Shooting Bricks with Orthogonal Laser Beams: A First Step towards Internal/External Map Labeling

Maarten Löffler* Martin Nöllenburg†

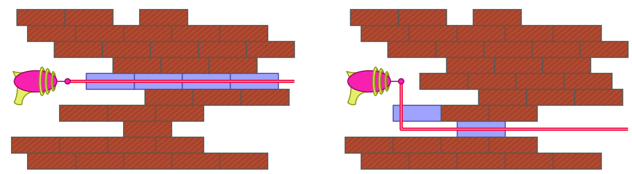
Abstract

We study several variants of a hybrid map labeling problem that combines the following two tasks: (i) a set A of points in a rectangle R needs to be labeled with rectangular labels on the right boundary of R using (axis-aligned) orthogonal one-bend polylines called *leaders* to connect points and labels; (ii) a maximum subset B' of a set B of fixed internal congruent rectangular labels in R needs to be selected such that B' is an independent set of labels and no leader intersects any label in B'. We also call the points in A *aliens*, the labels of B *bricks*, and the leaders *laser beams*. Then the problem translates into every alien shooting a laser beam so that in total as few bricks as possible are destroyed. We provide algorithms and NP-hardness results for different variants of the problem.

1 Introduction

Assume that we are given a rectangular map R with n points that are to be labeled by rectangular labels. It is required for readability of the map that no label overlaps another label or any of the input points. In *internal* labeling models, each label must be close to the labeled point, i.e., it usually needs to touch the point on its boundary. For instance, in the p-position model for $p \in \{1, 2, 4\}$ labels must touch the points at one of p admissible corners. Maximizing the number of labels that can be placed is NP-hard for $p = 2$ and $p = 4$ and in some cases even for $p = 1$ [4, 7].

An alternative *external* labeling model, in which all n input points can always be labeled (for sufficiently large R), is known as *boundary labeling* [1–3, 6]. In this model all labels are placed on one, two, or four sides of the boundary of R. Points are connected to their labels with arcs called *leaders*. In order to keep the map clean, leaders are usually required to be crossing-free and to have a prescribed simple shape, e.g., rectilinear



(a) (b)

Figure 1: An alien surrounded by bricks. (a) Shooting straight destroys four bricks. (b) The optimal solution.

polylines with at most one bend. Typical optimization criteria in boundary labeling are minimizing the total leader length or minimizing the total number of bends.

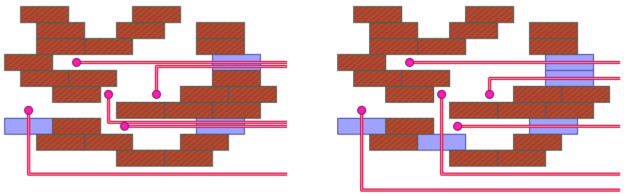
In the boundary labeling model, however, *all* labels must be on the boundary of R, even though some points would allow for an internal label, which is generally more favorable from the application's perspective. One of the open problems mentioned by Bekos et al. [1] and Kaufmann [6] is to study map labeling in a mixed model, where some points receive internal (fixed-position) labels and the remaining points are labeled externally. In such a mixed model, an additional requirement is that no leader intersects any of the internal labels.

In this paper, as a first step towards map labeling in the mixed model, we assume that the n input points are already partitioned into two sets A and B. The points in A must all be labeled externally using so-called *po*-leaders (they first run *p*arallel to and then *o*rthogonal to the labeled side of R) [2], which have by definition at most one bend. We consider the *one-sided* boundary labeling model, where all labels are placed on the right side of R. The points in B, on the other hand, can only be labeled internally using congruent rectangular labels with a fixed position. Depending on the placement of the leaders for A, some points in B may or may not be labelable. The goal is to label all points in A and to maximize the number of labeled points in B.

In the following we call the points in A *aliens*. We identify each point in B with its fixed label and call the labels in B *bricks*. Figure 1 shows an example. Since we do not want labels to occlude any points in A we can assume that no brick in B contains an alien in A. Instead of connecting each alien with a leader to the right

*Department of Computer Science, University of California, Irvine, mloffler@uci.edu

†Department of Computer Science, University of California, Irvine and Institute for Theoretical Informatics, Karlsruhe Institute of Technology, noellenburg@kit.edu



(a) (b)

Figure 2: An ABP instance with five aliens. (a) For non-spaced ABP at least three bricks need to be destroyed. (b) For spaced and crossing ABP at least five bricks need to be destroyed.

side of R, we say that each alien has a laser gun that can shoot a one-bend (axis-aligned) orthogonal *laser beam* to the right side of R. Each laser beam consists of a (possibly empty) vertical *laser segment* followed by a horizontal *laser ray*. Consequently, we can say that a laser beam *destroys* all bricks that it intersects. The map labeling problem then translates into the problem that every alien must shoot its laser gun exactly once in a way that minimizes the total number of destroyed bricks. We call this problem the *aliens-and-bricks problem* (ABP).

1.1 Parameters

Even in this simple setting (just one bend per leader and one position for internal labels), there are still several variants of the problem that can be of interest. We investigate the following three independent parameters and their effects on the complexity of the problem.

1. Bricks are either all disjoint or they overlap each other; in the latter case we need to find an independent set of bricks. Accordingly, we call an ABP instance *disjoint* or *overlapping*. We may further require that any selected brick does not occlude a given (e.g., the top left) corner of any other brick.

