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 1

Introduction

In this lecture we consider intruder and evader problems from an algorithmic point of view. We discuss problems in the context of motion planning in discrete and continuous environments. Discrete environments will be mainly defined by graphs whereras in the continuous case we will make use of the Euclidean plane or subsets of the Euclidean plane.

Generally, there is a set of searcher (or guards) and another set of intruder (or evaders) that compete with each other. The intruder tries to escape from the searcher in the given environment. The other way round the guards would like to protect a certain area from the intruder. Intruder and searcher can have different facilities or sensors. There is always some kind of dynamics, the searcher and/or the intruder can move or perform tasks over time.

The intruder search problem might also be considered as a scenario for the cleaning or the enclosure of a spreading contamination. For any moment in time the current contamination represents all possible remaining positions of the intruder. So the task of the searcher is to reduce the number or to bound the area of these positions and to give a guarantee where the intruder is located.

More precisely, from a theoretical point of view we would like to answer the following questions:

- Computational complexity (i.e. existence of a strategy, question for the number of guards required, running time for the computation of a strategy)
- Correctness or Failure (i.e. success of a given strategy, limits of the strategy)
- Efficiency (i.e. number of steps or path length required for a given strategy, area saved by the guards)
- Optimality (i.e. cost measures for the efficiency and its analysis)

We will also see that there are many interesting open questions. Sometimes simulations will be helpful for getting more insight into the problem and a conjecture for a solution. Additionally, the usage of simulation tools will be part of this lecture.

1.1 Introductory examples

In the following we would like to discuss some simple introductory examples and relate it to the above questions. This gives some insight into the nature of the problems we would like to discuss and also shows the variety of methods that will be applied.

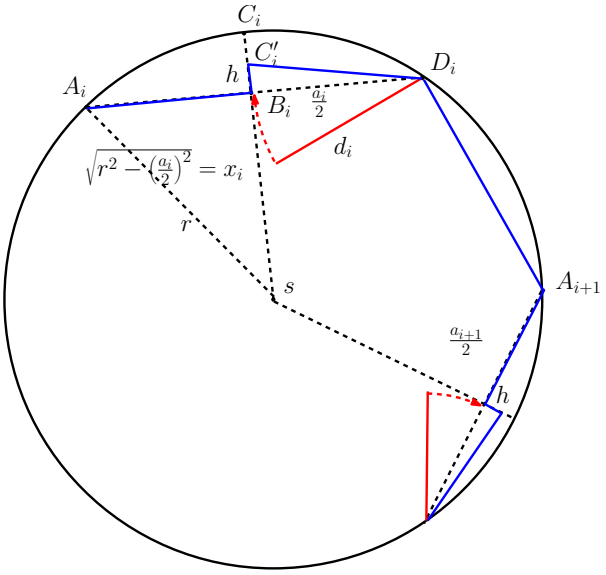

Figure 1.1: Constructing a polygon with doors for a set of n integer $a_1, a_2, \ldots, a_n$. The radius r and the speed v are chosen appropriately so that after $t + 0.5$ time steps the intruders can reach any point C'_i but before t time steps no point B_i can be reached.

1.1.1 Protecting parts of a polygonal area from a set of intruders

Let us assume that we have a simple room modeled by a polygonal area P (a simple polygon) and there is a set of n doors that can be closed successively. Additionally, there is a set of intruders starting at some point s. The intruder moves with speed v. For a given set of n doors each door can protect the area behind the door. The time for closing a single door d_i is given by a value a_i. Any door d_i protects a portion A_i of the polygon from the intruders. (Somehow it makes sense that the time a_i for closing the door is proportional to the area A_i that can be saved by d_i.) We would like to close as many doors as possible, thus maximizing the portion that cannot be entered by the intruders. The doors can only be closed one after the other.

The question is, which doors should be closed successively, so that the intruders cannot enter the part of the polygon behind the doors. We would like to maximize the area that will be safe. As a matter of computational complexity we show that this problem is NP-hard.

