\beta_2 < \pi/2$, since the case $\beta_2 = \pi/2$

already follows from part (i) of the Proposition. Next, observe that

$$\mu_*(\beta) = \sup \left\{ \lambda^* \left(L_{\text{horiz}} \cap \bigcap_{0\le t\le\beta} L_t(\mathbf{x}(t)) \cap \Big(\mathbf{x}(\beta) + \mathrm{Rot}_\beta(L_{\text{vert}})\Big) \right) : \right.$$

$$\left. \mathbf{x} \text{ is a } \beta\text{-rotation path} \right\}$$

$$\le \sup_{\mathbf{x}_1,\ldots,\mathbf{x}_{k+1}\in\mathbb{R}^2} \left[\lambda^* \left(H \cap \bigcap_{j=1}^k L_{\alpha_j}(\mathbf{x}_j) \cap \Big(\mathbf{x}_{k+1} + \mathrm{Rot}_\beta(V)\Big) \right) \right].$$

$$= \sup_{\mathbf{y}_1,\ldots,\mathbf{y}_k\in\mathbb{R}^2} \left[\lambda^* \left(H \cap \bigcap_{j=1}^k L_{\alpha_j}(\mathbf{y}_j) \cap \mathrm{Rot}_\beta(V) \right) \right],$$

where the last equality follows by expressing $\mathbf{x}_{k+1}$ in the form $\mathbf{x}_{k+1} = a(1,0)+b(-\sin\beta,\cos\beta)$, making the substitution $\mathbf{x}_j = \mathbf{y}_j+a(1,0)$, and using the facts that $H + a(1,0) = H$ and $\mathrm{Rot}_\beta(V) + b(-\sin\beta,\cos\beta) = \mathrm{Rot}_\beta(V)$. Finally, as noted after the definition of $B(\beta_1,\beta_2)$, this set has the property that if $\beta \in [\beta_1,\beta_2]$ then $\mathrm{Rot}_\beta(V) \subset B(\beta_1,\beta_2)$. We therefore get for such β that

$$\mu_*(\beta) \le \sup_{\mathbf{y}_1,\ldots,\mathbf{y}_k\in\mathbb{R}^2} \left[\lambda^* \left(H \cap \bigcap_{j=1}^k L_{\alpha_j}(\mathbf{y}_j) \cap B(\beta_1,\beta_2) \right) \right] = G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2},$$

which finishes the proof. □

Example. In the case of a vector $\boldsymbol{\alpha} = (\alpha)$ with a single angle $0 < \alpha < \pi/2$, a simple calculation, which we omit, shows that

$$G_{(\alpha)} = \sec\alpha + \csc\alpha. \qquad (10)$$

Taking $\alpha = \pi/4$ and using Proposition 4(ii), we get that the result $\mu_{\mathrm{MS}} \le 2\sqrt{2}$, which is precisely Hammersley's upper bound for μ_{MS} mentioned in the introduction (indeed, this application of the proposition is essentially Hammersley's proof rewritten in our notation).

We conclude this section with a result that makes precise the notion that the optimization problems involved in the definition of $G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$ are finite-dimensional approximations to the (infinite-dimensional) optimization problem that is the moving sofa problem.

Theorem 5 (Convergence to the moving sofa problem)**.** *Let $\boldsymbol{\alpha}^{(n)}$, $\beta_1^{(n)}$, and $\beta_2^{(n)}$, $n = 1, 2, \ldots$, be a sequence of lists of angles specifying instances of the optimization problem* (3)*. Furthermore, assume that the differences between angles — $\alpha_{i+1}^{(n)} - \alpha_i^{(n)}$, $\beta_1^{(n)} - \alpha_{k^{(n)}}^{(n)}$, and $\beta_2^{(n)} - \beta_1^{(n)}$ — all approach zero as $n \to \infty$, that $\beta_1^{(n)} \le \beta \le \beta_2^{(n)}$, and that $0 < \beta \le \pi/2$. Then $\lim_{n\to\infty} G_{\boldsymbol{\alpha}^{(n)}}^{\beta_1^{(n)},\beta_2^{(n)}} = \mu_*(\beta)$.*

Proof. First note that the sequence $G_{\boldsymbol{\alpha}^{(n)}}^{\beta_1^{(n)},\beta_2^{(n)}}$ is bounded from above. This can be seen for example from the fact that the explicit expression (10) for $G_{(\alpha)}$ implies that for any $0 < \gamma_1 < \gamma_2 < \pi/2$, the quantity $\max(\sec\gamma_1 + \csc\gamma_1, \sec\gamma_2 + \csc\gamma_2)$ is a uniform upper bound for $G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$ that holds for all vectors $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_k)$ for which $\alpha_i \in [\gamma_1, \gamma_2]$ for some i. In particular, it follows that $\limsup_{n\to\infty} G_{\boldsymbol{\alpha}^{(n)}}^{\beta_1^{(n)},\beta_2^{(n)}}$ exists. Let n_m denote the indices of a subsequence converging to the limit superior.

