Rheinische Friedrich-Wilhelms-Universität Bonn
Mathematisch-Naturwissenschaftliche Fakultät

Theoretical Aspects of Intruder Search

MA-INF 1318 Manuscript Wintersemsester 2015/2016

Elmar Langetepe

Bonn, 19. October 2015

The manuscript will be successively extended during the lecture in the Wintersemester. Hints and comments for improvements can be given to Elmar Langetepe by E-Mail `elmar.langetepe@informatik.uni-bonn.de`. Thanks in advance!

Chapter 5

Geometric firefighting

In this chapter we would like to discuss geometric fireighting settings in the Euclidean plane. We assume that there is a fire source s (a point in the plane) and the fire spreads in all directions with the same unit speed 1. This means that the fire can be considered to be an expanding circle, expanding with speed one over time.

The adversary of the fire expansions can build barriers with some speed b. In the geometric setting, a barrier is a curve in the plane with given start and endpoint. We consider different models. For example, we can assume that the barriers have to be build up successively one after the other or in parallel. We can also restrict the type of the barrier, barriers can be restricted to line segment or can be arbitrary curves. Additionally, we consider different types of environments.

5.1 Firefighting in a simple polygon

Let us assume that the fire spreads inside a simple polygon P at a single point s. Let us further assume that a set of m potential line segment barriers, $B = \{b_1, b_2, \ldots, b_m\}$, is given. Each barrier, b_i, has a start point and an end point t_i on the boundary of P.

All barriers can be build with the same speed b. If $|b_i|$ is the length of the barrier, its *relative completion time* is $\frac{|b_i|}{b}$. We also assume that the barriers have to be constructed one after the other. During the construction of the barrier the fire should never reach the current end point while the barrier is constructed. Any barrier saves a certain amount of the total area of the polygon. The goal of the barrier construction is, that we would like to find a schedule for a valid construction of the barriers, that saves a maximal portion of the total area of the polygon. An example for a polygon, a fire source s and a set of potential barriers is shown in Figure 5.1.

In contrast to the Intruder-Definition given in Section 1.1.1 we define the problem as a firefighter problems in a simple polygon.

Geometric Firefighter Problem in a simple polygon
Instance: A simple polygon, a fire that spreads from a given starting point $s \in P$ with speed 1, a set of m line segment *barriers*, $b_1, b_2, \ldots, b_m$, which can be successively constructed with the same speed B.
Output: Compute a valid sequence of barriers that is constructed successively so that the area protected from the fire is maximized.

As already shown in Theorem 1 in Section 1.1.1 this problem is NP-hard in general. It can also be shown that it is approximation hard, which means that there is no polynomial time approximation scheme. Therefore we have to consider constant approximations for the optimal schedule.

5.2 A general approximation algorithm

In this section we describe a general approach for the approximation of a scheduling problem. We also relate it to the above firefighting problem.

We consider m jobs $b_1, b_2, \ldots, b_m$ with precise latest starting time and duration. That is, the job b_i has to be started before time s_i and has a duration of d_i.

For a *valid schedule* of jobs after n processing steps of an algorithm with l_n entries, we have a consecutive list of jobs $J_n = (b_{n_1}, b_{n_2}, \ldots, b_{n_{l_n}})$ which can be scheduled one after the other. Let $s'_{n_k} \leq s_{n_k}$ be the precise starting time of the scheduled jobs. We require $\sum_{k=1}^{j} s'_{n_k} + d_{n_k} \leq s_{n_{j+1}}$ for $j = 1$ to $l_n - 1$. The next job ha to be started before its starting option ends. This means that the jobs can be started one after the other without conflicts of the starting times.

Each job b_i will contribute with a profit A_i to an overall profit A. Profit A_i will be attained, if the job is scheduled. These profits might *overlap* for different jobs. Thus, the profit of a job b_i may change over time, if another job b_j was scheduled before that also covers parts of the profit that can be covered by b_i.