2. Laser beams can either cross each other at 90-degree angles or they cannot cross each other. We call an ABP instance *crossing* or *non-crossing*, accordingly.

3. Horizontal laser rays can either come arbitrarily close to each other or they have to stay at least one unit apart. Accordingly, we call an ABP instance *spaced* or *non-spaced*. We may further extend the spacing requirement to the whole laser beams (including the vertical segments), i.e., a fixed-width strip along each laser beam must not intersect any other laser beam; we call such an ABP instance *fully spaced*.

Figure 2 shows two examples of different variants that we study.

The requirement in the overlapping variant that no brick may occlude the top left corner of any other brick relates to the 1-position model in map labeling, where all labels are placed with their top left corner on the corresponding point of interest. Then the requirement simply means that all points – whether labeled or not – must remain visible.

In the classical external map labeling setting, crossings can usually be avoided at no additional cost. Interestingly, this is not the case anymore in our problem, so we explicitly study the problem with and without allowing laser beams to cross.

Spaced variants of ABP have the advantage that we can immediately attach unit-height labels at their vertical midpoints to the leader ends without producing any label overlaps. In non-spaced versions of the problem, we either need to use extra bends in a track-routing area [2] in order to separate the labels vertically, or the labels must be placed in multiple columns to the right of R. Spacing of laser beams, in particular for both the vertical segments and the horizontal rays, also has a positive effect on the readability of the drawing as it makes different laser beams visually easier to distinguish.

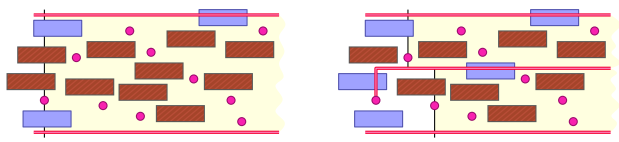
2 Non-Spaced Laser Beams

In this section we consider non-spaced laser beams. We note that in this case the issue of crossings is irrelevant: any crossing could be removed at no additional cost by shortening the vertical laser segment involved in a crossing so that the laser beam turns horizontal just before hitting the other horizontal laser ray. So we can restrict ourselves to non-crossing laser beams.

2.1 Disjoint Bricks

When the interiors of the bricks are all disjoint, we can solve the problem in polynomial time using dynamic programming. We define subproblems in half-strips that extend to the right and are bounded by two laser rays and a vertical line on the left that goes through an alien. Because we never use crossings, this alien must then shoot somewhere between the two rays, which subdivides the problem into two smaller subproblems.

Let $a \in A$ be an alien and consider the right-open half-strip S that is defined by the vertical line ℓ_a through a, the y-coordinate y_t of the lowest horizontal laser ray above a, and the y-coordinate y_b of the highest laser ray below a. Furthermore, let $B_a \subseteq B$ be the set of bricks that are stabbed by ℓ_a and lie completely between



(a) (b)

Figure 3: (a) A subproblem defined by a current alien (indicated by the black vertical line ℓ), two laser rays above and below it, and a marking of the bricks stabbed by ℓ. (b) A possible way for the current alien to shoot, and the resulting two subproblems.

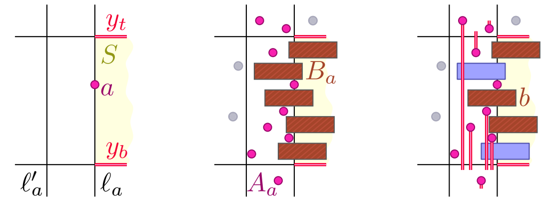
y_b and y_t. Then the subproblem for all points in S is in fact defined by S and additionally an indexing of B_a that indicates which bricks of B_a have already been shot by previous aliens. Figure 3 shows an example of a subproblem and how it divides into two independent smaller subproblems.

The algorithm is now standard dynamic programming. We store the solution of each subproblem, starting with the smallest ones (defined by the rightmost aliens). To solve a subproblem, we just go over the linear number of combinatorially different possible ways to shoot the current beam, look up the resulting two subproblems, and add the number of new bricks that we shot to their joint cost. We then choose the solution with minimum possible cost. The number of newly shot bricks can be computed by sweeping the horizontal laser ray over the half-strip of the subproblem and keeping track of the number of intersected bricks. This clearly takes linear time for each subproblem.

Since we use non-spaced laser beams there is an optimal solution in which no two laser beams intersect each other, even if crossings were allowed. This is because as soon as a vertical laser segment reaches an existing horizontal laser ray, it can follow that ray without destroying any other bricks. Therefore splitting a subproblem into two independent smaller subproblems along the current laser ray yields a correct algorithm that computes an optimal solution.

However, it is not so clear that the algorithm is efficient, since there could in principle be an exponential number of subsets of B_a that are marked as unshot, and therefore an exponential number of subproblems to solve. So, we proceed by showing that this is not the case and that it suffices to consider a linear number of subsets of B_a.

Lemma 1 *Let S be a half-strip defined as before by a leftmost alien a and the y-coordinates $y_b < y_t$ of two horizontal laser rays. Then the number of subsets of B_a that can be shot vertically by aliens on the left of ℓ_a is at most $|B_a|$.*



(a) (b) (c)

Figure 4: (a) An alien a and two horizontal rays at height y_t and y_b. (b) The set of aliens A_a in the vertical strip between ℓ'_a and ℓ_a, and the set of bricks B_a intersecting ℓ_a. (c) The leftmost brick b in B_a that is not shot determines the shooting direction of all aliens to the right of b.