Let P_n be a largest area connected component of

$$H \cap \bigcap_{j=1}^{k^{(n)}} L_{\alpha_j^{(n)}}\left(u_{2j-1}^{(n)}, u_{2j}^{(n)}\right) \cap B\left(\beta_1^{(n)}, \beta_2^{(n)}\right),$$

where $\left(u_1^{(n)}, \ldots, u_{2k^{(n)}}^{(n)}\right)$ is a point in $\mathbb{R}^{2k}$ where $g_{\boldsymbol{\alpha}^{(n)}}^{\beta_1^{(n)},\beta_2^{(n)}}$ attains its maximum. Let

$$\begin{aligned}
u^{(n)}(t) &= \max_{(x,y)\in P_n} x\cos t + y\sin t - 1, \\
v^{(n)}(t) &= \max_{(x,y)\in P_n} -x\sin t + y\cos t - 1, \\
T_n &= H \cap \bigcap_{t=0}^{\beta} L_t(u^{(n)}(t), v^{(n)}(t)) \cap \mathrm{Rot}_\beta(H).
\end{aligned}$$

Since (T_{n_m}) is a sequence of uniformly bounded compact sets, they have an accumulation point T_∞ in terms of the Hausdorff metric. Furthermore, since each T_n is an intersection $S_{\mathbf{x}}$ for some β-rotation path $\mathbf{x}$, so is T_∞. Since the Hausdorff distance of P_n to T_n approaches zero, T_∞ is also an accumulation point of (P_{n_m}), and is therefore connected. Since T_∞ is an intersection $S_{\mathbf{x}}$ for a β-rotation path and is connected, $\lambda(T) \le \mu_*(\beta)$. Since it is the accumulation point of P_{n_m}, $\lambda(T) = \limsup_{n\to\infty} G_{\boldsymbol{\alpha}^{(n)}}^{\beta_1^{(n)},\beta_2^{(n)}}$. Therefore, $\limsup_{n\to\infty} G_{\boldsymbol{\alpha}^{(n)}}^{\beta_1^{(n)},\beta_2^{(n)}} \le \mu_*(\beta)$.

We also have that $G_{\boldsymbol{\alpha}^{(n)}}^{\beta_1^{(n)},\beta_2^{(n)}} \geq \mu_*(\beta)$ from Proposition 4, so we conclude that $\lim_{n\to\infty} G_{\boldsymbol{\alpha}^{(n)}}^{\beta_1^{(n)},\beta_2^{(n)}} = \mu_*(\beta)$. $\square$

3 An algorithmic proof scheme for moving sofa area bounds

The theoretical framework we developed in the previous section reduces the problem of deriving upper bounds for μ_{MS} to that of proving upper bounds for the function $G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$. Since this function is defined in terms of solutions to a family of optimization problems in finite-dimensional spaces, this is already an important conceptual advance. However, from a practical standpoint it remains to develop and implement a practical, efficient algorithm for solving optimization problems in this class. Our goal in this section is to present such an algorithm and establish its correctness.

Our computational strategy is a variant of the **geometric branch and bound** optimization technique [13]. Recall from Lemma 3 that the maximum of the function $g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$ is attained in a box (a Cartesian product of intervals) $\Omega_{\boldsymbol{\alpha}}^{\beta_1,\beta_2} \subset \mathbb{R}^{2k}$. Our strategy is to break up $\Omega_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$ into sub-boxes. On each box E being considered, we will compute a quantity $\Gamma_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(E)$, which is an upper bound for $g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(\mathbf{u})$ that holds uniformly for all $\mathbf{u} \in E$. In many cases this bound will not be an effective one; in such a case the box will be subdivided into two further boxes E_1 and E_2, which will be inserted into a queue to be considered later. Other boxes lead to effective bounds and need not be considered further. By organizing the computation efficiently, practical bounds can be established in a reasonable time, at least for small values of k.