For a current schedule of jobs J_n we denote the current profit of a job $b_j \notin J_n$ by $A_j(J_n)$. This depends on the overlapping. More precisely for $J_n = (b_{n_1}, b_{n_2}, \ldots, b_{n_{l_n}})$ and $b_j \notin J_n$ we define

$$A_j(J_n) := A_j \setminus \left(\bigcup_{b_{n_k} \in J_n} A_{n_k} \right) .$$

(Note that here A_{n_k} is the initial overall potential profit of $b_{n_k} \in J_n$ but this profit is always covered by the profit of J_n, even though some of the profit A_{n_k} was intermediately already covered by another b_l in J_n.)

We start with an empty schedule J_0 and successively process one of the remaining jobs. This job will never be processed any more after that. So for m jobs we are done after m processing steps.

For the scheduling algorithm and a current schedule J_n after n steps we sort all non-rejected and non-scheduled jobs by decreasing relative profits $\frac{A_j(J_n)}{d_j}$. The profit of the job is considered in relation to its duration. We apply the following *GlobalGreedy* procedure to the current job b_j with maximal $\frac{A_j(J_n)}{d_j}$:

1. If a job b_j with current largest value $\frac{A_j(J_n)}{d_j}$ can be scheduled somewhere in J_n it is inserted into J_n for obtaining J_{n+1}.
2. If the job cannot be scheduled because some of the jobs overlap with its possible execution interval, we check whether we can delete such (consecutive) jobs out of J_n with the following property: For a given constant $\mu < 1$, the profit of the deleted jobs is smaller than μ times the profit $A_j(J_n)$ of job b_j. Additionally, after deleting the jobs, job b_j can be scheduled. All jobs that have been deleted will never be considered again. We build J_{n+1} by inserting b_j to the remaining list of J_n. There might be some options for such sequences.
3. If such a sequence of jobs does not exist, job b_j is rejected and will never be considered again. $J_{n+1} := J_n$.

For convenience for the profits in the procedure and also for the jobs we define corresponding colors. In the beginning all potential profits are colored red. Profits and jobs that will be scheduled by rule 1. or rule 2. are colored green. A profit (and a job) that was colored green in the schedule and is rejected later by rule 2. will be colored grey. For a schedule J_n let J'_n

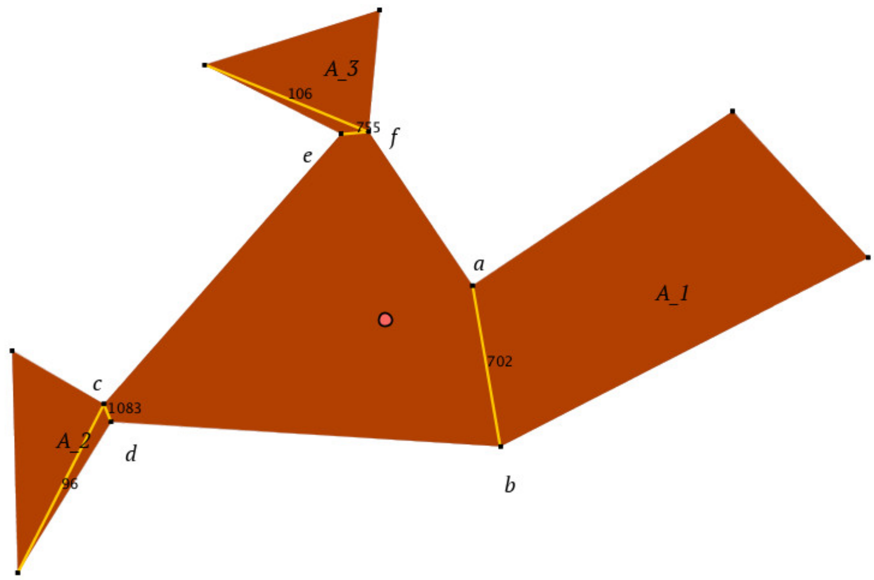

Figure 5.1: Considering the three barriers $b_1 = (a, b)$, $b_2 = (c, d)$ and $b_3 = (e, f)$ with $A_1 = 1053$, $A_2 = 162.45$ and $A_3 = 188.75$ and speed $b = 2$. We first the barriers by priority $p_1 = 702 = 2A_1/|b_1|$, $p_2 = 1083 = 2A_2/|b_2|$ and $p_3 = 755 = 2A_3/|b_3|$. The intermediate schedule $J_2 = (b_2, b_3)$ blocks the construction of b_1. Since $\mu \cdot A_1 > A_3$ and $(|b_1| + |b_3|)/2 < d(s, a)$ holds we delete b_3 and insert b_1.

denote the list of all jobs that have been inserted during the construction of J_n. The set J'_n also contains the deleted barriers of color grey and the current barriers of color green.