Proof. Consider the line ℓ'_a that lies one brick-width to the left of ℓ_a, see Figure 4(a). Observe that no alien on the left of ℓ'_a can shoot bricks in B_a without also hitting S. Therefore, we are only concerned with the set $A_a \subset A$ of aliens in the vertical strip between ℓ'_a and ℓ_a, see Figure 4(b). Since beams cannot go through S, any alien in A_a that lies between the lines $y = y_b$ and $y = y_t$ must either shoot above y_t or below y_b. Aliens below y_b that shoot above y_b must actually shoot above y_t and, similarly, aliens above y_t that shoot below y_t must actually shoot below y_b. Aliens below y_b that shoot down or aliens above y_t that shoot up cannot influence the subset of bricks being shot.

Now, let $b \in B_a$ be a brick, and assume that the configuration of laser beams belonging to aliens in A_a is such that b is the leftmost brick among those in B_a that is *not* shot. We claim that this completely determines the subset of bricks that are shot. Indeed, any aliens to the right of the left side of b cannot shoot through b, and therefore must shoot in the opposite direction, as illustrated in Figure 4(c). Note that an alien below y_b (or above y_t) could still shoot either up or down, but its laser beam must stay below y_b (or above y_t) and therefore this choice does not influence the subset of B_a. On the other hand, every alien to the left of b can still shoot in either direction, but by definition all bricks that extend further left than b are shot, and these aliens cannot influence the subset of bricks to the right of the left side of b. Note that it could be that the aliens to the left of b are not capable of shooting all bricks to the left of b; in that case b cannot be the leftmost brick that is not shot. We conclude that the entire subset of bricks of B_a that are shot is fixed.

Since there are $|B_a|$ bricks in B_a to choose as the leftmost brick that is not shot, we conclude that there are at most $|B_a|$ possible subsets that can be shot. $\square$

For a given alien a and the vertical line ℓ_a there is a linear number of possible locations y_t for the lowest horizontal beam above a and a linear number of locations

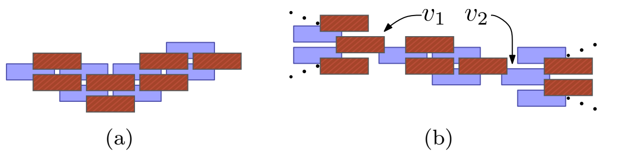


Figure 5: (a) An edge consists of a diagonal strip of width 2 of overlapping bricks that can make turns. An edge has two states, with the same cost. (b) A vertex is simply a single brick that overlaps up to three edges. For each edge, its two incident vertices overlap a different state of the edge; if v_1 is present then v_2 cannot be present too.

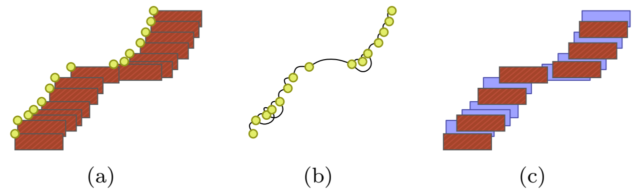


Figure 6: (a) When the bricks are congruent and cannot contain top left corners of other bricks, they can only form monotone chains. (b) The intersection graph of this sequence. (c) A maximum independent set can be found by greedily selecting bricks from left to right.

y_b for the highest beam below it. For each such triple (a, y_t, y_b) Lemma 1 tells us that we need to consider a linear number of subsets of B_a that can be shot vertically. So we have a total of $O(n^4)$ possible subproblems.

Looking up a specific subproblem in constant time is not trivial since it is not clear how to index a particular subset of bricks. We suggest to order all bricks in B, e.g., lexicographically from left to right and top to bottom and then use for each half-strip defined by the three parameters (a, y_t, y_b) a simple search tree according to the order of B to access the correct value corresponding to a particular subset of bricks of B_a. By Lemma 1, this tree has linear size, but its height can also be linear. Therefore, looking up the value of a particular subproblem takes linear time in the worst case. This yields the following time and space bounds:

Theorem 2 *The disjoint, non-spaced ABP can be solved in $O(n^6)$ time using $O(n^4)$ space.*

2.2 Overlapping Bricks

For overlapping bricks it is generally NP-hard to find a maximum independent set of bricks, even when there are no aliens. Klau and Mutzel [7] mention that this can be proven using a reduction from the NP-complete maximum independent set problem in planar graphs with maximum degree 3 [5], but they do not provide details, so we briefly sketch how this can be done. Given a planar input graph of degree at most 3, we "draw" the graph using edges made of diagonal double chains of overlapping bricks, as illustrated in Figure 5(a). Within each edge, there are exactly two maximum independent sets possible. Then, we use a single brick for each vertex as in Figure 5(b), where the two vertices incident to the same edge intersect two bricks belonging to different states of the edge. This ensures that we can only take the maximum number of bricks from the edge if not both incident vertices are selected. The reduction from maximum independent set is now immediate.

Theorem 3 *The overlapping, non-spaced ABP without further restrictions is NP-hard.*

However, if we assume that no brick in B contains a fixed corner (say, the top left one) of any other brick, then the structure of the overlapping bricks becomes much simpler. Let $B' \subseteq B$ be a maximal set of bricks such that the intersection graph of B' is connected. We call such a set of bricks a *chain*. The following lemma is illustrated in Figure 6(a).

