

Bounding an Optimal Search Path with a Game of Cop and Robber on Graphs

Frédéric Simard, Michael Morin^(✉), Claude-Guy Quimper, François Laviolette,
and Josée Desharnais

Department of Computer Science and Software Engineering, Université Laval,
Québec, QC, Canada

{Frederic.Simard.10,Michael.Morin.3}@ulaval.ca,
{Claude-Guy.Quimper,Francois.Laviolette,Josee.Desharnais}@ift.ulaval.ca

Abstract. In search theory, the goal of the Optimal Search Path (OSP) problem is to find a finite length path maximizing the probability that a searcher detects a lost wanderer on a graph. We propose to bound the probability of finding the wanderer in the remaining search time by relaxing the problem into a stochastic game of cop and robber from graph theory. We discuss the validity of this bound and demonstrate its effectiveness on a constraint programming model of the problem. Experimental results show how our novel bound compares favorably to the DMEAN bound from the literature, a state-of-the-art bound based on a relaxation of the OSP into a longest path problem.

Keywords: Optimal search path · Cop and robber · Constraint relaxation · Pursuit games

1 Introduction

The Optimal Search Path (OSP) problem [1, 2] is a classical problem from search theory [3] of which the SAROPS [4] and the SARPlan [5] are two examples of successful systems. The goal of an OSP is to find the best finite length path a searcher has to take in order to maximize the probability of finding a lost wanderer moving randomly on a graph. It is a common assumption, in search theory, to consider unconstrained searcher’s motion. The assumption holds, for instance, in contexts where the searcher is fast in comparison to the target of interest. There are, however, many cases, such as in land search and rescue [3], where the searcher is as slow as the moving wanderer. Washburn [6] also notes how bounded speeds (for searcher and wanderer) is used in modeling searches for hostile submarines or friendly vessels.

Most of the early work on the OSP problem consisted in computing bounds for a depth-first branch-and-bound algorithm initially developed by Stewart [2].¹ At the opposite of stochastic constraint programming, the OSP requires to compute a single scenario given by the decision variables and the probability that

¹ A review of the bounds developed prior to 1993 is found in the work of Washburn [6].

this scenario works is automatically computed through constraint propagation. Recent works on the OSP problem include generalization of the problem to consider, for instance, non-uniform travel time [7], searching from a distance [8], and path dependent detections [9]. Along with a novel OSP generalization, Lau et al. [7] propose the DMEAN bound. The bound is proved to effectively prune the branch-and-bound search tree.

In this paper, we present bounding techniques for a constraint programming (CP) model of the OSP problem introduced in [10]. We discuss, in Section 2, the DMEAN bound which has never been implemented in CP. Then, as our main contribution, we develop a bound on the OSP objective function (see Section 3). This bound, we call the Copwin bound, is obtained by relaxing the OSP into a game of cop and robber [11–14] which pertains to a well-studied game from graph theory (see Section 3). In Section 4, we show the benefits of using our novel Copwin bound by providing experimental results obtained on an existing CP model of the problem [10]. We conclude in Section 5.

2 The Optimal Search Path Problem

The OSP models a search operation where one searcher searches for a lost, and possibly moving, search object. The problem is known to be \mathcal{NP} -hard [1]. It is used, in search theory, as a model for path-constrained search operations [3] and can be formulated as follows. A searcher and a wanderer are moving on the vertices of a graph $G = (V(G), E(G))$. Their respective initial positions correspond to some vertex of $V(G)$ or of some subset of it. Each of their moves, that we can assume simultaneous, is performed along one of the edges $e \in E(G)$ of the graph and consumes one unit of time. There is a maximal number T of steps taken for the search operation. The wanderer \mathcal{W} is invisible to the searcher \mathcal{S} , but each time they share a vertex $v \in V(G)$, the latter has a given probability of detecting and removing him² from the graph:

$\text{pod}(v) :=$ probability of detecting \mathcal{W} when both \mathcal{W} and \mathcal{S} are on vertex v .

