axioms

MDPI

Article

An Algorithm Based on Compute Unified Device Architecture for Estimating Covering Functionals of Convex Bodies

Xiangyang Han, Senlin Wu * and Longzhen Zhang

School of Mathematics, North University of China, Taiyuan 030051, China; s202108097@st.nuc.edu.cn (X.H.); s202208059@st.nuc.edu.cn (L.Z.)
* Correspondence: wusenlin@nuc.edu.cn

Abstract: In Chuanming Zong's program to attack Hadwiger's covering conjecture, which is a longstanding open problem from Convex and Discrete Geometry, it is essential to estimate covering functionals of convex bodies effectively. Recently, He et al. and Yu et al. provided two deterministic global optimization algorithms having high computational complexity for this purpose. Since satisfactory estimations of covering functionals will be sufficient in Zong's program, we propose a stochastic global optimization algorithm based on CUDA and provide an error estimation for the algorithm. The accuracy of our algorithm is tested by comparing numerical and exact values of covering functionals of convex bodies including the Euclidean unit disc, the three-dimensional Euclidean unit ball, the regular tetrahedron, and the regular octahedron. We also present estimations of covering functionals for the regular dodecahedron and the regular icosahedron.

Keywords: Hadwiger's covering problem; covering functionals; CUDA; error estimation

MSC: 52C17; 52B11; 52B10; 52A15; 52A20

check for updates

Citation: Han, X.; Wu, S.; Zhang, L. An Algorithm Based on Compute Unified Device Architecture for Estimating Covering Functionals of Convex Bodies. *Axioms* **2024**, *13*, 132. https://doi.org/10.3390/axioms13020132

Academic Editor: Anna Maria Fino

Received: 12 January 2024
Revised: 14 February 2024
Accepted: 14 February 2024
Published: 19 February 2024

Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

A compact convex subset K of $\mathbb{R}^n$ having interior points is called a *convex body*, whose *boundary* and *interior* are denoted by bd K and int K, respectively. Let $\mathcal{K}^n$ be the set of convex bodies in $\mathbb{R}^n$. For each $K \in \mathcal{K}^n$, let $c(K)$ be the least number of translates of int K necessary to cover K. Regarding the least upper bound of $c(K)$ in $\mathcal{K}^n$, there is a long-standing conjecture:

Conjecture 1 (Hadwiger's covering conjecture)**.** *For each $K \in \mathcal{K}^n$, we have*

$$c(K) \leq 2^n;$$

the equality holds if and only if K is a parallelotope.

Classical results related to this conjecture can be found in [1,2]. While extensive research has been conducted (see e.g., refs. [3–7]), Conjecture 1 has only been conclusively resolved when $n = 2$.

For each $p \in \mathbb{Z}^+$, set $[p] = \{i \in \mathbb{Z}^+ \mid 1 \leq i \leq p\}$.

Let $K \in \mathcal{K}^n$. A set having the form $\lambda K + x$, where $\lambda \in (0,1)$ and $x \in \mathbb{R}^n$, is called a *smaller homothetic copy* of K. According to Theorem 34.3 in [1], $c(K)$ equals the least number of smaller homothetic copies of K required to cover K. Clearly, $c(K) \leq p$ for some $p \in \mathbb{Z}^+$ if and only if $\Gamma_p(K) < 1$, where

$$\Gamma_p(K) := \min\left\{\gamma > 0 \mid \exists\{x_i \mid i \in [p]\} \subseteq \mathbb{R}^n \text{ s.t. } K \subseteq \bigcup_{i\in[p]} (\gamma K + x_i)\right\}.$$

For each $p \in \mathbb{Z}^+$, the map

$$\begin{aligned} \Gamma_p(\cdot):\ \mathcal{K}^n &\to [0,1] \\ K &\mapsto \Gamma_p(K) \end{aligned}$$

is an affine invariant and is called the *covering functional with respect to* p. Let $K \in \mathcal{K}^n$ and $p \in \mathbb{Z}^+$. A set C of p points satisfying

$$K \subseteq \Gamma_p(K)K + C,$$

is called a *p-optimal configuration of* K.

In [8], Chuanming Zong proposed the first program based on computers to tackle Conjecture 1 via estimating covering functionals. Two different algorithms have been designed for this purpose. The first one is introduced by Chan He et al. (cf. [9]) based on the geometric branch-and-bound method (cf. [10]). The algorithm is implemented in two parts. The first part uses geometric branch-and-bound methods to estimate $\Gamma(K, C)$, where

$$\Gamma(K, C) = \min\{\gamma > 0 \mid K \subseteq C + \gamma K\}.$$

The second part also uses geometric branch-and-bound methods to estimate $\Gamma_p(K)$. When $n \geq 3$, computing $\Gamma(K, C)$ and $\Gamma_p(K)$ in this way exhibits a significantly high computational complexity. The other is introduced by Man Yu et al. (cf. [11]) based on the relaxation algorithm. Let $S \subseteq K$ be a discretization of K, P_K be a set containing a p-optimal configuration of K, and $V \subseteq P_K$. They transformed the problem of covering S by smaller homothetic copies of K into a vertex p-center problem, and showed that the solution of the corresponding vertex p-center problem is a good approximation of $\Gamma_p(K)$ by proving

$$\Gamma_p(S) - \varepsilon_1 \leq \Gamma_p(K) \leq \Gamma_p(S) + \varepsilon_2,$$

where

$$\Gamma_p(S) = \min\left\{ \gamma > 0 \mid \exists\{x_i \mid i \in [p]\} \subseteq \mathbb{R}^n \text{ s.t. } S \subseteq \bigcup_{i\in[p]} (\gamma K + x_i) \right\},$$

and ε_1 and ε_2 are two positive numbers satisfying

$$K \subseteq S + \varepsilon_1 K \text{ and } P_K \subseteq V + \varepsilon_2 K.$$

Clearly, finer discretizations of K are required to obtain more accurate estimates of $\Gamma_p(K)$, which will lead to higher computational complexity.

In this paper, we propose an algorithm utilizes Compute Unified Device Architecture (CUDA) and stochastic global optimization methods to accelerate the process of estimating $\Gamma_p(K)$. CUDA is a parallel computing platform, particularly well-suited for handling large-scale computational tasks by performing many computations in parallel (cf. [12]). When discretizing convex bodies, CUDA provides a natural discretization method and enables parallel computation for all discretized points, thereby accelerating the execution of algorithms. As show in Section 2, when calculating $\Gamma(S, C)$ for some $C \subseteq \mathbb{R}^n$, we need to obtain the maximum dissimilarity between a point in S and its closest point in C. The reduction technique provided by CUDA, which typically involves performing a specific operation on all elements in an array (summation, finding the maximum, finding the minimum, etc., cf. [13]), enables the efficient computation of $\Gamma(S, C)$. When facing large-scale optimization problems, stochastic algorithms have the capability to produce high-quality solutions in a short amount of time.

In Section 2, the problem of estimating $\Gamma_p(K)$ is transformed into a minimization optimization problem, and an error estimation is provided. Using ideas mentioned in [9], an algorithm based on CUDA for $\Gamma_p(S)$ is designed in Section 3. Results of computational experiments showing the effectiveness of our algorithm are presented in Section 4.