To make the idea precise, we introduce a few more definitions. Given two intervals $I = [a,b], J = [c,d] \subseteq \mathbb{R}$ and $\alpha \in [0,\pi/2]$, define

$$\widehat{L}_\alpha(I,J) = \bigcup_{u\in I, v\in J} L_\alpha(u,v).$$

Note that $\widehat{L}_\alpha(I,J)$ can also be expressed as a Minkowski sum of $L_\alpha(0,0)$ with the rotated rectangle $\mathrm{Rot}_\alpha(I\times J)$; in particular, it belongs to the class of planar sets known as **Nef polygons** (see the Appendix for further discussion of Nef polygons and their relevance to our software implementation of the algorithm). Now, for a box $E = I_1 \times \ldots \times I_{2k} \subset \mathbb{R}^2$, define

$$\Gamma_{\boldsymbol{\alpha}}(E) = \lambda^* \left(H \cap \bigcap_{j=1}^{k} \widehat{L}_{\alpha_j}(I_{2j-1}, I_{2j}) \cap B(\beta_1, \beta_2) \right). \tag{11}$$

Thus, by the definitions we have trivially that

$$\sup_{\mathbf{u} \in E} g_{\boldsymbol{\alpha}}(\mathbf{u}) \leq \Gamma_{\boldsymbol{\alpha}}^{\beta_1, \beta_2}(E). \tag{12}$$

Next, given a box $E = I_1 \times \ldots \times I_{2k}$ where $I_j = [a_j, b_j]$ as before, let

$$P_{\mathrm{mid}}(E) = \left(\frac{a_1 + b_1}{2}, \ldots, \ldots, \frac{a_{2k} + b_{2k}}{2} \right)$$

denote its midpoint. We also assume that some rule is given to associate with each box E a coordinate $i = \mathrm{ind}(E) \in \{1, \ldots, 2k\}$, called the **splitting index** of E. This index will be used by the algorithm to split E into two sub-boxes, which we denote by $\mathrm{split}_{i,1}(E)$ and $\mathrm{split}_{i,2}(E)$, and which are defined as

$$\begin{aligned} \mathrm{split}_{i,1}(E) &= I_1 \times \ldots \times I_{i-1} \times \left[a_i, \tfrac{1}{2}(a_i + b_i)\right] \times I_{i+1} \times \ldots I_{2k}, \\ \mathrm{split}_{i,2}(E) &= I_1 \times \ldots \times I_{i-1} \times \left[\tfrac{1}{2}(a_i + b_i), b_i\right] \times I_{i+1} \times \ldots I_{2k}. \end{aligned}$$

We assume that the mapping $E \mapsto \mathrm{ind}(E)$ has the property that, if the mapping $E \mapsto \mathrm{split}_{\mathrm{ind}(E),j}(E)$ is applied iteratively, with arbitrary choices of $j \in \{1, 2\}$ at each step and starting from some initial value of E, the resulting sequence of splitting indices $i_1, i_2, \ldots$ contains each possible coordinate infinitely many times. A mapping satisfying this assumption is referred to as a **splitting rule**.

The algorithm is based on the standard data structure of a **priority queue** [2] used to hold boxes that are still under consideration. Recall that in a priority queue, each element of the queue is associated with a numerical value called its priority, and that the queue realizes operations of pushing a new element into the queue with a given priority, and popping the highest priority element from the queue. In our application, the priority of each box E will be set to a value denoted $\Pi(E)$, where the mapping $E \mapsto \Pi(E)$ is given and is assumed to satisfy

$$\Pi(E) \geq \Gamma_{\boldsymbol{\alpha}}^{\beta_1, \beta_2}(E). \tag{13}$$

Aside from this requirement, the precise choice of mapping is an implementation decision. (A key point here is that setting $\Pi(E)$ *equal* to $\Gamma_{\boldsymbol{\alpha}}^{\beta_1, \beta_2}(E)$

is conceptually the simplest choice, but from the practical point of view of minimizing programming complexity and running time it may not be optimal; see the Appendix for further discussion of this point.) Note that, since boxes are popped from the queue to be inspected by the algorithm in decreasing order of their priority, this ensures that the algorithm pursues successive improvements to the upper bound it obtains in a greedy fashion.

The algorithm also computes a lower bound on $G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$ by evaluating $g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(\mathbf{u})$ at the midpoint of every box it processes and keeping track of the largest value observed. This lower bound is used to discard boxes in which it is impossible for the maximum to lie. The variable keeping track of the lower bound is initialized to some number ℓ_0 known to be a lower bound for $G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$. In our software implementation we used the value

$$\ell_0 = \begin{cases} 0 & \text{if } \beta_2 < \pi/2, \\ 11/5 & \text{if } \beta_2 = \pi/2, \end{cases}$$

this being a valid choice thanks to the fact that (by Proposition 4(iii)) $G_{\boldsymbol{\alpha}}^{\beta_1,\pi/2} \geq \mu_*(\pi/2) \geq \mu_{\mathrm{G}} = 2.2195\ldots > 2.2 = 11/5$. Note that simply setting $\ell_0 = 0$ in all cases would also result in a valid algorithm, but would result in a slight waste of computation time compared to the definition above.

