# Pursuit-Evasion with Imprecise Target Location*

Günter Rote†

October 16, 2002

**Abstract**

We consider a game between two persons where one person tries to chase the other, but the pursuer only knows an approximation of the true position of the fleeing person. The two players have identical constraints on their speed. It turns out that the fugitive can increase his distance from the pursuer beyond any limit. However, when the speed constraints are given by a polyhedral metric, the pursuer can always remain within a constant distance of the other person.

We apply this problem to buffer minimization in an on-line scheduling problem with conflicts.

## 1 Problem Statement and Overview

We consider a game between two players, the *sheriff* $S$ and the *thief* $T$. The sheriff tries to catch the thief or at least remain close to him, whereas the thief tries to escape. The thief does not have to reveal his true location but only an approximate location $R$ that is constrained within a constant radius of the true position.[1]

The game is played in continuous time $t$. The sheriff can control his position $S(t)$ after observing $R(t)$. The thief can control $T(t)$ and $R(t)$. To make the game fair, identical velocity constraints are imposed on the motions of $S$ and $T$:

$$\|S(t_2) - S(t_1)\|_V \le |t_2 - t_1| \text{ and } \|T(t_2) - T(t_1)\|_V \le |t_2 - t_1|,$$

where $\|\cdot\|_V$ is the norm that is used to measure speed. The revealed position $R$ is subject only to the distance constraint

$$\|R(t) - T(t)\|_D \le 1.$$

where $\|\cdot\|_D$ is another norm, possibly different from $\|\cdot\|_V$. There is no velocity constraint on $R$.[2]

*The results of Section 3 about Euclidean speed constraints were obtained after the paper was accepted for this conference, during my visit to the DIMACS Special Focus on Computational Geometry and Applications, supported by grant NSF EIA 00 87022.

†Freie Universität Berlin, Institut für Informatik, Takustraße 9, D-14195 Berlin, Germany, rote@inf.fu-berlin.de

[1]We may for example think of the thief as submerged under water, and the sheriff moving on the surface, being able to see only the tip of the thief's air-tube, which the thief can move at will within a limited radius. One could also regard the deviation between $R$ and $T$ as resulting from a sensing error on the side of the sheriff, as suggested by the title of the paper, but we prefer the metaphor where $R$ is controlled by a personal adversary (the thief) instead of an anonymous entity like "Nature".

[2]It can however be shown that it does not "make sense" for the thief to move $R$ faster than a certain bound.

It is clear that the thief can move on a straight line away from $S$ and thus make it impossible for $S$ to ever come closer to him, even if the thief always reveals his true position $R(t) = T(t)$. The question is whether the possibility of faking his position gives the thief essentially more power.

- Can the thief eventually get further and further away from the sheriff? If so, at what rate can he increase his distance?
- Or can the sheriff ensure that he remains within a bounded distance of the thief? If yes, what is this distance (assuming that they are close enough at the start)?

It is clear that the distance norm $\|\cdot\|_D$ can only affect the quantitative questions, but not the qualitative result of the game. However, the velocity norm $\|\cdot\|_V$ has a more profound influence on the problem.

For the Euclidean norm—this corresponds to the usual notion of a speed limit—we are going to see in Section 3 that the thief has a strategy which moves him further and further away from the sheriff. His distance increases like $\|T(t) - S(t)\| = \Omega(\sqrt[3]{t})$. The sheriff can ensure that the distance growth can asymptotically be no worse than this, by following the most natural "greedy" approach of always moving towards the current apparent target position $R$. The easy proof will be given in Section 2. On the other hand, if the velocity is constrained by a *polyhedral* metric, there is a bounded-distance strategy for the sheriff, see Section 4. This result holds in any dimension.

Such problems have been treated in recreational mathematics and mathematical games, like the lion-and-man problem [10, pp. 114–117], [13], but they also have applications in *search theory*, which deals with the planning of rescue searches [4]. Many other settings are possible, like cooperating players that try to meet despite incomplete information (*rendezvous search*), see [2]. Pursuit-evasion games have also been studied in graphs [1, 12] and inside polygonal regions [7, 9, 11, 14], with various capabilities of the pursuer and the fugitive, or with several pursuers instead of one.

We apply our techniques to a different field: to an on-line multi-processor scheduling problem with conflicts that was proposed in [5]. More details are given in Section 5. In fact, we originally developed the above pursuit-evasion game as a geometric abstraction of this scheduling problem. We can establish a bound of $n$ on the competitive ratio for the case when the conflict graph is a bipartite graph with $n$ vertices.

The geometric and polyhedral approach that is applied to a special on-line scheduling problem in this paper may prove to be useful for other problems.

# 2 Euclidean Speed Bound: A Greedy Strategy for the Sheriff

In this section and the next, the velocity norm $\|\cdot\|_V$ is the Euclidean norm $\|\cdot\|_2$, which we denote simply by $\|\cdot\|$ if no confusion is possible. We will also assume without loss of generality that $\|x\|_D \geq \|x\|_2$. This can always be achieved by choosing a different time scale and space scale.