2. Covering Functional and Error Estimation

As in [9], we put $Q_n = [-1,1]^n$. For each $K \in \mathcal{K}^n$, let

$$\alpha(K) = \max\{\alpha > 0 \mid \exists T \in \mathcal{A}^n \text{ s.t. } \alpha Q_n \subseteq T(K) \subseteq Q_n\},$$

where $\mathcal{A}^n$ is the set of all non-singular affine transformations on $\mathbb{R}^n$. We can apply an appropriate affine transformation, if necessary, to ensure that

$$\alpha(K)Q_n \subseteq K \subseteq Q_n.$$

Remark 1. *In general, it is not easy to calculate $\alpha(K)$. If K is symmetric about the origin, then $1/\alpha(K)$ is the Banach–Mazur distance between K and Q_n. By Proposition 37.6 in [14], when K is the n-dimensional Euclidean unit ball B_2^n, $1/\alpha(K) = \sqrt{n}$. For our purpose, it will be sufficient to choose an α, as large as possible, such that $\alpha Q_n \subseteq K \subseteq Q_n$.*

Definition 1 (cf. [11])**.** *Let $K \in \mathcal{K}^n$. For $k, c \in \mathbb{R}^n$, the number*

$$d(k,c) = \min\{\gamma \geq 0 \mid k \in \gamma K + c\}. \tag{1}$$

is called the dissimilarity *of k and c.*

If $K = B_2^n$, the dissimilarity between any two points is precisely the Euclidean distance between these two points. In general, the dissimilarity is not symmetric. If K is an n-dimensional convex polytope, the dissimilarity between any two points can be computed by Lemma 1.

Lemma 1. *Let $K \in \mathcal{K}^n$ be an n-dimensional convex polytope with $o \in$ int K determined by*

$$K = \{x \in \mathbb{R}^n \mid Ax \leq B\}, \tag{2}$$

where A is an m-by-n matrix and B is an m-dimensional column vector whose elements are all 1. For any $u, c \in \mathbb{R}^n$, we have

$$d(u,c) = \max_{1 \leq i \leq m} p_i, \tag{3}$$

where $(p_i)_{m \times 1} = A(u-c)$.

Proof. Since K is bounded, $\max\limits_{1 \leq i \leq m} p_i \geq 0$. Let

$$\gamma_0 = \min\{\gamma \geq 0 \mid u \in c + \gamma K\}.$$

For positive γ, we have

$$u - c \in \gamma K \iff \frac{1}{\gamma}(u-c) \in K \iff \frac{1}{\gamma}A(u-c) \leq B \iff A(u-c) \leq \gamma B.$$

Since $B = (1, 1 \ldots, 1)^T$, we have $\max\limits_{1 \leq i \leq m} p_i \leq \gamma$, which implies that

$$\max_{1 \leq i \leq m} p_i \leq \gamma_0 = d(u,c). \tag{4}$$

Since $p_i \leq \max\limits_{1 \leq j \leq m} p_j$, $\forall i \in [m]$, we have $A(u-c) \leq \left(\max\limits_{1 \leq i \leq m} p_i\right) B$. Thus,

$$u - c \in \left(\max_{1 \leq i \leq m} p_i\right) K,$$

which implies that

$$\max_{1 \leq i \leq m} p_i \geq \gamma_0 = d(u,c). \tag{5}$$

The desired equality (3) follows from (4) and (5). □

Remark 2. *Clearly, each convex polytope K containing the origin 0 in its interior can be represented as* (2).

Let $i \geq 2$ and $j \in [n]$ be two integers. We denote by $p_j(x)$ the j-th coordinate of a point $x \in \mathbb{R}^n$ and

$$S_i = \left\{ (\gamma_1, \ldots, \gamma_n) \in \mathbb{R}^n \mid \gamma_j \in \left\{ -1 + \frac{2k}{i-1} \mid k \in [i-1] \cup \{0\} \right\}, \forall j \in [n] \right\}.$$

Lemma 2. $Q_n \subseteq S_i + \frac{1}{i-1} Q_n$.

Proof. Suppose that $x = (\beta_1, \ldots, \beta_n) \in Q_n$. Let $y \in \mathbb{R}^n$ be the point satisfying

$$p_j(y) = \begin{cases} \frac{\lfloor \beta_j(i-1) \rfloor}{i-1}, & \lfloor \beta_j(i-1) \rfloor + i - 1 \text{ is even}, \\ \frac{\lfloor \beta_j(i-1) \rfloor + 1}{i-1}, & \lfloor \beta_j(i-1) \rfloor + i - 1 \text{ is odd}, \end{cases} \quad \forall j \in [n].$$

Then, $y \in S_i$. Let $j \in [n]$. If $\lfloor \beta_j(i-1) \rfloor + i - 1$ is even, then

$$|\beta_j - p_j(y)| = \frac{\beta_j(i-1) - \lfloor \beta_j(i-1) \rfloor}{i-1} \leq \frac{1}{i-1}.$$

Otherwise, we have

$$|\beta_j - p_j(y)| = \frac{|\beta_j(i-1) - \lfloor \beta_j(i-1) \rfloor - 1|}{i-1} \leq \frac{1}{i-1}.$$

It follows that $\|x - y\|_\infty \leq \frac{1}{i-1}$. Therefore,

$$x \in y + \frac{1}{i-1} Q_n \subseteq S_i + \frac{1}{i-1} Q_n. \quad \square$$

Theorem 1. *Let $\alpha > 0$ and $i \geq 2$ be an integer. If $K \in \mathcal{K}^n$ satisfies $\alpha Q_n \subseteq K \subseteq Q_n$, then*

$$K \subseteq \left(S_i \cap \left(\frac{1 + \alpha(i-1)}{\alpha(i-1)} K \right) \right) + \frac{1}{i-1} Q_n.$$

Proof. By Lemma 2, we have

$$Q_n \subseteq S_i + \frac{1}{i-1} Q_n.$$

It follows that

$$K \subseteq \left(S_i + \frac{1}{i-1} Q_n \right) \cap K.$$

Let $x \in K$. There exists a point $p \in S_i$ such that

$$x \in p + \frac{1}{i-1} Q_n.$$

Thus,

$$p \in x - \frac{1}{i-1} Q_n \subseteq K + \frac{1}{i-1} Q_n \subseteq K + \frac{1}{\alpha(i-1)} K = \frac{1 + \alpha(i-1)}{\alpha(i-1)} K.$$

Therefore, we have

$$x \in \left(S_i \cap \left(\frac{1 + \alpha(i-1)}{\alpha(i-1)} K \right) \right) + \frac{1}{i-1} Q_n. \quad \square$$

Let $\alpha > 0$, $i \geq 2$ be an integer, K be a convex body satisfying $\alpha Q_n \subseteq K \subseteq Q_n$,

$$S = S_i \cap \left(\frac{1 + \alpha(i-1)}{\alpha(i-1)} K \right), \tag{6}$$

and $C \subseteq \mathbb{R}^n$. Put

$$\Gamma(S, C) = \inf\{\gamma > 0 \mid S \subseteq C + \gamma K\},$$

and

$$\Gamma_p(S) = \min\{\Gamma(S, C) \mid C \text{ contains at most } p \text{ points}\}.$$

