Analysis of Helmut's Algorithms

Jonathan Lenchner
lenchner@us.ibm.com

Initial version: February 24, 2005
1st revision: February 27, 2005
2nd revision: March 15, 2005

The Setup: We are given a set of N points in the plane, a horizontal line L, and are asked to cover the points by squares centered somewhere along the line such that either (i) the sum of the edge lengths of the covering squares is minimized or (ii) the sum of the areas of the covering squares is minimized.

The final evening of the Bellairs workshop Helmut discussed the following algorithm:

Algorithm (HELMUT1): Consider the points in order of decreasing distance from the line L. Without loss of generality assume that all points lie above L (i.e. we may reflect all points below L so they are above L and we are faced with essentially the same problem). First find the furthest point p_1 from L, cover p_1 with two adjacent squares S_L and S_R exactly of the same height as p_1 but such that S_L has p_1 at its upper right corner, and S_R has p_1 at its upper left corner. There are no points above S_R and S_L so we may remove all points covered by S_R, S_L from consideration and recurse, finding the next furthest point from L and so forth. In the case where two points are precisely the same distance from L we break ties arbitrarily.

Helmut argued that since the optimal solution (henceforth referred to as OPT) must contain a square which includes p_1, of the same height as p_1, and similarly for each point p_i that we process, HELMUT1 must be a factor 2 approximation to OPT. He then very cleverly went on to show that one could extend the factor 2 approximation to the optimal horizontal line location problem. While the extension of the constant factor approximation to the line location problem appears to hold, the exact constant factor is not correct. If one takes the case of two points both of unit height above L and separated by a distance of $2+\epsilon$, HELMUT1 requires the use of two boxes of edge length 2 with abutting edges at each point, for a total of four such boxes, while OPT requires just the use of a single box of edge length $2+\epsilon$ centered precisely half way between the projection of the two points on L. Hence we see that HELMUT1 is at best a factor 4 approximation.

The problem in Helmut's original argument is the following: if we are considering boxes around a highest remaining point p_i, it is true that OPT must contain p_i in a box of height at least that of p_i and edge length twice $height(p_i)$, and so include at least half as much area as that used in the two boxes S_{Li}, S_{Ri} - however - we cannot conclude that this results in a factor 2 approximation since as we work through the points p_i we may end up with overlapping boxes. It is a simple matter though to conclude that HELMUT1 is in fact a factor 4 approximation - we just observe that HELMUT1 never results in more than 2 squares overlapping above a given point on the line L. There are two cases (i) where we consider overlap at the projection on L of a point p_i used in the algorithm, (ii) where we consider overlap at other points. In case (i) we have the situation in Figure 1.

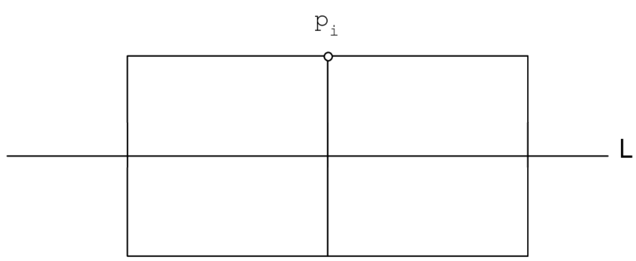

FIGURE 1. Case where overlap occurs at the projection on L of a point p_i used in the HELMUT1 algorithm.

The question is whether any other box used in HELMUT1 can overlap at the projection of p_i on L. Clearly if such a box were bigger than either of the two abutting boxes, it would contain p_i. Also, the box, since bigger, would have been placed prior to considering the point p_i so we would never have had the chance to process p_i in the first place. On the other hand, if the box were smaller than either of the abutting boxes, and contained the projection of p_i, then it would be consumed by the two boxes in Figure 1 and hence could not have been placed. It follows in this case that we cannot have additional overlap.

Case (ii) is depicted in Figure 2.