Let $\pi_t = v_1, v_2, \dots, v_t$ be a path of length $t \leq T$ representing the t first moves of the searcher ($v_i \in V(G)$, $i = 1, \dots, t$). We define $\pi_{t[.k]} := v_1, v_2, \dots, v_k$ for $k \leq t$. We define the following probabilities for a searcher \mathcal{S} and a wanderer \mathcal{W} :

$\text{poc}_{\pi_t}(v) :=$ probability³ of \mathcal{W} being on vertex v at time t and not having previously been detected, given that \mathcal{S} 's first t steps were along π_t ;
 $\text{cos}_{\pi_t} :=$ Cumulative probability Of Success of detecting \mathcal{W} up to time t when \mathcal{S} follows π_t , i.e., it is given as $\sum_{i=1}^t \text{poc}_{\pi_{t[.t-1]}}(v_i)\text{pod}(v_i)$, where $\text{poc}_{\pi_{t[.0]}}$ is the initial distribution of the wanderer;

² Following the convention of cop and robber games on graphs, we suppose that the searcher (or cop) is a woman and that the wanderer (or robber) is a man.

³ $\text{poc}_t(v)$ also stands for the probability of containment [10].

$\text{cos}_{\pi_t}^*$:= maximal cumulative probability of success up to T if \mathcal{S} 's t first steps are along π_t , that is, $\max\{\text{cos}_{\pi_T} \mid \pi_T \text{ has prefix } \pi_t\}$.

The goal of the searcher is to find the path π_T with maximal cumulative probability of success cos_{π_T} . This probability is actually $\text{cos}_{\pi_0}^*$, where π_0 is the empty path.

The wanderer is passive in that he does not react to the searcher's moves. It is the usual assumption to model the wanderer's motion by using a Markovian transition matrix M with source vertices on rows and destination vertices on columns. Thus, $M(r, r')$ is the probability of a wanderer's move from vertex r to vertex r' within one time step. Each time the searcher searches a vertex r' along path π_t and the search operation does not stop, then either the wanderer is not there, or he is undetected; the latter happens with probability $1 - \text{pod}(r')$. That is, as long as the search goes on, the wanderer is not found and the knowledge about his location evolves according to:

$$\begin{aligned} \text{poc}_{\pi_t}(r') &= \text{poc}_{\pi_{t[\dots, t-1]}}(v_t)(1 - \text{pod}(v_t))M(v_t, r') \\ &+ \sum_{r \in V(G) \setminus \{v_t\}} \text{poc}_{\pi_{t[\dots, t-1]}}(r)M(r, r'). \end{aligned} \quad (1)$$

The next section summarizes the CP model of the OSP used in [10].

2.1 Modeling the OSP in Constraint Programming

A CP model for the OSP problem follows directly from the problem's definition. We define, for each step t ($1 \leq t \leq T$), a *decision variable* $PATH_t$ that represents the searcher's position. The $PATH_t$ domain is an enumerated domain such that $\text{dom}(PATH_t) = V(G)$. The searcher's path is constrained by a possible limited choice for its initial position $PATH_1$ and by the fact that

$$(PATH_{t-1}, PATH_t) \in E(G) \quad (2)$$

for any steps $t \in \{2, \dots, T\}$. The encoding in the CP-framework is done via a sequence of TABLE constraints [15].

We define two sets of *implicit variables* with bounded domains for probabilities: $POC_t(v)$ with $\text{dom}(POC_t(v)) = [0, 1]$, and $POS_t(v)$ with $\text{dom}(POS_t(v)) = [0, 1]$ for $1 \leq t \leq T$ and for $v \in V(G)$. If the searcher is located in a vertex v , then she obtains a success probability as modeled by the following constraint set defined over all time steps $t \in \{1, \dots, T\}$ and vertices $v \in V(G)$, where $\text{pod}(v)$ is the known probability of detection in vertex v :

$$PATH_t = v \Rightarrow POS_t(v) = POC_t(v)\text{pod}(v). \quad (3)$$

If the searcher is not located in vertex v at step t , then the probability of detecting the object in v at step t is null. This is modeled by the following constraint set defined over all time steps $t \in \{1, \dots, T\}$ and vertices $v \in V(G)$:

$$PATH_t \neq v \Rightarrow POS_t(v) = 0. \quad (4)$$

The value of the $POC_1(r)$ variables is fixed given the initial distribution on the wanderer’s location, a value known for all $r \in V(G)$. For all subsequent time steps $t \in \{2, \dots, T\}$ and vertices $v \in V(G)$, the evolution of the searcher’s knowledge on the wanderer’s location is modeled by the following constraint set:

$$POC_t(v) = \sum_{r \in V(G)} M(r, r') (POC_{t-1}(r) - POS_{t-1}(r)). \tag{5}$$

A bounded domain variable Z with $\text{dom}(Z) = [0, 1]$ is added to represent the objective value to maximize which corresponds to the cumulative overall probability of success as defined above by cos_{π_T} where π_T is an assignment to the $PATH_t$ variables. Following [10], we observe that the searcher is in one vertex at a time leading to

$$Z = \sum_{t=1}^T \max_{v \in V(G)} POS_t(v) \tag{6}$$

and improving filtering over the usual double sum definition of the objective. The objective is finally

$$\max_{PATH_1, \dots, PATH_T} Z, \tag{7}$$

subject to Constraints (2) to (6) which leads to $\text{cos}_{\pi_0}^*$ value.

2.2 The DMEAN Bound from the Literature

Martins [16] proposes to bound the objective function using the MEAN bound. Lau et al. [7] improve on this bound by presenting the discounted MEAN bound (DMEAN) which tightens the MEAN at a reasonable computational cost. In the literature, these bounds are usually applied in an OSP-specific depth-first branch-and-bound algorithm. In this section, we show how to use the DMEAN bound in our OSP CP model. We chose to implement DMEAN among other bounds from the literature since both DMEAN and our novel bound are based on graph theory problems. While DMEAN aims at solving a longest path problem, the Copwin bound, as will be explained, is based on pursuit games. Just as DMEAN, MEAN is based on the maximization of the expected number of detections in the remaining time. The core idea of the MEAN bound is to construct a directed acyclic graph (DAG) at time $t \in \{1, \dots, T\}$ on vertices consisting of pairs (r, k) where $r \in V(G)$ and $t < k \leq T$. A directed edge is added from vertex (r, k) to vertex $(r', k + 1)$ if and only if $(r, r') \in E(G)$. The maximal expected number of detections Z_{DAG} from step t to T is obtained by following the longest path on the DAG, which can be computed in time linear on $|V(G)| + |E(G)|$. Given the beginning π_t of a path, it is sufficient to know $\text{poc}_{\pi_t}(v)$ for all $v \in V(G)$ to be able to compute the expected detection at step k with $t \leq k \leq T$ in vertex $r \in V(G)$. An incident edge into a node $(r', k + 1)$ is weighted as follows:

$$R_{\text{MEAN}}(r', k + 1) = \sum_{v \in V(G)} \text{poc}_{\pi_t}(v) M^{k+1-t}(v, r') \text{pod}(r') \tag{8}$$

where M^{k+1-t} is the transition matrix M to the power of $k+1-t$.

The DMEAN bound is based on the additional observation that a wanderer moving from vertex r to r' has survived a detection on r to possibly being captured on vertex r' . On each outgoing edge from a node (r, k) into a node $(r', k+1)$, if $k = t$ we let the weight become $R_{\text{DMEAN}}(r', k+1) = R_{\text{MEAN}}(r', k+1)$. Otherwise, if $k > t$, we let:

$$R_{\text{DMEAN}}(r', k+1) = R_{\text{MEAN}}(r', k+1) - R_{\text{MEAN}}(r, k)M(r, r')\text{pod}(r'). \quad (9)$$

That is, the wanderer has a probability $R_{\text{MEAN}}(r', k+1)$ of reaching vertex r' on time step $k+1$ and being detected there, from which we subtract his probability $R_{\text{MEAN}}(r, k)M(r, r')\text{pod}(r')$. This last term is the probability the wanderer had of being on vertex r at the preceding time step k , transiting on r' and being captured there. The $R_{\text{DMEAN}}(r', k+1)$ value thus corresponds to the probability that the wanderer reaches r , survives an additional search, reaches r' and gets caught. Given the beginning of a path π_t , an admissible⁴ bound for the OSP is obtained by first solving a longest path problem of which Z_{DAG} is the optimal value and then by summing Z_{DAG} and \cos_{π_t} [7].