Proposition 1. *Let K, α, i, S be as above, $p \in \mathbb{Z}^+$. Then,*

$$\Gamma_p(K) \leq \Gamma_p(S) + \frac{1}{\alpha(i-1)}. \tag{7}$$

Proof. By Theorem 1, we have

$$K \subseteq S + \frac{1}{i-1} Q_n \subseteq S + \frac{1}{\alpha(i-1)} K.$$

Let C' be a p-element subset of $\mathbb{R}^n$ such that $S \subseteq C' + \Gamma_p(S)K$. We have

$$K \subseteq S + \frac{1}{\alpha(i-1)} K \subseteq C' + \Gamma_p(S)K + \frac{1}{\alpha(i-1)} K,$$

which completes the proof. □

3. An Algorithm Based on CUDA for $\Gamma_p(S)$

Let $K \in \mathcal{K}^n$, $p \in \mathbb{Z}^+$, and C be a set of p points. First, we use CUDA to obtain S defined by (6) and compute the minimum dissimilarity from each point in S to C. Then, we employ a CUDA-based reduction algorithm to obtain $\Gamma(S, C)$. Finally, we use different stochastic global optimization algorithms to estimate $\Gamma_p(S)$ and select an appropriate optimization algorithm through comparison. Figure 1 shows the overall framework of the algorithm.

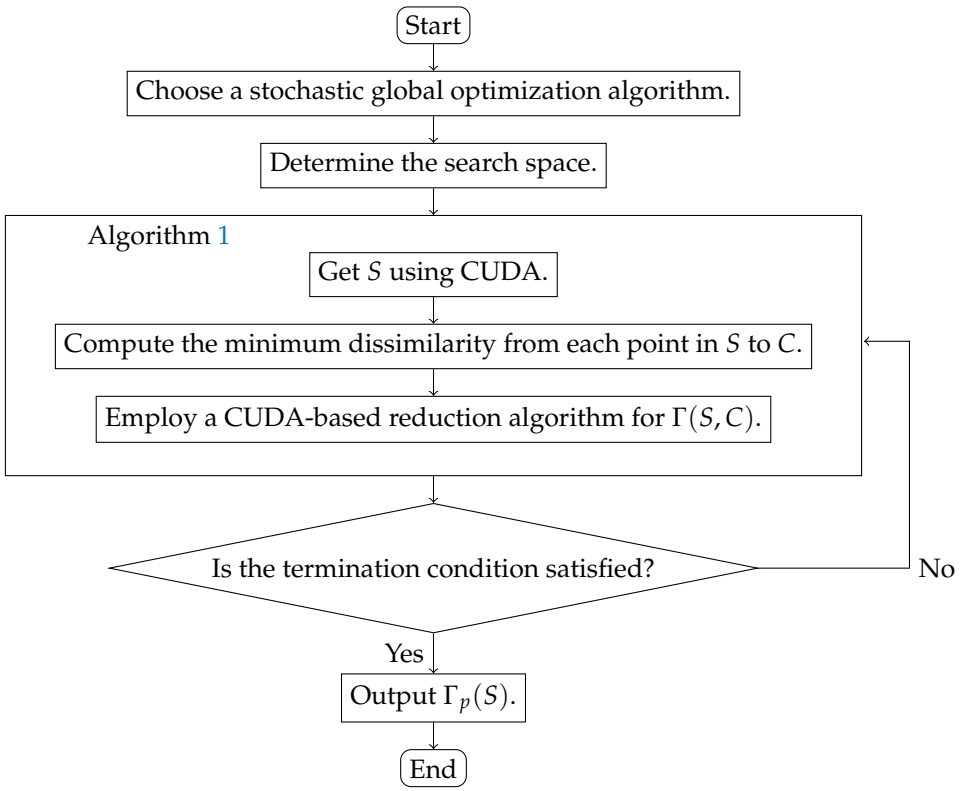


Figure 1. The overall framework.

3.1. An Algorithm Based on CUDA for $\Gamma(S, C)$

CUDA organizes threads into a hierarchical structure consisting of grids, blocks, and threads. The grid is the highest-level organization of threads in CUDA, and a grid represents a collection of blocks. A block, identified by a unique block index within its grid, is a group of threads that can cooperate with each other and share data using shared memory. Threads are organized within blocks, and each thread is identified by a unique thread index within its block. The number of blocks and threads per block can be specified when launching a CUDA kernel. The grid and block dimensions can be one-dimensional, two-dimensional, or three-dimensional, depending on the problem to address. For more information about CUDA, we refer to [15–18].

The organization of threads within blocks and grids provides a natural way to discretize Q_3. First, we discretize $[-1,1] \times [-1,1] \times \{0\}$ into a set P of (gridDim.x) $\times$ (gridDim.y) points. Each point p in P corresponds to a block B_p in CUDA. And B_p contains a collection of blockDim.x threads, each one of which corresponds to a point in $p + \{0\} \times \{0\} \times [-1,1]$. See Figure 2, where gridDim.x, girdDim.y, and blockDim.x are set to be 5.

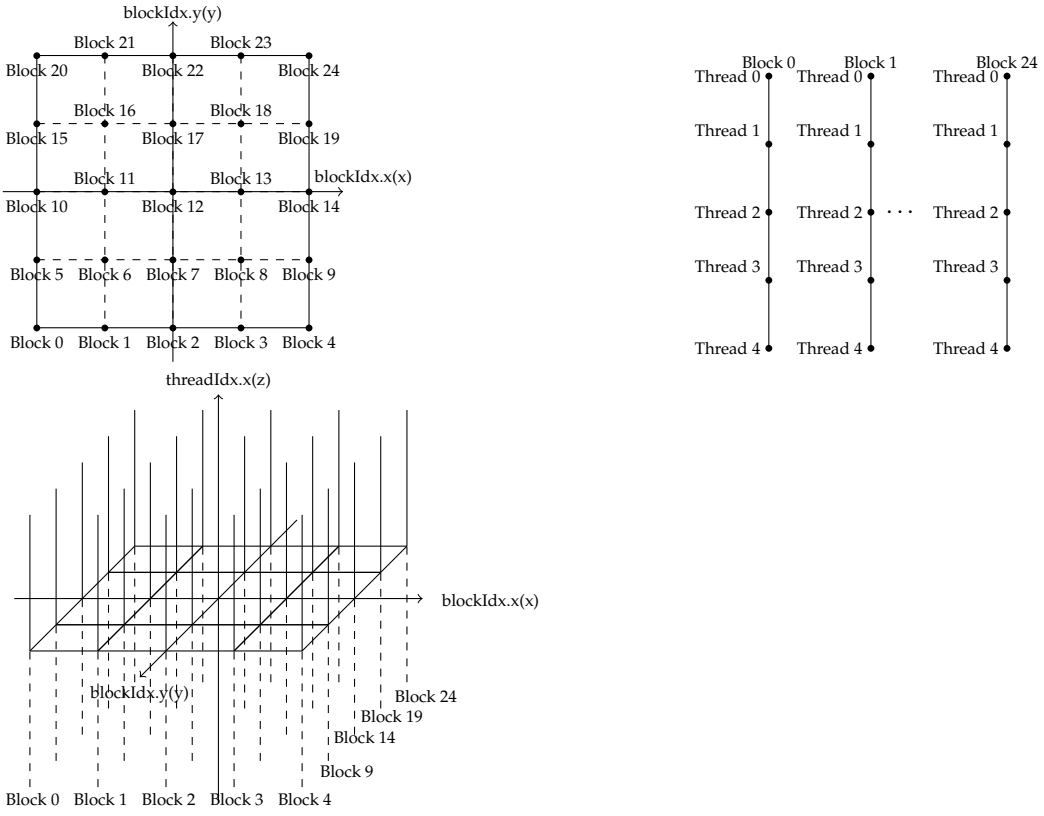


Figure 2. Discretize $[-1,1]^3$ with CUDA.