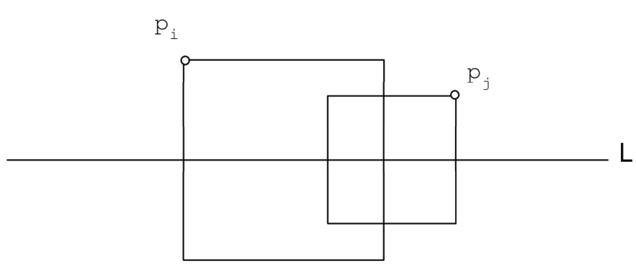

FIGURE 2. More generic overlap case for HELMUT1

Without loss of generality we start with the square, call it S_i, corresponding to point p_i, with p_i pinned to the top left corner, and consider the possible overlaps. We assume that S_i is as large as, or larger than, the other initially overlapping square. (A symmetrical argument can be made if S_i is smaller.) Considerations akin to those in case (i) allow us to conclude that S_i cannot be overlapped by a square whose left-most extent extends further to the left than p_i. Hence we necessarily have the situation depicted in the Figure. Now, suppose the overlap area is overlapped by an additional square, induced by a point p_k. As noted, p_k cannot lie to the left of S_i. It cannot lie inside S_i since then it would have been engulfed by S_i and never processed. Analogously, it cannot lie above S_i since then it would have been processed before p_i and so p_i would never have been processed. The same considerations go when considering whether p_k can lie within or above the box S_j corresponding to p_j. Finally, p_k cannot lie to the right of p_j for the same reasons that it could not lie to the left of p_i. We thus conclude that at most two boxes ever overlap when we run HELMUT1. Now consider the boxes placed running HELMUT1. At most half of the area of any two boxes placed by HELMUT1 lies outside OPT. If we consider $H_A = \sum Area(S_i)$ for all boxes S_i placed by HELMUT1 then at most $.5H_A$ lies outside OPT. Of the at least $.5H_A$ that lies inside OPT there can be at worst a factor 2 overlap. Hence H_A, which is the HELMUT1 approximation for area, is at worst a factor 4 approximation to OPT. We know the factor 4 is tight by the example of two points of unit height above L separated by a distance of $2 + \epsilon$. Precisely the same argument applies if we replace cumulative area by cumulative edge length.

We have thus proved:

Lemma 1. *HELMUT1 is a factor 4 approximation to OPT for square covering where cost is measured in terms of the sum of the area of the squares, or sum of the edge lengths of the squares.* □

Helmut also mentioned in passing the following algorithm.

Algorithm (HELMUT2): Again consider the points in order of decreasing distance from the line L. Without loss of generality assume that all points lie above L. First find the furthest point p_1 from L, cover p_1 with a square S_1 exactly of the same height as p_1 centered at the projection of p_1 on L. There are no points above S_1 so we may remove all points covered by S_1 from consideration and recurse, finding the next furthest point from L and so forth. As in the prior algorithm, in the case where two points are precisely the same distance from L we break ties arbitrarily.

One is tempted to argue that HELMUT2 is a factor 2 approximation to OPT: For any point p_i processed using HELMUT2, either the right half of the box centered at p_i or the left half of the box centered at p_i must be entirely contained in OPT - of course the problem again is overlap. If we consider three points of unit height, separated successively by distances of $1 + \epsilon$, then HELMUT2 gives a covering by three squares of edge length 2, while OPT covers all three points with a single box of edge length $2 + 2\epsilon$. This example is due to Estie in an email message to me. We thus see that HELMUT2 is at best a factor 3 approximation to OPT. In fact we have the following:

Lemma 2. *HELMUT2 is a factor* 3 *approximation to OPT for square covering where cost is measured in terms of the sum of the area of the squares, or sum of the edge lengths of the squares.*

Proof. One can see that HELMUT2 results in at most a double covering (overlapping) of points, just like we saw for HELMUT1. Of course this argument alone leads only to a factor 4 approximation so we must do a more careful accounting. To this end, consider a square S in OPT. We consider those squares in HELMUT2 corresponding to points $\{p_{i_j} : p_{i_j} \in S\}$ and argue that these squares cannot contain more than three times the area of S, and analogously cannot have total edge length which is more than three times the edge length of S. The same will then follow for all of HELMUT2 and all of OPT.

