Constrained Minkowski Sums: A Geometric Framework for Solving Interval Problems in Computational Biology Efficiently

**Thorsten Bernholt · Friedrich Eisenbrand ·
Thomas Hofmeister**

Abstract In this paper, we introduce the notion of a constrained Minkowski sum: for two (finite) point-sets $P, Q \subseteq \mathbb{R}^2$ and a set of k inequalities $Ax \geq b$, it is defined as the point-set $(P \oplus Q)_{Ax \geq b} = \{x = p + q \mid p \in P, q \in Q, Ax \geq b\}$. We show that typical interval problems from computational biology can be solved by computing a set containing the vertices of the convex hull of an appropriately constrained Minkowski sum. We provide an algorithm for computing such a set with running time $O(N \log N)$, where $N = |P| + |Q|$ if k is fixed. For the special case $(P \oplus Q)_{x_1 \geq \beta}$ where P and Q consist of points with integer x_1-coordinates whose absolute values are bounded by $O(N)$, we even achieve a linear running time $O(N)$. We thereby obtain a linear running time for many interval problems from the literature and improve upon the best known running times for some of them. The main advantage of the presented approach is that it provides a general framework within which a broad variety of interval problems can be modeled and solved.

1 Introduction

The *Minkowski sum* of two (finite) point-sets $P \subseteq \mathbb{R}^2$ and $Q \subseteq \mathbb{R}^2$ is defined as $P \oplus Q = \{p + q \mid p \in P, q \in Q\}$. Minkowski sums are a fundamental concept in

T. Bernholt · T. Hofmeister
Informatik 2, University of Dortmund, 44221 Dortmund, Germany

F. Eisenbrand (✉)
EPFL, 1015 Lausanne, Switzerland
e-mail: friedrich.eisenbrand@epfl.ch

algorithmic geometry, in particular in robot motion planning [12, 13, 15, 18] and placement problems [1, 6]. The convex hull of $P \oplus Q$ can be computed in linear time [12] if the points in P and Q are sorted w.r.t. the value of some given linear function, for example, the value of their x_1-coordinate. The convex hull of $P \oplus Q$ has at most $N = |P| + |Q|$ vertices.

In this paper, we introduce the notion of a *constrained* Minkowski sum. For a matrix $A \in \mathbb{R}^{k \times 2}$ and a vector $b \in \mathbb{R}^k$, the *constrained Minkowski sum* $(P \oplus Q)_{Ax \geq b}$ is defined as the point-set

$$(P \oplus Q)_{Ax \geq b} = \{x \in P \oplus Q \mid Ax \geq b\}.$$

For $k = 1$, the system $Ax \geq b$ consists of one linear inequality $a^T x \geq \beta$, and we write $(P \oplus Q)_{a^T x \geq \beta}$. We call a constraint $a^T x \geq \beta$ *linearly sortable* if all $|a^T p|$, $p \in P$, and $|a^T q|$, $q \in Q$, are integers bounded by $O(N)$.

Our motivation for studying constrained Minkowski sums comes from a very practical application. A large class of interval problems from computational biology can be solved by maximizing a quasiconvex function over the points of a constrained Minkowski sum. Recall that a function $f : D \to \mathbb{R}$ is called *quasiconvex* if for all points $s_1, s_2 \in D$ and all $\lambda \in [0, 1]$, one has $f(\lambda \cdot s_1 + (1 - \lambda) \cdot s_2) \leq \max\{f(s_1), f(s_2)\}$, where $D \subseteq \mathbb{R}^2$ is a nonempty convex set. The function is quasiconvex if and only if its domain D and all its *sublevel sets* $S_\alpha = \{s \in D \mid f(s) \leq \alpha\}$, $\alpha \in \mathbb{R}$, are convex, see, e.g., [5]. If $R \subseteq \mathbb{R}^2$ is a finite set of points, then a quasiconvex function f attains its maximum over R on one of the vertices of the *convex hull* conv(R) of R.

1.1 Contributions of This Paper

Our main result is an algorithm which computes a set $R \subseteq (P \oplus Q)_{Ax \geq b}$ containing all vertices of conv($(P \oplus Q)_{Ax \geq b}$) in time $O(N \log N)$ if the number of constraints is fixed. If the number of constraints is k, then this algorithm runs in time $O(k \log k + k \cdot N \log N)$. This shows that a quasiconvex function which can be evaluated in constant time can be maximized over $(P \oplus Q)_{Ax \geq b}$ in time $O(k \cdot \log k + k \cdot N \log N)$. As a consequence, we obtain for many interval problems from the literature linear-time algorithms and improve upon the best known running times for some of them. These results are achieved via the following steps:

(i) First, we show that the number of vertices of the convex hull of a Minkowski sum with one constraint is linear. In fact, we provide a tight bound.
(ii) This result is exploited to derive a linear-time algorithm which outputs a set R containing all the vertices of conv($(P \oplus Q)_{a^T x \geq \beta}$) if the points of P and Q are sorted w.r.t. the linear function $a^T x$.
(iii) Next, we describe a divide and conquer algorithm which computes a set $R \subseteq (P \oplus Q)_{Ax \geq b}$ containing all vertices of conv($(P \oplus Q)_{Ax \geq b}$) in time $O(N \log N)$ if $Ax \geq b$ consists of two constraints.
(iv) If $Ax \geq b$ describes a triangle, we show how to reduce the computation of such a set R to the case described in (iii)), which implies the main result by triangulation of the convex polygon described by the system $Ax \geq b$.

We close this section by arguing why our result for a fixed number of constraints is optimal in the *algebraic decision-tree* model. Ben-Or [3] showed that the *set-disjointness* problem has a lower bound of $\Omega(n \log n)$ in this model of computation. Set disjointness is defined as follows. Given two sets $A = \{a_1, \ldots, a_n\} \subseteq \mathbb{R}$ and $B = \{b_1, \ldots, b_n\} \subseteq \mathbb{R}$, one has to decide whether $A \cap B = \emptyset$ holds. Set-disjointness can be reduced to the problem of maximizing a quasiconvex, even linear, function over a constrained Minkowski sum in linear time as follows. Construct the point-sets $P = \{(0, -a) \mid a \in A\}$ and $Q = \{(0, b) \mid b \in B\}$. The point $(0, 0)$ is contained in $P \oplus Q$ if and only if A and B are not disjoint. Thus, the maximum of the objective function $-x_2$ over the constrained Minkowski sum $(P \oplus Q)_{x_2 \geq 0}$ is equal to 0 if and only if A and B are not disjoint. We therefore have the following theorem.