Let T be the set of all threads invoked by CUDA. Then, the cardinality of T is

$$\text{gridDim.x} \times \text{gridDim.y} \times \text{blockDim.x}.$$

For $i \in [\text{gridDim.x} - 1] \cup \{0\}$, $j \in [\text{gridDim.y} - 1] \cup \{0\}$, and $k \in [\text{blockDim.x} - 1] \cup \{0\}$, there is a thread $t := t(i,j,k) \in T$ indexed by

$$(j \times \text{gridDim.x} + i) \times \text{blockDim.x} + k,$$

which corresponds to the point

$$p_t = \left(-1 + \frac{2i}{\text{gridDim.x} - 1}, -1 + \frac{2j}{\text{gridDim.y} - 1}, -1 + \frac{2k}{\text{blockDim.x} - 1}\right) \in Q_3.$$

Put

$$S = \left\{ t \in T \mid p_t \in \left(1 + \frac{1}{\alpha(\text{gridDim.x} - 1)}\right) K \right\},$$

where α is a positive number satisfying $\alpha Q_3 \subseteq K \subseteq Q_3$. For each $t \in S$, denote by $d(p_t, C)$ the minimum dissimilarity from point p_t to C, i.e.,

$$d(p_t, C) = \min\{d(p_t, c) \mid c \in C\}.$$

If $t \notin S$, we set $d(p_t, C) = 0$. The CUDA thread corresponding to $t \in T$ computes $d(p_t, C)$. Then, a CUDA-based reduction algorithm will be invoked to obtain $\max_{t \in T} d(p_t, C)$.

The idea of the reduction algorithm based on CUDA is to divide the original data into multiple blocks, and then perform a local reduction operation on each block to obtain the local reduction result, and, finally, a global reduction operation is performed on the local reduction results to obtain the final reduction result (cf. [19]).

Algorithm 1 with parameters K, C, p, blockDim.x, gridDim.x, gridDim.y, and α calculates $\Gamma(S, C)$. It is more efficient than the geometric branch-and-bound approach proposed in [9]. For example, take $K = B_2^3$, $p = 6$,

$$\begin{aligned} C_1 = \Big\{ & (0.125, 0.125, 0.125), (-0.125, -0.125, -0.125), (0.25, 0.25, 0.25), \\ & (-0.25, -0.25, -0.25), (0.5, 0.5, 0.5), (-0.5, -0.5, -0.5), \\ & (0.0375, 0.0375, 0.0375), (-0.0375, -0.0375, -0.0375) \Big\}, \text{ and} \\ C_2 = \Big\{ & (0, 0.57, -0.38), (0.074, -0.185, 0.618), (-0.05, -0.66, 0.13), \\ & (0.32, -0.25, -0.58), (-0.52, -0.15, -0.38), (-0.587, 0.123, 0.22), \\ & (0.662, -0.024, 0.013), (0.136, 0.589, 0.319) \Big\}. \end{aligned}$$

Algorithm 1 An algorithm based on CUDA to compute $\Gamma(S, C)$

Require: a convex body K, a set C of p points in $\mathbb{R}^n$, a positive number α, blockDim.x, gridDim.x and gridDim.y
Ensure: *max* as an estimation of $\Gamma(S, C)$

```
Host and device allocate memory, initialize and copy host data to device
bid ← blockIdx.y * gridDim.x + blockIdx.x
t ← bid * blockDim.x + threadIdx.x
x[t] ← −1 + 2/(gridDim.x − 1) * blockIdx.x
y[t] ← −1 + 2/(gridDim.y − 1) * blockIdx.y
z[t] ← −1 + 2/(blockDim.x − 1) * threadIdx.x
p[t] ← (x[id], y[id], z[id])
if (p[t] ∈ (1 + 1/(α(gridDim.x−1)))K then:
    Calculate d(p[t], C) by Lemma 1
    P[t] ← d(p[t], C)
else
    P[t] ← 0
end if
sData[threadIdx.x] ← P[t]
Using the reduction algorithm to find the maximum value
Dd[blockIdx.x] ← sData[0]
Copy the final reduction result to the host, D[0] ← Dd[blockIdx.x]
max ← D[0]
return max
```

Algorithm 1 yields good estimations of $\Gamma_p(S, C_1)$ and $\Gamma_p(S, C_2)$ much faster, as seen in Table 1. Both algorithms run on a computer equipped with an AMD Ryzen 9 3900X

12-core processor and the NVIDIA A4000 graphics processor. For Algorithm 1, we take gridDim.x = gridDim.y = blockDim.x = 1024, and the accuracy is given by Proposition 1. For the geometric branch-and-bound algorithm, we set the relative accuracy ε to be 0.0001 (cf. [9] for the usage of the relative accuracy). The execution time of the geometric branch-and-bound approach exhibits substantial variability among different Cs, whereas the algorithm based on CUDA shows relatively consistent execution times across various cases.

Table 1. Comparison of results between the algorithm based on CUDA and the geometric branch-and-bound approach.

Algorithm	Algorithm 1	The Geometric Branch-and-Bound Approach
accuracy	$\frac{\sqrt{3}}{1023}$	0.0001
$\Gamma(S, C_1)$	1.002750...	1.002089...
Time	42.258720 ms	32.594993 s
$\Gamma(S, C_2)$	0.748103...	0.748099...
Time	42.232449 ms	0.303316 s

3.2. Different Stochastic Global Optimization Algorithms for $\Gamma_p(S)$

We choose to employ stochastic global optimization algorithms for several reasons. In the program proposed by Chuanming Zong (cf. [8]), after appropriately selecting a positive real number β and constructing a β-net $\mathcal{N}$ for $\mathcal{K}^n$ endowed with the Banach–Mazur metric, we only need to verify that $\Gamma_{2^n}(K) \leq c_n$ holds for each $K \in \mathcal{N}$, where c_n is a reasonably accurate estimate of the least upper bound of $\Gamma_{2^n}(K)$. For this purpose, we do not need to determine exact values of covering functionals of convex bodies in $\mathcal{N}$. Stochastic global optimization algorithms demonstrate a low time complexity and high algorithmic efficiency. Moreover, based on the results presented in Table 2, it is evident that stochastic global optimization algorithms provide satisfactory estimates for covering functionals.

Table 2. Numerical estimations of $\Gamma_p(B_2^3)$.

i	LB	UB	p	$\Gamma_p(S_i)$	Ranges of $\Gamma_p(B_2^3)$	Known Exact Value
1024	−2	2	4	0.942359...	$\leq 0.9441\ldots$	0.943... [8]
1024	−2	2	5	0.894146...	$\leq 0.8959\ldots$	0.894... [8]
1024	−2	2	6	0.816220...	$\leq 0.8180\ldots$	0.816... [8]
1024	−2	2	7	0.777139...	$\leq 0.7789\ldots$	0.778... [8]
1024	−2	2	8	0.744846...	$\leq 0.7466\ldots$	

The NLopt (Non-Linear Optimization) library is a rich collection of optimization routines and algorithms, which provides a platform-independent interface for global and local optimization problems (cf. [20]). Algorithms in the NLopt library are partitioned into four categories: non-derivative-based global algorithms, derivative-based global algorithms, non-derivative-based local algorithms, and derivative-based local algorithms. We use several non-derivative-based stochastic global algorithms here. All global optimization algorithms require bound constraints to be specified as optimization parameters (cf. e.g., ref. [21]).