Remark 1. Given a graph with n vertices and m edges, DMEAN asks for the construction of a DAG with nT new vertices and mT new edges. Then, if the current time is k , the longest path can be solved in $\mathcal{O}((T-k+1)(n+m))$ steps.

It is convenient, to apply the DMEAN bound in our CP model of the OSP, to order the path variables in a static order of the searcher's positions in time. This is the natural order to solve an OSP since the PATH_t variables are the only decision variables and since the implicit variables representing the probabilities, including the objective variable Z that represents $\cos_{\pi_0}^*$, are entirely determined by the searcher's path. When opening a node for a searcher's position at a time t , we are able to compute, using the chosen bound, a tighter upper bound on the domain of the objective variable. The DMEAN bound is proved admissible [7], that is, it never underestimates the real objective value. Whenever the bound computed on the opened node of the searcher's position at a time t is lower than the best known lower bound on the objective value (e.g., the objective value of the current incumbent solution) that searcher's move is proven unpromising.

3 Bounding the Optimal Search Path Using Search Games on Graphs

The cop and robber game on a graph consists in finding a winning strategy for a cop to catch a robber, considering perfect information for both players (i.e., each player knows her/his position and her/his adversary's position). We focus on a recent variant where the robber is random (or drunk) [17–19]. We show how a valid upper bound on the objective function of the OSP can be derived

⁴ A bound is *admissible* if and only if it never underestimate (resp. overestimate) the value of the objective to maximize (resp. minimize).

by considering, after t steps, the best possible scenario for the searcher that corresponds to allowing her the ability of seeing the wanderer for the remainder of the game.

3.1 A Game of Cop and Drunk Robber

In this game, a cop and a robber move in turn on a graph G , the cop moving first. In contrast with the OSP, the cop has probability 1 of catching the robber when sharing the same vertex, and she *sees* the robber, who, as does the wanderer in the previous game, walks randomly according to a stochastic transition matrix M . The value $M(r, r')$ is the probability that the robber moves to r' provided he is in r at the beginning of the turn. It is positive only if $(r, r') \in E(G)$. The game ends whenever the cop and the robber share the same vertex. The following definition generalizes Nowakowski and Winkler's [11] relational characterization for the classic cop and robber game to the stochastic one. We want to define the relation $w_n^M(r, c)$ as standing for the probability that the cop catches the robber within n moves given their positions (r, c) and the robber's random movement model M . Its definition is based upon the observation that in order to maximize her capture probability, the cop only needs to average her probability of capture at the next turn on the robber's possible transitions.

Definition 1. *Given r, c the respective positions of the robber and the cop in G , M the robber's random walk matrix, we define:*

$$w_0^M(r, c) := 1 \text{ if } r = c; \text{ otherwise, it is } 0;$$

$$w_n^M(r, c) := \begin{cases} 1 & \text{if } c \in N[r], n \geq 1; \\ \max_{c' \in N[c]} \sum_{r' \in N[r]} M(r, r') w_{n-1}^M(r', c') & \text{if } c \notin N[r], n \geq 1. \end{cases} \quad (10)$$

where $N[z] := \{z\} \cup \{v \in V(G) \mid (v, z) \in E(G)\}$ is the closed neighbourhood of z .

The following proposition gives sense to the previous definition.

Proposition 1. *If the cop plays first on G and M governs the robber's random walk, then $w_n^M(r, c)$ is the probability a cop starting on vertex c captures the drunk robber on his start vertex r in n steps or less.*

Proof. By induction. The base case, with $n = 0$, is clear. Suppose the proposition holds for $n - 1 \geq 0$ with $n > 0$ and let us now prove it for n . If $c \in N[r]$, then $w_n^M(r, c) = 1$ and the result follows because the cop indeed catches the robber. If $c \notin N[r]$, then let the cop move to some vertex c' . The position of the robber at the end of the round is r' with probability $M(r, r')$. The probability that the cop catches the robber depends on this last one's next move and on w_{n-1} following the expression $\sum_{r' \in N[r]} M(r, r') w_{n-1}^M(r', c')$. Hence, the best possible move for the cop is $\operatorname{argmax}_{c' \in N[c]} \sum_{r' \in N[r]} M(r, r') w_{n-1}^M(r', c')$. The wanted probability is thus $w_n^M(r, c)$. □

Proposition 1 could provide a trivial upper bound on the probability of finding the wanderer, but a tighter one is presented in the following section. The bound we will derive is inspired from Proposition 1 but is closer to the OSP's objective function.