**Theorem 1.** *When the sheriff always moves straight toward $R$, the distance increases by a rate of at most*

$$\|T(t) - S(t)\| \leq \sqrt{3/2} \cdot \sqrt[3]{t + \hat{t}} = O(\sqrt[3]{t}),$$

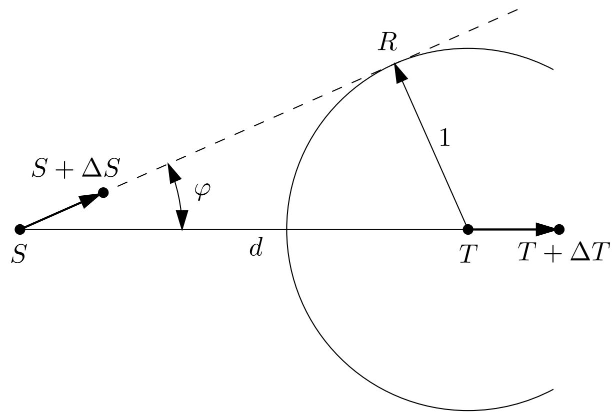


Figure 1: The distance increase of the greedy algorithm from time $t$ to time $t + \Delta t$.

*where the constant $\hat{t}$ is chosen to fulfill $\|T(0) - S(0)\| = \sqrt{3/2} \cdot \sqrt[3]{\hat{t}}$.*

*Proof.* If the direction between the direction $SR$ of movement and the true direction $ST$ to the target is $\varphi$, then the distance $d(t) := \|T(t) - S(t)\|$ increases at most by

$$\dot{d}(t) \leq 1 - \cos\varphi$$

if the sheriff always moves at full speed, see Figure 1. (The best that the thief can do is to move straight away from the sheriff.) $\dot{d}$ denotes the derivative with respect to $t$. From the constraint $\|T - R\| \leq 1$ we get $\sin\varphi \leq 1/d$, see Figure 1. Thus we have

$$\dot{d}(t) \leq 1 - \sqrt{1 - \frac{1}{d^2}} \leq 1 - \left(1 - \frac{1}{2d^2}\right) = \frac{1}{2d^2}, \tag{1}$$

using the inequality $\sqrt{1+x} \leq 1 + x/2$. The family of functions $f(t) = \sqrt{3/2} \cdot \sqrt[3]{t + \hat{t}}$, for any constant $\hat{t}$, satisfies this inequality as an equation: $\dot{f}(t) = 1/(2f(t)^2)$. □

# 3 Euclidean Speed Bound: How the Thief Can Escape

We will give an escape strategy for the thief which achieves the same asymptotic distance growth $\Theta(\sqrt[3]{t})$ as in Theorem 1. In fact, the growth is governed by a very similar relation as (1), only with a different constant. Similar as in the previous section, we will assume that $\|x\|_D \leq \|x\|_2$. (This is the reverse inequality of the previous section.)

The thief's strategy that we describe is against a known deterministic strategy for the sheriff. The thief knows what the sheriff will do, based on his observation of $R$. Alternatively, we can view the sheriff as an "off-line" player who need not specify $T$, but who must reveal only the positions of $R$, subject to the condition that a compatible path of $T$ exists. (This path is selected at the end after the game is over.) It is also possible to derive a randomized strategy that achieves the same growth rate.

**Theorem 2.** *The thief has a strategy which guarantees a distance increase of at least*

$$\|T(t) - S(t)\| \geq \sqrt[3]{\frac{3}{8} \cdot (t + \hat{t}\,)} - O\left(\frac{1}{\sqrt[3]{t}}\right) = \Theta(\sqrt[3]{t}),$$

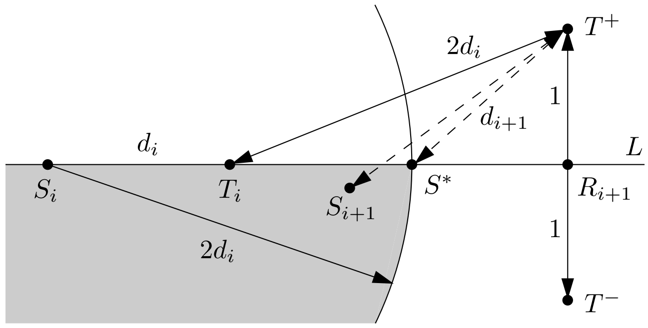


Figure 2: The escape strategy for the murderer.