The following is the framework of a stochastic optimization algorithm based on NLopt.

- Define the objective function and boundary constraints.
- Declare an optimizer for NLopt.
- Set algorithm and dimension.
- Set termination conditions. NLopt provides different termination condition options including: value tolerance, parameter tolerance, function value stop value, iteration number, and time.

Proposition 2. *If* $K \in \mathcal{K}^n$, $K \subseteq Q_n$, *and* $(x + K) \cap K \neq \emptyset$, *then* $x \in 2Q_n$.

Proof. Suppose that $y \in (x+K) \cap K$. Then $z := y - x \in K \subseteq Q_n$. Thus,

$$\frac{1}{2}x = \frac{1}{2}y + \frac{1}{2}(-z) \in Q_n,$$

which shows that $x \in 2Q_n$. □

Remark 3. *By Proposition 2, when $K \subseteq Q_n$, we only need to search $2Q_n$ for points in a p-optimal configuration of K.*

We utilized different stochastic global optimization algorithms to choose a more efficient one. Optimization algorithms under consideration include Controlled Random Search with local mutation (GN_CRS2_LM) (cf. [22]), evolutionary strategy (GN_ESCH) (cf. [23]), and evolutionary constrained optimization (GN_ISRES) (cf. [24]). See Algorithm 2.

Algorithm 2 A stochastic optimization algorithm for $\Gamma_p(S)$ based on NLopt.

Require: K, C, p, blockDim.x, gridDim.x, gridDim.y, an estimation $\Gamma_p(K) \leq \gamma$, lower bound LB and upper bound UB of the search domain
Ensure: *min* as an estimation of $\Gamma_p(S)$.

```
Function Min(unsigned n, const double * x, double * grad, void * data)
begin
    x ← C
    max ← procedure(K, C, p, blockDim.x, gridDim.x, gridDim.y)        ▷Algorithm 1
end
opter ← nlopt_create(NLOPT_GN_CRS2_LM, p × n)
opter_set_lower_bounds(lb) ← LB
opter_set_upper_bounds(ub) ← UB
nlopt_set_min_objective(opter, Min, NULL)
nlopt_set_xtol_rel(opter, tol)
min ← γ
i ← 0
repeat
    stopval ← min
    nlopt_set_stopval(opter, stopval)
    nlopt_get_stopval(opter)
    result ← nlopt_optimize(opter, x, &f_min)
    if result then
        min ← f_min
    end if
    i ← i + 1
until i ≥ 40
return min
```

Let p be a positive integer, $K = B_2^3$, LB = -2, UB = 2, and

$$\text{gridDim.x} = \text{gridDim.y} = \text{blockDim.x} = 1024.$$

Table 3 shows a comparison between these three stochastic algorithms. It can be seen that GN_CRS2_LM is better than the other ones.

Table 3. Comparison between different stochastic algorithms.

Algorithm	GN_CRS2_LM	GN_ESCH	GN_ISRES
$\Gamma_5(B_2^3)$	0.894120...	0.896774...	0.895573...
Time	3655 s	7863 s	35231 s
$\Gamma_6(B_2^3)$	0.817386...	0.818881...	0.819020...
Time	3332 s	8036 s	34921 s

4. Computational Experiments

All algorithms were coded in CUDA 12.2 and gcc 13.2.1. The computer's graphics card is an NVIDIA RTX A4000.

4.1. Covering Functional of the Euclidean Unit Disc

Let B_2^2 be the Euclidean unit disc. In this situation, $d(x,y)$ is the Euclidean distance between x and y. The discretization of Q_2 is performed using threadIdx.x for the x axis and blockIdx.x for the y axis. For a positive integer i, let $i = \text{blockDim.x} = \text{gridDim.x}$; then, the values of threadIdx.x and blockIdx.x range from 0 to $i-1$. Let

$$S_i^2 = \left\{(x,y) \mid x,y \in \left\{-1 + \frac{2k}{i-1} \mid k \in [i-1] \cup \{0\}\right\}\right\}.$$

It is clear that

$$\frac{\sqrt{2}}{2} Q_2 \subseteq B_2^2 \subseteq Q_2.$$

Let

$$S = S_i^2 \cap \left(\frac{1 + \frac{\sqrt{2}}{2}(i-1)}{\frac{\sqrt{2}}{2}(i-1)} K\right) = S_i^2 \cap \left(\frac{2 + \sqrt{2}(i-1)}{\sqrt{2}(i-1)} K\right).$$

By Theorem 1, we have $B_2^2 \subseteq S + \frac{1}{i-1} Q_2$. By Corollary 3.1 in [9], there is a p-optimal configuration of B_2^2 contained in B_2^2. Thus, we can take LB $= -1$ and UB $= 1$. Proposition 1 shows that

$$\Gamma_p(B_2^2) \leq \Gamma_p(S) + \frac{\sqrt{2}}{i-1}.$$

Numerical estimates of $\Gamma_p(B_2^2)$ are summarized in Table 4.

Table 4. Numerical estimations of $\Gamma_p(B_2^2)$.

i	LB	UB	p	$\Gamma_p(S_i)$	Ranges of $\Gamma_p(B_2^2)$	Known Exact Value
1024	−1	1	3	0.865850...	≤0.8673...	0.866... [5]
1024	−1	1	4	0.706002...	≤0.7074...	0.707... [5]
1024	−1	1	5	0.609076...	≤0.6105...	0.609... [5]
1024	−1	1	6	0.555575...	≤0.5570...	0.555... [5]
1024	−1	1	7	0.499461...	≤0.5009...	0.5 [25]
1024	−1	1	8	0.444873...	≤0.4463...	0.445... [25]

4.2. Covering Functionals of Three-Dimensional Convex Bodies

For each $K \in \mathcal{K}^3$ satisfying $\alpha Q_3 \subseteq K \subseteq Q_3$, let

$$i = \text{gridDim.x} = \text{gridDim.y} = \text{blockDim.x}$$

be a positive integer; then, the values of threadIdx.x, blockIdx.x and blockIdx.y range from 0 to $i-1$. Let

$$S_i^3 = \left\{(x,y,z) \mid x,y,z \in \left\{-1 + \frac{2k}{i-1} \mid k \in [i-1] \cup \{0\}\right\}\right\},$$

and

$$S = S_i^3 \cap \left(\frac{1 + \alpha(i-1)}{\alpha(i-1)} K\right).$$

By Theorem 1 and Proposition 1, we have

$$\Gamma_p(K) \leq \Gamma_p(S) + \frac{1}{\alpha(i-1)}. \tag{8}$$

4.2.1. Covering Functionals of B_2^3

By Remark 3, we set LB $= -2$ and UB $= 2$. Clearly,

$$\frac{\sqrt{3}}{3}Q_3 \subseteq B_2^3 \subseteq Q_3.$$

From (8), we have

$$\Gamma_p(B_2^3) \leq \Gamma_p(S) + \frac{\sqrt{3}}{i-1}.$$

Numerical estimations of $\Gamma_p(B_2^3)$ for $p \in \{4,5,6,7,8\}$ are summarized in Table 2.