With this setup, we can now describe the algorithm, given in pseudocode in Listing 1.

The next proposition is key to proving the algorithm's correctness.

Proposition 6. *Any box $\tilde{E}$ which is the highest priority box in the queue* `box_queue` *at some step satisfies*

$$\sup\{g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(\mathbf{u}) : \mathbf{u} \in \tilde{E}\} \leq G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2} \leq \Pi(\tilde{E}). \tag{14}$$

Proof. First, the lower inequality holds for *all* boxes in the queue, simply because the value $g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(\mathbf{u})$ for any $\mathbf{u} \in \Omega_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$ is a lower bound on its maximum over all $\mathbf{u} \in \Omega_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$.

Next, let Q_n denote the collection of boxes in the priority queue after n iterations of the `while` loop (with Q_0 being the initialized queue containing the single box $\Omega_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$), and let D_n denote the collection of boxes that were discarded (not pushed into the priority queue during the execution of the `if` clause inside the `for` loop) during the first n iterations of the `while` loop. Then we first note that for all n, the relation

$$\Omega_{\boldsymbol{\alpha}}^{\beta_1,\beta_2} = \bigcup_{E \in Q_n \cup D_n} E \tag{15}$$

```
box_queue ← an empty priority queue of boxes
initial_box ← box representing Ω_α^{β1,β2}, computed according to the
                function of α, β1, β2 described in Lemma 3
push initial_box into box_queue with priority Π(initial_box).
best_lower_bound_so_far ← the initial lower bound ℓ0
while true do
    pop highest priority element of box_queue into current_box
    current_box_lower_bound ← g_α^{β1,β2}(P_mid(current_box))
    best_upper_bound_so_far ← Π(current_box).
    if current_box_lower_bound > best_lower_bound_so_far then
        best_lower_bound_so_far ← current_box_lower_bound
    end if
    i ← ind(current_box)
    for j = 1, 2 do
        new_box ← split_{i,j}(current_box)
        if Π(new_box) ≥ best_lower_bound_so_far then
            push new_box into box_queue with priority
                    Π(new_box)
        end if
    end for
    Reporting point: print the values of best_upper_bound_so_far
                     and best_lower_bound_so_far.
end while
```

Listing 1: The algorithm for computing bounds for $G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$.

holds. Indeed, this is easily proved by induction on n: if we denote by X the highest priority element in Q_n, then during the $(n+1)$th iteration of the `while` loop, X is subdivided into two boxes $X = X_1 \cup X_2$, and each of X_1, X_2 is either pushed into the priority queue (i.e., becomes an element of Q_{n+1}) or discarded (i.e., becomes an element of D_{n+1}), so we have that $\Omega_{\boldsymbol{\alpha}}^{\beta_1,\beta_2} = \bigcup_{E\in Q_{n+1}\cup D_{n+1}} E$, completing the inductive step.

Second, note that for any box $X \in D_n$, since X was discarded during the kth iteration of the `while` loop for some $1 \leq k \leq n$, we have that $\Pi(X)$ is smaller than the value of `best_lower_bound_so_far` during that iteration. But `best_lower_bound_so_far` is always assigned a value of the form $g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(\mathbf{u})$ for some $\mathbf{u} \in R$ and is therefore bounded from above by $G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$, so we have established that

$$\Pi(X) < G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2} \qquad (X \in D_n). \tag{16}$$

The relations (4), (12), (13), (15), and (16) now imply that

$$\begin{aligned} G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2} &= \max\left\{ g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(\mathbf{u}) \,:\, \mathbf{u} \in \Omega_{\boldsymbol{\alpha}}^{\beta_1,\beta_2} \right\} \\ &= \max_{E\in Q_n\cup D_n} \left(\sup\left\{ g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(\mathbf{u}) \,:\, \mathbf{u} \in E \right\} \right) \leq \max_{E\in Q_n\cup D_n} \Gamma_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(E) \\ &\leq \max_{E\in Q_n\cup D_n} \Pi(E) = \max_{E\in Q_n} \Pi(E). \end{aligned}$$

Finally, $\max_{E\in Q_n} \Pi(E) = \Pi(\tilde{E})$, since $\tilde{E}$ was assumed to be the box with highest priority (that is, highest value of $\Pi(\cdot)$) among the elements of Q_n, so we get the upper inequality in (14), which finishes the proof. □

We immediately have the correctness of the algorithm as a corollary:

Theorem 7 (Correctness of the algorithm)**.** *Any value of the variable* `best_upper_bound_so_far` *reported by the algorithm is an upper bound for* $G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$.