3.2 Markov Decision Process (MDP) and the OSP Bound

In Section 3.1, we presented a game where the robber is visible, a setting that we will use as a bound on the OSP's objective function. We formulate this game as a MDP from which we derive the Copwin bound and the proof that it is admissible. As a recall, a MDP [20–22] is a stochastic decision process generalizing Markov chains. In a MDP, an agent evolves in a state space S . Every time the agent finds itself in a state $s \in S$, it takes an action a in a set A of possible actions. The system then randomly transits to another valid state s' according to a given transition probability $\mathbb{P}[s' \mid s, a]$ (where \mid denotes a conditional). In order to guide the agent's choice of actions, a reward function $R : S \times A \times S \rightarrow \mathbb{R}$ is defined that assigns to every triple (s, a, s') a real number. The goal of the agent is then to maximize its expected reward within a time bound T . We first formulate the game of cop and robber defined above as an MDP and then deduce a valid OSP bound. The following definition presents the MDP; its solution encodes the optimal strategy for the cop.

Definition 2 (MDP Cop and Drunk Defending Robber). *Let G be a graph, M a stochastic transition matrix on $V(G)$ and T the maximal number of time steps. We define a MDP $\mathcal{M} = (S, A, P, R)$ as follows. A state is a triplet (r, c, t) consisting in both positions of the robber and the cop $r, c \in V(G)$ in addition to the current time $t \in \{1, \dots, T + 1\}$.*

$$S := (V(G) \cup \{\text{jail}\}) \times V(G) \times \{1, 2, \dots, T + 1\}.$$

The set of actions is $V(G)$, the vertices on which the cop moves.

$$A := V(G).$$

From a pair of positions of the cop and the robber at a certain time, once the cop chooses an action another state is chosen randomly with probability P .

$$P\left[(r', c', t') \mid (r, c, t), a\right] := 0 \text{ whenever } a \neq c' \text{ or } c' \notin N[c] \text{ or } t' \neq t + 1$$

otherwise, P is defined as:

$$P\left[(r', c', t + 1) \mid (r, c, t), c'\right] := \begin{cases} 1 & \text{if } r = r' = \text{jail}; \\ \text{pod}(r) & \text{if } r = c', r' = \text{jail}; \\ (1 - \text{pod}(r))M(r, r') & \text{if } r = c', r' \neq \text{jail}; \\ M(r, r') & \text{if } r \notin \{c, c', \text{jail}\}. \end{cases} \quad (11)$$

$$R((r', c', t') \mid (r, c, t), a) := \begin{cases} 1 & \text{if } r' = \text{jail} \neq r, t \leq T; \\ 0 & \text{otherwise.} \end{cases} \quad (12)$$

The game is initialized as follows: the cop chooses her initial position c on a subset X of the graph vertices, and then the initial position r of the robber is picked at random according to the probability distribution poc_{π_0} , which results in an initial state $(r, c, 1)$ for the MDP. A turn starts with a cop move. If the cop transits to the robber state ($r = c'$), then there is probability $\text{pod}(r)$ that she catches the robber, which results in the robber going to jail ($r' = \text{jail}$) and staying there for the rest of the game. The cop then receives a reward of 1. If the catch fails ($r' \neq \text{jail}$, with probability $1 - \text{pod}(r)$) or if the cop did not transit to the robber state ($r \neq c'$), the robber is still free to roam, following M . Note that the state transition probabilities (11) are non-null only when valid moves are considered (time goes up by one and $a = c' \in N[c]$). Note also that, when the robber is in jail, no more reward is given to the cop. In the MDP \mathcal{M} , the cop's goal is to find a strategy, also called policy, to maximize her expected reward, that is, the probability to capture the robber before a total of T steps is reached. A strategy in \mathcal{M} consists in first choosing an initial position (for the cop) and then in following a function $f : S \rightarrow A$ that, given the current state, tells the cop which action to take, that is, which state to transit to. Note that, since the current position of the robber is included in the current state of the MDP, the cop therefore has full information on the system when she is elaborating her strategy.