4.2.2. Covering Functional of the Regular Tetrahedron

Let

$$T = \left\{ x \in \mathbb{R}^3 \mid p_i(x) \geq 0 \text{ and } \sum_{i\in[3]} p_i(x) \leq 1 \right\}.$$

Thus, T is a regular tetrahedron affinely equivalent to

$$T_1 := \left\{ x \in \mathbb{R}^3 \mid p_i(x) \geq -\frac{1}{5} \text{ and } \sum_{i\in[3]} p_i(x) \leq \frac{3}{5} \right\}.$$

Clearly, $T_1 = \{x \in \mathbb{R}^n \mid Ax \leq B\}$, where

$$A = \begin{bmatrix} \frac{5}{3} & \frac{5}{3} & \frac{5}{3} \\ -5 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & -5 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix}.$$

Let $v_i(x) = p_i(x) - p_i(c)$. Then,

$$A[x-c] = \begin{bmatrix} \frac{5}{3} & \frac{5}{3} & \frac{5}{3} \\ -5 & 0 & 0 \\ 0 & -5 & 0 \\ 0 & 0 & -5 \end{bmatrix} \cdot \begin{bmatrix} v_1(x) \\ v_2(x) \\ v_3(x) \end{bmatrix} = \begin{bmatrix} \frac{5}{3}(v_1(x)) + \frac{5}{3}(v_2(x)) + \frac{5}{3}(v_3(x)) \\ -5(v_1(x)) \\ -5(v_2(x)) \\ -5(v_3(x)) \end{bmatrix}.$$

By Lemma 1, the dissimilarity $d(x,c)$ between x and c is given by

$$d(x,c) = \max\left\{ \frac{5}{3}v_1(x) + \frac{5}{3}v_2(x) + \frac{5}{3}v_3(x), -5v_1(x), -5v_2(x), -5v_3(x) \right\}.$$

It can be verified that

$$\frac{1}{5}Q_3 \subseteq T_1 \subseteq Q_3,$$

see Figure 3, where

$$w_1 = \left(-\frac{1}{5}, -\frac{1}{5}, -\frac{1}{5}\right),\ w_2 = \left(-\frac{1}{5}, \frac{1}{5}, -\frac{1}{5}\right),\ w_3 = \left(\frac{1}{5}, -\frac{1}{5}, \frac{1}{5}\right),$$
$$w_4 = \left(1, -\frac{1}{5}, -\frac{1}{5}\right),\ w_5 = \left(-\frac{1}{5}, -\frac{1}{5}, 1\right), \text{ and } w_6 = \left(-\frac{1}{5}, 1, -\frac{1}{5}\right).$$

By Remark 3, we take LB $= -2$ and UB $= 2$. By (8), we have

$$\Gamma_p(T) \leq \Gamma_p(S) + \frac{5}{i-1}.$$

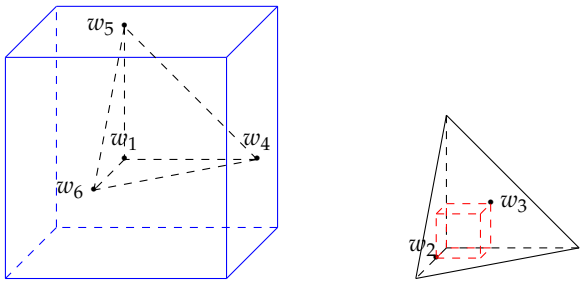


Figure 3. Inscribed cube and circumscribed cube of the regular tetrahedron.

Numerical estimations of $\Gamma_p(T)$ for $p \in \{4,5,6,7,8\}$ are summarized in Table 5.

Table 5. Numerical estimations of $\Gamma_p(T)$.

i	LB	UB	p	$\Gamma_p(S_i)$	Ranges of $\Gamma_p(T)$	Known Exact Value or Estimations
1024	-2	2	4	0.754079...	$\leq$0.7590...	$\frac{3}{4}$ [8]
1024	-2	2	5	0.697766...	$\leq$0.7027...	$\frac{9}{13}$ [8]
1024	-2	2	6	0.678219...	$\leq$0.6832...	$\frac{2}{3}$ [26]
1024	-2	2	7	0.643850...	$\leq$0.6488...	$[\frac{3}{5}, \frac{7}{11}]$ [26]
1024	-2	2	8	0.630140...	$\leq$0.6351...	$\leq \frac{8}{13}$ [26]

4.2.3. Covering Functional of the Regular Octahedron

Let

$$B_1^3 = \left\{ x \in \mathbb{R}^3 \mid \sum_{i \in [3]} |p_i(x)| \leq 1 \right\}.$$

Then, B_1^3 is a regular octahedron. By Lemma 1, the dissimilarity between x and c with respect to B_1^3 is given by

$$d(x,c) = |v_1(x)| + |v_2(x)| + |v_3(x)|.$$

It can be verified that

$$\frac{1}{3} Q_3 \subseteq B_1^3 \subseteq Q_3,$$

see Figure 4, where

$$w_1 = (1,0,0),\ w_2 = (0,0,1),\ w_3 = \left(-\frac{1}{3}, -\frac{1}{3}, -\frac{1}{3}\right), \text{ and } w_4 = \left(\frac{1}{3}, \frac{1}{3}, \frac{1}{3}\right).$$

By Remark 3, we can set LB $= -2$ and UB $= 2$. From (8), we have

$$\Gamma_p(B_1^3) \leq \Gamma_p(S) + \frac{3}{i-1}.$$

Numerical estimations of $\Gamma_p(B_1^3)$ for $p \in \{6,7,8\}$ are summarized in Table 6.

Table 6. Numerical estimations of $\Gamma_p(B_1^3)$.

i	LB	UB	p	$\Gamma_p(S_i)$	Ranges of $\Gamma_p(B_1^3)$	Known Exact Value
1024	-2	2	6	0.667671...	$\leq$0.6707...	$\frac{2}{3}$ [8,27,28]
1024	-2	2	7	0.667666...	$\leq$0.6706...	$\frac{2}{3}$ [8,27,28]
1024	-2	2	8	0.667657...	$\leq$0.6706...	$\frac{2}{3}$ [8,27,28]

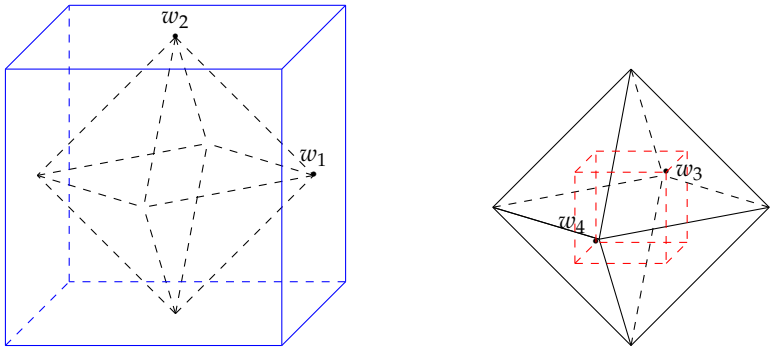


Figure 4. Circumscribed and inscribed cubes of the regular octahedron.