Note that the correctness of the algorithm is not dependent on the assumption we made on the splitting index mapping $E \mapsto \mathrm{ind}(E)$ being a splitting rule. The importance of that assumption is explained by the following result, which also explains one sense in which assuming an equality in (13) rather than an inequality provides a benefit (of a theoretical nature at least).

Theorem 8 (Asymptotic sharpness of the algorithm)**.** *Assume that the priority mapping $E \mapsto \Pi(E)$ is taken to be*

$$\Pi(E) = \Gamma_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(E). \tag{17}$$

Then the upper and lower bounds output by the algorithm both converge to $G_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}$.

Proof. As one may easily check, the upper bound used in the calculation under the assumption (17), $\Gamma_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(E)$, approaches the actual supremum of $g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(\mathbf{u})$ over E as the diameter of E approaches zero. That is $|\Gamma_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(E) - \sup\{g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(\mathbf{u}) : \mathbf{u} \in E\}|$ is bounded by a function of the diameter of E that approaches zero when the diameter approaches zero. The same is true of the variation in each box, $|\sup\{g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(\mathbf{u}) : \mathbf{u} \in E\} - \inf\{g_{\boldsymbol{\alpha}}^{\beta_1,\beta_2}(\mathbf{u}) : \mathbf{u} \in E\}|$.

When using a valid splitting rule, the diameter of the leading box approaches zero as n approaches infinity, and Proposition 6 completes the proof. $\square$

As with the case of the choice of priority mapping and the value of the initial lower bound ℓ_0, the specific choice of splitting rule to use is an implementation decision, and different choices can lead to algorithms with different performance. A simple choice we tried was to use the index of the coordinate with the largest variation within E (i.e., the "longest dimension" of E). Another choice, which we found gives superior performance and is the rule currently used in our software implementation `SofaBounds`, is to let the splitting index be the value of i maximizing $\lambda(D_i \cap S(E))$, where $S(E)$ is the argument of λ^* in Eq. (11), and

$$D_i = \begin{cases} \displaystyle\bigcup_{u \in I_{2j-1}} \widehat{L}_{\alpha_j}(u, I_{2j}) \setminus \bigcap_{u \in I_{2j-1}} \widehat{L}_{\alpha_j}(u, I_{2j}) & \text{if } i = 2j-1, \\ \displaystyle\bigcup_{u \in I_{2j}} \widehat{L}_{\alpha_j}(I_{2j-1}, u) \setminus \bigcap_{u \in I_{2j}} \widehat{L}_{\alpha_j}(I_{2j-1}, u) & \text{if } i = 2j. \end{cases}$$

4 Explicit numerical bounds

We report the following explicit numerical bounds obtained by our algorithm, which we will then use to prove Theorems 1 and 2.

Theorem 9. *Define angles*

$$\begin{aligned}
\alpha_1 &= \sin^{-1}\tfrac{7}{25} \approx 16.26^\circ,\\
\alpha_2 &= \sin^{-1}\tfrac{33}{65} \approx 30.51^\circ,\\
\alpha_3 &= \sin^{-1}\tfrac{119}{169} \approx 44.76^\circ,\\
\alpha_4 &= \sin^{-1}\tfrac{56}{65} = \pi/2 - \alpha_2 \approx 59.59^\circ,\\
\alpha_5 &= \sin^{-1}\tfrac{24}{25} = \pi/2 - \alpha_1 \approx 73.74^\circ,\\
\alpha_6 &= \sin^{-1}\tfrac{60}{61} \approx 79.61^\circ,\\
\alpha_7 &= \sin^{-1}\tfrac{84}{85} \approx 81.2^\circ.
\end{aligned}$$

Then we have the inequalities

$$G_{(\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha_5)} \leq 2.37, \tag{18}$$

$$G^{\alpha_4,\alpha_5}_{(\alpha_1,\alpha_2,\alpha_3)} \leq 2.21, \tag{19}$$

$$G^{\alpha_5,\alpha_6}_{(\alpha_1,\alpha_2,\alpha_3)} \leq 2.21, \tag{20}$$

$$G^{\alpha_6,\alpha_7}_{(\alpha_1,\alpha_2,\alpha_3,\alpha_4)} \leq 2.21. \tag{21}$$

Proof. Each of the inequalities (18)–(21) is certified as correct using the `SofaBounds` software package by invoking the `run` command from the command line interface after loading the appropriate parameters. For (18), the parameters can be loaded from the saved profile file `thm9-bound1.txt` included with the package (see the Appendix below for an illustration of the syntax for loading the file and running the computation). Similarly, the inequalities (19), (20), (21) are obtained by running the software with the profile files `thm9-bound2.txt`, `thm9-bound3.txt`, and `thm9-bound4.txt`, respectively. Table 1 in the Appendix shows benchmarking results with running times for each of the computations. □