Arguing as we did regarding overlaps, it is easy to see that at most two of the boxes S_{i_j} associated with the points $p_{i_j} \in S$ processed by HELMUT2 actually protrude outside of S, one on the left and one on the right. If we write FC for the fraction of the area of the boxes S_{i_j} which is actually contained in S, $OVERLAP$ for the area that is covered by two boxes from the $\{S_{i_j}\}$ inside S, $NONOVERLAP$ for the area covered by precisely one box from the $\{S_{i_j}\}$ inside S, and $LEFTPRO$ and $RIGHTPRO$ for the areas associated with boxes protruding from the left and right we have

(1)
$$FC = \frac{OVERLAP + NONOVERLAP}{LEFTPRO + RIGHTPRO + 2\,OVERLAP + NONOVERLAP}$$

The key observation now is that

$$LEFTPRO \leq \frac{OVERLAP + NONOVERLAP}{2}, \tag{2}$$

$$RIGHTPRO \leq \frac{OVERLAP + NONOVERLAP}{2}. \tag{3}$$

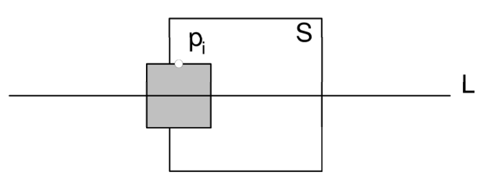

FIGURE 3. Analysis of the case where a box protrudes to the left from a box in OPT in the HELMUT2 algorithm.

See Figure 3. Since we must have an entire square of edge length 2 $height(p_i)$ inside S, the protrusion of the shaded box in Figure 3 outside S must be matched by at least twice as much inside S. Equations (2) and (3) follow. Substituting equations (2) and (3) into (1) gives

$$(4) \qquad FC \geq \frac{OVERLAP + NONOVERLAP}{3\ OVERLAP + 2\ NONOVERLAP} \geq \frac{1}{3}$$

which is what we sought to prove. Precisely the same argument holds if we replace cumulative area by cumulative edge length of squares. □

Finally, we observe that it is easy to modify either HELMUT1 or HELMUT2 to get an algorithm that is a factor 2 approximation in edge length. For example, when processing the successive points p_i encountered in either HELMUT1 or HELMUT2, if the associated square S_i would result in an overlap with already existing square S_{i-k} then just grow S_{i-k} enough to include p_i, keeping the vertical edge furthest from p_i at the same point on L. If placing S_i would overlap two squares, S_{i-k} and $S_{i-k'}$ say, grow the one which requires the smallest edge extension. Call the resulting algorithms HELMUT1a and HELMUT2a respectively. For the sake of simplicity of some of our later arguments, we only grow the square S_{i-k} to capture the point p_i if the squares S_i and S_{i-k} would intersect with non-zero area.

Lemma 3. *HELMUT1a and HELMUT2a are factor 2 approximations to OPT for cumulative edge lengths of squares.*

Proof. We give the proof just for HELMUT2a; the argument for HELMUT1a is analogous.

As we process points p_i using HELMUT2a attribute to each point p_i a line segment s_i along L as follows. If processing p_i resulted in the placement of a square S_i centered at the projection of p_i in L then attribute to p_i the projection on L of a horizontal edge of S_i If, on the other hand, processing of p_i resulted in the growing of a prior square S_{i-j} to just capture p_i, attribute to p_i the projection on L of the portion of the horizontal edge of the expanded S_{i-j} needed to capture p_i. (This amount is at most the distance of p_i to L since S_{i-j} only grows to capture p_i in the event there would otherwise have been overlap with the square centered at p_i.) We must show that the lengths of the segments is no more than twice the edge lengths of squares in OPT.

It suffices to show that for any square S in OPT, the segments s_i associated with points $p_i \in S$ processed by HELMUT2a cannot have total edge length which exceeds twice the edge length of S. Equations (1), (2) and (3) from earlier remain the same, but now there is no OVERLAP, so we have

$$FC = \frac{NONOVERLAP}{LEFTPRO + RIGHTPRO + NONOVERLAP}$$

$$LEFTPRO \leq \frac{NONOVERLAP}{2}$$

$$RIGHTPRO \leq \frac{NONOVERLAP}{2}$$

so

$$FC \geq \frac{NONOVERLAP}{2\ NONOVERLAP} = \frac{1}{2}$$

as needed. □

Unfortunately it is equally easy to see that HELMUT1a and HELMUT2a are not constant factor approximations to OPT for area. For example, if we are using HELMUT2a, consider n consecutive points at height 1 separated one from the next by a distance of $1+\epsilon$. If we process the points left to right using HELMUT2a we cover all points with one square of edge length $n+(n-1)\epsilon$, and so area $O(n^2)$, while we can clearly cover all points with n overlapping squares each of edge length 2, so with total area $4n$.