Optimal-Closing-Sequence
Instance: A simple polygon, a set of n intruders with starting positions and speed 1, a set of m *doors* that can be closed successively in time c_i and saves area A_i.
Output: Compute the optimal sequence of doors that has to be closed for maximizing the area saved.

Theorem 1 *Computing the optimal closing-sequence for a set of doors in a polygon is NP-hard in general.*

Proof. The subset-sum problem can be reduced to our problem. We have n integer numbers $a_1, a_2, \ldots, a_n$ and a integer threshold t and we are searching for a subset-sum, so that the sum does not exceed t but is as close as possible to t. [1] The subset-sum problem will be transformed into our problem as depicted in Figure 1.1. We make use of a circle C_r of radius r. Let s be the center of the circle. Along the boundary we use non-intersecting secants of length a_i with endpoints A_i and D_i in clockwise order. A ray starting at s and running through the midpoint

[1] Note that partition can be reduced to the subset-sum problem.

B_i of each secant hits the boundary of C_r at some point C_i. Along the segment B_iC_i we use a point C_i' a fixed distance h away from B_i. For the construction of the polygon we use the chain A_i,B_i,C_i',D_i and connect D_i with A_{i+1} and also finally D_n with A_1. There is a door d_i for each pair of points B_i and D_i. This door can be closed within a_i time steps and saves a region of area $\frac{a_i \cdot h}{4}$ proportional to a_i.

Now we choose the speed v so that $v(t+0.5) = r$ holds. This means that after $t+0.5$ time steps an intruder can reach any C_i'. Additionally, we have to take care that after t time steps, none of the entry points B_i closest to s can be reached by an intruder. Therefore, we choose r large enough so that $vt < x_i = \sqrt{r^2 - \left(\frac{a_i}{2}\right)^2}$ holds. Substituting v by $\frac{r}{(t+0.5)}$ gives

$$\left(\frac{a_i}{2}\right)^2 < \left(1 - \frac{t^2}{(t+0.5)^2}\right) r^2$$

which can be fulfilled for any a_i.

Altogether, after $t+0.5$ time steps any entrance point B_i could be passed but after t time step none of the entrance points have been visited. If we would like to save the maximum area from the intruders, we have to compute a maximum sum of values a_i not exceeding $t+0.5$ but beeing as close as possible to t. This gives a solution for the subset-sum problem.

Conversely, for a solution of the subset-sum problem we obtain the maximum sum of values a_i smaller than t, thus we save a maximum area if we close the corresponding doors successively.

□

Exercise 1 *Make use of the* `GeoGebra`*-tool and compute an interactive construction of the above reduction for a finite set of elements $a_1, a_2, \ldots, a_n$. Give examples $a_1, a_2, \ldots, a_n$ and t where the subset-sum questions is answered with Yes and with No, respectively.*

1.1.2 Catching an evader in a grid world

The next example considers a discrete problem variant and a different intention.

There is some evader that can move in a rectangular grid world and a guard would like to enclose and catch the evader. For the enclosement and for catching the evader the guard can block and check cells of the grid while the evader is moving. The evader starts at a single cell and can visit the cells of the 4Neighborship in one step or could stay in its current cell. Simultaneously, the guard can block or check k of the cells. Any cell that was blocked cannot be entered by the evader anymore. In order to synchronize the movements we first let the evader move and then the guard can choose its k blocking and checking cells; see Figure 1.2.

We do not have an idea of the strategy of the evader. Therefore, simply all possible locations of the single evader will be considered and marked by red circles. The blocked cells and checked cells are marked by black circles. We would like to find an efficient strategy for the guard. That is, we would like to enclose the evader as fast as possible. After that the guard simply checks all the remaining red cells. The game ends when there is no single red cell any more.

Evader-Enclosement (Grid-Graph):
Instance: A rectangular grid, a start vertex s of the evader and k protecting guards per time step.
Output: Compute a protection strategy that encloses the evader (and finally find the evader).