4.2.4. Covering Functional of the Regular Dodecahedron

Let P be the regular dodecahedron having the form:

$$P = \left\{ x \in \mathbb{R}^3 \mid \varphi|p_1(x)| + |p_2(x)| \leq 1, \varphi|p_2(x)| + |p_3(x)| \leq 1, \varphi|p_3(x)| + |p_1(x)| \leq 1 \right\},$$

where $\varphi = \frac{\sqrt{5}-1}{2}$. By Lemma 1, the dissimilarity between x and c with respect to P is given by

$$\begin{aligned} d(x,c) = \max\{&\varphi v_1(x) + v_2(x),\ \varphi v_1(x) - v_2(x),\ -\varphi v_1(x) + v_2(x),\ -\varphi v_1(x) - v_2(x),\\ &\varphi v_2(x) + v_3(x),\ \varphi v_2(x) - v_3(x),\ -\varphi v_2(x) + v_3(x),\ -\varphi v_2(x) - v_3(x),\\ &\varphi v_3(x) + v_1(x),\ \varphi v_3(x) - v_1(x),\ -\varphi v_3(x) + v_1(x),\ -\varphi v_3(x) - v_1(x)\}. \end{aligned}$$

It can be verified that

$$\varphi Q_3 \subseteq P \subseteq Q_3,$$

see Figure 5, where

$$w_1 = (-1, 0, \varphi - 1),\ w_2 = (-1, 0, 1 - \varphi),\ w_3 = (-\varphi, -\varphi, -\varphi),\ \text{and}\ w_4 = (\varphi, \varphi, \varphi).$$

By Remark 3, we can set LB $= -2$ and UB $= 2$. From (8), we have

$$\Gamma_p(P) \leq \Gamma_p(S) + \frac{1}{\varphi(i-1)}.$$

Numerical estimations of $\Gamma_p(P)$ for $p \in \{5, 6, 7, 8\}$ are summarized in Table 7. The estimate of $\Gamma_8(P)$ is much better than that given in [29].

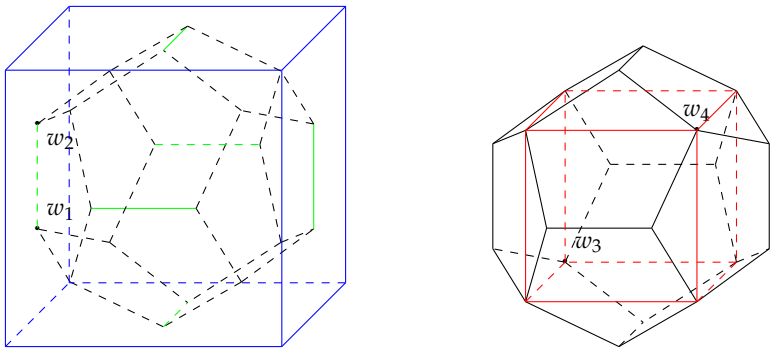


Figure 5. Inscribed cube and circumscribed cube of the regular dodecahedron.

Table 7. Numerical estimations of $\Gamma_p(P)$.

i	LB	UB	p	$\Gamma_p(S_i)$	Ranges of $\Gamma_p(P)$
1024	-2	2	5	0.840639...	$\leq$0.8423...
1024	-2	2	6	0.761430...	$\leq$0.7631...
1024	-2	2	7	0.745951...	$\leq$0.7476...
1024	-2	2	8	0.711539...	$\leq$0.7132...

4.2.5. Covering Functional of the Regular Icosahedron

Let M be the regular icosahedron having the following form:

$$M = \Big\{x \in \mathbb{R}^3 \mid \varphi|p_1(x)| + \varphi|p_2(x)| + \varphi|p_3(x)| \leq 1,\ (\varphi-1)|p_1(x)| + |p_3(x)| \leq 1,$$
$$|p_1(x)| + (\varphi-1)|p_2(x)| \leq 1,\ |p_2(x)| + (\varphi-1)|p_3(x)| \leq 1\Big\},$$

where $\varphi = \frac{\sqrt{5}-1}{2}$. By Lemma 1, the dissimilarity between x and c with respect to M is given by

$$d(x,c) = \max\{-\varphi v_1(x) + \varphi v_2(x) + \varphi v_3(x),\ -\varphi v_1(x) + \varphi v_2(x) - \varphi v_3(x),$$
$$\varphi v_1(x) + \varphi v_2(x) + \varphi v_3(x),\ \varphi v_1(x) + \varphi v_2(x) - \varphi v_3(x),\ -\varphi v_1(x) - \varphi v_2(x) + \varphi v_3(x),$$
$$\varphi v_1(x) - \varphi v_2(x) + \varphi v_3(x),\ \varphi v_1(x) - \varphi v_2(x) - \varphi v_3(x),\ -\varphi v_1(x) - \varphi v_2(x) - \varphi v_3(x),$$
$$(\varphi-1)v_1(x) + v_3(x),\ (\varphi-1)v_1(x) - v_3(x),\ (1-\varphi)v_1(x) + v_3(x),\ (1-\varphi)v_1(x) + v_3(x),$$
$$(\varphi-1)v_2(x) + v_1(x),\ (\varphi-1)v_2(x) - v_1(x),\ (1-\varphi)v_2(x) + v_1(x),\ (1-\varphi)v_2(x) + v_1(x),$$
$$(\varphi-1)v_3(x) + v_2(x),\ (\varphi-1)v_3(x) - v_2(x),\ (1-\varphi)v_3(x) + v_2(x),\ (1-\varphi)v_3(x) + v_2(x)\}.$$

It can be verified that

$$\frac{1+\varphi}{3}Q_3 \subseteq M \subseteq Q_3,$$

see Figure 6, where

$$w_1 = (\varphi-1, 0, 1),\ w_2 = (1-\varphi, 0, 1),$$
$$w_3 = \left(-\frac{1+\varphi}{3}, \frac{1+\varphi}{3}, -\frac{1+\varphi}{3}\right), \text{ and } w_4 = \left(\frac{1+\varphi}{3}, -\frac{1+\varphi}{3}, \frac{1+\varphi}{3}\right).$$

By Remark 3, we take LB $= -2$ and UB $= 2$. By (8), we have

$$\Gamma_p(M) \leq \Gamma_p(S) + \frac{3}{(1+\varphi)(i-1)}.$$

Numerical estimations of $\Gamma_p(M)$ for $p \in \{4,5,6,7,8\}$ are summarized in Table 8. Compared to the estimate $\Gamma_6(M) \leq 0.873$ given in [29], our estimate here is much better.

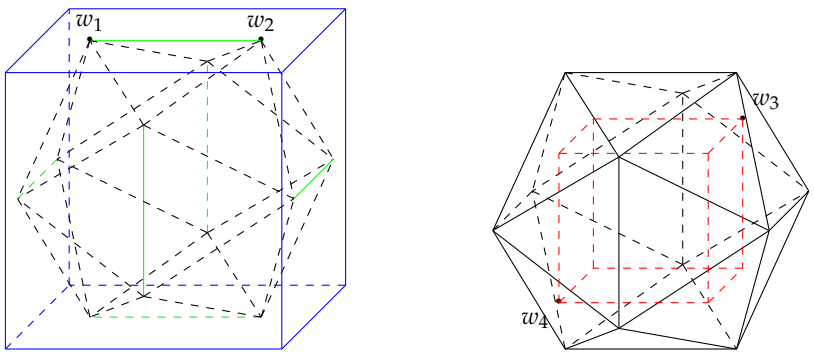


Figure 6. Inscribed cube and circumscribed cube of the regular icosahedron.