In the case of area, it is possible that OPT contains overlapping squares, as illustrated in Figure 4.

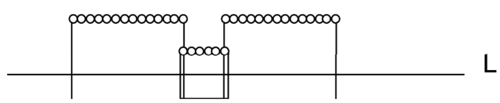

FIGURE 4. OPT for area may have to contain overlapping squares.

Thus, we could try to modify HELMUT2, say, so that when we process successive points p_i, instead of placing the associated square S_i we check first if we can more cheaply expand a neighboring square S_{i-k} to cover p_i, and if so, expand S_{i-k} and keep processing successive points. However, this strategy can be counterproductive.

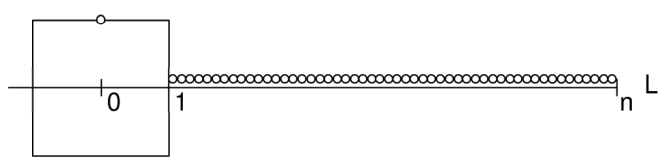

FIGURE 5. Simple greedy absorption of points by area may be inefficient.

In Figure 5 there is a point at height 1 at the 0 point along L and a large collection of points all at height $1/n$ beginning at the point 1 unit along L which are very close together (i.e. at a distance $\Delta \ll 1/n$, one from the next, out to a distance n along L). If we apply greedy absorption by area, processing first the points at unit height and then the points at height $1/n$ from left to right, we obtain a single square of height n and so area $4n^2$, while OPT consists of a square at height 1 with left edge at 0 and $(n-2)/2$ squares of edge length $2/n$ for a total cost of $4+(\frac{2}{n})^2\ \frac{n-2}{2}$ so greedy absorption by area is not even a constant factor approximation to OPT.

A more clever area absorption policy might be to absorb not just the point p_i, but the hypothetical square S_i, if S_{i-k} can be grown so to cover

S_i but at lower total cost. However, if we employ this policy, and consider again the example of three points, each at unit height, but at separations of $1+\epsilon$, we are not led to do any absorption, and so the algorithm remains a factor 3 approximation to OPT.

Running Time Analysis and the Line Location Problem

We now consider the running time of HELMUT2 and HELMUT2a. First consider HELMUT2. Sort the points by x-coordinate and separately by distance from the line L in time $O(n \log n)$ and process the points in order of decreasing distance from L. As the point p_i at distance d_i from L is processed, we throw away points which are within horizontal distance d_i from p_i. This takes time $O(\log n + k_i)$ time where k_i is the number of points within d_i from p_i. Since we do this up to n times with $k_1 + \cdots + k_l = n$ the total running time is $O(n \log n)$.

For HELMUT2a we do the same as before, but as we process points, along with throwing away points within horizontal distance d_i of p_i, we also keep a sorted list $\mathcal{S}$ of all endpoints e_{i_L}, e_{i_R}, which are the left and right crossings of L by the square S_i corresponding to the point p_i, along with a pairing so that we know which e_{i_L} corresponds to which e_{i_R}. To consider whether to keep a point p_i which would introduce new endpoints e_{i_L}, e_{i_R} we must check to see whether there is an earlier point p_{i-j} with $e_{i_L} < e_{(i-j)_R} < e_{i_R}$ or $e_{i_L} < e_{(i-j)_L} < e_{i_R}$. To do this, we just need to consider the points in $\mathcal{S}$ to the immediate left and right of p_i. If, say $e_{i_L} < e_{(i-j)_R} < e_{i_R}$, then we do not add p_i but instead set $e_{(i-j)_R} = e_{i_R}$. This additional step just adds constant time at each step to the time taken for HELMUT2 so we conclude that HELMUT2a can also be performed in time $O(n \log n)$.