Proof of Theorem 1. For angles $0 \leq \beta_1 < \beta_2 \leq \pi/2$ denote

$$M(\beta_1,\beta_2) = \sup_{\beta_1 \leq \beta \leq \beta_2} \mu_*(\beta).$$

By (5), we have

$$\mu_{\mathrm{MS}} = M(0,\pi/2) = \max\Big(M(0,\alpha_5), M(\alpha_5,\pi/2)\Big). \tag{22}$$

By Proposition 4(iii), $M(0,\alpha_5)$ is bounded from above by $G^{\alpha_4,\alpha_5}_{(\alpha_1,\alpha_2,\alpha_3)}$, and by Proposition 4(i), $M(\alpha_5,\pi/2)$ is bounded from above by $G_{(\alpha_1,\alpha_2,\alpha_3,\alpha_4,\alpha_5)}$. Thus, combining (22) with the numerical bounds (18)–(19) proves (1). □

Proof of Theorem 2. Using the same notation as in the proof of Theorem 1 above, we note that

$$M(0,\alpha_7) = \max\Big(M(0,\alpha_4), M(\alpha_4,\alpha_5), M(\alpha_5,\alpha_6), M(\alpha_6,\alpha_7)\Big).$$

Now, by Gerver's observation mentioned after the relation (5), we have that $M(0,\alpha_4) \le \sec(\alpha_4) < \sec(\pi/3) = 2$. By Proposition 4(iii) coupled with the numerical bounds (19)–(21), the remaining three arguments $M(\alpha_4,\alpha_5)$, $M(\alpha_5,\alpha_6)$, and $M(\alpha_6,\alpha_7)$ in the maximum are all bounded from above by 2.21, so in particular we get that $M(0,\alpha_7) \le 2.21 < \mu_{\mathrm{G}} \approx 2.2195$. On the other hand, we have that

$$\mu_{\mathrm{MS}} = M(0,\pi/2) = \max\Big(M(0,\alpha_7), M(\alpha_7,\pi/2)\Big) \ge \mu_{\mathrm{G}}.$$

We conclude that $\mu_{\mathrm{MS}} = M(\alpha_7,\pi/2)$ and that $\mu_*(\beta) \le 2.21 < \mu_{\mathrm{MS}}$ for all $\beta < \alpha_7$. This proves that a moving sofa of maximal area has to undergo rotation by an angle of at least α_7, as claimed. $\square$

5 Concluding remarks

The results of this paper represent the first progress since Hammersley's 1968 paper [8] on deriving upper bounds for the area of a moving sofa shape. Our techniques also enable us to prove an improved lower bound on the angle of rotation a maximal area moving sofa shape must rotate through. Our improved upper bound of 2.37 on the moving sofa constant comes much closer than Hammersley's bound to the best known lower bound $\mu_{\mathrm{G}} \approx 2.2195$ arising from Gerver's construction, but clearly there is still considerable room for improvement in narrowing the gap between the lower and upper bounds. In particular, some experimentation with the initial parameters used as input for the `SofaBounds` software should make it relatively easy to produce further (small) improvements to the value of the upper bound.

More ambitiously, our hope is that a refinement of our methods—in the form of theoretical improvements and/or speedups in the software implementation, for example using parallel computing techniques—may eventually be used to obtain an upper bound that comes very close to Gerver's bound, thereby providing supporting evidence to his conjecture that the shape he found is the solution to the moving sofa problem. Some supporting evidence of this type, albeit derived using a heuristic algorithm, was reported in a recent paper by Gibbs [6]. Alternatively, a failure of our algorithm (or improved versions thereof) to approach Gerver's lower bound may provide clues that his conjecture may in fact be false.

Our methods should also generalize in a fairly straightforward manner to other variants of the moving sofa problem. In particular, in a recent paper [12], one of us discovered a shape with a piecewise algebraic boundary that is a plausible candidate to being the solution to the so-called **ambidextrous moving sofa problem**, which asks for the largest shape that can be moved around a right-angled turn *either to the left or to the right* in a hallway of width 1 (Fig. 3(a)). The shape, shown in Fig. 3(b), has an area given by the intriguing explicit constant

$$\mu_{\mathrm{R}} = \sqrt[3]{3+2\sqrt{2}} + \sqrt[3]{3-2\sqrt{2}} - 1 + \arctan\left[\frac{1}{2}\left(\sqrt[3]{\sqrt{2}+1} - \sqrt[3]{\sqrt{2}-1}\right)\right]$$
$$= 1.64495521\ldots$$

As with the case of the original (non-ambidextrous) moving sofa problem, the constant μ_{R} provides a lower bound on the maximal area of an ambidextrous moving sofa shape; in the opposite direction, any upper bound for the original problem is also an upper bound for the ambidextrous variant of the problem, which establishes 2.37 as a valid upper bound for that problem. Once again, the gap between the lower and upper bounds seems like an appealing opportunity for further work, so it would be interesting to extend the techniques of this paper to the setting of the ambidextrous moving sofa problem so as to obtain better upper bounds on the "ambidextrous moving sofa constant."