We first discuss failure and correctness for different values of k, we will see that some machinery is necessary.

Lemma 2 *Catching an evader in a grid world by setting $k = 1$ blocking cells after each movement of the evader cannot succeed in general.*

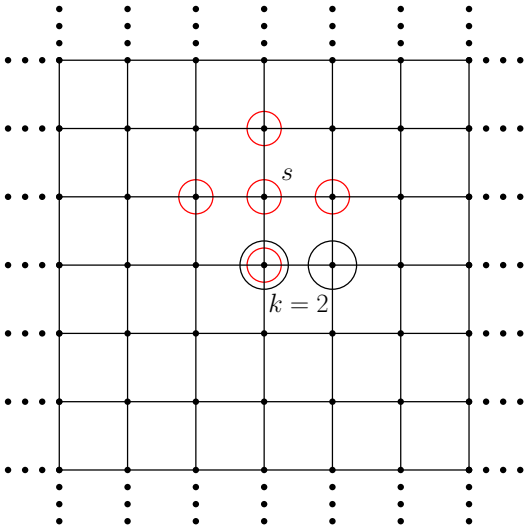

Figure 1.2: The evader starts at s. In the first step he can move to the adjacent vertices of the grid. For $k = 2$ the guard blocks and checks two cells afterwards. Now the evader can move on.

Proof. We show that the evader always has an escape strategy. Or the other way round, a guard can never cancel out all cells marked red.

For a formal proof we show that it is already impossible to catch the evader in a single quadrant of cells. The evader starts at the source of the quadrant, also called root in the following; see Figure 1.3. For any step $l \geq 1$ let D_l denote the set of vertices that are distance l away from the root vertex. This means, that these cells lie on the l-th diagonal of the quadrant.

Let r_l denote the number of blocked cells in D_{l+1}, D_{l+2},... at the end of the l-th step. The blocked cells in D_{l+1}, D_{l+2},... can be considered to build a reserve for the future. Let $B_l \subseteq D_l$ denote the number of red cells at the end of the l-th step distance l away from the root. We have $r_0 = 0$.

By induction we show that $B_l \geq 1 + r_l$ holds for all $l \geq 1$. In the first step $l = 1$ the evader can move to both vertices of D_1 and the guard can block a single cell, either in D_t for $t = 0, 1$ or in D_t for $t > 1$. The latter one means $r_1 = 1$. In any case we have $B_1 \geq 1 + r_1$.

Let us assume that the statement holds after $l \geq 1$ steps. We can assume $B_l \geq 1 + r_l$. This means that in step l the evader can visit $r_l + 1 + 1$ cells in diagonal D_{l+1}. But some of them might already be blocked by the reserve and another one can be blocked by the guard in this step. Let us assume that there are $x \leq r_l$ blocked cells in D_{l+1}.

This means that at first (the evader moves) we have $B_{l+1} \geq 1 + 1 + r_l - x$ red cells in D_{l+1}. Now the guard can block another cell in step l also. Either in D_t for $t \leq l$, in D_t for $t = l + 1$, or in D_t for $t > l + 1$.

In the latter case we have $r_{l+1} = 1 + r_l - x$ and in the former cases we have $r_{l+1} = r_l - x$. Now the guard can only reduce B_{l+1} by blocking a vertex in D_{l+1}. In any case we have $B_{l+1} \geq 1 + r_{l+1}$ which gives the conclusion. □

Note that for the pure enclosement of the evader it seems reasonable that in step l the guard should never block a cell that was marked by a red circle in one of the previous rounds $< l$.