We now consider the line location problem, initially using HELMUT2 in the case where we are trying to minimize the edge lengths of the covering squares. We want to claim that we can find a horizontal line and a covering of points by squares which has total edge length which is at most a factor 3 from the the optimal covering by squares given the optimal horizontal line location. To this end, consider a horizontal line coming in from $+\infty$. At heights sufficiently far above any points, all points will be covered by a single box just covering the lowest point (or just covering any one of the lowest points if there are more than one). As we move the line down, the single box eventually no longer covers one or more points. An example of such a critical point is illustrated in Figure 6. Let us denote the horizontal distance between the two points p and q by δ and the vertical distance between the points by λ. Then the situation in Figure 6 will occur so long as $\delta \geq \lambda$.

As the line L moves vertically downward, while the point p remains furthest from L, the point q is ultimately no longer captured by the square S_p corresponding to p. The case where q is at the corner of S_p is not essentially different. As we move the line L downward, with all points still lying below

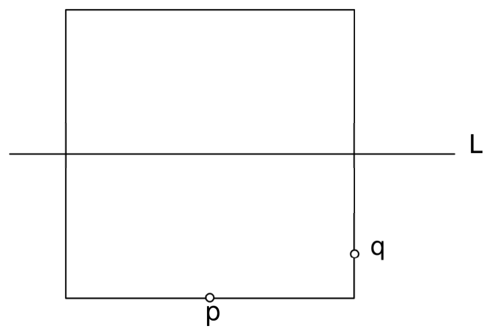

FIGURE 6. A critical point as the line L moves down from $+\infty$, for the case $\delta \geq \lambda$.

L it is not hard to see that the critical points as illustrated in Figure 6 are the only points when new squares appear in the running of HELMUT2. Once L moves below some points there are additional critical points. The point q, though still on the right edge say, could be on the opposite side of L from p. Such a situation occurs if $\frac{1}{2}\lambda \leq \delta \leq \lambda$. Finally, in the case $\delta \leq \frac{1}{2}\lambda$, we obtain a critical point when, as illustrated in Figure 7, a point q which had resided inside the box S_p, resides on the border of S_p at a distance to L equal to that from p to L. As L continues downward, the box S_p disappears, being replaced by a box S_q, which then contains p in its interior.

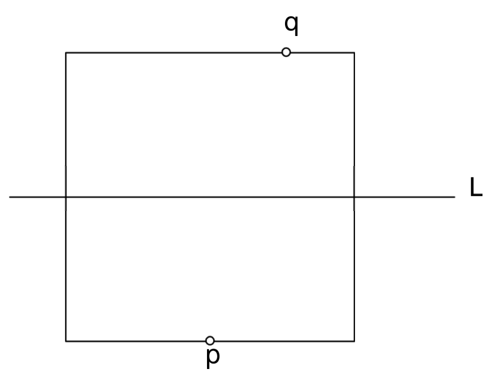

FIGURE 7. Additional critical points as line L moves through some points. The case $\delta \leq \frac{1}{2}\lambda$.

When the line L is below all points, we return to critical points as depicted in Figure 6, but with the defining points flipped, and again $\delta \geq \lambda$.

In total then, critical points can occur when the horizontal line L is (vertically) equidistant from two points, or when a point is as far from L as it is horizontally from another point. There are $2\binom{n}{2} = O(n^2)$ such possible line positions to consider. Between critical line positions, if we have covering squares $S_1, ..., S_k$, with $S_1, ..., S_j$ growing and $S_{j+1}, ..., S_k$ shrinking as we move L horizontally downward, then between critical line positions $y = K$

and $y = K'$ we can write for the corresponding edge lengths e_i,

$$\begin{aligned} e_1 &= e_1^* + \Delta \\ &\cdot \\ &\cdot \\ e_j &= e_j^* + \Delta \\ e_{j+1} &= e_{j+1}^* - \Delta \\ &\cdot \\ &\cdot \\ e_k &= e_k^* - \Delta \end{aligned}$$

for constants $e_1^*, ..., e_k^*$ and Δ varying from 0 to $|K - K'|$. Clearly then, $\sum e_i$ being linear in Δ, is minimized at one of the endpoints, $y = K$ or $y = K'$.