The above algorithm is a general scheme for a corresponding scheduling problem. We would like to apply it to our geometric firefighter problem and first give a very simple example. In Figure 5.1 there is a fire source and a set of potential barriers which are diagonals.

In this setting for convenience we require that the barrier has to be fully constructed before the fire reaches any point of it. This requirement is not necessary in general but by considering the closest point from b_i to s and the duration d_i of the barrier we can easily compute the a constraint for b_i. In general the fire should not have reached the current construction point of the barrier.

We consider only the barriers $b_1 = (a, b)$, $b_2 = (c, d)$ and $b_3 = (e, f)$ as depicted in Figure 5.1, where each barrier will be started with its first vertex. We assume that the building speed is $b = 2$. Each barrier subdivides the polygon into two parts and potentially would save the part A_i that does not contain the fire source. Let $|b_i|$ denote the lenght of each b_i, we assume $|b_1| = 3$, $|b_2| = 0.3$ and $|b_3| = 0.5$. The distance $d_P(s, a)$ from s to a in P is assumed to be 1.8. We further have calculated $A_1 = 1053$, $A_2 = 162.45$ and $A_3 = 188.75$ which are the profits of the barriers in the beginning. For the scheduling algorithm we order them relatively by priority $p_i = \frac{2A_i}{|b_i|}$ because $d_i = \frac{|b_i|}{2}$ is the duration for constructing b_i. The values in the figure denote the corresponding priority values p_i This means that $p_1 = 702$, $p_2 = 1083$ and $p_3 = 755$ holds. We assume that $\mu = 0.2$.

The scheduling algorithm first chooses b_2 and b_3 in this order because of the priorities and of the fact that both barriers can be indeed scheduled in this order. The fire will not reach c and e after $\frac{0.3+0.5}{2} = 0.4$ time steps. Because $(|b_1| + |b_2| + |b_3|)/2 = 1.9$ exceeds the distance from s to a in P, in the third step of the process we would like to apply rule 2. Since $\mu \cdot A_1 > A_3$ holds, we delete b_3 out of the schedule and build b_1 instead. The final schedule is $J_3 = (b_2, b_1)$ which saves A_2 and A_1. The resulting corresponding color scheme is shown in Figure 5.2.

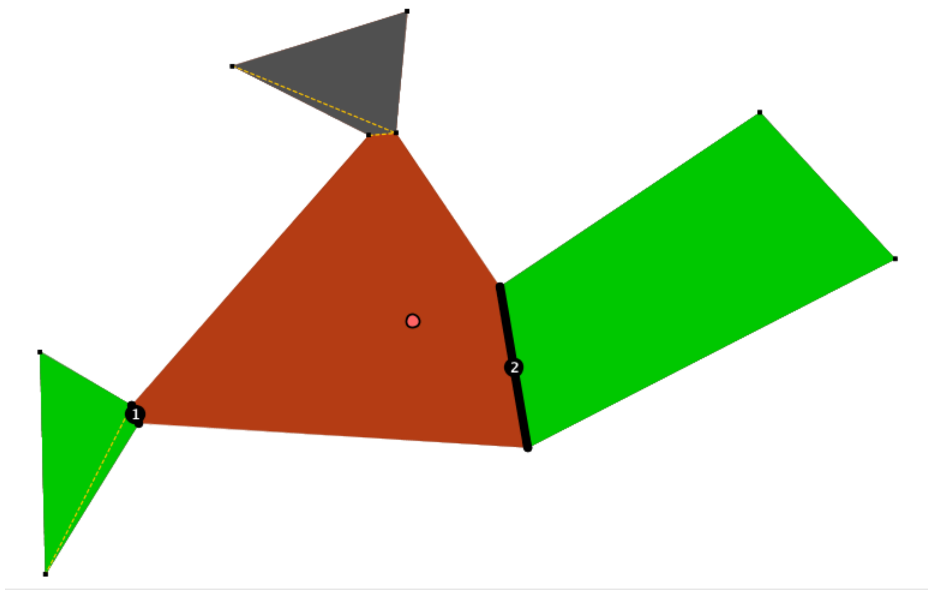

Figure 5.2: The color scheme of the scheduling algorithm for Figure 5.1. After b_1 exchanges b_3 the profit A_3 is colored grey. The part that was not protected remains red-colored. The barriers are build in the order 1. and 2. as depicted.