Theorem 1 *The problem of maximizing a quasiconvex objective function f over the constrained Minkowski sum $(P \oplus Q)_{Ax \geq b}$ requires time $\Omega(N \log N)$ in the algebraic decision tree model even if f is a linear function and $Ax \geq b$ consists of only one constraint.*

2 Interval Problems from Computational Biology

Numerous interval problems that arise in computational biology can be formulated in the following abstract form.

Given an array $a_1, \ldots, a_n$ of real numbers and an objective function f, compute an interval $[i, j]$ such that the interval $a_i, a_{i+1}, \ldots, a_j$ satisfies some given constraints and maximizes f.

Often, the function $f = f(\ell, s)$ depends on the sum $s = a_i + \cdots + a_j$ of the interval and its length, $\ell = j - i + 1$. Here are just a few examples from the literature which fit into this framework:

(a) The *maximum-sum segment problem* [10]: Given L and U, find an interval with length between L and U such that its sum is as large as possible.
(b) The *maximum-density segment problem* [11]: In addition to the array, weights $w_1, \ldots, w_n > 0$ and bounds L, U are given. Among all intervals $[i, j]$ with weight $L \leq w_i + \cdots + w_j \leq U$, find one with the largest density $(a_i + \cdots + a_j)/(w_i + \cdots + w_j)$.
(c) The *longest biased interval* [2]: Given a bias $0 \leq b \leq 1$, find an interval $[i, j]$ which has an average $(a_i + \cdots + a_j)/(j - i + 1) \geq b$ and which is as long as possible. Allison [2] uses this problem in the context of "preferred characters," where we additionally have that $a_i \in \{0, 1\}$, as one can use a_i as an indicator for whether a character in the array is "preferred" or not.
(d) The *length-constrained heaviest segments* [16]: Given a bound L, find an interval $[i, j]$ with length at least L which has maximum average $(a_i + \cdots + a_j)/(j - i + 1)$. This is in fact a special case of problem (b). (Set all $w_i = 1$ and $U = n$.)
(e) *DNA copy number data analysis* [17]: Here, the objective is to find an interval $[i, j]$ such that $|a_i + \cdots + a_j|/\sqrt{j - i + 1}$ is as large as possible. Problem (e) also has an application in statistics, see the multiresolution criteria problem in [7].

(Please note that originally, Lipson et al. [17] consider the value $(a_i + \cdots + a_j)/\sqrt{j - i + 1}$ without the absolute value $|\cdot|$, but this poses no problem for the application, as pointed out in [4].)

We now show that these problems can be solved by maximizing a quasiconvex function over the points of a constrained Minkowski sum. An interval $[i, j]$ has length $\ell(i, j) = j - i + 1$ and sum $s(i, j) = a_i + \cdots + a_j$. If we map each interval $[i, j]$ to the two-dimensional point $(\ell(i, j), s(i, j))$, we obtain a point-set Z. Problem (e), for example, is the problem of maximizing the quasiconvex function $f(\ell, s) = |s|/\sqrt{\ell}$ over Z.

It remains to describe how the point-set Z can be seen as a constrained Minkowski sum. For $1 \leq i, j \leq n$, define the points $p_j = (j, a_1 + \cdots + a_j)$ and $q_i = (-i + 1, -(a_1 + \cdots + a_{i-1}))$. For $i \leq j$, we then have $p_j + q_i = (j - i + 1, a_i + \cdots + a_j)$, i.e., the first coordinate of $p_j + q_i$ corresponds to the length of the interval $[i, j]$, and the second corresponds to the sum of the interval. If $i > j$, then the sum $p_j + q_i$ does not correspond to an interval in the same way.

Now let $P = \{p_1, \ldots, p_n\}$, $Q = \{q_1, \ldots, q_n\}$. The constrained Minkowski sum $(P \oplus Q)_{x_1 \geq 1}$ contains all the points to which intervals of the array are mapped to. The constraint $x_1 \geq 1$ guarantees that we omit the meaningless intervals with negative or zero length. If in fact the interval problem requires that only intervals of length at least, say, L be considered, then we can replace the constraint by $x_1 \geq L$. Thus, problem (e) can be understood as maximizing the quasiconvex function $f(x_1, x_2) = |x_2|/\sqrt{x_1}$ over $(P \oplus Q)_{x_1 \geq 1}$.

Instead of evaluating f on all points of $(P \oplus Q)_{x_1 \geq 1}$, we first apply an algorithm for computing a point-set $R \subseteq (P \oplus Q)_{x_1 \geq 1}$ which contains all vertices of the convex hull of the constrained Minkowski sum. We then evaluate f on the points of R and choose a point with maximum value. If evaluating f on a point takes time $O(1)$, which is a reasonable assumption, then the time for evaluating f on R is bounded by the running time for computing R.

In the case that additional constraints are needed, like the constraint that we should only consider intervals which have a length bounded by U, we can add the corresponding constraint like $x_1 \leq U$ and compute the constrained Minkowski sum with two or more constraints. Problem (a), for example, is the problem of maximizing $f(x_1, x_2) = x_2$ under the constraints $L \leq x_1 \leq U$. For some problems, the modeling is immediate; for others, extra precautions have to be taken. A detailed modeling is given in the appendix of this paper.

The system $Ax \geq b$ typically consists of one constraint which either is linearly sortable or is of the form $\alpha \leq a^{\mathrm{T}} x \leq \beta$, where both constraints are linearly sortable. In the first case, Theorem 5 below shows that the maximization problem can be solved in linear time, whereas Theorem 7 below shows linear running time in the second case. We therefore obtain the following corollary.

Corollary 1 *Problems* (a)–(e) *can be solved in time* $O(n)$.

In particular we improve upon the best known running times of $O(n \log n)$ [4] and $O(n^2)$ [2] for problems (e) and (c), respectively. More generally, we have the following.

