Well-separated pair decomposition and t-spanners

Frédérik Paradis - fpara058@uottawa.ca

University of Ottawa

October 30, 2015

Well-separated pair decomposition (WSPD)

- A well-separated pair decomposition (WSPD) for a point set $S \subseteq \mathbb{R}^d$ is a set of pairs $\{\{A_1, B_1\}, \{A_2, B_2\}, \ldots, \{A_m, B_m\}\}$ where:
  - for any $p, q \in S$, there exist a pair $\{A_i, B_i\}$, $1 \leq i \leq m$, such that $p \in A_i$ and $q \in B_i$, or $p \in B_i$ and $q \in A_i$;
  - for any pair $\{A_i, B_i\}$, $1 \leq i \leq m$, A_i and B_i are well-separated with respect to a separation ratio s.
- Two sets A and B are well-separated with respect to s if there are two circles of the same radius ρ such that these two circles are respectively enclosing A and B and that the distance between the two circles is at least $s\rho$.

Example of a well-separated pair

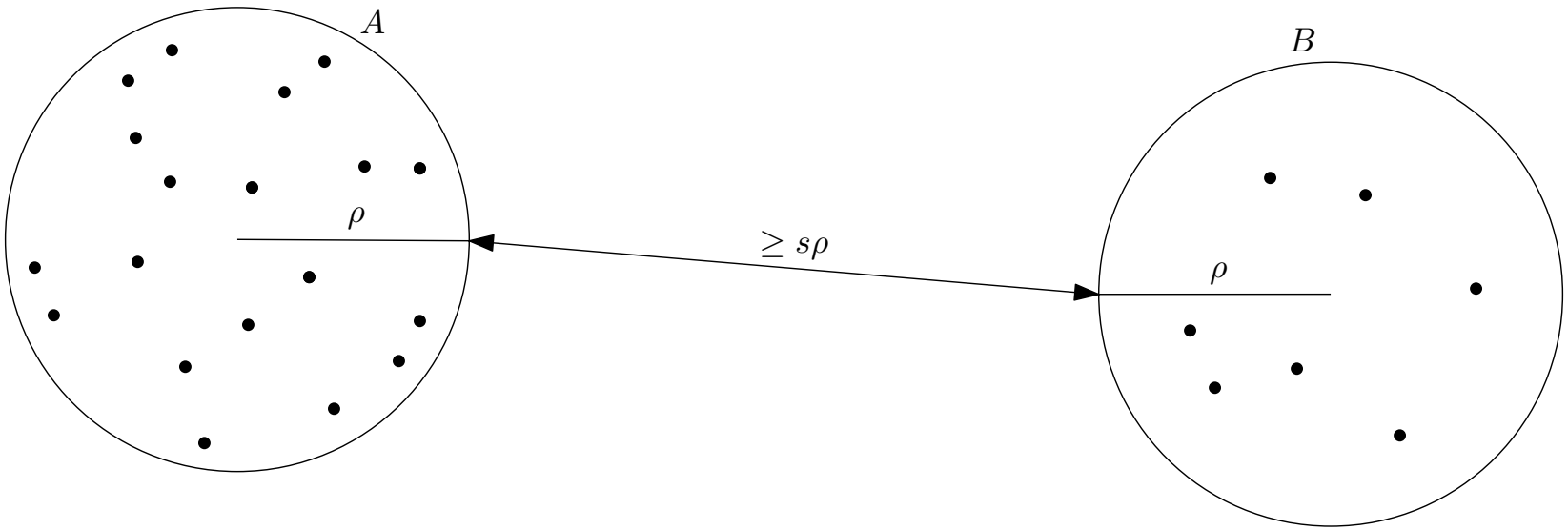


Figure: Visual representation of well-separated pair

Example of a WSPD

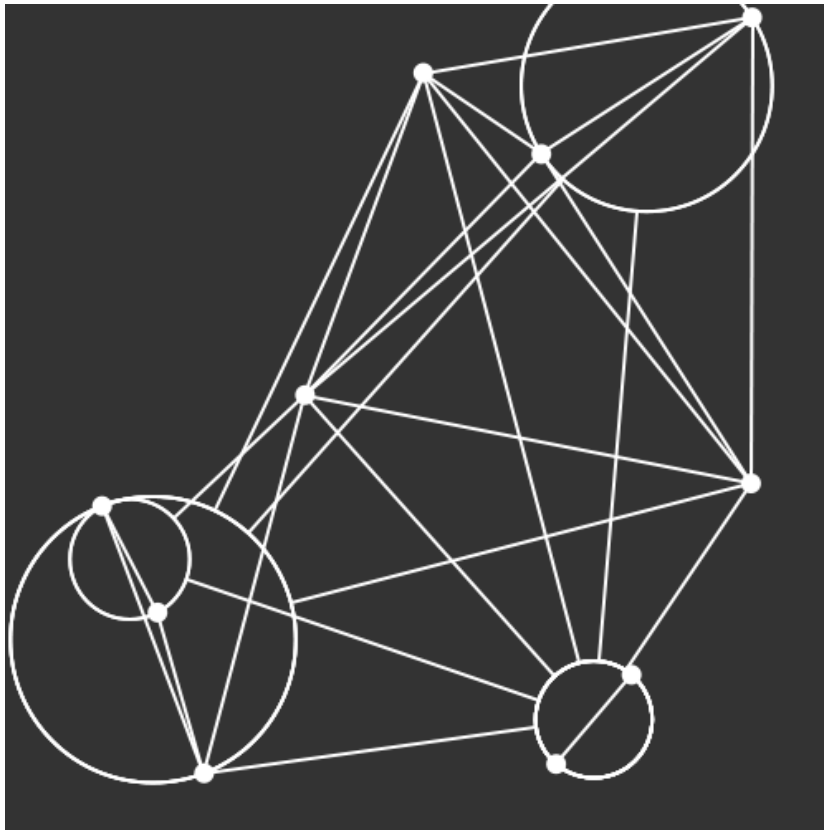
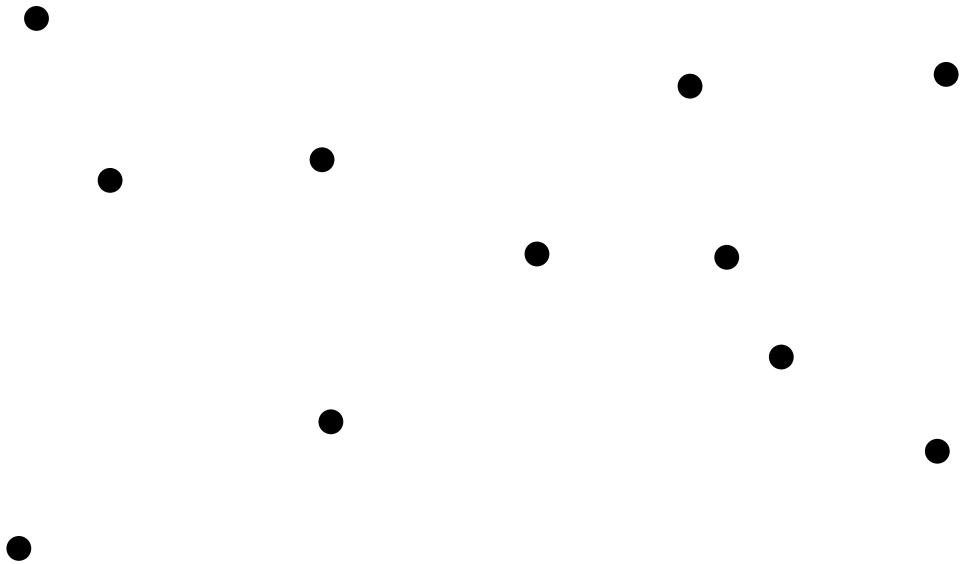


Figure: Visual representation of WSPD with $s = 2$. Figure done with an applet in [1].