The following Lemma relates the green profit with the grey profit, so that we do not loose to much by deletions. By rule 2. a profit gets color grey, if it was deleted for insertion of a green profit larger than μ times the grey profit. Therefore we can prove the following relationship. Let J_m denote the final schedule for a set of jobs $b_1, \ldots, b_m$ and let $J_n(grey)$ and $J_n(green)$ denote the profits that are colored green and grey during the construction of J_n.

Lemma 52 *By the* GlobalGreedy *procedure we finally have*

$$J_m(grey) \leq \frac{\mu}{1-\mu} J_m(green)\,. \tag{5.1}$$

Proof. By induction on number of jobs processed during GlobalGreedy. The lemma holds in the beginning for the empty schedule J_0. Let us assume that the lemma holds after n steps for J_n. Consider step $n+1$.

If no job is deleted by rule 1. or rule 3. only the green profit will increase from J_n to J_{n+1} that is

$$J_n(grey) = J_{n+1}(grey) \leq \frac{\mu}{1-\mu} J_n(green) \leq \frac{\mu}{1-\mu} J_{n+1}(grey)\,.$$

If by rule 2. some of the jobs where cancelled out of J_n for inserting b_j in the $n+1$ step. The overall profit of the cancelled jobs (that becomes grey now) is smaller than μ times $A_j(J_n)$, the profit of b_j right now. We conclude

$$\begin{aligned} \frac{\mu}{1-\mu} J_{n+1}(green) &\geq \frac{\mu}{1-\mu}(J_n(green) + (1-\mu)A_j(J_n)) \\ &\geq \frac{\mu}{1-\mu} J_n(green) + \mu A_j(J_n) \geq J_n(grey) + \mu A_j(J_n) \geq J_{n+1}(grey), \end{aligned}$$

where we used the induction hypothesis for the secon $\geq$ sign in the second line. □

For the relationsship between the profit of the final schedule J_m and an optimal schedule J_{opt} we make use of another color, blue.

For the profits, all profit elements in the optimal schedule J_{opt} that are still have color red after the application of J_m will be colored blue. So none of the grey and green profits of J_m become blue. The blue profit part depends on J_m and J_{opt}, let us denote this profit, by $J_m(\textit{blue})$. It is clear that

$$|J_{\text{opt}}| \leq J_m(\textit{blue}) + J_m(\textit{green}) + J_m(\textit{grey})\,.$$

holds, we would like to express $J_m(\textit{blue})$ in terms of $J_m(\textit{green})$ and $J_m(\textit{grey})$.

We can assume that the jobs of J_m are colored green and grey by the schedule. Likewise, we assign the blue property to the jobs of J_{opt}. The blue property is assigned to the first job of J_{opt} that covers a portion of the blue profit. Such a job cannot be green or grey, because otherwise it cannot participate at a the red colored profit at the end of the schedule J_m. Thus, all three color classes of jobs are pairwise disjoint.

As already mentioned J'_m denote the list of all jobs that have been inserted during the construction of J_m. Any green and grey jobs of J'_m has obtained a unique execution time, that will not be changed by the process afterwards (although grey jobs will no longer be scheduled at the end).

We make use of a scheme that pays money from green and grey jobs to the blue jobs w.r.t. the execution intervals of the jobs $b_i \in J'_m$ and $b_j \in J_{\text{opt}}$.

1. If the execution interval of $b_j \in J_{\text{opt}}$ is fully included in the execution interval of $b_i \in J'_m$, the job b_i pays its green or grey profit times $\frac{d_j}{d_i} < 1$ to b_j.
2. If the execution interval of $b_j \in J_{\text{opt}}$ overlaps with the execution interval of $b_i \in J'_m$, the job b_i pays its green or grey profit times $\frac{1}{\mu}$ to b_j.