*for some constant $\hat{t}$.*

*Proof.* The strategy proceeds in rounds. Let $S_i$ and $T_i$ be the positions at the start of the round, (at time $t_i$), and let let $d_i = d(t_i)$ be their distance $\|S_i - T_i\|$. The round will take time $2d_i$.[3] Let $L$ be the line $S_iT_i$, see Figure 2. The thief moves $R$ on line $L$ straight from $T_i$ away from $S_i$. $T$ moves on a straight line at a small angle with $L$, in such a way that at the end of the round, at time $t_{i+1} = t_i + 2d_i$, it has reached a point at distance 1 from $L$. There are two symmetric possibilities for the final position, which we denote by $T^+$ and $T^-$. Now from observing $R$, the sheriff cannot distinguish whether the thief moves to $T^+$ or $T^-$. So, at the end of the round it will be at some definite point $S_{i+1}$ inside the circle of radius $2d_i$ around $S_i$. If $S_{i+1}$ is below the line $L$ (in the shaded half-disk indicated in Figure 2), then the thief chooses to go to $T^+$, otherwise he goes to $T^-$. In this way, the thief always forces the sheriff to make a small detour. The minimum distance is achieved when the $S_{i+1}$ is the point $S^*$ on line $L$.

Let us now analyze the distance increase that the thief has gained in this way. Elementary calculations give

$$\begin{aligned} d_{i+1} &= \|T_{i+1} - S^*\| = \sqrt{(\sqrt{(2d_i)^2 - 1} - d_i)^2 + 1} \\ &= \sqrt{5d_i^2 - 2d_i\sqrt{4d_i^2 - 1}} = d_i \cdot \sqrt{5 - 4\sqrt{1 - \tfrac{1}{4d_i^2}}} \geq d_i \cdot \sqrt{5 - 4\left(1 - \tfrac{1}{8d_i^2}\right)} \\ &= d_i \cdot \sqrt{1 + \tfrac{1}{2d_i^2}} = d_i \cdot (1 + \tfrac{1}{4d_i^2} - O(\tfrac{1}{d_i^4})) = d_i + \tfrac{1}{4d_i} - O(\tfrac{1}{d_i^3}), \end{aligned}$$

using the inequalities $\sqrt{1+x} \leq 1 + x/2$ and $\sqrt{1+x} \geq 1 + x/2 - O(1/x^2)$. This defines a sequence of pairs $(t_i, d_i)$ with $d_i = d(t_i)$ by the relations $t_{i+1} = t_i + 2d_i$ and

$$d_{i+1} = \sqrt{5d_i^2 - 2d_i\sqrt{4d_i^2 - 1}} = d_i + \tfrac{1}{4d_i} - O(\tfrac{1}{d_i^3}). \tag{2}$$

For simplicity, we ignore the $O(\frac{1}{d_i^3})$ error term in the following computation. The complete calculation will appear in the full version of the paper. It is easy to check that the function $f(t) := \sqrt[3]{\frac{3}{8} \cdot (t + \hat{t}\,)}$, for any constant $\hat{t}$, satisfies the relation

$$f(t + 2f(t)) \leq f(t) + \frac{1}{4f(t)},$$

[3]The constant 2 in $2d$ is the optimal choice for this proof.

by a similar calculation as above, using $\sqrt[3]{1+x} \le 1 + x/3$.

$$\begin{aligned}
f(t) &= \sqrt[3]{ct} \\
f(t+2f(t)) &= \sqrt[3]{c(t+2\sqrt[3]{ct})} = \sqrt[3]{ct} \cdot \sqrt[3]{1+2c(ct)^{-2/3}} \\
&\le \sqrt[3]{ct} \cdot (1 + 2c/3 \cdot (ct)^{-2/3}) = \sqrt[3]{ct} + 2c/3 \cdot (ct)^{-1/3} = \sqrt[3]{ct} + \frac{8c/3}{4\sqrt[3]{ct}}
\end{aligned}$$

Now it follows by induction that $d(t) \ge f(t)$ holds for the sequence of values $t_0, t_1, t_2, \dots$ if we choose $\hat{t}$ so that it holds initially at $t = t_0$. For the inductive step from $i$ to $i+1$, find a value $\tilde{t}$ such that $f(\tilde{t}) = d(t_i)$. By the inductive assumption $d(t_i) \ge f(t_i)$ and by the monotonicity of $f$ we have $\tilde{t} \ge t$. Thus,

$$\begin{aligned}
d(t_{i+1}) &= d(t_i) + \tfrac{1}{4d(t_i)} = f(\tilde{t}) + \tfrac{1}{4f(\tilde{t})} \ge f(\tilde{t} + 2f(\tilde{t})) \\
&= f(\tilde{t} + 2d(t)) \ge f(t + 2d(t)) = f(t_{i+1})
\end{aligned}$$

We have proved that $\|T(t) - S(t)\| \ge \sqrt[3]{\frac{3}{8} \cdot (t + \hat{t}\,)}$ holds at the beginning and at the end of each round (for the times $t_0, t_1, \dots$). During a round, the distance may temporarily decrease, since the thief moves at an angle from the line $ST$. This, and the $O(\frac{1}{d_i^3})$ error term in (2), accounts for the term $-O(1/\sqrt[3]{t})$ in Theorem 2. □

The possible decrease of $\|S - T\|$ during a round comes at the expense of an *additional* increase at the end of the round. Thus, it might be possible to get rid of the error term $O(1/\sqrt[3]{t})$, but we have not analyzed this. It is probably easier to increase the constant $\sqrt[3]{3/8}$ by a more refined algorithm.