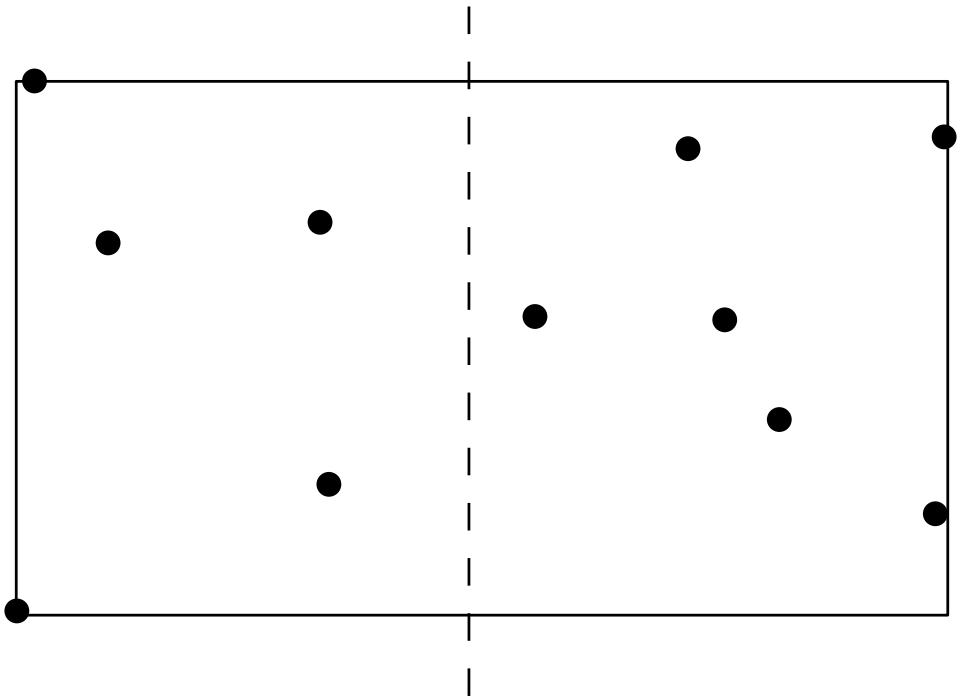
Split tree

- The split tree is a data structure that represents the action of recursively splitting a point set S in two using the bounding box of the point set.
- Bounding box $R(S)$ is the smallest axes-parallel d-dimensional rectangle containing S.
  - Let $L_i(R(S))$ be the length of the side in the i-th dimension.
  - Let $L_{max}(R(S))$ be length of the longest side of $R(S)$.

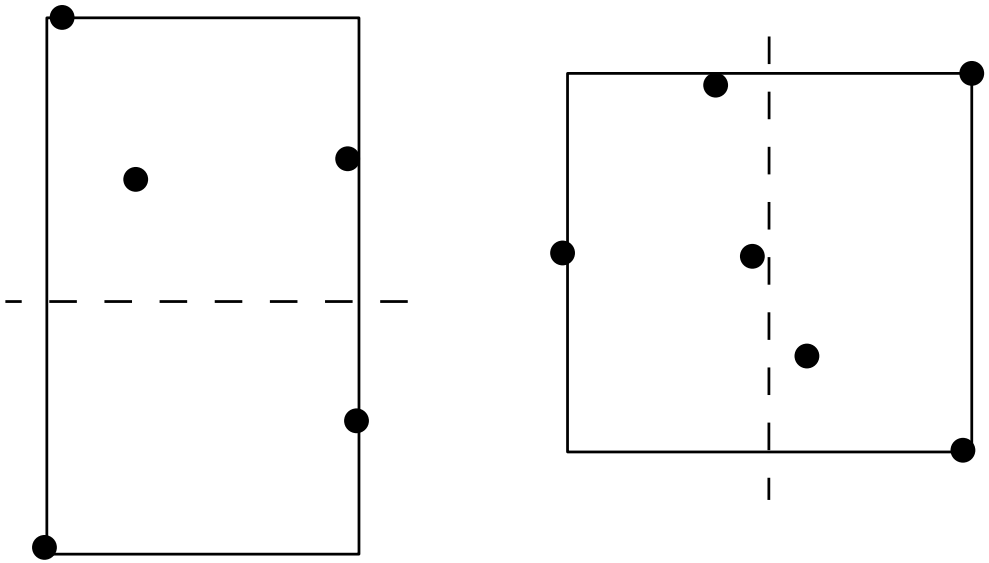
Example of the algorithm



Example of the algorithm



Example of the algorithm



The algorithm of the split tree

function SplitTree(S)
    Let u be the node for S
    if $|S| = 1$ **then**
        $R(u) := R(S)$ ▷ $R(S)$ is a rectangle which each side has a length of zero.
        Store in u the only point in S.
    else
        Compute $R(S)$
        Let the i-th dimension be the one where $L_{max}(R(s)) = L_i(R(S))$
        Split $R(S)$ along the i-th dimension in two same-size rectangles and obtain S_v and S_w.
        $v :=$ SplitTree(S_v)
        $w :=$ SplitTree(S_w)
        Store v and w as the left and right children of u.
        $R(u) := R(S)$
    end if
    return u
end function

Time complexity

- Because the tree can be very unbalanced, this algorithm runs in $O(n^2)$ time.
- It can be reduced to $O(n \log(n))$ with the following technique.