The payment of each green and grey job to the blue jobs has an upper bound w.r.t. their profit.

Lemma 53 *Any single green or grey job from J'_m pays in total at most $1 + \frac{2}{\mu}$ times its profit to the blue jobs.*

Proof. A job $b_i \in J'_m$ has a fixed execution interval I_i and the start- and endtime of the interval can only be located inside at most two intervals of jobs from J_{opt} which are also executed successively. This gives at most 2 times $\frac{1}{\mu}$ the profit of b_i. For all jobs from $b_j \in J_{\text{opt}}$ fully inside the execution interval I_i of b_i a portion $\frac{d_j}{d_i}$ of b_i's profit is payed for $d_j < d_i$. In total this can only sum up to at most 1. □

The other way round, any blue job is payed by at least its own profit from the green and grey jobs.

Lemma 54 *Any single blue job from J_{opt} achieves at least a payment in the size of its blue profit from the green and grey jobs.*

Proof. We consider a blue job $b_j \in J_{\text{opt}}$. This job was rejected at some step $k+1$ and let $J_k = (b_{k_1}, b_{k_2}, \ldots, b_{k_{l_k}})$ be the corresponding schedule of the algorithm. For the final exection time interval of $b_j \in J_{\text{opt}}$ consider an consecutive subset $\overline{J_k}$ of the jobs of J_k of mimimal size whose intervals overlap with the execution time of b_j, The deletion of $\overline{J_k}$ would allows to schedule b_j.

The total sum profit of the jobs of $\overline{J_k}$ is larger than μ times the current profit $A_j(J_k)$ of b_j which is currently red. Some of this red part $A_j(J_k)$ will become blue at then end but may be not all of it. In any case μ times the current red part $A_j(J_k)$ of b_j is larger than μ times the final blue part of b_j. On the other hand, the job b_j had priority less than all previously inserted jobs of J_k which is $\frac{A_i(J_k)}{d_i} \geq \frac{A_j(J_k)}{d_j}$.

We consider the size of $\overline{J_k}$. If $|\overline{J_k}| = 1$ holds for the single job, say b_i, the execution time of $b_j \in J_{\text{opt}}$ might be fully inside the execution time of b_i, which gives

$$A_i(J_k)\frac{d_j}{d_i} \geq A_j(J_k)\frac{d_i}{d_i} = A_j(J_k)$$

where $A_i(J_k)$ will be the green or grey profit of b_i for the final J_m and $A_j(J_k)$ is larger than the blue profit of b_j.

For $|\overline{J_k}| \geq 1$, the execution time interval of b_j is not fully inside an execution interval of a job in $\overline{J_k}$, but the execution interval of b_j overlaps with all execution intervals of the jobs in $\overline{J_k}$, This gives:

$$\frac{1}{\mu}\sum_{b_i \in \overline{J_k}} A_i(J_k) \geq \frac{1}{\mu}(\mu A_j(J_k)) = A_j(J_k)\,.$$

In any case a blue job gets paid with at least its profit form the green and grey jobs. □

By the above Lemmata we can now estimate the profit of $J_m(\mathit{blue})$ in comparison to $J_m(\mathit{green})$ and $J_m(\mathit{grey})$. The blue profit is upper bounded by the payment from green and grey jobs, in turn the payment of green and grey jobs is upper bounded by the profits of green and grey jobs.

We have

$$J_m(\mathit{blue}) \leq \left(1 + \frac{2}{\mu}\right)(J_m(\mathit{green}) + J_m(\mathit{grey}))\,. \tag{5.2}$$

By Lemma 52 we can now express the profit of J_{opt} in terms of $J_m(\mathit{green})$, only.

$$\begin{aligned}
|J_{\text{opt}}| &\leq J_m(\mathit{blue}) + J_m(\mathit{green}) + J_m(\mathit{grey}) && (5.3)\\
&\leq \left(2 + \frac{2}{\mu}\right)(J_m(\mathit{green}) + J_m(\mathit{grey})) && (5.4)\\
&\leq \frac{2(\mu+1)}{\mu}\left(J_m(\mathit{green}) + \frac{\mu}{1-\mu}J_m(\mathit{green})\right) && (5.5)\\
&\leq \frac{2(\mu+1)}{\mu}\frac{1}{1-\mu}J_m(\mathit{green}) && (5.6)\\
&\leq 2\frac{\mu+1}{\mu(1-\mu)}J_m(\mathit{green}) \leq 2\frac{\mu+1}{\mu(1-\mu)}|J_m|\,. && (5.7)
\end{aligned}$$

The factor $f(\mu) := 2\frac{\mu+1}{\mu(1-\mu)}$ is minimized for $\mu = \sqrt{2}-1$ and gives $f(\mu) = 6+4\sqrt{2} \approx 11.657$ and our final result.