Corollary 2 *Let $a_1, \ldots, a_n$ be an array of numbers, let $f(\ell, s)$ be a quasiconvex function, and L, U natural numbers. The problem of finding an interval $[i, j]$ whose value $f(\ell(i, j), s(i, j))$ is maximum among all intervals whose length satisfies $L \leq \ell(i, j) \leq U$ can be solved in linear time $O(n)$. If additionally, a fixed number of linear constraints on ℓ and s are given that the interval has to satisfy, then the problem can be solved in time $O(n \log n)$.*

3 Minkowski Sums with One Constraint

Before we inspect Minkowski sums with one constraint, we first have to recall a well-known fact about unconstrained Minkowski sums, see, e.g., [8]. See Fig. 1 for an illustration.

Theorem 2 *Let P and Q be finite point-sets in the plane, and let $Z = \text{conv}(P \oplus Q)$ be the convex hull of the Minkowski sum of P and Q. Then the sequence of vertices of Z in clockwise order can be written as*

$$p_{i_1} + q_{j_1}, \ldots, p_{i_k} + q_{j_k}, \tag{1}$$

where each appearance of each p and each q in (1) is consecutive. In other words, if $p \in P$ appears in a sum of (1), then there exist an index ℓ_p and an integer μ_p such that all appearances of p are in the positions $\ell_p, \ell_p + 1, \ldots, \ell_p + \mu_p$ in (1), where all these integers are taken modulo k. Similarly, if $q \in Q$ appears in a sum in (1), then there exist an index ℓ_q and an integer μ_q such that all appearances of q are in the positions $\ell_q, \ell_q + 1, \ldots, \ell_q + \mu_q$ in (1), where all these integers are taken modulo k.

How can one compute such a sequence as it is described in Theorem 2? First, one computes the clockwise order of the vertices of the convex hull of P and Q individually. Let p_l and q_l be the leftmost vertices of P and Q, respectively. The sequence is initiated with $p_l + q_l$. Let $p + q$ be the most recent element in the sequence, and let p' and q' be the successors of p and q in the clockwise order of the vertices of $\text{conv}(P)$
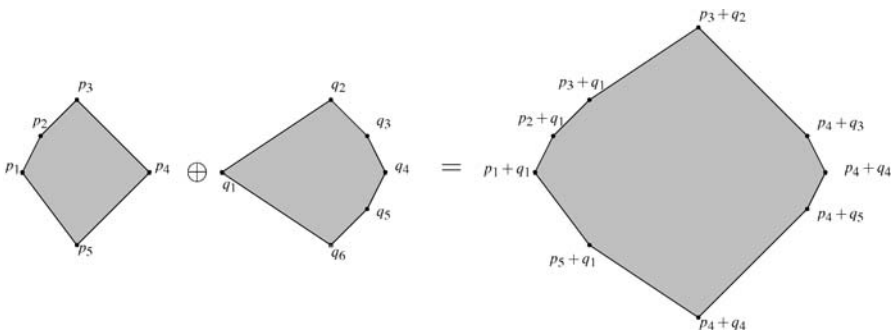

Fig. 1 An illustration of Theorem 2. The point p_3, for example, appears more than once in the convex hull of the Minkowski sum. These appearances are however consecutive. Similarly, the point q_1 appears four times consecutively

and conv(Q), respectively. If the polygonal curve defined by $p + q$, $p + q'$, $p' + q'$ turns to the right, one chooses $p + q'$ as the next point in the sequence. Otherwise, the next point is $p' + q$.

Theorem 2 implies that the number of vertices of conv($P \oplus Q$) is bounded by $|P| + |Q|$. We want to find an efficient algorithm which computes a set R containing all the vertices of $Z = \text{conv}((P \oplus Q)_{a^{\mathrm{T}} x \geq \beta})$. Clearly this depends on the number of vertices of Z. How large is this number? It turns out that we can answer this question exactly.

3.1 A Tight Bound on the Number of Vertices

We begin with a lower bound. The left part of Fig. 2 shows the part of the unit circle in which the first coordinate x_1 is nonnegative. This half-circle is closed with the line segment from $(0, -1)$ to $(0, 1)$. In addition, for some small number $\varepsilon > 0$, we have sketched the constraint $x_1 \geq -\epsilon$ by a line which is located to the left of the half-circle. Now bend the line segment of the half-circle a little bit outside so that the result is a curve which, from bottom to top, turns to the right and is symmetric around the x_1-axis. Place distinct points $p_1, \ldots, p_n$ on the upper half of the half-circle. Place the points $p'_1, \ldots, p'_n$ on the lower part of the half-circle so that p_i and p'_i are symmetric around the x_1-axis. For each of the points p_i and p'_i, there exists a vector q_i which is parallel to the x_1-axis such that $p_i + q_i$ and $p'_i + q_i$ are on the curve closing the half-circle.

Finally, let $P = \{p_1, \ldots, p_n, p'_1, \ldots, p'_n\}$ and $Q = \{0, q_1, \ldots, q_n\}$. Then conv($(P \oplus Q)_{x_1 \geq -\epsilon}$) has the vertices $P \cup \{p_1 + q_1, \ldots, p_n + q_n, p'_1 + q_1, \ldots, p'_n + q_n\}$. This shows that conv($(P \oplus Q)_{x_1 \geq -\epsilon}$) can have $|P| + 2 \cdot |Q| - 2$ vertices. This proves the following theorem.

Theorem 3 *For each $n \in \mathbb{N}$, there exist point-sets P and Q with $|P| \geq n$ and $|Q| \geq n$ and a constraint $a^{\mathrm{T}} x \geq \beta$ such that the number of vertices of $(P \oplus Q)_{a^{\mathrm{T}} x \geq \beta}$ is at least*

$$\min\{2 \cdot |P| + |Q|, |P| + 2 \cdot |Q|\} - 2.$$

Fig. 2 A lower bound construction for the number of vertices
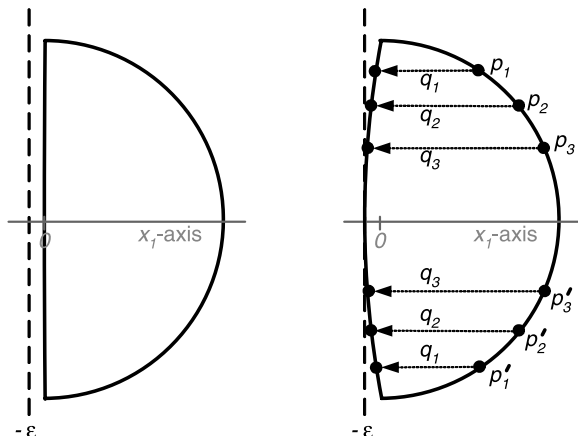
Next, we now show that this lower bound is tight. Without loss of generality we can assume that the constraint $a^T x \geq \beta$ is $x_1 \geq 0$. Now let $P = \{p_1, \ldots, p_n\}$ and $Q = \{q_1, \ldots, q_m\}$, where the p_i and q_j are sorted nondecreasingly according to their x_1 coordinates. For $i \in \{1, \ldots, n\}$, the number $J(i)$ denotes the first index such that $p_i + q_{J(i)}$ is a valid point. Clearly one has

$$(P \oplus Q)_{x_1 \geq 0} = \bigcup_{i=1}^{n} (\{p_i, \ldots, p_n\} \oplus \{q_{J(i)}, \ldots, q_m\}). \tag{2}$$