Appendix: The `SofaBounds` software

We implemented the algorithm described in Section 3 in the software package `SofaBounds` we developed, which serves as a companion package to this paper and whose source code is available to download online [9]. The package is a Unix command line tool written in `C++` and makes use of the open source computational geometry library `CGAL` [1]. All computations are done in the exact rational arithmetic mode supported by `CGAL` to ensure that the bounds output by the algorithm are mathematically rigorous. For this reason, the software only works with angles γ for which the vector $(\cos\gamma, \sin\gamma)$ has rational coordinates, i.e., is a rational point $(a/c, b/c)$ on the unit circle; clearly such angles are parametrized by Pythagorean triples (a, b, c) such that $a^2 + b^2 = c^2$, and it is using such triples that the angles are entered

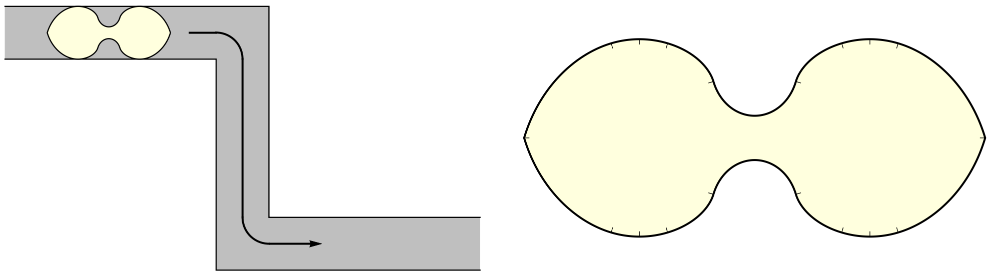

Figure 3: (a) The ambidextrous moving sofa problem involves maximization of the area of moving sofa shapes that can navigate a hallway with right-angled turns going both ways, as shown in the figure; (b) a shape discovered by Romik [12] that was derived as a possible solution to the ambidextrous moving sofa problem. The boundary of the shape is a piecewise algebraic curve; the tick marks in the figure delineate the transition points between distinct parts of the boundary.

into the program as input from the user. For example, to approximate an angle of 45 degrees, we used the Pythagorean triple $(119, 120, 169)$, which corresponds to an angle of $\sin^{-1}(119/169) \approx 44.76°$.

The software uses the **Nef polygon** geometric primitive implemented in `CGAL`; recall that Nef polygons are planar sets that can be obtained from a finite set of half-planes by applying set intersection and complementation operations. It is easy to see that all the planar sets manipulated by the algorithm belong to this family and can be readily calculated using elementary geometry and the features of `CGAL`'s Nef polygon sub-library [14].

Our implementation uses the priority rule

$$\Pi(E) = \lambda \left(H \cap \bigcap_{j=1}^{k} \widehat{L}_{\alpha_j}(I_{2j-1}, J_{2j}) \cap B(\beta_1, \beta_2) \right),$$

i.e., we use the total area of the intersection as the priority instead of the area of the largest connected component as in (11); this is slightly less ideal from a theoretical point of view, since Theorem 8 does not apply, but simplified the programming and in practice probably results in better computational performance.

The software runs our algorithm on a single Unix thread, since the parts of the CGAL library we used are not thread-safe; note however that the

Bound	Saved profile file	Num. of iterations	Computation time
(18)	`thm9-bound1.txt`	7,724,162	480 hours
(19)	`thm9-bound2.txt`	917	2 minutes
(20)	`thm9-bound3.txt`	26,576	1:05 hours
(21)	`thm9-bound4.txt`	140,467	6:23 hours

Table 1: Benchmarking results for the computations used in the proof of the bounds (18)–(21). The computations for (18) were performed on a 2.3 GHz Intel Xeon E5-2630 processor, and the computations for (19)–(21) were performed were performed on a 3.4 GHz Intel Core i7 processor.

nature of our algorithm lends itself fairly well to parallelization, so a multithreading or other parallelized implementation could yield a considerable speedup in performance, making it more practical to continue to improve the bounds in Theorems 1 and 2.