The thief can use a *randomized* strategy which does not assume any knowledge about the strategy of the sheriff. If the thief simply chooses the points $T^+$ and $T^-$ with probability 1/2 each, then $d_i$ becomes a random variable, and one can show that (2) holds for the *expected* distance $d_{i+1}$. Thus, Theorem 2 holds for the expected distance of this randomized algorithm.

# 4 Polyhedral Distance Functions

A polyhedral norm $\|\cdot\|_V$ in $\mathbb{R}^n$ is given by some polytope $V$ which is centrally symmetric about the origin and contains the origin in its interior, see Figure 3a.[4] $V$ is then the unit ball of this norm, and

$$\|x\|_V := \min\{\, \lambda \ge 0 \mid x \in \lambda \cdot V \,\},$$

see Figure 3b. The polytope $V$ can be written as an intersection of half-spaces

$$V = \{\, x \mid a_i \cdot x \le 1 \text{ for } i = 1, \dots, k \,\}$$

Each half-space corresponds to a facet $F_i$ of $V$.

$$F_i := \{\, x \in V \mid a_i \cdot x = 1 \,\}$$

[4] The requirement that $V$ is symmetric is only required to satisfy the axioms of a norm. The results of this paper don't require symmetry. A non-symmetric polytope $V$ gives rise to a non-symmetric distance function.

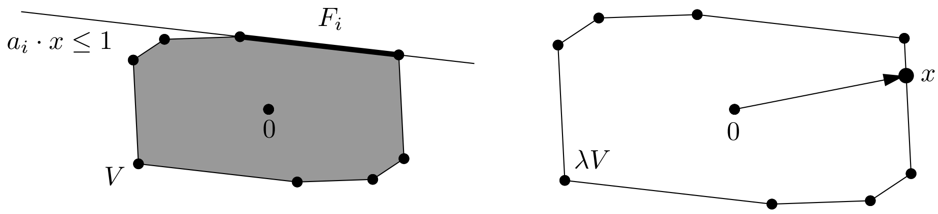


Figure 3: The polyhedral norm induced by $V$

Now suppose that the sheriff is at position $S$ and wants to reach a fixed target point $T$ as fast as possible. The shortest time is given by the distance $\|T - S\|_V$, and it can for example be reached by moving on a straight line from $S$ to $T$. However, this is not the only possibility, see Figure 4: We can draw the ray from $S$ to $T$ and determine the facet $F_i$ at which this ray intersects the polytope $V$ centered at $S$. Now, a small move towards *any* point on $F_i$ will decrease distance from $S$ to $T$ at the optimal rate.

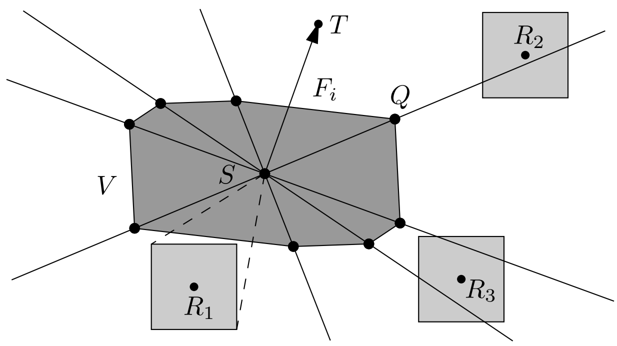


Figure 4: Various ways how the rays from $S$ to $R + D$ can lie with respect to $V$

**Lemma 1.** *The set of directions in which $S$ can move on a fastest route to $T$ is given by the facet or the facets of $V$ that are hit by the ray from $S$ to $T$.* □

Now, consider the case when the location of the target $T$ is not known precisely, but it is known in some neighborhood $R + D$ of a point $R$, where $D$ is the unit ball of the metric $\|\cdot\|_D$, see Figure 4. It may happen that the ray $ST$ exits $S + V$ in a unique facet $F_i$, regardless of where $T$ is chosen, as shown in the case of $R_1 + D$. Even if the facet $F_i$ is not uniquely determined, all possible facets might still meet in a common point, such as $Q$ in the case of $R_2 + D$. $Q$ can always be chosen as a vertex of $V$. This point specifies the direction in which $S$ has to move. It may happen that there is no such direction, as in $R_3 + D$.

The crucial idea is that the third case cannot occur when $R$ is sufficiently far away from $S$.

**Theorem 3.** *If speed is measured according to a polyhedral distance function $V$, there is a constant $c$ such that whenever the distance between the sheriff $S$ and the revealed position $R$ of the thief is larger than $c$, the sheriff has a strategy which ensures that the distance $\|T - S\|_V$ does not increase.*

*Proof.* If we succeed to find a direction $Q$ which is valid according to Lemma 1 for all choices of $T$ in $R + D$, we can ensure that we decrease the distance $\|T - S\|_V$ at rate 1,

i. e., we would decrease it by $\varepsilon$ in time $\varepsilon$ if $T$ were static. On the other hand, $T$ may move away at maximum speed, increasing $\|T - S\|_V$ at a rate at most 1. Thus, the distance can only decrease or stay the same.