Theorem 4 *The polygon $Z = \mathrm{conv}((P \oplus Q)_{x_1 \geq 0})$ has at most $\min\{2 \cdot |P| + |Q|, |P| + 2 \cdot |Q|\} - 2$ vertices.*

Proof For symmetry reasons, it is enough to show that Z has at most $2 \cdot |P| + |Q| - 2$ vertices. Clearly, this holds if $n = 1$, since then Z has at most $|Q|$ vertices.

For $n > 1$, we argue by induction. Consider the Minkowski sum

$$M = \{p_1, \ldots, p_n\} \oplus \{q_{J(1)}, \ldots, q_m\}.$$

Let $p_1 + q_1^*, \ldots, p_1 + q_\mu^*$ be the vertices involving p_1 from the clockwise-order representation of $\mathrm{conv}(M)$ as in Theorem 2, see Fig. 3. Let K be the set $K = \{q_2^*, \ldots, q_{\mu-1}^*\}$. The vertex-representation of $\mathrm{conv}(M)$ does not contain a pair $p_i + q$ for $i \geq 2$ and $q \in K$, since the appearance of such a $q \in K$ then would not be consecutive. Thus, the convex hull of M is the convex hull of the point-set

$$(\{p_1\} \oplus \{q_1^*, \ldots, q_\mu^*\}) \cup (\{p_2, \ldots, p_n\} \oplus (\{q_{J(1)}, \ldots, q_m\} \setminus K)). \tag{3}$$

Now $(P + Q)_{x_1 \geq 0}$ is equal to

$$M \cup (\{p_2, \ldots, p_n\} \oplus \{q_1, \ldots, q_{J(1)-1}\})_{x_1 \geq 0}.$$

Therefore, $(P + Q)_{x_1 \geq 0}$ is the convex hull of the union of the three sets

$$\{p_1\} \oplus \{q_1^*, \ldots, q_\mu^*\} \cup \{p_2, \ldots, p_n\} \oplus (\{q_{J(1)}, \ldots, q_m\} \setminus K)$$

$$\cup (\{p_2, \ldots, p_n\} \oplus \{q_1, \ldots, q_{J(1)-1}\})_{x_1 \geq 0}.$$

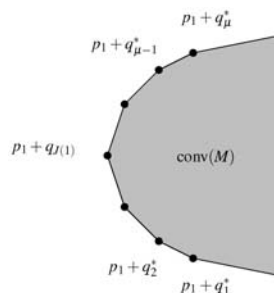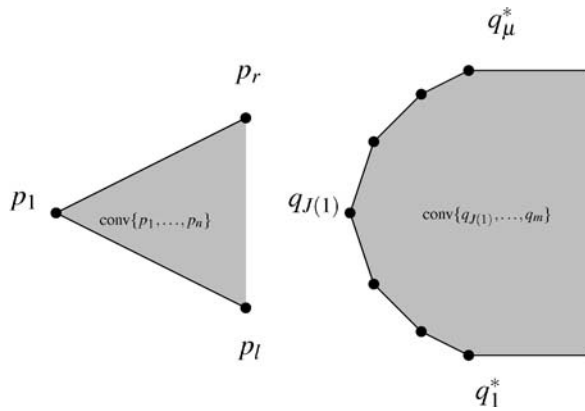**Fig. 3** An illustration of the proof of Theorem 4

Fig. 4 Computing the set $\{p_1 + q_1^*, \ldots, p_1 + q_\mu^*\}$

Every point in the second set above is valid. This shows that

$$(P + Q)_{x_1 \geq 0} = \big(\{p_1\} \oplus \{q_1^*, \ldots, q_\mu^*\}\big) \cup \big(\{p_2, \ldots, p_n\} \oplus Q \setminus K\big)_{x_1 \geq 0}. \tag{4}$$

The polygon $\mathrm{conv}(\{p_1\} \oplus \{q_1^*, \ldots, q_\mu^*\})$ has $|K| + 2$ vertices, whereas, by induction, the convex hull of $(\{p_2, \ldots, p_n\} \oplus Q \setminus K)_{x_1 \geq 0}$ has at most $2 \cdot (|P| - 1) + |Q| - |K| - 2$ vertices. This proves the claim. $\qquad\qquad\square$

Remark 1 Notice that we can augment the set K in the proof above with the convexly dependent points of $\{q_{J(1)}, \ldots, q_m\}$. Let $D \subseteq \{q_{J(1)}, \ldots, q_m\}$ denote the subset of points which are not vertices of $\mathrm{conv}(\{q_{J(1)}, \ldots, q_m\})$. Each occurrence of K in the proof above, starting from (3), can be replaced by $K \cup D$. This means that we have the following strengthening of (4):

$$(P + Q)_{x_1 \geq 0} = \big(\{p_1\} \oplus \{q_1^*, \ldots, q_\mu^*\}\big) \cup \big(\{p_2, \ldots, p_n\} \oplus Q \setminus (K \cup D)\big)_{x_1 \geq 0}.$$

We need this in the following linear-time algorithm to compute a superset of the vertices of $(P \oplus Q)_{x \geq 0}$.

3.2 A Linear-Time Algorithm

The proof of Theorem 4 also suggests an algorithm to compute a set $R \subseteq (P \oplus Q)_{x_1 \geq 0}$ containing all vertices of $Z = \mathrm{conv}((P \oplus Q)_{x_1 \geq 0})$ in linear time if the points are sorted nondecreasingly according to their x_1-values.