Because of \mathcal{M} 's Markov property, whenever a strategy f is fixed, one can compute the *value* $u_f(r, c, t)$ of each state (r, c, t) of the MDP, that is, the expected reward the cop can obtain from that state when following the strategy f . The optimal strategy, noted u^* , is therefore the one that gives the highest value on all states (r, c, t) . The cop's optimal strategy consists in moving to the robber's position if possible, and then trying to capture him. If the robber is not positioned on one of the cop's neighbours, the cop moves to the position that maximizes her probability of capture in the remaining time allowed. Similarly to Proposition 1, the value of this optimal strategy is:

$$u^*(r, c, t) = \begin{cases} \max_{c' \in N[c]} \sum_{r' \in N[r]} M(r, r') u^*(r', c', t + 1) & \text{if } r \notin N[c], t < T; \\ 1 - (1 - \text{pod}(r))^{T+1-t} & \text{if } r \in N[c], t \leq T; \\ 0 & \text{if } r \notin N[c], t = T. \end{cases} \quad (13)$$

If $r \notin N[c]$, the cop, who moves first, must choose a next state that will result in the best probability of eventual capture, given the robber's present position, and knowing that the robber's next move is governed by M . If $r \in N[c]$, the cop tries to catch the robber with probability of success $\text{pod}(r)$; if she fails, the robber will transit to one of his neighbours, and hence the cop can keep on trying to catch the robber until success or until the maximal time has been reached. It is important to note here that the robber is completely visible and the game is not played simultaneously, hence why the cop can follow the robber. Equation (13) is analogous to (10) with time steps reversed and $\text{pod}(r) = 1$ for all vertices. The formula follows from the fact that at the beginning of time step t , the cop has $T + 1 - t$ remaining attempts.

Since the optimal probability of capture in the OSP problem is always lower than the optimal probability of capture in the cop and robber game, we have the following proposition:

Proposition 2. *The probability $\text{cos}_{\pi_0}^*$ of finding the wanderer is always at most that of catching the robber:*

$$\text{cos}_{\pi_0}^* \leq \max_{c \in X} \sum_{r \in V(G)} \text{poc}_{\pi_0}(r) u^*(r, c, 1),$$

where X is the subset of $V(G)$ of possible initial positions of the cop.

Proof. Clearly, $\text{cos}_{\pi_0}^*$ is bounded by the optimal probability of capture of the cop and robber game. In the MDP, the optimal probability of capture is obtained if the cop's first choice maximizes his probability of capture considering that at that moment the robber is not yet positioned on the graph but will be according to the probability distribution poc_{π_0} . \square

Unfortunately, Proposition 2's bound is of no use in a branch-and-bound attempt for solving the OSP problem, because a bound for each π_t and each $t = 0, \dots, T$ is needed. The next proposition generalizes it appropriately.

Proposition 3. *Let $\pi_t = v_1, v_2, \dots, v_t$. Then*

$$\text{cos}_{\pi_t}^* \leq \text{cos}_{\pi_t} + \max_{c' \in N[v_t]} \sum_{r' \in V(G)} \text{poc}_{\pi_t}(r') u^*(r', c', t + 1), \quad (14)$$

where $N[v_0]$ is interpreted as the possible initial positions of the cop.

Proof. As in the preceding proof, $\text{cos}_{\pi_t}^*$ is bounded by the optimal reward obtained when first playing the OSP game along π_t and then (if the wanderer is not yet detected) switching to the cop and robber game: this is done by making the wanderer (robber) visible to the searcher (cop). When starting this second phase (at the $t + 1$ step), the cop must choose his next position in order to maximize the probability of capture; at this moment, the robber is not yet visible but his next position is governed by poc_{π_t} . If the cop chooses c' , his probability of capture will be $\sum_{r' \in V(G)} \text{poc}_{\pi_t}(r') u^*(r', c', t + 1)$ and the result follows. \square

Remark 2. An important aspect of Copwin is its ability to be mostly precomputed. For any vertices $r, c \in V(G)$ and time $t \in \{1, \dots, T + 1\}$, the values $u^*(r, c, t)$ are computed recursively (starting with $t = T$) and then stored. Then, when the bound is called the next value of Equation (14) depends on the neighbours of the searcher's position and the vertices of the graph, requiring $\mathcal{O}((\Delta + 1)n)$ extra operations on an n vertex graph of maximal degree Δ . Since DMEAN's complexity is $\mathcal{O}((T - k + 1)(n + m))$, Copwin is faster on most time steps and on many instances.