In the case where we are minimizing area, the situation is not much different. The algorithm behaves the same and so has boxes appearing at the same critical points. Between critical points, we are interested in the behavior of the function which is the sum of the edges squared. We are thus interested in the minimum of the function

$$(5) \quad P(\Delta) = (e_1^* + \Delta)^2 + \cdots + (e_j^* + \Delta)^2 + (e_{j+1}^* - \Delta)^2 + \cdots + (e_k^* - \Delta)^2$$

which is just a quadratic polynomial in the single variable Δ. Therefore, if we are concerned about minimizing area, for each pair of successive candidate line positions considered initially, we must also possibly consider an additional line position given by differentiating the polynomial $P(\Delta)$ and setting the resultant equation equal to zero. Since the earlier Lemma 2 holds for arbitrary non-zero powers of edge length the same essential argument as above gives a variant of HELMUT2 that approximates the total cost for optimal line location if cost is measured as $\sum e_i^3$, $\sum e_i^4$, or $\sum e_i^5$. The case $\sum e_i^3$ yields up to two additional line locations between original pairs, $\sum e_i^4$ yields up to three additional line locations, and $\sum e_i^5$ yields up to four additional line locations.

In all cases we have $O(n^2)$ line placements to test, and at each candidate line placement we run the initial HELMUT2 algorithm, resulting in a total running time of $O(n^3 \log n)$. If we call the result of running HELMUT2 for any particular horizontal line L PSEUDO-OPT, then PSEUDO-OPT is at most 3 times OPT for L, whether we are considering PSEUDO-OPT and OPT for edge length, or PSEUDO-OPT and OPT for area. The optimal line location then finds the minimum of PSEUDO-OPT. If L^* is the line which minimizes OPT, then PSEUDO-OPT over all lines is less than or equal to PSEUDO-OPT over L^*, so PSEUDO-OPT over all line placements is also a factor 3 approximation to OPT over all line placements.

Finally, we turn our attention to the analysis of the algorithm HELMUT2a in the context of optimal line location. Recall that HELMUT2a gives a factor

2 approximation to OPT for edge length, but is not a constant approximation to OPT for area - hence we confine our attention to the edge length problem. The critical point analysis (perhaps more properly described as critical line location analysis) is a little bit more involved than it was for HELMUT2. We still have the critical points as described in HELMUT2 but in addition we have critical points like those illustrated in Figure 8 when two squares are just about to overlap, and instead HELMUT2a tells us to choose growth of the larger square.

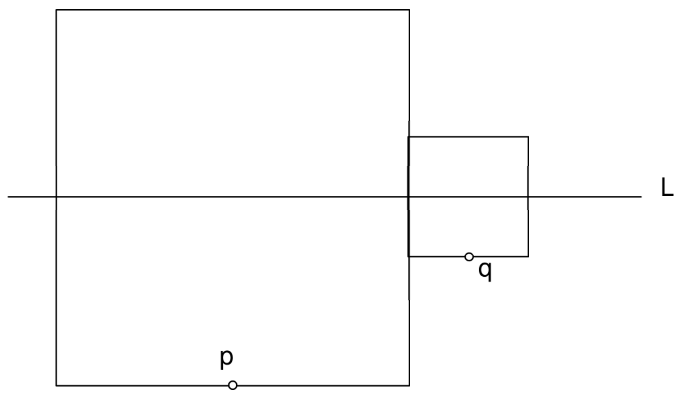

FIGURE 8. Special critical point when running HELMUT2a in the case $\delta > \lambda$.

As before, denoting the horizontal separation of the points p and q by δ and the vertical separation by λ, if in addition we let d denote the distance of the closer point to L, then as long as $\delta > \lambda$ such a critical point occurs when

$$\lambda + 2d = \delta.$$

Solving this equation for d actually yields two candidate line locations, one when q is the closer point to L (as drawn) and both points lie below L and one when p is the closer point to L and both points lie above L. If $\delta < \lambda$ there are no special critical points associated with p and q as a result of running HELMUT2a (though we still have those from running HELMUT2, discussed earlier). On the other hand, if $\delta = \lambda$, we may place L anywhere in the vertical separation between p and q, and find that the squares S_p and S_q intersect only at their edges, and so HELMUT2a does not call out special action beyond that from HELMUT2. Figure 9 illustrates this point. Once L moves beyond one of the points, the square centered at the other point captures the other point, so these become the critical points.