Exercise 2 *Give a formal argument for the fact that a fast guard strategy should first* enclose *the evader and then check the remaining cells. This means that in step l the guard strategy will not choose a red marked vertex that was marked in a step $< l$ until the evader cannot move anymore.*

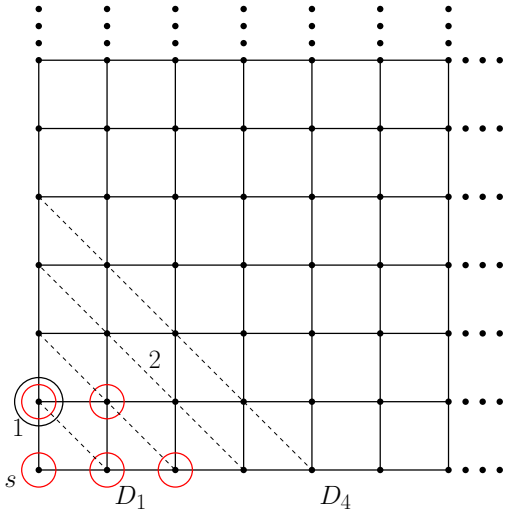

Figure 1.3: The situation for $k = 1$ after step 2. We have $r_2 = 1$ and $B_2 = 1 + r_2 = 2$.

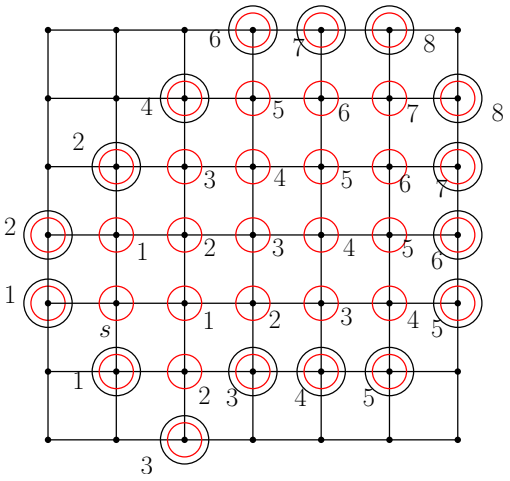

Figure 1.4: For $k = 2$ there is a guard strategy that encloses the evader after 8 steps. Now the evader can be catched by checking the remaining 18 cells.

Fortunately for $k = 2$ there is a successful strategy. We can just give the sequence of steps for the enclosement as presented in Figure 1.4. After that the remaining cells can be cleaned.

Lemma 3 *There is a successful guard strategy for $k = 2$. The evader can be enclosed after 8 steps. With 9 additional steps the evader will be catched.*

Proof. By construction; see Figure 1.4. □

The above situation can also be considered as the problem of containing a spreading fire. We slightly change the rules of the game. Let us assume that the firemen first blocks k cells and then the fire can spread from the source s to its neighboring cells. A burning cell can never be blocked any more.

In this sense and for $k = 2$ a firemen can enclose a spreading fire in 8 steps as depicted in Figure 1.4. Here only 18 vertices get burned. We give a formal proof that this is optimal.

Lemma 4 *For the outbreak of a fire on a single source in a grid and the usage of two firefighters*

per time step any successul strategy encloses an area of at least 18 burning vertices. This bound is tight.

Proof. For the proof of the statement we make use of an Integer Linear Program as follows. We already know that it is possible to enclose 18 burning cells by the strategy given in Figure 1.4 in 8 time steps. Therefore we make use of a fixed constant $T > 8$ for the number of time steps. Additionally, we use a grid of size $l \times l$ for another fixed constant l.

Let $L = \{(x, y)||x| \leq l \text{ and } |y| \leq l\}$ and $0 \leq t \leq T$. For $v \in L$ let $N(v)$ denote its 4Neighborship in L. For the Integer Program we use the following variables for $v \in L$.

$$b_{v,t} = \begin{cases} 1 & : \text{ vertex } v \in L \text{ burns before or at time } t \\ 0 & : \text{ otherwise} \end{cases}$$

$$d_{v,t} = \begin{cases} 1 & : \text{ vertex } v \in L \text{ is defended before or at time } t \\ 0 & : \text{ otherwise} \end{cases}$$

We would like to minimize the number of vertices that become burned.