The algorithm of the split tree

- The goal is to always call the recursion on at most $n/2$ points.
- The method is to walk along the point list from the beginning and the end simultaneously and stop when one of the sides reaches the line that separates the rectangle in two.

The algorithm of the split tree

- The goal is to always call the recursion on at most $n/2$ points.
- The method is to walk along the point list from the beginning and the end simultaneously and stop when one of the sides reaches the line that separates the rectangle in two.
- Let S_i^j be the j-th coordinate of the i-th point in S such that S is sorted for each dimension and $p(S_i^j)$ be the point.
- Let $h(R(S))$ be the line that splits $R(S)$ in two along his longest side.

The algorithm of the split tree

Let the node u.
$size := |S|$
repeat
  $j := 1$, $k := |S|$
  Let the i-th dimension be the one where $L_{max}(R(S)) = L_i(R(S))$
  while $S_i^{j+1} < h(R(S))$ **and** $S_i^{k-1} > h(R(S))$ **do**
    $size := size - 1$, $j := j + 1$, $k := k - 1$
  end while
  Let v and w be the left and right children of u.
  if $S_i^{j+1} > h(R(S))$ **then**
    Recursively call this function for v for the points on the sides of j, remove them from S and let $u := w$.
  else if $S_i^{k-1} < h(R(S))$ **then**
    Recursively call this function for w for the points on the sides of k, remove them from S and let $u := v$.
  end if
until $size \leq \frac{n}{2}$
Recursively call this function for u for the remaining points.

Time complexity

Because each call to the recursion involves less than or equal to $n/2$ points and runs in $O(n)$ time, this algorithm should run in $O(n \log(n))$.

WSPD computation

The bounding boxes are used to compute the WSPD. Deciding if two pairs are well-separated or not is then done in constant time.