In fact, the set K from the proof can be computed in time $O(|K|)$ if the convex hull of $\{q_{J(1)}, \ldots, q_m\}$ and the two neighbors of p_1 on the convex hull of $\{p_1, \ldots, p_n\}$ are known. This works as follows. Let p_r and p_l be the neighboring vertices of p_1 in clockwise and counterclockwise direction of $\mathrm{conv}(\{p_1, \ldots, p_n\})$, see Fig. 4. Assume for simplicity that all x_1-values of points in P and Q respectively are different. The point $q_{J(1)}$ is a vertex of $\mathrm{conv}(\{q_{J(1)}, \ldots, q_m\})$, and p_1 is a vertex of $\mathrm{conv}(\{p_1, \ldots, p_n\})$. In fact those points are the unique leftmost vertices respectively. With these points at hand, the points $q_1^*, \ldots, q_\mu^*$ can easily be computed in time $O(k)$ by following the neighbors of $q_{J(1)}$ clockwise along the convex hull

of $\{q_{J(1)}, \ldots, q_m\}$ until the slope on the upper hull is less than the slope of the line-segment p_1, p_r and counterclockwise until the slope on the lower hull is more than the slope of the line-segment p_1, p_l.

We are now ready to describe the complete algorithm, which we call CONSTR-MINKOWSKI, to compute a superset R of the vertices of $(P \oplus Q)_{x_1 \geq 0}$. With the Graham scan algorithm (from right to left) for convex hulls [8], we compute for each point p_i its two neighbors on the convex hull of $\{p_i, \ldots, p_n\}$. The set R is initialized with the empty set.

The algorithm now proceeds recursively. If the set P contains only one point, then we compute $(P \oplus Q)_{\geq 0}$ directly.

Else, we compute the convex hull of $\{q_{J(1)}, \ldots, q_m\} \subseteq Q$ with the incremental Graham scan algorithm from right to left. Then, we compute the points $q_1^*, \ldots, q_\mu^*$. We store each point of $\{p_1\} \oplus \{q_1^*, \ldots, q_\mu^*\}$ in R and delete each point in $K = \{q_2^*, \ldots, q_{\mu-1}^*\}$ from Q. Using the notation of Remark 1, recall that the Graham scan deletes the convexly dependent points D of $\{q_{J(1)}, \ldots, q_m\}$. We connect q_1^* and q_μ^* to obtain the convex hull representation of $\{q_{J(1)}, \ldots, q_m\} \setminus (K \cup D)$.

With the thereby updated set $Q := Q \setminus ((K \cup D)$, we recursively compute a superset of the vertices of $(\{p_2, \ldots, p_n\} \oplus Q)_{x_1 \geq 0}$ and add these points to R.

Theorem 5 *The above described algorithm* CONSTR-MINKOWSKI *correctly computes a set R containing all the vertices of $(P \oplus Q)_{x_1 \geq 0}$ in linear time, provided that the points in P and Q are sorted according to their x_1-value.*

Proof Correctness follows from Theorem 4 and Remark 1. The total running time for convex hull computations on the set of points in Q is bounded by $O(|Q|)$. This is because the convex-hull representation of $\{q_{J(1)}, \ldots, q_m\}$ after the deletion of the points in K and the deletion of convexly dependent points of $\{q_{J(1)}, \ldots, q_m\}$ can be repaired in constant time by connecting q_1^* with q_μ^*. Notice that the deletion of the convexly dependent points is done by the Graham scan anyway. The succeeding convex hulls of points in Q can be computed by continuing the Graham scan. Thus, the time which is required by the Graham scan is in total $O(m)$. This shows the claim.   □

4 Minkowski Sums with More than One Constraint

In this section, we show how to compute a set $R \subseteq (P \oplus Q)_{Ax \geq b}$ containing the vertices of $\mathrm{conv}(P \oplus Q)_{Ax \geq b}$ in time $O(N \log N)$ if the number of constraints in $Ax \geq b$ is fixed. If the number k of constraints is not fixed, our algorithm has a running time of $O(k \cdot \log k + k \cdot (N \log N))$.

First, we present an algorithm for the case of two constraints. Then we consider the case with three constraints, i.e., where the convex polygon $T = \{x \in \mathbb{R}^2 \mid Ax \geq b\}$ is a triangle. We then show how to reduce the computation of a set R containing the vertices of $\mathrm{conv}((P \oplus Q) \cap T)$ to a sequence of a fixed number of Minkowski sum computations with two constraints. Finally, for larger k, we triangulate the polygon $U = \{x \in \mathbb{R}^2 \mid Ax \geq b\}$ into k triangles, compute sets R_i, $i = 1, \ldots, k$, containing the vertices of the constrained Minkowski sums yielded by these triangles and then return the union of the R_i.

4.1 Minkowski Sums with Two Constraints

Consider the constrained Minkowski sum $(P \oplus Q)_{Ax \geq b} = \{x \mid x \in P \oplus Q, Ax \geq b\}$, where $Ax \geq b$ consists of two linear inequalities $a_1^T x \geq b_1$ and $a_2^T x \geq b_2$.

First, we sort P and Q w.r.t. increasing values of $a_1^T x$ and $a_2^T x$. This can be done in time $O(N \log N)$. After this preprocessing, any subset $P' \subseteq P$ and $Q' \subseteq Q$ can be sorted in time $O(N)$.

Consider the first inequality $a_1^T x \geq b_1$. By translation, we can assume that $b_1 = 0$. For a given $\gamma \in \mathbb{R}$, define the sets

$$P_L = \{p \in P \mid a_1^T p < -\gamma\},$$
$$P_= = \{p \in P \mid a_1^T p = -\gamma\} \quad \text{and}$$
$$P_R = \{p \in P \mid a_1^T p > -\gamma\}.$$

Likewise, define $Q_L, Q_=, Q_R$ according to whether $a_1^T q < \gamma, a_1^T q = \gamma$, or $a^T q > \gamma$. The number γ can be chosen in such a way that $|P_L| + |Q_R| \leq \lceil n/2 \rceil$ and $|P_R| + |Q_L| \leq \lfloor n/2 \rfloor$, and this can also be done in time $O(|P| + |Q|)$ by starting with a large γ and decreasing it in a plane-sweep manner.

Observe that the points in $P_L \oplus (Q_L \cup Q_=)$ and in $(P_L \cup P_=) \oplus Q_L$ do not satisfy the inequality $a_1^T x \geq 0$ and hence do not satisfy the system $Ax \geq b$. On the other hand, every point in $(P_R \cup P_=) \oplus (Q_R \cup Q_=)$ satisfies $a_1^T x \geq 0$, but it might or might not satisfy $a_2^T x \geq b_2$. This constraint still needs to be checked.