$$\begin{array}{lrcll} \text{Minimize} & \sum_{v \in L} b_{v,T} & & & \\ \text{so that} & b_{v,t} + d_{v,t} - b_{w,t-1} & \geq & 0 & : \forall v \in L, v \in N(w), 1 \leq t \leq T \\ & b_{v,t} + d_{v,t} & \leq & 1 & : \forall v \in L, 1 \leq t \leq T \\ & b_{v,t} - b_{v,t-1} & \geq & 0 & : \forall v \in L, 1 \leq t \leq T \\ & d_{v,t} - d_{v,t-1} & \geq & 0 & : \forall v \in L, 1 \leq t \leq T \\ & \sum_{v \in L}(d_{v,t} - d_{v,t-1}) & \geq & 2 & : \forall 1 \leq t \leq T \\ & b_{v,0} & = & 1 & : v \in L \text{ is the origin } (0,0) \\ & b_{v,0} & = & 0 & : v \in L \text{ is not the origin } (0,0) \\ & d_{v,0} & = & 0 & : \forall v \in L \\ & d_{v,t}, b_{v,t} & \in & \{0,1\} & : \forall v \in L, 1 \leq t \leq T \end{array}$$

The first inequality triggers the spread of the fire. The neighbor of a burning vertex is either blocked or it burns in the next step. Note that with this condition any vertex can also burn accidently. But this will not happen in the optimal solution. The next inequality prevents a defended vertex from burning and a burning vertex from beeing defended. The next two inequalities take care that the status of the vertices remains the same after burning or beeing defended. The sum condition refers to the number $k = 2$ in any step. Initially only one vertex is burning and no vertex is defended. Finally, the program gets binary.

The corresponding program was run for example by Develin and Hartke [] and for $l = 6$ and $T = 9$ and the origin $(0, 0)$ the solution of Figure 1.4 was achieved. □

The above proof is an example where we make use of a computer aided process for the conclusion of a theoretical proof. With similar arguments it is also possible to show that 18 vertices cannot be enclosed within less than 8 steps.

1.1.3 Enclosing a fire by a single circle

In the last section we have seen that catching an evader and the enclosement of a fire are very similar problems which can be formulated in a similar model. Let us consider the fire containment problem in the following continuous setting.

The fire is already burning for a while and the fire spreads into any direction with the same unit speed 1. There is a single firefighter which can build a firebreak with speed $v > 1$. Geometrically

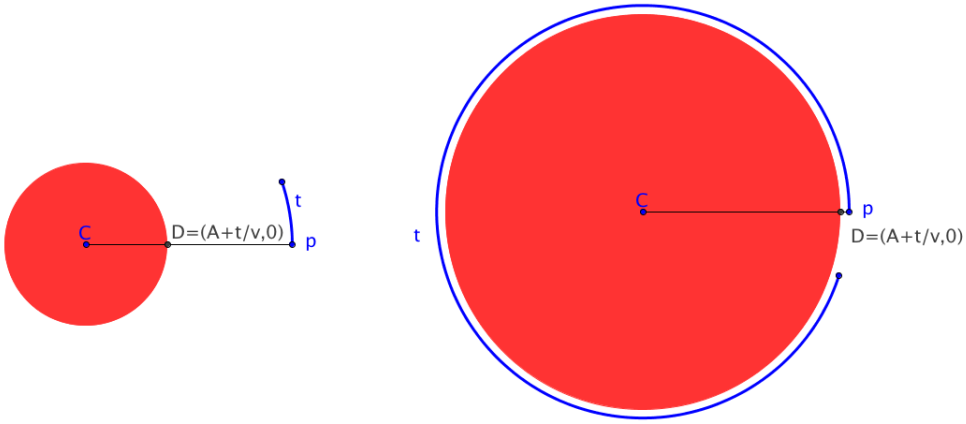

Figure 1.5: The firefighter builds a firebreak of length t with speed $v > 1$ and the fire expands from the source with speed 1. The circular strategy is successful here.

we can assume that in the beginning of the game the fire is a circle of radius A from a source C and the circle grows with speed 1 while the firefighter builds a path t with speed $v > 1$; see Figure 1.5. We call this the *geometric firefighter problem*.