Lemma 4 *Let B' be a chain of bricks. Then the point set consisting of the top left corners of all bricks in B' forms an xy-monotone sequence.*

Proof. Let p and q be two points in this set, and assume that the lemma is not true, i.e., the x-coordinate of p is smaller than the x-coordinate of q, while the y-coordinate of p is larger than the y-coordinate of q. Clearly, the bricks of p and q cannot overlap, otherwise p's brick would contain q. However, as the intersection graph of B' is connected, there must be a path from p to q in the intersection graph. Let P be the shortest such path. If this path contains another pair of points that are in the same configuration as p and q, we recurse on that subpath, so we assume that this is not the case: all consecutive pairs in the path are to the bottom left / top right of each other. But since q is not to the top right of p, the path must contain a triple of consecutive points u, v, w such that the direction changes from u, v to v, w, e.g., v is to the top right of u, but w is to the bottom left of v. Now we claim that if the bricks are congruent, then u must intersect w, meaning P was not the shortest path, contradiction. □

We call a chain that satisfies Lemma 4 a *monotone* chain. For monotone chains the independent set problem can be easily solved by picking an independent set of bricks greedily from one end of the chain, as seen in Figure 6(c).

Lemma 5 *A maximum independent set for a monotone chain $B' \subseteq B$ can be computed in linear time after sorting the bricks.*

Proof. The algorithm picks greedily the leftmost brick in the sequence, removes it together with all bricks that it intersects and continues to pick the next leftmost brick. It is clear that the set of chosen bricks is independent. To show that it is also a maximum independent set, assume we are given a maximum independent set M. If the leftmost of the bricks in M is actually the leftmost brick in B' the two sets agree in the first element and we recurse. Otherwise, we can replace the leftmost brick b in M by the leftmost brick b' in B'. Since b' is below and to the left of b it cannot intersect any other brick in M and thus the new set is again a maximum independent set of B'. Now the greedy independent set and M agree on the first element and we recurse. This process is a cardinality-preserving transformation of M into the greedy independent set, which concludes the proof. □

We would now like to use the previous DP-algorithm again. We need to make a few modifications. The value of any subproblem, which is still defined as before, is the number of unshot bricks in its half-strip including those intersecting ℓ_a. Whenever we need to compute the value of a particular location for the current laser beam, we need to consider a maximum independent set M of the unshot bricks that are in the current subinstance but in none of the two subproblems. So for every chain of unshot bricks we begin to build an independent set greedily from the left as in Lemma 5. If any such chain extends into the strip of a subproblem we mark those bricks intersecting its left vertical strip boundary as either present or absent depending on the greedy independent set. This is exactly the information necessary to extend the independent set of the chain into the subproblem so that in fact the subproblem needs no information about the bricks to its left.

However, to bound the number of subsets of B_a that can be present in a subproblem, we now not only need to consider which bricks are shot by aliens, but also which bricks are intersected by other bricks that were greedily picked to be in an independent set. Which of those are chosen again depends on which other bricks in the same chain were shot by aliens, though. Unfortunately, it is not possible to immediately generalize Lemma 1. As Figure 7 indicates, for subinstances defined as before, monotone chains that intersect the vertical line ℓ_a may originate below the lower boundary $y = y_b$ of the current subinstance with half-strip S. Now if these chains can be shot by laser beams that stay below y_b their status (i.e., the actual maximum independent subset of the bricks) is independent of the definition of the current

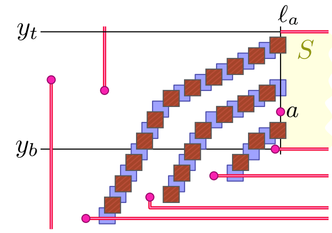


Figure 7: In the worst case an exponential number of subsets of bricks can intersect the line ℓ_a.

subinstance. So if there are k such independent chains $\Omega(2^k)$ different subsets of B_a are possible, where k may be linear in n.

In order to keep the number of subsets polynomially bounded we can bound the maximal length of a chain of overlapping bricks by some constant k. Then no more than k independent chains of bricks originating below y_b can intersect the line ℓ_a, because the topmost chain must intersect ℓ_a at least k units above y_b and thus needs at least k bricks in order to extend below y_b. We can consider all combinations of their statuses, which is a constant number.

To show that for a constant maximum chain length the number of subproblems is still polynomial, we extend Lemma 1 to the situation where we consider all chains that intersect the vertical line ℓ_a instead of only the bricks intersecting ℓ_a themselves. Let C_a be the set of all chains that intersect the line ℓ_a between y_b and y_t, where a brick that does not overlap any other brick is also considered a chain. We define $|C_a|$ to be the number of bricks in the union of all chains in C_a.