For the remaining points, even $Ax \geq b$ has to be checked. We have shown that the following formula holds:

$$(P \oplus Q)_{Ax \geq b} = \big((P_L \cup P_R \cup P_=) \oplus (Q_L \cup Q_R \cup Q_=)\big)_{Ax \geq b}$$
$$= (P_L \oplus Q_R)_{Ax \geq b} \cup (P_R \oplus Q_L)_{Ax \geq b}$$
$$\cup \big((P_R \cup P_=) \oplus (Q_R \cup Q_=)\big)_{a_2^T x \geq b_2}.$$

It leads to the following recursive algorithm CONSTR-MINKOWSKI$_2$ (Fig. 5) for $k = 2$.

The algorithm uses a *global* array R of points to which the recursive calls add points. When the algorithm terminates, R contains all vertices of conv$((P \oplus Q)_{Ax \geq b})$. We set $R = \emptyset$ in the beginning.

Theorem 6 *Algorithm* CONSTR-MINKOWSKI$_2$ *(Fig.* 5*) computes a set of points* $R \subseteq (P \oplus Q)_{Ax \geq b}$ *which contains all vertices of conv$((P \oplus Q)_{Ax \geq b})$. Its running time is* $O(N \log N)$, *where* $N = |P| + |Q|$. *The convex hull of a constrained Minkowski sum with two constraints has* $O(N \log N)$ *vertices.*

Proof Let $T(n)$ be the running time of the algorithm excluding the time required to sort P and Q w.r.t. $a_1^T x$ and $a_2^T x$ for the first time. Notice that the sorting in the recursion can be done in time $O(N)$. Likewise step (10) requires time $O(N)$, by Theorem 5. Neglecting floors and ceilings, we obtain the recursion

$$T(N) \leq 2 \cdot T(N/2) + O(N).$$

Algorithm: CONSTR-MINKOWSKI$_2$ $(P, Q, Ax \geq b)$
Input: Point-sets $P, Q \subseteq \mathbb{R}^2$ and 2 linear constraints given by $Ax \geq b$.
        The first constraint is $a_1^T x \geq b_1$, the other constraint is $a_2^T x \geq b_2$.
Output: A set of points $R \subseteq (P \oplus Q)_{Ax \geq b}$ which contains all
        vertices of $\text{conv}((P \oplus Q)_{Ax \geq b})$.

(1) Use translation on P and Q to obtain $b_1 = 0$.
(2) Sort the sets P and Q in increasing order w.r.t. $a_1^T x$ and $a_2^T x$.
(3) **if** $(|P| + |Q| \leq 2)$, **then**
(4)   add each point in $(P \oplus Q)_{Ax \geq b}$ to R
(5) **else**
(6)   Determine $\gamma \in \mathbb{R}$ such that $|P_R| + |Q_L| \leq \lfloor n/2 \rfloor$ and $|P_L| + |Q_R| \leq \lceil n/2 \rceil$.
(7)   The following calls add points to the global set R:
(8)   CONSTR-MINKOWSKI$_2$ $P_R, Q_L, Ax \geq b$
(9)   CONSTR-MINKOWSKI$_2$ $P_L, Q_R, Ax \geq b$
(10)   CONSTR-MINKOWSKI $P_R \cup P_=, Q_R \cup Q_=, a_2^T x \geq b_2$

Fig. 5 Algorithm for computing the Minkowski sum with two constraints.

This shows the claim.   □

We obtain the following corollary.

Corollary 3 *Let $Ax \geq b$ be a system of two constraints. The convex hull of $(P \oplus Q)_{Ax \geq b}$ can be computed in time $O(N \log^2 N)$ in general and in time $O(N \log N)$ if one of the two constraints is linearly sortable.*

4.2 Two Parallel Constraints

If the two constraints from above are of the form $a^T x \geq L$ and $a^T x \leq U$, then we can obtain a running time of $O(N)$ if the constraints are linearly sortable or the sets P and Q are pre-sorted, as we describe now. This is particularly the case if the constraints in the interval problems of Sect. 2 are bounding the length of the interval from above and below. Together with this extra trick, we obtain linear running times for all listed interval problems.

Suppose without loss of generality that the constraints are $L \leq x_1$ and $x_1 \leq U$. The parallel constraints form a *vertical strip* with a *width* of $w = U - L \geq 0$. We will show how to split the point-sets P and Q so that we obtain a number of Minkowski problems with only one constraint. The main idea is as follows: Split all points of P and Q into disjoint subsets $P_1, \ldots, P_u$ and $Q_1, \ldots, Q_v$ such that each subset is contained in a vertical strip of width $w/4$.

Now consider $P_i \oplus Q_j$. It follows that the resulting points are contained in a vertical strip of width $w/2$. Since the distance between L and U is w, we can conclude that at most one constraint L or U is located in this vertical strip of width $w/2$, or the

strip is either completely inside or outside of $L \leq x_1 \leq U$, and therefore we obtain a
Minkowski problem with one or zero constraints. We do not know the number $u + v$
of subsets of P and Q, but one subset P_i can only be combined with six subsets of Q,
since the subsets are disjoint and all other possible Minkowski sums contain points
that violate $L \leq x_1$ or $x_1 \leq U$. We thus have the following theorem.

Theorem 7 *Suppose that the points in P and Q are sorted w.r.t. $a^T x$. Then one can
compute a set containing the vertices of* $\mathrm{conv}((P \oplus Q)_{\alpha \leq a^T x \leq \beta})$ *in linear time.*

4.3 Minkowski Sums with an Arbitrary Number of Constraints

Suppose now that the system $Ax \geq b$ contains an arbitrary number k of constraints.
First, we compute the vertex representation of the polygon $U = \{x \in \mathbb{R}^2 \mid Ax \geq b\}$
and then triangulate U into at most $k - 2$ triangles. (By adding constraints, if neces-
sary, we can assume that U is bounded.) We thus reduce the problem of computing
a superset of the vertices of the convex hull of $(P \oplus Q)_{Ax \geq b}$ to the computation of k
such supersets for triangles in time $O(k \cdot \log k)$.