We assume for convenience that $S$ is the origin. Let $F_1, \ldots, F_l$ be the facets of $V$ which are hit by the radial projection of $R + D$ onto $V$. Let us assume that these facets don't intersect in a common vertex $Q$. We have to show that this can only happen for $\|R - S\| \leq c$. (We can measure the distance $\|R - S\|$ by any convenient norm.)

We project the facets $F_i$ onto the unit sphere $S^{n-1}$ in $\mathbb{R}^n$, yielding spherical polytopes $\hat{F}_1, \ldots, \hat{F}_l$. Let $K$ be the smallest ball on $S^{n-1}$ which intersects each of $\hat{F}_1, \ldots, \hat{F}_l$. $K$ must have a positive radius $r_K > 0$, because otherwise $\hat{F}_1, \ldots, \hat{F}_l$ would contain a common point. Therefore, when $R$ is sufficiently far away, we can ensure that the projection of $R + D$ onto the unit sphere is contained in a sphere smaller than $r_K$. This means that, if $\|R - S\| > c$ for an appropriate constant $c$, $F_1, \ldots, F_l$ cannot be the facets which are hit by the projection of $R + D$ onto $V$.

Such a constant $c$ can be determined for each set $F_1, \ldots, F_l$ of facets which have no common vertex. By taking the maximum $c$ over all these finitely many choices, we obtain the theorem. □

We will briefly discuss how one can make this proof more effective. In principle, for small examples, one can determine the optimum value of the constant $c$.

For each facet $F_i$ of $V$, the rays from the origin $S$ through the points of $F_i$ form a polyhedral cone which we denote by $SF_i$. This cone is given by

$$SF_i = \{\, x \in \mathbb{R}^n \mid a_i \cdot x \leq a_j \cdot x \text{ for all } j = 1, \ldots, k,\ j \neq i \,\}. \tag{3}$$

(It is sufficient to take the indices $j$ of the facets adjacent to $F_i$, if the structure of the polytope $V$ is known.)

We can directly write the constraints that the cones $SF_i$ intersect $R + D$ by postulating the existence of a point $T_i$:

$$T_i \in SF_i \quad \text{and} \quad T_i \in R + D$$

If $D$ is a polytope, all these constraints can be written as linear inequalities in the unknown coordinates $T_i$ and $R$. By checking all vertices of this polytope, one can determine the largest value of $\|R - S\|$. If $\|R - S\|$ is measured by a polyhedral norm, the largest value can also be found by a sequence of linear programming problems.

This procedure would have to be repeated for all sets $F_1, \ldots, F_l$ of facets which have no common vertex.

By Helly's theorem, applied to the convex sets $SF_i - \{S\}$, there is a subfamily of at most $n + 1$ facets without common vertex. It is therefore sufficient to check all subsets consisting of $2, 3, \ldots, n + 1$ facets.

**Using history.** By using information from previous values of $R$, it is often possible to restrict $T$ to a smaller set than $R + D$ whenever $R$ moves fast. By the speed constraint we know that $T(t) \in R(t') + D + (t - t') \cdot V$, for all $t' \leq t$. If this is exploited in a systematic way, it might lead to better bounds.

# 5 On-line Buffer Minimization

We consider the scheduling problem for a sequence of tasks in a multi-processor system with conflicts due to the fact that two or more processors share common resources that can only be accessed by one processor at a time. This can be modeled as an undirected graph $G = (P, E)$, where $P$ is the set of processors and the edges in $E$ represent conflicts. The elements of $P$ will be called the *nodes* of the graph to distinguish them from vertices of polytopes. Processors that may run simultaneously form an independent set in this graph.

At certain times new tasks arrive in the system, where each task specifies the amount of work (processing time) added to each processor's workload. Each processor stores this workload in its input buffer. Our objective is to schedule task execution, obeying the conflict constraints, and minimizing the maximum buffer size of all processors.

An on-line algorithm processes the workload from the buffers without knowledge of future tasks.

We evaluate on-line algorithms by comparing their buffer size to that of an optimal off-line algorithm. An on-line algorithm is $c$-competitive for a graph $G$ if, for any task sequence that has an off-line schedule with maximum buffer size $B$, it constructs a feasible schedule with buffer size at most $cB$. The competitive ratio of $G$, denoted $\mathit{buf}(G)$, is the smallest value of $c$ for which there is a $c$-competitive on-line algorithm on $G$.

This problem was introduced by Chrobak, Csirik, Imreh, Noga, Sgall, and Woeginger [5]. They showed that $\mathit{buf}(G)$ is finite for all graphs $G$, and they calculated $\mathit{buf}(G)$ for complete graphs and complete multipartite graphs. For trees of diameter $d$ they proved an upper bound of $\mathit{buf}(G) \leq 1 + d/2$.