Lemma 6 *Let S be a half-strip defined as before by a leftmost alien a and the y-coordinates $y_b < y_t$ of two horizontal laser rays. If the maximum length of a chain of overlapping bricks is bounded by a constant k then there are at most $2^k|C_a|$ subsets of B_a that can be either shot vertically by aliens on the left of ℓ_a or removed because they overlap greedily chosen bricks.*

Proof. We follow the same argument as in Lemma 1. However, since we consider chains rather than single bricks, all aliens A_a to the left of ℓ_a may influence the subset of B_a if their laser beams intersect the left-open horizontal half-strip S' defined by ℓ_a, $y = y_b$, and $y = y_t$. Let b be a brick above y_b in one of the chains c in C_a and assume that the configuration of laser beams of the aliens in A_a is such that b is the leftmost brick among all bricks above y_b of chains in C_a that is *not* shot. Since no alien in A_a can shoot any brick of c that is to the right of the left side of b we know for all aliens to the right of the left side of b how they shoot within S'. This determines which bricks of chains in C_a are shot. We use Lemma 5 to compute a maximum independent set of the remaining bricks. This induces, for each of the at most 2^k different intersection patterns between ℓ_a and the chains extending below y_b, a unique subset of unshot bricks of B_a. Since the number of choices for b is $|C_a|$, the number of subsets of B_a that can be shot is at most $2^k|C_a|$. □

So to actually compute the value of a subproblem, we add for any possible location of the current laser beam the values of the two induced subproblems and the size of the left-greedy independent set M. Now, for each of the $O(n^4)$ subproblems (by Lemma 6) we have a linear number of possible laser beams, for each of which we greedily compute a maximum independent set in linear time. So we spend quadratic time per subproblem (which we needed anyway for looking up the subproblems) and this yields the same $O(n^6)$ time and $O(n^4)$ space bounds as in Theorem 2.

Theorem 7 *The overlapping, non-spaced ABP, where bricks cannot contain top left corners of other bricks and the maximum length of a chain of overlapping bricks is bounded by a constant, can be solved in $O(n^6)$ time using $O(n^4)$ space.*

We note that if the maximum length of chains of overlapping bricks is unbounded we can at least obtain a polynomial-time approximation scheme as follows. For an integer k we split all chains of overlapping bricks with an independent set larger than k into a minimum number of subchains whose maximum independent sets contain at most k bricks. Applying Theorem 7 to the induced instance yields a $(1 - 1/k)$-approximation.

3 Spaced Laser Beams

If we add the vertical spacing requirement for horizontal laser rays, surprisingly the problem becomes NP-hard, at least if we allow crossings. Otherwise, for non-crossing laser beams, the previous algorithm can still be applied.

3.1 Crossing Laser Beams

We show that the ABP is NP-hard if we require vertically spaced horizontal laser rays and allow crossings between laser beams. This result is independent of having overlapping or disjoint bricks.

Let I be an instance of Max2Sat with n variables $x_1, \ldots, x_n$ and m clauses $c_1, \ldots, c_m$ such that each variable x_i appears in at most three clauses. The problem

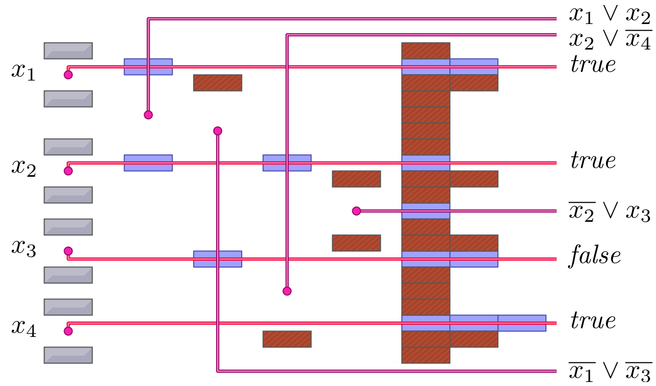


Figure 8: Aliens (points) and bricks representing a Max2Sat formula with four variables and four clauses.

of finding a truth value assignment for the variables that maximizes the number of satisfied clauses in I is NP-complete [8]. We will now describe a reduction from Max2Sat to the spaced and crossing ABP. Figure 8 shows an example of our reduction with four variable and four clause gadgets.

The variable gadget consists of an alien whose laser beam is vertically blocked by two indestructible bricks. These indestructible bricks are actually blocks of 4×4 regular bricks, but distinguishing two types of bricks saves space in the figure. Shooting an indestructible brick thus destroys at least four regular bricks. Towards the right, each variable alien has an upper and a lower choice of shooting its laser ray. Both choices destroy exactly three bricks. We assign the value *true* to the upper choice and *false* to the lower choice. We order all variable gadgets from top to bottom by their variable index. Furthermore, there is a "wall" of bricks to the far right that must be crossed by any horizontal laser ray between the gadgets for x_1 and x_n.

On its way to the right, each laser ray passes through up to three clause-connector bricks before hitting the wall. In a variable's *true* state, the clause-connector bricks in the upper row are destroyed and in its *false* state, the bricks in the lower row are destroyed. This is used to encode the truth values for the clause gadgets as follows. For a clause c_k containing variables x_i and x_j, $i < j$, we place an alien a in a new row between the gadgets for x_i and x_{i+1}. Now if x_i is a positive (negative) literal of c_k, we add a clause-connector brick vertically above a in the upper (lower) row of x_i. Analogously, we add a clause-connector brick for the literal of x_j vertically below a. Additionally, we add a dummy brick in the respective other row to the right of the wall so that we always destroy an equal number of bricks in both truth states. For every clause we use a new column so that a vertical line through any clause alien intersects exactly the two connector bricks for that clause, see Figure 8.