Let the triangle T be given by $T = \{x \in \mathbb{R}^2 \mid a_i^T x \leq b_i, \ i = 1, 2, 3\}$. In the fol-
lowing we explain how to reduce the computation of $\mathrm{conv}((P \oplus Q) \cap T)$ to the con-
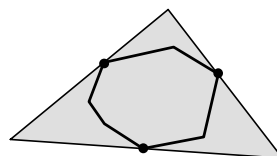strained Minkowski sum computation with two constraints.

Suppose that each edge of the triangle contains a point of $P \oplus Q$, see Fig. 6. In this
case, each vertex of $\mathrm{conv}((P \oplus Q) \cap T)$ is a vertex of $\mathrm{conv}((P \oplus Q) \cap C_i)$, where C_i is
the cone $C_i = \{x \in \mathbb{R}^2 \mid a_i^T x \leq b_i, a_{i+1}^T x \leq b_{i+1}\}$, where indices are taken modulo 3.
Therefore, we only need to show how to transform T into a triangle T' so that the
following two conditions hold:

(a) $(P \oplus Q) \cap T' = (P \oplus Q) \cap T$.
(b) Each edge of T' contains a point of $(P \oplus Q)$.

Consider the cone $C_1 = \{x \in \mathbb{R}^2 \mid a_1^T x \leq b_1, a_2^T x \leq b_2\}$, see Fig. 7(a). Now com-
pute a set R which contains all the vertices of $(P \oplus Q) \cap C_1$. Let x be the ver-
tex of the triangle defined by the constraints $a_1^T x \leq b_1$ and $a_3^T x \leq b_3$. Now we can
determine in time linear in R the point $r \in R \cap T$ which is first hit if we rotate
the constraint $a_1^T x \leq b_1$ around x in such a way that the vertex of T defined by
$a_1^T x = b_1$ and $a_2^T x = b_2$ is invalid, see Fig. 7(b). Clearly, we can replace the con-
straint $a_1^T x \leq b_1$ by its rotated variant and thereby obtain a triangle $\tilde{T}$ which satisfies
$\tilde{T} \cap (P \oplus Q) = T \cap (P \oplus Q)$. The edge defined by the new constraint contains a point
of $P \oplus Q$.

By repeating this operation above for each edge of T we can thus construct a
triangle T' which satisfies (a) and (b). Together with Theorem 6, this shows that we

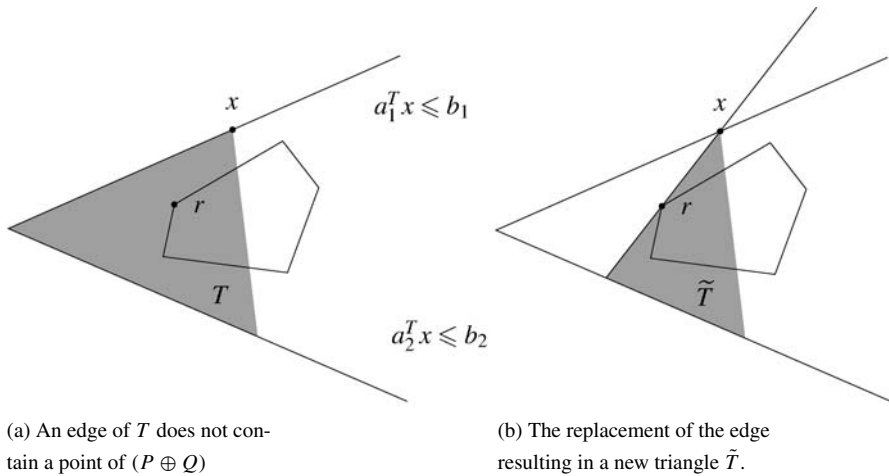**Fig. 6** Each edge contains a
point of $(P \oplus Q)$

(a) An edge of T does not contain a point of $(P \oplus Q)$

(b) The replacement of the edge resulting in a new triangle $\tilde{T}$.

Fig. 7 Reducing to two constraints when $r \notin T$

can compute a set R which contains all the vertices of $\mathrm{conv}((P \oplus Q) \cap T)$ in time $O(N \log N)$. Summarizing, we obtain the following theorem.

Theorem 8 *Given a set of k linear inequalities $Ax \geq b$ and point-sets $P, Q \subseteq \mathbb{R}^2$, one can compute a set $R \subseteq (P \oplus Q)_{Ax \geq b}$ containing all vertices of $\mathrm{conv}((P \oplus Q)_{Ax \geq b})$ in time $O(k \cdot \log k + k \cdot N \log N)$, with $N = |P| + |Q|$. The number of vertices is bounded by $O(k \cdot N \log N)$.*

4.4 Final Remarks

We close with some open problems. It is an interesting question whether the running time for 2 constraints can be further improved to linear time if all constraints are linearly sortable.

An interesting structural question is how many vertices $\mathrm{conv}((P \oplus Q)_{Ax \geq b})$ can have if $k \geq 2$. We only have the bound $O(k \cdot N \log N)$, which follows from our divide-and-conquer algorithm and triangulation. However, we suspect that this is not an exact bound. A related question is the following. If S is an arbitrary subset of $P \oplus Q$, how many vertices can $\mathrm{conv}(S)$ have? This question was also considered by Halman et al. [14]. Eisenbrand et al. [9] have shown that any subset $S \subseteq P \oplus Q$ has at most $O(N^{4/3})$ vertices.

Appendix: Modeling the Interval Problems

In this appendix, we show in more details how the interval problems can be modeled.

(a) This is the problem of maximizing the quasiconvex function $f(\ell, s) = s$ under the constraints $L \leq \ell \leq U$. These two constraints are linearly sortable, the constraint

$x_1 \geq 1$ is replaced by $x_1 \geq L$, hence our algorithm gives a running time $O(n)$ (Theorem 7) which matches the running time $O(n)$ in [10].