When $G$ is the complete graph on $n$ nodes, there is essentially a single processor which has to process the tasks from all queues. This situation has later been independently analyzed by [3, 8]. The optimal competitive ratio $H_n = 1 + 1/2 + \cdots + 1/n$ is achieved by a greedy algorithm. In Theorem 4 we will give an upper bound of $n$ on the competitive ratio for a bipartite graph on $n$ nodes.

**The stable set polytope, buffer minimization, and fractional chromatic number.** Suppose that the current load vector is $w$. Then the minimum time needed to empty all buffers (without any new tasks arriving) is given by the following linear program, where we use $I$ to denote independent sets in $G$.

$$\begin{aligned} \min \lambda = &\sum_{I} y_I \\ \text{s.t.} &\sum_{I \ni j} y_I = w_j \text{ for } j \in P \\ &y_I \geq 0 \text{ for each } I \end{aligned} \tag{4}$$

The variables $y_I$ indicate the amount of time for which $I$ should be run in this schedule.

If $w = \mathbf{1}$ (the vector of all 1's) and if the variables $y_I$ are restricted to integral values, then the optimal solution of the resulting integer program is just the chromatic number $\chi(G)$: Every independent set $I$ with $y_I = 1$ forms a color class. When the integrality constraints are relaxed, the optimal value is called the *fractional chromatic number* $\chi_f(G)$. The above linear program is a weighted generalization, and we denote the optimal value by $\chi_f(G, w)$. The chromatic number and the fractional chromatic number are closely related, see [6].

The *stable set polytope* $\mathrm{STAB}(G)$ is the convex hull of the incidence vectors of the independent sets $I$. It is the set of weight vectors $w$ for which the linear program

(4) has a solution of value 1. Therefore, $\chi_f(G,w)$ equals just the norm $\|w\|_V$ for $V = \mathrm{STAB}(G)$:

$$\chi_f(G,w) = \min\{\, \lambda \mid w \in \lambda \cdot \mathrm{STAB}(G) \,\}$$

(To satisfy the requirements of a norm, we would have to extend this polytope $V$, which is defined to lie in the nonnegative orthant, to all other orthants. But we will compute the norm only for nonnegative vectors $w$.)

We model the scheduling problem as a pursuit-evasion game in $n$ dimensions as follows. $R(t)$ is the cumulative workload that has arrived until time $t$. $S(t)$ represents the total work that has been done on each processor by the on-line algorithm. The work that is stored in the buffers is $R - S$. Thus, we reduce the work stored in buffers by moving $S$ closer to $R$. The maximum buffer size is given by the maximum norm $\|R - S\|_\infty$. $T(t)$ represents the total work that has been done on all processors by the off-line algorithm, increased by $B \cdot \mathbf{1}$, where $B$ is the maximum buffer size of the off-line algorithm. The addition of $B \cdot \mathbf{1}$ ensures that $S \le T$.

This fits the framework of our pursuit-evasion game, with the following modifications:

- $S$ and $T$ are restricted to move monotonically in each direction, with the speed limit given by $V = \mathrm{STAB}(G)$.
- $R$ is restricted by $R \le T$ and $\|R - T\|_\infty \le B$, and it can only move in discrete time steps.
- $S$ is restricted by $S \le R$, and we want to minimize $\|S - R\|_\infty$.

Apart from the restriction $S \le R$, all constraints can be accommodated in the setting of Section 4, and thus, finiteness of *buf*$(G)$ can be obtained as a consequence of Theorem 3, with appropriate modifications to account for the constraint $S \le R$. In Theorem 4 we will show that, for bipartite graphs $G$ with $n$ nodes, *buf*$(G)$ is bounded by $n$.

The stable set polytope $V = \mathrm{STAB}(G)$ of a bipartite graph $G = (P, E)$ is given by the inequalities

$$x_i + x_j \le 1, \text{ for all edges } ij \in E$$

and the nonnegativity constraints $x_i \ge 0$. Thus, it has a facet $F_{ij} = \{\, x \in \mathrm{STAB}(G) \mid x_i + x_j = 1 \,\}$ for each edge $ij \in E$, and we have

$$\|w\|_V = \max\{\, w_i + w_j \mid ij \in E \,\}$$

for all $w \ge 0$.

The on-line algorithm which we propose is the attempt to carry out the program of the proof of Theorem 3 for this particular problem and this particular polytope $V$.

The steps of the algorithm will perhaps seem completely unmotivated at first sight. Their purpose is to guarantee the existence of a vertex $Q$ of $V$ which is a "good direction" provided that $R$ is far enough from $S$. It will be seen that the algorithm fits perfectly together with the lemmas in the ensuing proof of competitiveness, which relies on the facet structure of $F$.