Now the reduction works as follows. If one of the literals of a clause is *true* then its corresponding clause-connector brick is destroyed by the variable's laser ray. Hence the clause alien can shoot its vertical laser segment through that brick at no cost until it reaches beyond the top- or bottommost variable, where it can turn to the right and reach the boundary of R. If, however, both literals of a clause are *false*, then no matter how the alien shoots its laser beam, it must destroy at least one brick.

In summary, we know that in an optimal solution (where all clauses are satisfied) the variable laser beams destroy exactly $3n$ bricks. Every non-satisfied clause causes exactly one more brick to be shot. So the MAX2SAT instance I has an optimal solution that satisfies at least K clauses if and only if the aliens destroy at most $3n+m-K$ bricks. The reduction clearly takes polynomial time.

Theorem 8 *The spaced and crossing ABP is NP-hard. This is independent of the disjointness of the bricks.*

3.2 Non-Crossing Laser Beams

For the non-crossing variant we can still exploit the fact that a laser beam from the leftmost alien in any subproblem splits that instance into two independent smaller subproblems. So we keep the same definitions for a subproblem as in Section 2. Again we assume that bricks can overlap, but not contain top left corners of other bricks and each chain is of constant size. Note that the problem for disjoint bricks is a special case of this. The only additional constraint is that we only consider locations for the new laser ray that stay at least one unit away from the boundaries of the subproblem. If no such location is available the cost will be ∞. We conclude:

Theorem 9 *The spaced non-crossing ABP, where bricks cannot contain top left corners of other bricks and the maximum length of a chain of overlapping bricks is bounded by a constant, can be solved in $O(n^6)$ time using $O(n^4)$ space.*

3.3 Fully Spaced Laser Beams

When we require laser beams to be horizontally spaced as well as vertically, we assume that bricks have a constant width w (and unit height), which means that at most w beams can cross the same brick vertically. We then require the space between any pair of vertical laser segments or horizontal laser rays to be at least 1. Note that this also solves the case where different amounts of spacing are desired by simply stretching the input appropriately.

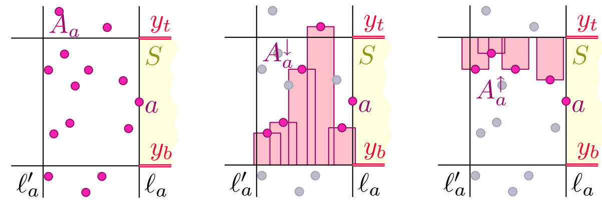


Figure 9: (a) The set of aliens A_a in the vertical strip of width w. (b) Only w aliens can potentially influence the situation for this subproblem while shooting down through the line $y = y_b$, since they must have disjoint rectangles of width 1. (c) Similarly, only w aliens can influence the subproblem while shooting up.

We first observe that a solution is not always possible: if there are too many aliens in too small a space, they can simply never shoot their laser beams without some of them getting too close to each other. Therefore, we assume that some solution exists (and if this is not the case, an algorithm should report this). If we do allow crossings, then the NP-hardness construction of the previous section can be used unchanged to show that this problem is also hard.

However, without crossings and for disjoint bricks we can find an optimal solution (if it exists) in $O(n^4)$ time. We use essentially the same dynamic programming approach as before. To speed it up, though, we now bound the number of subsets of B_a for a given subproblem in a different way.

For a given subproblem defined by an alien $a \in A$ and two rays at height y_b and y_t, let ℓ_a be the vertical line through a and ℓ'_a be the line w to the left of ℓ_a, and let A_a be the set of aliens between ℓ'_a and ℓ_a, as in Section 2.1. For each alien $a' \in A$ let $(x_{a'}, y_{a'})$ be the point, where a' is located. Even though A_a can have arbitrarily many aliens, we now argue that only $2w$ of them can actually influence the subset of B_a. For each alien $a' \in A_a$, consider the rectangle $[x_{a'}-1, x_{a'}+1] \times [y_b, y_{a'}]$. If this rectangle contains no other aliens of A_a, we call a' *free to shoot down.* Let $A_a^{\downarrow} \subseteq A_a$ be the set of all aliens that are free to shoot down. Similarly, we call a' *free to shoot up* if the rectangle $[x_{a'}-1, x_{a'}+1] \times [y_{a'}, y_t]$ contains no other aliens of A_a, and denote this set by $A_a^{\uparrow}$. Figure 9 shows an example of these subsets.

Lemma 10 *The number of subsets of B_a that can be shot by aliens in A_a is the same as the number of subsets of bricks that can be shot by aliens in $A_a^{\downarrow} \cup A_a^{\uparrow}$.*

Proof. Recall that any alien whose laser beam influences the subset of B_a that is shot must start shooting vertically up or down, and the vertical laser segment must continue until it is outside the extended strip containing S (between y_b and y_t). Now, let $a' \in A_a$ be an alien that shoots down and influences the subset (in some configuration). This means that its vertical segment must at least span the y-interval $[y_b, y_{a'}]$. But by the spacing requirement, this is only allowed if there are no other aliens close to that segment, i.e. if the rectangle $[x_{a'} - 1, x_{a'} + 1] \times [y_b, y_{a'}]$ is empty, i.e. if $a' \in A_a^{\downarrow}$. The argument for $A_a^{\uparrow}$ is symmetric. □

In other words, we can ignore any aliens not in $A_a^{\downarrow} \cup A_a^{\uparrow}$. Next, we will show how to bound the number of subsets of B_a in terms of $|A_a|$ rather than $|B_a|$, replacing Lemma 1.