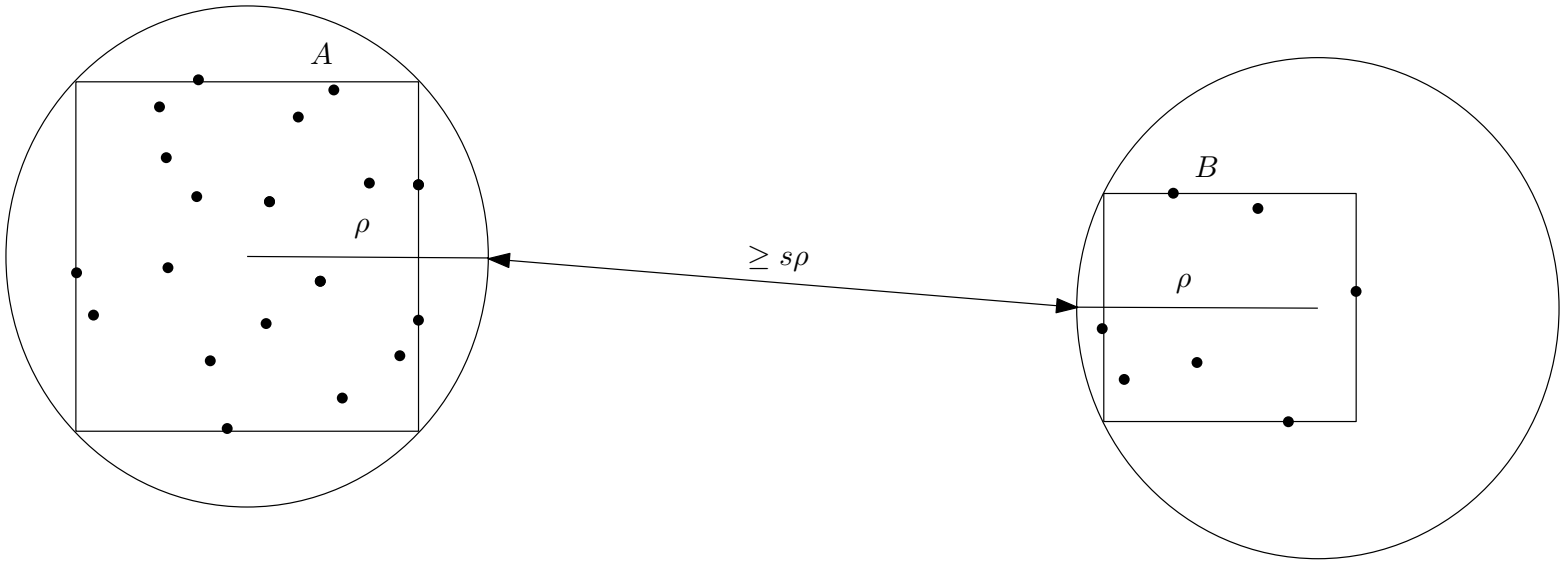


Figure: Visual representation of a well-separated pair computed with the bounding boxes

WSPD algorithm

From two nodes u and v in the split tree that have disjoint point sets, this algorithm finds well-separated pairs of sets so that, for each pair $\{A, B\}$, $A \subseteq S_v$ and $B \subseteq S_w$.

function FindPairs(v, w)
  if S_v and S_w are well-separated with respect to s **then**
    return The set $\{\{u, v\}\}$
  else if $L_{max}(R(v)) \leq L_{max}(R(w))$ **then**
    Let w_l and w_r be the left and right children of w.
    FindPairs(v, w_l)
    FindPairs(v, w_r)
  else
    Let v_l and v_r be the left and right children of v.
    FindPairs(v_l, w)
    FindPairs(v_r, w)
  end if
end function

WSPD algorithm

From the split tree, a WSPD with respect to s is found.

function FindWSPD(T, s)
    for each node u that is not a leaf in the split tree T **do**
        Let v and w be the left and right children of u.
        FindPairs(v, w)
    end for
end function

Correctness

- Each leaf of the split tree has a point associated to. Because FindPairs is called on each internal node, the two children of each node that is a common ancestor of each pair of leaves are involved in a call to FindPairs.
- Because each pair of leaves has only one lowest common ancestor, the algorithm finds unique well-separated pair for each pair of points.

Time complexity

- Each time the recursion tree splits in two, there is one more pair added to the decomposition. So, the algorithm run-time is in the number of pairs in the final decomposition.
- The number of pairs is in $O(s^d n)$. Callahan and Kosaraju proved this in their original paper [3].

t-spanner computation using a WSPD

- A t-spanner of the point set S is a graph such that S is the vertex set of the graph and that the euclidean length of a path between two points p and q in S is less than or equal to $t|pq|$ where $|pq|$ is the euclidean distance between p and q.
- For each pair $\{A, B\}$, add an edge between them. Choose it arbitrarily.
- Then, you have a t-spanner where $t = (s + 4)/(s - 4)$.
- The complete graph which clearly is a t-spanner has $\binom{n}{2}$ edges. With this construction, we have a t-spanner which has fewer edges than the complete graph. Actually, this construction gives a graph with a linear number of edges (i.e $O(s^d n)$).

Two lemmas on WSPD

Lemma

Let $\{A, B\}$ be a well-separated pair with respect to s. Let $p, p' \in A$ and $q \in B$. Then, $|pp'| \leq (2/s)|pq|$.

Proof.

We know that $|pp'| \leq 2\rho$ where ρ is the radius of the enclosing circles of A and B. We also know that $|pq| \geq s\rho$. Using these two inequalities, we obtain the result of the lemma. □

Two lemmas on WSPD

Lemma

Let $\{A, B\}$ be a well-separated pair with respect to s. Let $p, p' \in A$ and $q, q' \in B$. Then, $|p'q'| \leq (1 + 4/s)|pq|$.

Proof.

By the triangle inequality, we have:

$$|p'q'| \leq |p'p| + |pq| + |qq'|$$

From Lemma 1, we obtain:

$$\begin{aligned} |p'q'| &\leq (2/s)|pq| + |pq| + (2/s)|pq| \\ &= (1 + 4/s)|pq| \end{aligned}$$

Correctness of the t-spanner computation

- Let $\{A, B\}$ be a well-separated pair with respect to s in a WSPD. Let $[a, b]$ be the edge connecting A to B. Let any point $p \in A$ and $q \in B$.
- This proof is by induction on the length of the path P_{pq}. Suppose there is a t-path between any pair of points which distance is less than $|pq|$ and that $s > 2$.