*Algorithm.* Let $P = P_1 \cup P_2$ be the decomposition of the node set into the two color classes. By $a_i := R_i - S_i$ we denote the buffered workload for processor $i$. Let $N := \{\, i \in P \mid a_i > 0 \,\}$ be the set of processors with some work to do and $Z := P - N$ the set of processors with zero work. We have to select an independent subset $I \subseteq N$

for processing. For a parameter $\beta \geq 0$, let $H(\beta) := \{ ij \in E \mid a_i + a_j > \beta \}$ denote the edges with "high load". We determine the smallest value $\beta$ such that the graph $G[H(\beta)]$ with node set $P$ and edge set $H(\beta)$ contains no path from a node in $P_1 \cap Z$ to a node in $P_2 \cap Z$. Then in each component $C$ of $G[H(\beta)]$, we select the nodes of $I$ as follows: If $C$ contains a node in $P_1 \cap Z$, we put the nodes of $C \cap P_2$ into $I$. If $C$ contains a node in $P_2 \cap Z$, we put the nodes of $C \cap P_1$ into $I$. If $C$ contains no node of $Z$, we arbitrarily choose a color class of $C$ and put it into $I$. Clearly, $I$ is an independent set and contains no node of $Z$. Furthermore, every edge $ij \in H(\beta)$ is *covered* by $I$, in the sense that either $i \in I$ or $j \in I$. We process $I$ until a new job arrives or until the graph $H(\beta)$ changes due to the decrease of the workloads $a_i$.

**Theorem 4.** *If $G$ is bipartite and the longest simple odd path in $G$ has length $2\Delta + 1$, then $buf(G) \leq 2\Delta + 2$.*

**Corollary 1.** *If $G$ is bipartite with $n_1 + n_2$ nodes in its two color classes, then $buf(G) \leq 2 \cdot \min\{n_1, n_2\}$.*

For trees with diameter $d$, the bound that we get from Theorem 4 is about twice as large as the bound $1 + d/2$ of [5].

The proof of Theorem 4 proceeds in a couple of lemmas. Let $\alpha := \|R - S\|_V = \max\{ a_i + a_j \mid ij \in E \}$. We start by bounding the value of $\beta$ that is determined in the algorithm.

**Lemma 2.**

$$\beta \leq \alpha \cdot \frac{\Delta}{\Delta + 1}$$

*Proof.* Let $\alpha' := \alpha\Delta/(\Delta+1)$ and assume that $G[H(\alpha')]$ contains an odd path $i_0, i_1, \ldots, i_{2k+1}$ from a node $i_0 \in Z$ to a node $i_{2k+1} \in Z$. We have the relations $a_0 + a_1 > \alpha'$, $a_2 + a_3 > \alpha'$, $\ldots$, and $a_1 + a_2 \leq \alpha$, $a_3 + a_4 \leq \alpha$, $\ldots$, as well as $a_0 = a_{2k+1} = 0$. By the assumption we know that $k \leq \Delta$, and by subtracting the second class of inequalities from the first class one derives the contradiction $(k + 1)\alpha' < k\alpha$. □

*Remark.* The definition of $\Delta$ by the longest path is actually more general than what is needed in this proof. Only the following very special condition is necessary: We consider certain kinds of alternating paths in $G$ with a designated subset of nodes $Z$ and a designated subset of edges $H$: paths from a node in $Z \cap P_1$ to a node in $Z \cap P_2$, such that every odd-numbered edge of such a path belongs to $H$. For every choice of $H$ and $Z$, whenever such an alternating path exists between a pair of nodes of $Z \cap P_1$ and $Z \cap P_2$, then there must be such an alternating path of length at most $2\Delta + 1$ between those two nodes.

For example, when $G$ is a complete bipartite graph, one can take $\Delta = 1$.

Let $B$ denote the buffer size of the off-line algorithm.

**Lemma 3.** *Let $I$ be the independent set found by the algorithm, and let $x^I$ be the corresponding vertex of $V = \mathrm{STAB}(G)$. If $\alpha = \|R - S\|_V \geq 2(\Delta + 1)B$ then every ray from $S$ to a point in the cube $[R, R + B]$ goes through one of the facets $F_{ij}$ of $V$ which contain the vertex $x^I$.*

*Proof.* $[R, R + B]$ denotes the cube in which $T$ is known to lie. The facets $F_{ij}$ incident to $x^I$ correspond precisely to the edges $ij$ that are covered by $I$, i. e., $i \in I$ or $j \in I$.

For deriving a contradiction, assume that a ray to a point $T \in [R, R+B]$ goes through a face $F_{kl}$, where $kl$ is disjoint from $I$.

Regarding $S$ as the origin and writing $t_i$ for the coordinates of $T - S$, we then have

$$t_k + t_l > t_i + t_j, \text{ for all } ij \in E, ij \neq kl$$

by (3) and

$$a_i \leq t_i \leq a_i + B, \text{ for all } i \in P$$

This leads to

$$a_k + a_l > a_i + a_j - 2B, \text{ for all } ij \in E$$

and hence $a_k + a_l > \alpha - 2B \geq \alpha\Delta/(\Delta + 1) \geq \beta$, by Lemma 2. Hence $kl \in H(\beta)$ and therefore the edge $kl$ is covered by $I$, a contradiction. □

Under the conditions of Lemma 3, it follows that $S$ can decrease his distance to $T$ at full speed. As in Theorem 3, it follows that the invariant $\|R - S\|_V \leq 2(\Delta + 1)B$ will be maintained. From this we conclude that for all $i$, $a_i \leq \max\{ a_i + a_j \mid ij \in E \} = \|R - S\|_V \leq 2(\Delta + 1)B$, i. e., the algorithm is $2(\Delta + 1)$-competitive.