Lemma 11 *Let S be a half-strip defined as before by a leftmost alien a and the y-coordinates $y_b < y_t$ of two horizontal laser rays. Let $A' \subset A$ be a set of aliens on the left of ℓ_a. Then the number of subsets of B_a that can be shot by aliens in A' is at most $|A'|^2$.*

Proof. Let $A_t \subseteq A'$ be the aliens above y_t, $A_b \subseteq A'$ the aliens below y_b, and $A_c \subseteq A'$ the aliens between y_b and y_t, see Figure 10(a). Now, we define a subdivision $\mathcal{C}$ of the center region bounded by all four lines. For each alien $z \in A_c$, we draw a horizontal line segment between z and ℓ'_a. Then, for each alien $z \in A_c$, we draw a vertical line segment through z that extends up and down as far as possible without crossing any of the horizontal segments or the lines $y = y_t$ and $y = y_b$. This divides the rectangle into a set of smaller rectangles, the *cells* of $\mathcal{C}$. Figure 10(b) shows $\mathcal{C}$ for the example.

Consider an assignment of shooting directions for all aliens in A' between ℓ_a and ℓ'_a, and let V be the region visible from S when looking to the left, where a point is "visible" if the horizontal ray from that point to the right does not intersect a laser beam. Figure 10(c) shows an example. First, we claim that V can only have $O(|A_c| \cdot |A_t \cup A_c \cup A_b|)$ different shapes. Observe that the boundary of V consists of a leftmost segment, and two staircases connecting it to the corners of S. Once this leftmost segment is fixed, the whole shape of V is fixed, since any aliens to the right of it must shoot away in order not to block the segment, and any aliens to the left of it are by definition not contributing to V. Now, the leftmost segment of V must lie inside a cell $C \in \mathcal{C}$, and V will be the union of the right half of C and all cells to the right of C. Clearly, there are $O(|A_c|)$ cells in $\mathcal{C}$. Furthermore, there are at most $O(|A_t \cup A_c \cup A_b|)$ choices for the x-coordinate of the segment, since it must come from some alien shooting vertical.

Now, consider the bricks in B_a. We first observe that the subset of B_a that is shot by the aliens is exactly the

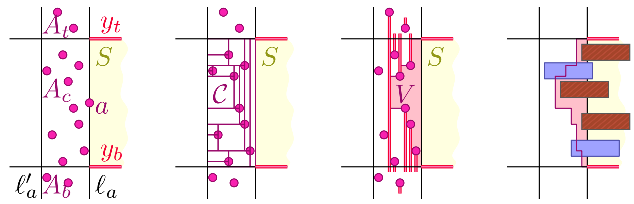


Figure 10: (a) An alien a and two horizontal rays at height y_t and y_b define three regions of interest. The aliens in the vertical strip between ℓ'_a and ℓ_a are divided into three groups. (b) The subdivision $\mathcal{C}$ indicates the possible locations for the leftmost segment of V. (c) A possible shape of the visibility region V. (d) The bricks shot by aliens are exactly the bricks that are not completely inside $S \cup V$.

subset that intersects the left boundary of V, because all these bricks intersect ℓ_a. Figure 10(d) illustrates this. Hence, the number of subsets that can be shot cannot be larger than the number of shapes that v can have, which is $O(|A'|^2)$. □

Now, observe that there cannot be more than w aliens in $A_a^{\downarrow}$. For this, consider the narrower rectangles $[x_{a'} - \frac{1}{2}, x_{a'} + \frac{1}{2}] \times [y_b, y_{a'}]$ of the aliens in $A_a^{\downarrow}$. These rectangles must be disjoint, otherwise the alien defining one of these rectangles would be contained in the broader rectangle of the other. Since A_a is contained in a vertical strip of width w and these narrow rectangles have width 1, there can be at most w of them. Similarly, there are only w aliens in $A_a^{\uparrow}$.

Let $A' = A_a^{\downarrow} \cup A_a^{\uparrow}$. Clearly $|A'| \leq 2w$, and by Lemma 11 there can be at most $O(w^2)$ subsets of B_a in any subproblem. (Note that this means, in particular, that $|A_c| \leq 2w$, since all aliens in A_c must shoot either up or down inside the extended strip. If this is not the case then this subproblem has no valid solutions.)

The rest of the algorithm remains the same. If we assume w is a constant, then we arrive at the following time and space bounds for the algorithm, which are actually a bit better than for the non-spaced case:

Theorem 12 *The fully spaced disjoint non-crossing ABP can be solved in $O(n^4)$ time using $O(n^3)$ space.*

Furthermore, in the case of overlapping bricks we can also extend the algorithm in a similar way. In this case we cannot use the region V defined above to argue about the number of subsets, but we can still say that at most $2kw$ aliens can be involved in the subsets if at most k bricks can be part of any chain, since these aliens would have to be in a strip of width kw. Then, these aliens together can clearly not lead to more than 2^{2kw} different configurations. Additionally, as in Section 2.2, there can be at most k chains that extend outside the current strip, and these can lead to another 2^k different configurations. In total, there are at most $2^{O(kw)}$ subsets of B_a per subproblem, which is still constant for constant k and w, and the problem can still be solved in $O(n^4)$ time and $O(n^3)$ space.