Correctness of the t-spanner computation

From the triangle inequality, the assumptions and the fact that $|pa| \leq (2/s)|pq| \Rightarrow |pa| < |pq|$ and $|bq| \leq (2/s)|pq| \Rightarrow |bq| < |pq|$ according to Lemma 1, we have:

$$|P_{pq}| \leq t|pa| + |ab| + t|bq|$$

From Lemma 1 and 2, we obtain:

$$\begin{aligned} |P_{pq}| &\leq t(2/s)|pq| + (1 + 4/s)|pq| + t(2/s)|pq| \\ &= \left(\frac{4(t+1)}{s} + 1\right)|pq| \end{aligned}$$

Then, t must be equal to $(s+4)/(s-4)$ so that P_{pq} is a t-path between p and q.

t-spanners with bounded diameter

- The diameter of a graph is the minimum number of edges for which it is possible to have a path from any point to any other point in the graph.
- Computing such t-spanners is done by choosing closely the edge between each pair in the WSPD. The two endpoints of the chosen edge are called representatives of their respective point set.

Construction

- The construction is from the split tree.
- For each node of the split tree, the edge leading to a child with a bigger subtree is called **heavy edge** and the other one is called **light edge**.
- The representative of each point set associated to a node is the point associated to the leaf reachable with a chain containing only heavy edges.

Claim

Let any well-separated pair $\{A, B\}$. Suppose that from any point in A to the representative of A, there is a path containing at most $\lg(n)$ edges in G. Suppose the same for B. So, the diameter of a path P_{pq} between $p \in A$ and $q \in B$ is at most $2\lg(n) - 1$.

Proof.

The diameter of the path P_{pq} noted $||P_{pq}||$ is:

$$\begin{aligned} ||P_{pq}|| &\leq \lg(n/2) + \lg(n/2) + 1 \\ &= \lg(n) - \lg(2) + \lg(n) - \lg(2) + 1 \\ &= 2\lg(n) - 1 \end{aligned}$$

Proof of the assumption

Theorem

From any point $p \in A$ to the representative of $a \in A$, there is a path containing at most $\lg(n)$ edges in G.

Proof.

- Let the pair $\{C, D\}$ be the one where $p \in C$ and $a \in D$ without loss of generality. This proof is by induction on the size of A.
- a is also the representative of D because D is on the chain of heavy edges between a and A.
- The size of C is less than or equal to $|A|/2$. By induction, the number of edges on the path from p to the representative of C is at most $\lg(|A|/2) \leq \lg(n) - 1$.

t-spanners with other specific properties

- t-spanners with degrees bounded to a constant. These can contain a lot of edges.
- t-spanners with diameter bounded to a constant. These can have vertices with high degrees.
- t-spanners with degrees and diameter bounded $O(\log n)$.
- There are many other possibilities...

References I


French introduction to WSPD with some good applets.
`http://homepages.ulb.ac.be/~slanger/cg/P/WSPD/index.html`.


S. Arya, D.M. Mount, and M. Smid.
Randomized and deterministic algorithms for geometric spanners of small diameter.
In *Foundations of Computer Science, 1994 Proceedings., 35th Annual Symposium on*, pages 703–712, Nov 1994.


Paul B. Callahan and S. Rao Kosaraju.
A decomposition of multi-dimensional point sets with applications to k-nearest-neighbors and n-body potential fields.
J. ACM, 42:546–556, 1992.

References II


Paul B. Callahan and S. Rao Kosaraju.
Faster algorithms for some geometric graph problems in higher dimensions.
In *Proceedings of the Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, SODA '93, pages 291–300, Philadelphia, PA, USA, 1993. Society for Industrial and Applied Mathematics.


Michiel Smid.
The well-separated pair decomposition and its applications.
In Teofilo Gonzalez, editor, *Handbook of Approximation Algorithms and Metaheuristics*, pages 53–1 – 53–12. Chapman & Hall/CRC, Boca Raton, 2007.