Finally, we need to mention what happens once a square has grown to absorb another point on one of its sides, say to its right. Call the point that has been thus captured r. The further interaction of the square S which has just captured r and points further to the right which have not been processed, is not impacted by anything else but the location of r. In this

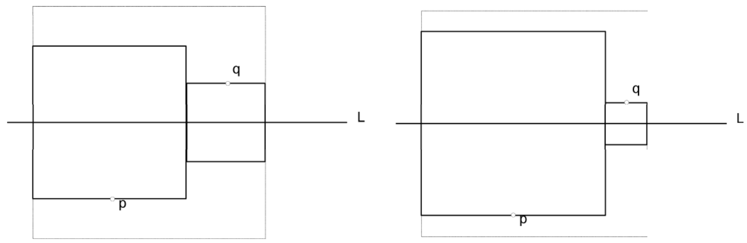

FIGURE 9. The analysis of HELMUT2a: the special case where the horizontal separation δ between p and q is the same as the vertical separation λ. L may be placed anywhere in the vertical separation between p and q and the boxes S_p and S_q intersect just at their edges.

case, a point r' lying to the right of r and having a horizontal separation δ from r will give rise to a critical line location when the line is precisely a distance δ from r'. This situation is illustrated in Figure 10.

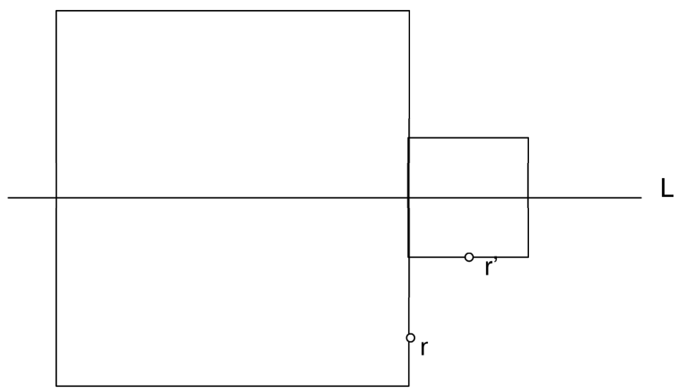

FIGURE 10. A critical point after the square on the left has grown to absorb the point r.

In total, then, we see that the running of HELMUT2a introduces only at most $O(n^2)$ new critical line locations to those already considered in the running of HELMUT2. Between critical line locations the pseudo-optimal solutions given by HELMUT2a clearly change linearly with the edge lengths of some squares growing linearly and the edge length of others shrinking linearly, just as in HELMUT2. Hence the OPT of the PSEUDO-OPTs is found by running HELMUT2a just at the critical locations, and so HELMUT2a gives at worst a factor 2 approximation to OPT for edge length, and can be computed in $O(n^3 \log n)$ time.

From Boxes to Discs

So far we have talked just of OPT for squares, while the more realistic problems have to do with discs. The points along L are thought to be broadcast locations, and we are trying to reach sensors or receiving stations in the plane. Broadcasts are assumed to be made radially. Thus, we state the more realistic variant of what we have studied thus far:

More Realistic Problem: We are given a set of N points in the plane, a horizontal line L, and are asked to cover the points by discs centered somewhere along the line such that either (i) the sum of the radii of the covering discs is minimized or (ii) the sum of the areas of the covering discs is minimized.

It is also reasonable to consider cost models that are functions of other powers of the broadcast radius.