4 Discussion

We have studied the ABP in which a set of aliens A and a set of bricks B, both inside a rectangle R, are given, and we want each alien in A to shoot an orthogonal laser beam to the right side of R. In total, the aliens must destroy as few bricks in B as possible. We studied six main variants of the problem, and presented algorithms or hardness results for each variant. There is a trade-off between the different versions: by adding more restrictions (such as not allowing crossings or requiring laser beams with spacing) the resulting drawing may look cleaner, but obviously fewer bricks can be placed. Perhaps surprisingly, the least and most restricted variants of the problem can both be solved in polynomial time using essentially the same algorithm, while one of the in-between variants is NP-hard. Apart from the variants we studied, many more variants are conceivable by considering different ways for internal labeling (more or different label positions) or external labeling (more bends per leader, labels at multiple sides of R) that would be interesting to study.

Our polynomial-time algorithms have a rather high dependency on n. Partly this is due to the worst-case nature of the analysis: the linear number of subsets of B_a is in fact only linear in the number of bricks and aliens in a small subregion of R which may be expected to be much smaller than n. Additionally, we have chosen n to be the total number of input points in A and B combined, but some of the factors can be separately bounded by either of those numbers. Nonetheless, the number of half-strips that need to be considered is really as big as $O(n^3)$, but we have found no evidence that using this much time and space should be required.

4.1 One-Class ABP

We have assumed that the sets of aliens A and bricks B are given as two separate sets. However, in the map labeling formulation of the problem, it may be more realistic to assume that all input points belong to a single set, and that every point is labeled either internally or externally.We then want to maximize the number of internally labeled points. This version of the problem was suggested by [1, 6], and mostly remains open.

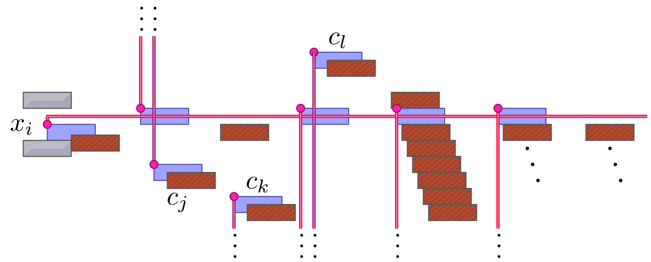


Figure 11: Changes to the gadgets in order to cover the case where destroyed bricks become aliens.

One easy observation is that, if no two bricks overlap, the problem becomes trivial since in the optimal solution all points will be labeled internally. Although it seems plausible that the dynamic programming algorithm still works in some settings, we no longer have the equivalent of Lemma 1, so the number of subproblems may be exponential.

The hardness result of Theorem 8 does extend to the case where destroyed bricks become aliens that shoot their own laser gun. The previous gadgets need to be adapted slightly to accommodate the additional laser beams as shown in Figure 11. First of all, we replace each variable and clause alien of Figure 8 by two overlapping bricks such that the upper left one contains the top-left corner of the lower right one. That way, the upper left brick must always be an alien. The wall and all dummy bricks on the right-hand side are diagonally shifted such that each destroyed brick among them turns into an alien that can shoot vertically downwards without hitting any other brick. Moreover, the clause-connector bricks, once destroyed, also become aliens that can shoot vertically up- or downward without hitting any other brick: in each column there are exactly two such bricks so that the upper one can always shoot upward and the lower one always downward. In that way we obtain a behavior that is identical to the reduction in the proof of Theorem 8.

Other than these relatively simple observations though, this problem remains open. We hope that our results in this paper constitute a first step towards solving it.

Acknowledgments. M. Nöllenburg is supported by the German Research Foundation under grant NO 899/1-1 and M. Löffler is supported by the Office of Naval Research under MURI grant N00014-08-1-1015.

References

[1] M. Bekos, M. Kaufmann, M. Nöllenburg, and A. Symvonis. Boundary labeling with octilinear leaders. *Algorithmica*, 2009. Available online.

[2] M. A. Bekos, M. Kaufmann, A. Symvonis, and A. Wolff. Boundary labeling: Models and efficient algorithms for rectangular maps. *Comput. Geom. Theory Appl.*, 36:215–236, 2007.

[3] M. Benkert, H. Haverkort, M. Kroll, and M. Nöllenburg. Algorithms for multi-criteria boundary labeling. *J. Graph Algorithms Appl.*, 13(3):289–317, 2009.

[4] M. Formann and F. Wagner. A packing problem with applications to lettering of maps. In *Proc. 7th Ann. ACM Symp. Comput. Geom. (SoCG'91)*, pages 281–288, 1991.

[5] M. R. Garey and D. S. Johnson. The rectilinear steiner tree problem is NP-complete. *SIAM J. Appl. Math.*, 32(4):826–834, 1977.

[6] M. Kaufmann. On map labeling with leaders. In S. Albers, H. Alt, and S. Näher, editors, *Efficient Algorithms*, volume 5760 of *Lecture Notes Comput. Sci.*, pages 290–304. Springer-Verlag, 2009.

[7] G. W. Klau and P. Mutzel. Optimal labeling of point features in rectangular labeling models. *Mathematical Programming*, 94(2):435–458, 2003.

[8] V. Raman, B. Ravikumar, and S. S. Rao. A simplified NP-complete MAXSAT problem. *Inform. Process. Lett.*, 65:1–6, 1998.