To illustrate the use of the software, Listing 2 shows a sample working session in which the upper bound 2.5 is derived for $G_{\boldsymbol{\alpha}}$ with

$$\boldsymbol{\alpha} = \left(\sin^{-1}\frac{33}{65}, \sin^{-1}\frac{119}{169}, \sin^{-1}\frac{56}{65}\right) \approx (30.51^\circ, 44.76^\circ, 59.49^\circ). \tag{23}$$

The numerical bounds (18)–(21) used in the proofs of Theorems 1 and 2 were proved using `SofaBounds`, and required several weeks of computing time on a desktop computer. Table 1 shows some benchmarking information, which may be useful to anyone wishing to reproduce the computations or to improve upon our results.

Additional details on `SofaBounds` can be found in the documentation included with the package.

```
Users/user/SofaBounds$ SofaBounds
SofaBounds version 1.0

Type "help" for instructions.

> load example-30-45-60.txt
File 'example-30-45-60.txt' loaded successfully.
> settings

Number of corridors: 3

Slope 1:         33       56       65      (angle: 30.5102 deg)
Slope 2:        119      120      169      (angle: 44.7603 deg)
Slope 3:         56       33       65      (angle: 59.4898 deg)
Minimum final slope: 1 0 1                 (angle: 90 deg)
Maximum final slope: 1 0 1                 (angle: 90 deg)

Reporting progress every: 0.01 decrease in upper bound
> run
<iterations=0>
<iterations=1 | upper bound=3.754 | time=0:00:00>
<iterations=7 | upper bound=3.488 | time=0:00:01>
<iterations=9 | upper bound=3.438 | time=0:00:01>
```

[... *54 output lines deleted* ...]

```
<iterations=1776 | upper bound=2.560 | time=0:08:43>
<iterations=2188 | upper bound=2.550 | time=0:10:48>
<iterations=2711 | upper bound=2.540 | time=0:13:23>
<iterations=3510 | upper bound=2.530 | time=0:18:18>
<iterations=4620 | upper bound=2.520 | time=0:24:54>
<iterations=6250 | upper bound=2.510 | time=0:34:52>
<iterations=8901 | upper bound=2.500 | time=0:50:45>
```

Listing 2: A sample working session of the `SofaBounds` software package proving an upper bound for $G_{\boldsymbol{\alpha}}$ with $\boldsymbol{\alpha}$ given by (23) . User commands are colored in blue. The session loads parameters from a saved profile file `example-30-45-60.txt` (included with the source code download package) and rigorously certifies the number 2.5 as an upper bound for $G_{\boldsymbol{\alpha}}$ (and therefore also for μ_{MS}, by Proposition 4(ii)) in about 50 minutes of computation time on a laptop with a 1.3 GHz Intel Core M processor.

References

[1] Multiple contributors. The Computational Geometry Algorithms Library. Online resource: `http://www.cgal.org`. Accessed June 15, 2017.

[2] T. H. Cormen, C. E. Leiserson, R. L. Rivest, C. Stein. Introduction to Algorithms (3rd edition). MIT Press, 2009.

[3] H. T. Croft, K. J. Falconer, R. K. Guy. Unsolved Problems in Geometry. Springer-Verlag, 1991, pp. 171–172.

[4] S. Finch. Mathematical Constants. Cambridge University Press, 2003.

[5] J. L. Gerver. On moving a sofa around a corner. *Geom. Dedicata* **42** (1992), 267–283.

[6] P. Gibbs. A computational study of sofas and cars. Preprint (2014), `http://vixra.org/pdf/1411.0038v2.pdf`.

[7] S. Har-Peled, S. Roy. Approximating the maximum overlap of polygons under translation. *Algorithmica* **78** (2017), 147–165.

[8] J. M. Hammersley. On the enfeeblement of mathematical skills by "Modern Mathematics" and by similar soft intellectual trash in schools and universities. *Bull. Inst. Math. Appl.* **4** (1968), 66–85.

[9] Y. Kallus, D. Romik. `SofaBounds`: A software package for computer-assisted proofs in the moving sofa problem. Online resource: `https://github.com/ykallus/SofaBounds`. Accessed June 15, 2017.

[10] L. Moser. Problem 66-11: Moving furniture through a hallway. *SIAM Rev.* **8** (1966), 381.

[11] D. M. Mount, R. Silverman, A. Y. Wu. On the area of overlap of translated polygons. Computer Vision and Image Understanding: CVIU, 64 (1996), 53–61.

[12] D. Romik. Differential equations and exact solutions in the moving sofa problem. To appear in *Exper. Math.*

[13] H. Ratschek, J. Rokne. New Computer Methods for Global Optimization. Ellis Horwood, 1988.

[14] Michael Seel. 2D Boolean Operations on Nef Polygons. In: *CGAL User and Reference Manual.* CGAL Editorial Board, 4.10 edition, 2017. Online resource: `http://doc.cgal.org/latest/Nef_2/index.html`. Accessed June 15, 2017.