The analysis of the more realistic problem is facilitated by the fact that squares are "not that far from" circles in a sense in which we now make precise. First consider the problem of minimizing the sum of the radii of covering discs. Let OPT_{DISC} denote the optimum such solution, i.e. $OPT_{DISC} = min(\sum_i r_i)$ where r_i are the radii of discs, whose union covers all points, and the min is taken over all such coverings. Analogously, let OPT_{BOX} denote the optimum such solution for boxes (squares), i.e. $OPT_{BOX} = min(\sum_i r_i)$ where r_i is half the edge length of squares, whose union covers all points, and again the min is taken over all coverings. Given OPT_{DISC}, we can find a covering with squares of the same total "radii" by simply taking axis aligned squares which circumscribe the circles. Hence $OPT_{BOX} \leq OPT_{DISC}$. Additionally, any square covering **induces** a circle covering with a factor $\sqrt{2}$ greater total radii, where we just replace each square with its circumscribing circle. Hence

$$OPT_{BOX} \leq OPT_{DISC} \leq \sqrt{2}\, OPT_{BOX}.$$

Theorem 1. *Given a set of N points in the plane, and a horizontal line L, if we are trying to cover the points by a set of discs centered on L such that the sum of the radii is minimized, then*

(i) Using HELMUT1, PSEUDO-OPT is at worst a factor 4 approximation to OPT_{DISC} and induces a disc covering that is at worst a factor $4\sqrt{2}$ approximation to OPT_{DISC}.
(ii) Using HELMUT2, PSEUDO-OPT is at worst a factor 3 approximation to OPT_{DISC} and induces a disc covering that is at worst a factor $3\sqrt{2}$ approximation to OPT_{DISC}.
(iii) Using HELMUT2a, PSEUDO-OPT is at worst a factor 2 approximation to OPT_{DISC} and induces a disc covering that is at worst a factor $2\sqrt{2}$ approximation to OPT_{DISC}.

The same constant factor approximations hold for the case of the optimal line location problem.

Proof. Consider the case (iii) of HELMUT2a, the other cases are analogous. If $OPT_{BOX} = OPT_{DISC}$ the result obviously holds, so we need only concern ourselves with the other extreme, when $OPT_{BOX} = \frac{1}{\sqrt{2}} OPT_{DISC}$. PSEUDO-OPT is not an underestimate of OPT_{BOX} so can only be as small as $\frac{1}{\sqrt{2}} OPT_{DISC}$, and as large as $2OPT_{DISC}$. In all cases then it is at worst a factor 2 approximation. The fact that the induced covering is a factor $2\sqrt{2}$ approximation follows immediately from the fact that $OPT_{DISC} \leq \sqrt{2}\, OPT_{BOX}$. □

Next consider the problem of minimizing the area of the covering discs. Again use the notation OPT_{DISC} and OPT_{BOX}, but this time to denote the optimal coverings with respect to area. Circumscribing the discs of OPT_{DISC} with axis aligned squares shows that $OPT_{BOX} \leq \frac{4}{\pi} OPT_{DISC}$ and then circumscribing the squares of OPT_{BOX} with discs gives

$$\frac{\pi}{4} OPT_{BOX} \leq OPT_{DISC} \leq \frac{\pi}{2} OPT_{BOX}.$$

Our final result is then:

Theorem 2. *Given a set of N points in the plane, and a horizontal line L, if we are trying to cover the points by a set of discs centered on L such that the cumulative area of all covering discs is minimized, then*

(i) Using HELMUT1, PSEUDO-OPT is at worst a factor $\frac{16}{\pi}$ approximation to OPT_{DISC} and induces a disc covering that is at worst a factor 8 approximation to OPT_{DISC}.
(ii) Using HELMUT2, PSEUDO-OPT is at worst a factor $\frac{12}{\pi}$ approximation to OPT_{DISC} and induces a disc covering that is at worst a factor 6 approximation to OPT_{DISC}.

The same constant factor approximations hold for the case of the optimal line location problem.

Proof. HELMUT2a is not a finite approximation to OPT for area, hence we have just the two cases. Consider just the case (ii) of HELMUT2, since the other case is analogous. For area, $OPT_{BOX} \leq \frac{4}{\pi} OPT_{DISC}$ so PSEUDO-OPT $\leq 3OPT_{BOX} \leq \frac{12}{\pi} OPT_{DISC}$. Also, PSEUDO-OPT $\geq OPT_{BOX} \geq \frac{2}{\pi} OPT_{DISC}$. So in both cases PSEUDO-OPT is a factor $\frac{12}{\pi}$ approximation. By virtue of $OPT_{DISC} \leq \frac{\pi}{2} OPT_{BOX}$, the induced covering adds another multiplicative factor of $\frac{\pi}{2}$ yielding, in total, a factor 6 approximation. □