Theorem 55 *For the geometric firefighter problem inside a simple polygon with non-intersecting barriers there is an approximation algorithms that saves at least* $\frac{1}{6+4\sqrt{2}} = \frac{3}{2} - \sqrt{2} \approx 0.086$ *times the area of the optimal barrier solution.*

Exercise 21 *Verify the computation of the optimization value* $\mu = \sqrt{2}-1$ *and* $f(\mu) = 6+4\sqrt{2} \approx 11.657$.

The result is restriced to non-intersecting barriers. Additionally, we require that any point in the polygon that can be covered, can already be covered by a single barrier. Otherwise some barriers $b_1, b_2, \ldots, b_l$ could help each other for covering a part of the polygon that is not covered

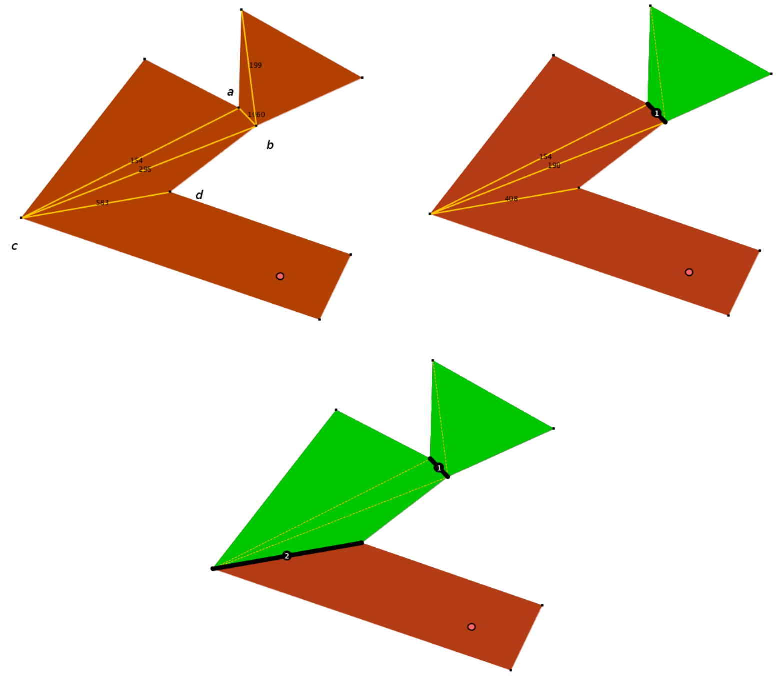

Figure 5.3: After the usage of barrier (a, b), the priority of barrier (c, d) decreases because (a, b) already protects parts of the area related to (c, d).

by a single barrier. In this case we can protect new areas A' which are not related to single barriers. This is obviously not represented in the algorithm and the analysis.

In principle the algorithm could also run for intersecting barriers but there is one major problem. The execution time for building a barrier is not independent from the order of the barrier constructed so far. A barrier might have a release times that depends on the barriers constructed before. For example a barrier b_1 blocks the fire and extends the release time of a barrier b_2 that intersects b_1.

So there might be a barrier of the optimal solution, that contributes to the optimal profit with an arbitrary large amount of its profit but cannot be scheduled in the approximation since it depends on the construction of a very less important (w.r.t. area profit) barrier. The analysis might fail.

Exercise 22 *Construct an example, where the construction of a barrier b_2, depends on the construction of a barrier b_1.*

Exercise 23 *If intersections are allowed, where is the main problem in the proof of the above approximation result?*

Finally, we end the section with an example where the profit of a job is reduced by overlapping, as shown in Figure 5.3.