Table 8. Numerical estimations of $\Gamma_p(M)$.

i	LB	UB	p	$\Gamma_p(S_i)$	Ranges of $\Gamma_p(M)$
1024	−2	2	4	0.873606...	$\leq$0.8755...
1024	−2	2	5	0.848224...	$\leq$0.8501...
1024	−2	2	6	0.746998...	$\leq$0.7489...
1024	−2	2	7	0.747460...	$\leq$0.7493...
1024	−2	2	8	0.731832...	$\leq$0.7337...

5. Conclusions

This paper proposes an algorithm based on CUDA for estimating covering functionals of three-dimensional convex polytopes, which is more efficient than the geometric branch-and-bound method presented in [9]. The theoretical accuracy of our algorithm is given by (8). Restrictions on the dimensions of each grid makes the implementation of our algorithm in higher dimensions difficult.

Author Contributions: Conceptualization, S.W.; methodology, X.H. and S.W.; software, X.H.; validation, S.W. and L.Z.; writing—original draft preparation, X.H.; writing—review and editing, S.W. and L.Z.; funding acquisition, S.W. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China (grant number 12071444), the Fundamental Research Program of Shanxi Province (grant numbers 202103021223191 and 202303021221116), and the 19th Graduate Science and Technology Project of North University of China (grant number 20231944).

Data Availability Statement: Data sharing is not applicable, since no dataset was generated or analyzed during the current study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Boltyanski, V.; Martini, H.; Soltan, P.S. *Excursions into Combinatorial Geometry, Universitext*; Springer: Berlin/Heidelberg, Germany, 1997.
2. Martini, H.; Soltan, V. Combinatorial problems on the illumination of convex bodies. *Aequationes Math.* **1999**, *57*, 121–152. [CrossRef]
3. Bezdek, K. *Classical Topics in Discrete Geometry, CMS Books in Mathematics/Ouvrages de Mathématiques de la SMC*; Springer: New York, NY, USA, 2010.
4. Brass, P.; Moser, M.; Pach, J. *Research Problems in Discrete Geometry*; Springer: New York, NY, USA, 2005.
5. Bezdek, K.; Khan, M.A. The geometry of homothetic covering and illumination. In *Discrete Geometry and Symmetry*; Springer: Cham, Switzerland, 2018; Volume 234, pp. 1–30.
6. Rogers, C.A.; Zong, C. Covering convex bodies by translates of convex bodies. *Mathematika* **1997**, *44*, 215–218. [CrossRef]
7. Papadoperakis, I. An estimate for the problem of illumination of the boundary of a convex body in E^3. *Geom. Dedicata* **1999**, *75*, 275–285. [CrossRef]
8. Zong, C. A quantitative program for Hadwiger's covering conjecture. *Sci. China Math.* **2010**, *53*, 2551–2560. [CrossRef]
9. He, C.; Lv, Y.; Martini, H.; Wu, S. A branch-and-bound approach for estimating covering functionals of convex bodies. *J. Optimiz. Theory. App.* **2023**, *196*, 1036 – 1055. [CrossRef]
10. Scholz, D. *Deterministic Global Optimization: Geometric Branch-and-Bound Methods and Their Applications*; Springer: New York, NY, USA, 2012.
11. Yu, M.; Lv, Y.; Zhao, Y.; He, C.; Wu, S. Estimations of covering functionals of convex bodies based on relaxation algorithm. *Mathematics* **2023**, *11*, 2000. [CrossRef]
12. Han, T.D.; Abdelrahman, T.S. Hicuda: High-level GPGPU programming. *IEEE T. Parall. Distr.* **2011**, *22*, 78–90. [CrossRef]
13. Harris, M. Optimizing Parallel Reduction in CUDA. Available online: https://developer.download.nvidia.cn/assets/cuda/files/reduction.pdf (accessed on 14 February 2024).
14. Tomczak-Jaegermann, N. Banach-Mazur distances and finite-dimensional operator ideals. In *Pitman Monographs and Surveys in Pure and Applied Mathematics*; John Wiley & Sons, Inc.: New York, NY, USA, 1989; Volume 38.
15. Dind, K.; Tan, T.A. Review on general purpose computing on GPU and its applications in computational intelligence. *CAAI Trans. Intell. Syst.* **2015**, *10*, 1–11.
16. Cheng, J.; Grossman, M.; Mckercher, T. *Professional CUDA C Programming*; John Wiley & Sons, Inc.: New York, NY, USA, 2014.

17. Ma, X.; Han, W. A parallel multi-swarm particle swarm optimization algorithm based on CUDA streams. In Proceedings of the 2018 Chinese Automation Congress (CAC), Xi'an, China, 30 November–2 December 2018; pp. 3002–3007.
18. Guide, D. *CUDA C Programming Guide*. Available online: https://docs.nvidia.com/cuda/archive/9.1/pdf/CUDA_C_Programming_Guide.pdf (accessed on 14 February 2024).
19. Zhang, Y.; Chen, L.; An, X.; Yan, S. Study on performance optimization of reduction algorithm targeting GPU computing platform. *Comput. Sci.* **2019**, *46*, 306–314.
20. Steven, G.J. *NLopt Documentation*. Available online: https://nlopt.readthedocs.io/en/latest/ (accessed on 14 February 2024).
21. Tseng, L.Y.; Chen, C. Multiple trajectory search for large scale global optimization. In Proceedings of the 2008 IEEE Congress on Evolutionary Computation (IEEE World Congress on Computational Intelligence), Hong Kong, China, 1–6 June 2008; pp. 3052–3059.
22. Wang, H.; Shao, H. System optimization strategy based on genetic algorithm and controlled random search. *Control. Theory Appl.* **2000**, *6*, 907–910.
23. Hansen, N.; Ostermeier, A. Completely derandomized self-adaptation in evolution strategies. *Evol. Comput.* **2001**, *9*, 159–195. [CrossRef] [PubMed]
24. Branke, J.; Deb, K.; Miettinen, K.; Slowinski, R. *Multi-Objective Optimization: Interactive and Evolutionary Approaches*; Springer: Berlin/Heidelberg, Germany, 2008.
25. Fejes Tóth, G. Thinnest covering of a circle by eight, nine, or ten congruent circles. In *Combinatorial and Computational Geometry*; Cambridge University Press: Cambridge, UK, 2005; Volume 52, pp. 361–376.
26. Yu, M.; Gao, S.; He, C.; Wu, S. Estimations of covering functionals of simplices. *Math. Inequal. Appl.* **2023**, *26*, 793–809. [CrossRef]
27. Lian, Y.; Zhang, Y. Covering the crosspolytope with its smaller homothetic copies. *arXiv* **2021**, arXiv:2103.10004v2.
28. Joós, A. Covering the crosspolytope with crosspolytopes. *arXiv* **2023**, arXiv:2305.00569.
29. Wu, S.; He, C. Covering functionals of convex polytopes. *Linear Algebra. Appl.* **2019**, *577*, 53–68. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.