(b) To model this problem, we proceed as follows: Since the objective function does not depend on ℓ but on the weights instead, we have to use a different mapping which sets $p_j = (w_1 + \cdots + w_j, a_1 + \cdots + a_j)$ and $q_i = (-(w_1 + \cdots + w_{i-1}), -(a_1 + \cdots + a_{i-1}))$. Since all weights w_i are positive, we can discard intervals with negative or zero length by the constraint $x_1 \geq w_{\min}$, where $w_{\min} = \min\{w_i \mid i = 1, \ldots, n\}$. In order to discard intervals with a weight which is too small, we use the constraint $x_1 \geq c$, where $c = \max\{w_{\min}, L\}$. We then compute the constrained Minkowski sum $(\{p_1, \ldots, p_n\} \oplus \{q_1, \ldots, q_n\})_{c \leq x_1 \leq U}$ and maximize the quasiconvex function $f(x, y) = y/x$. Notice that since the weights are all positive, the numbers $w_1 + \cdots + w_i$ are sorted, and therefore the running time we obtain is $O(n)$ (Theorem 7), which matches [11].

(c) The function $f(\ell, s) = \ell$ is quasiconvex, the linear constraint is $s \geq b \cdot \ell$. The bias b is a constant, say $b = b_1/b_2$ for natural numbers b_1 and b_2. Then, the constraint is equivalent to $b_2 \cdot s - b_1 \cdot \ell \geq 0$. In the application, s counts the number of "preferred" characters, i.e., $s \in \{0, \ldots, n\}$. The sets P and Q used to model that problem thus have integer coordinates with absolute values $O(n)$. $b_2 \cdot s - b_1 \cdot \ell$ is linearly sortable, as its absolute values are also bounded by $O(n)$. This means that the problem is solved by maximizing $f(\ell, s) = \ell$ under the constraint $b_2 \cdot s - b_1 \cdot \ell \geq 0$. The constraint $x_1 \geq 1$ in the Minkowski sum can be omitted, since we are maximizing ℓ: When the array contains at least one preferred character (the other case is trivial), then the maximum ℓ is at least 1, and it does not matter that we are allowing intervals of zero or negative length. Our (worst-case) running time for this problem thus is $O(n)$, improving upon the (worst-case) trivial running time $O(n^2)$ in [2].

(d) This is the problem of maximizing the quasiconvex function $f(\ell, s) = s/\ell$ under the constraint $\ell \geq L$. We can model the problem as a constrained Minkowski sum with only one constraint $x_1 \geq L$ (replacing $x_1 \geq 1$). Since P and Q have x_1-coordinates which are integers with absolute values $O(n)$, the constraint is linearly sortable, and we obtain a linear-time algorithm for the problem. This improves upon the result from [16], where the running time is $O(n \log L)$, but only equals the running time $O(n)$, which results from the algorithm for the more general problem (b) in [10].

(e) Here one wants to maximize the quasiconvex function $f(\ell, s) = |s|/\sqrt{\ell}$ without extra constraint. Our algorithm yields running time $O(n)$, improving the results from [4], where *unconstrained* Minkowski sums were used to obtain an $O(n \log n)$ bound. We could now even generalize problem (e) by allowing extra parameters L and U. The task would be to find an interval with length at least L and at most U such that $f(\ell, s)$ is as large as possible. We obtain a running time of $O(n)$ for this generalized problem.

References

1. Agarwal, P.K., Amenta, N., Sharir, M.: Largest placement of one convex polygon inside another. Discrete Comput. Geom. **19**(1), 95–104 (1998)

2. Allison, L.: Longest biased interval and longest non-negative sum interval. Bioinf. Appl. Note **19**(10), 1294–1295 (2003)
3. Ben-Or, M.: Lower bounds for algebraic computation trees. In: Proc. of the 15th Annual ACM Symposium on Theory of Computing (STOC '83), pp. 80–86 (1983)
4. Bernholt, T., Hofmeister, T.: An algorithm for a generalized maximum subsequence problem. In: Proc. of the 7th Latin American Symposium on Theoretical Informatics (LATIN 2006), pp. 178–189 (2006)
5. Boyd, S., Vandenberghe, L.: Convex Optimization. Cambridge University Press, Cambridge (2004)
6. Chew, L.P., Kedem, K.: A convex polygon among polygonal obstacles: placement and high-clearance motion. Comput. Geom. **3**(2), 59–89 (1993)
7. Davies, P.L., Kovac, A.: Local extremes, runs, strings and multiresolution (with discussion). Ann. Stat. **29**(1), 1–65 (2001)
8. de Berg, M., van Kreveld, M., Overmars, M., Schwarzkopf, O.: Computational Geometry Algorithms and Applications. Springer, Berlin (1997)
9. Eisenbrand, F., Pach, J., Rothvoß, T., Sopher, N.B.: Convexly independent subsets of the Minkowski sum of planar point sets. Electron. J. Comb. **15**(1), 4 of Note 8 (2008)
10. Fan, T.-H., Lee, S., Lu, H.-I., Tsou, T.-S., Wang, T.C., Yao, A.: An optimal algorithm for maximum-sum segment and its application in bioinformatics. In: Proc. of the 8th International Conference on Implementation and Application of Automata (CIAA 2003), pp. 251–257 (2003)
11. Goldwasser, M.H., Kao, M.-Y., Lu, H.-I.: Linear-time algorithms for computing maximum-density sequence segments with bioinformatics applications. J. Comput. Syst. Sci. **70**(2), 128–144 (2005)
12. Guibas, L.J., Ramshaw, L.H., Stolfi, J.: A kinetic framework for computational geometry. In: Proc. of the 24th IEEE Symposium on the Foundations of Computer Science (FOCS '83), pp. 100–111 (1983)
13. Guibas, L.J., Sharir, M., Sifrony, S.: On the general motion-planning problem with two degrees of freedom. Discrete Comput. Geom. **4**(5), 491–521 (1989)
14. Halman, N., Onn, S., Rothblum, U.: The convex dimension of a graph. Discrete Appl. Math. **155**, 1373–1383 (2007)
15. Latombe, J.C.: Robot Motion Planning. Kluwer, Dordrecht (1991)
16. Lin, Y.-L., Jiang, T., Chao, K.-M.: Efficient algorithms for locating the length-constrained heaviest segments with applications to biomolecular sequence analysis. J. Comput. Syst. Sci. **65**(3), 570–586 (2002)
17. Lipson, D., Aumann, Y., Ben-Dor, A., Linial, N., Yakhini, Z.: Efficient calculation of interval scores for DNA copy numbers. In: Proc. of the 9th International Conference on Research in Computational Molecular Biology (RECOMB 2005), pp. 83–100 (2005)
18. Lozano-Perez, T., Wesley, M.A.: An algorithm for planning collision-free paths among polyhedral obstacles. Commun. ACM **22**, 560–570 (1979)