This concludes the proof of Theorem 4. □

# 6 Open Questions

We have seen a striking difference between the Euclidean norm and polyhedral norms with respect to the pursuit evasion-game. The proof of Theorem 3 is somehow in accordance with this. The tendency is that the bound degrades as the polytope $V$ has more and more small facets with small angles between them. It would be interesting to find the exact boundary between those norms that allow the thief to escape, like the Euclidean norm, and those norms that behave like polyhedral norms.

The main problem is of course to improve the bounds by using more sophisticated algorithms than the simple greedy-style algorithms of Section 2 and 4, and to come up with lower bounds as well.

# References


[1] Micah Adler, Harald Räcke, Naveen Sivadasan, Christian Sohler, and Berthold Vöcking. Randomized pursuit-evasion in graphs. In *Automata, Languages, and Programming*, Proc. 29th Int. Coll., ICALP 2002, Malaga, July 8–13, 2002, (P. Widmayer et al., eds.) Lect. Notes Comput. Sci., Vol. 2380, Springer-Verlag 2002, pp. 901–912.

[2] Edward J. Anderson and Sándor P. Fekete. Asymmetric rendezvous in the plane. In *Proc. 14th Ann. Symp. Computational Geometry*, SCG '98, Minneapolis, (R. Janardan, ed.) ACM Press, 1998, pp. 365–373.

[3] Amotz Bar-Noy, Ari Freund, Shimon Landa, and Joseph (Seffi) Naor. Competitive on-line switching policies. In *Proc. 13th Ann. ACM-SIAM Symp. Discrete Algorithms*, SODA 2002, San Francisco, Jan. 6–8, 2002, (David Eppstein, ed.) ACM Press, 2002, pp. 525–534.

[4] S. J. Benkoski, M. G. Monticino, and J. R. Weisinger. A survey of the search theory literature. *Naval Research Logistics* **38** (1991), 469–494.

[5] Marek Chrobak, János Csirik, Csanád Imreh, John Noga, Jiří Sgall, and Gerhard J. Woeginger. The buffer minimization problem for multiprocessor scheduling with conflicts. In *Automata, Languages, and Programming*, Proc. 28th Int. Coll., ICALP 2001, Crete, July 8–12, 2001, (F. Orejas, P. G. Spirakis, and J. van Leeuwen, eds.) Lect. Notes Comput. Sci., Vol. 2076, Springer-Verlag 2001, pp. 862–874.

[6] M. Grötschel, L. Lovász, and A. Schrijver. The ellipsoid method and its consequences in combinatorial optimization. *Combinatorica* **1** (1981), 169–197.

[7] L. J. Guibas, J.-C. Latombe, S. M. LaValle, D. Lin, and R. Motwani. A visibility-based pursuit-evasion problem. *Int. J. Comput. Geom. Appl.* **9** (1999), 471–493.

[8] Hisashi Koga. Balanced scheduling toward loss-free packet queuing and delay fairness. In *Algorithms and Computation*. Proc. 12th International Symposium, ISAAC 2001, Christchurch, New Zealand, December 19–21, 2001, (P. Eades and T. Takaoka, eds.) Lecture Notes in Computer Science, Vol. 2223, Springer-Verlag 2001, pp. 61–73.

[9] Steven M. LaValle and John E. Hinrichsen. Visibility-based pursuit-evasion: the case of curved environments. In *Proc. IEEE Int. Conf. Robotics Automation*, 1999, pp. 1677–1682.

[10] John E. Littlewood. *A Mathematician's Miscellany*, Methuen & Co., London 1953. Revised edition published as *Littlewood's Miscellany*, Cambridge University Press, 1986.

[11] Sang-Min Park, Jae-Ha Lee, and Kyung-Yong Chwa. Visibility-based pursuit-evasion in a polygonal region by a searcher. In *Automata, Languages and Programming*. Proc. 28th Int. Coll., ICALP 2001, Crete, July 2001, (F. Orejas, P. G. Spirakis, and J. van Leeuwen, eds.) Lect. Notes Comput. Sci., Vol. 2076, Springer-Verlag, 2001, pp. 456–468.

[12] T. D. Parsons. Pursuit-evasion in a graph, in *Theory and Applications of Graphs* (Y. Alavi and D. Lick, eds.) Lect. Notes Math., Vol. 642, Springer-Verlag, Berlin, 1976, pp. 426–441.

[13] Jiří Sgall. A solution of David Gale's man and lion problem. *Theoret. Comput. Sci.* **259** (2001), 663–670.

[14] I. Suzuki and M. Yamashita. Searching for a mobile intruder in a polygonal region. *SIAM J. Comput.* **21** (1992), 863–888.