Geometric Firefigther Problem
Instance: A circle with center C of radius A that grows with unit speed. An agent who builds a firebreak with speed $v > 1$
Output: Compute a strategy that finally fully enclose the spreading fire.

A simple question is: How fast must the firefighter be in order to enclose the fire by a single circular loop around the source of the fire? Let us assume that we can choose an arbitrary starting point p for the firefigther.

Lemma 5 *Enclosing a fire of extension A with a single circular loop around the source of the fire is possible, if and only if the speed v of the firefigther is larger than 2π.*

Proof. We assume that the fire source is given by the origin and we choose a point $p = (A+x, 0)$ away from the fire. A single circular loop around the origin has length $2\pi(A + x)$ and takes $\frac{2\pi(A+x)}{v}$ time. We can enclose the expanding circle if this is smaller than or equal to x. This gives $\frac{2\pi A}{x} + 2\pi \leq v$. For $v > 2\pi$ there will always be an x so that the inequality holds. For $v \leq 2\pi$ we can never fulfill the required inequality. □

Exercise 3 *Consider the above circular strategy. Assume that it is allowed to move along an arbitrary circle (around an arbitrary center). Is it possible to improve the above bound? I.e., is the above bound tight for arbitrary circles or can we always succeed with a circle and speed less than 2π?*

1.1.4 Simulation and conjecture for a discrete spiral strategy

Finally, we would like to motivate that in some settings it is helpful to make some simulation. We would like to enclose a spreading contamination in a grid world and apply a strategy that always moves close to the boundary of the contamination. The agent has to move from one cell to the other and can build a wall as in the previous setting. The agent either moves clockwise or counterclockwise around the starting contamination. We assume that the contamination is a grid square in the beginning.

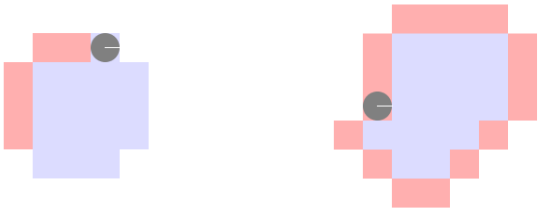

Figure 1.6: The fire spreads after 30 steps and the agent build a wall cell within 5 time steps. The lefthandside figure shows the situation after 30 time steps. The contaminations is enclosed in a single loop after a few more steps.

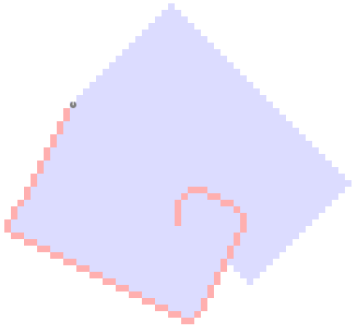

Figure 1.7: The fire spreads after 30 steps and the agent build a wall cell within 15 time steps. After a while it seems that the spiral strategy will not succeed.

Parameters: Moving from one cell to the other takes one time step. Building a wall element and cleaning a cell takes b time steps. After n time steps the contamination *spreads*. This means that the 4Neighborship of any contamintated cell becomes contaminated but cells that contain a wall element cannot be contaminated anymore.

Discrete Firefigther Problem
Instance: A grid contamination some size B that spreads to its 4Neighborship after n time steps. An agent who cleans a cell, builds a wall and leaves the celle within b time steps.
Output: Compute a strategy that finally fully enclose the spreading fire.

Figure 1.6 shows an example for a 3×3 starting contamination with a single agent, $b = 5$ and $n = 30$ after exactly 30 time steps and when the task is finished. Figure 1.7 shows the case where $n = 30$ and $b = 15$, the spiral strategy does not succeed.

Experiments from the thesis of Smiegilski [] give the following conjecture.

Conjecture 1 *For a grid fire thats spreads after n time steps and an agent that builds a wall within b time steps, the spiral strategy only succeeds if $b < \frac{n-1}{2}$ holds.*