We note that Pralat and Kehagias [18] also formulated the game of cop and visible drunk robber as an MDP, but instead of using this framework to compute the probability of capture of the robber it was formulated to obtain the expected number of time steps before capture.

Applying the Copwin bound in CP requires filtering the upper bound of the objective function variable Z . The solver can be configured to branch in a static order of the $PATH_t$ variable, which is, as discussed in Section 2.2, the natural order of the decision variables to solve an OSP. We proved that our bound is admissible in that it never underestimates the real objective value. In the next section, we use the Copwin bound jointly with a CP model of the OSP and compare its performance to that of the DMEAN bound from literature.

4 Experiments

All experiments were run on the supercomputer Colosse from Université Laval using the Choco 2.1.5 solver [23] along with the Java Universal Network/Graph (JUNG) 2.0.1 framework [24]. The total allowed CPU time is 5 minutes (or 5 million backtracks) per run. We used a total of six graphs representing different search environments. In the grid G^+ , the nodes are connected to their North, South, East, and West neighbour. The grid G^* has the same connections than G^+ but also adds connectivity to the North-East, North-West, South-East, and South-West neighbours. The last grid is a typical hexagonal grid we call G^H with the searcher moving on its edges. Grid-like accessibility constraints are common in search theory problems in that they model aerial or marine search operations where the searcher has few accessibility constraints but her own physical constraints. This justifies the presence of grids in our OSP benchmarks. We also added a fourth grid with supplementary constraints on accessibility. This last grid G^V is a hexagonal grid with a supplementary node in the center of each hexagon. We then had removed edges randomly to simulate supplementary accessibility constraints. The last two graphs, G^L and G^T , are real underground environments. The G^L graph is the graph of the Université Laval's tunnels. The G^T graph is the graph of the London Underground subway (also known as the Tube). Both neighbours of an intermediate station, i.e., a node of degree 2, were connected together while the node was removed. This practice results in a more interesting graph environment for searches.

For each instance, we selected a $p \in \{0.3, 0.6, 0.9\}$ and put $\text{pod}(r) = p$ for all vertex r . For the wanderer's Markovian transition matrix, we pick $\rho \in \{0.3, 0.6, 0.9\}$ and considered that at any position v , the wanderer has probability ρ of staying in v and probability $1 - \frac{\rho}{\text{deg}(v)}$ to move to one of his $\text{deg}(v)$ neighbours. We uniformly distributed the remaining probability mass $1 - \rho$ on the neighbouring vertices. Finally, we chose $T \in \{5, 7, 9, 11, 13, 15, 17, 19, 21, 23, 25\}$. This leads to a total of 594 instances.

We conducted the experiment in three parts. In the first two parts, we compare our novel bound to the DMEAN bound we reviewed in Section 2.2. First, we implemented the two bounds as separate constraints on the upper bound of

the objective value. Both bounds are evaluated on the same problem instances. This enables us to assess the bounds performance with respect to the objective value of the best incumbent solution found by the solver and to the time required to attain it.

In the second part, we implemented both bounds as a single constraint on the upper bound of the objective value. Both bounds were computed simultaneously by the constraint. The minimal value of both bounds was used to update the upper bound of the domain of the Z variable representing the objective if required. That is, whenever that minimal value is lower than the upper bound on the domain of the Z variable. If the domain of Z is emptied, then one of the bounds produced a value that is lower than the best known lower bound on the objective thus leading to a backtrack. For each bounding technique (DMEAN and Copwin), we count how many times, during the search, the bounding of the objective value causes a backtrack. We gave a single point to a bound that leads to a backtrack that way. We note that on occasions both bounds could receive a point. In all cases, the solver branches on the $PATH_t$ decision variables in their natural order, i.e., in ascending order of the time steps. We thus were able to record the exact time step at which each backtrack occurred. Such an experiment enables one to empirically assess which bound is more efficient at pruning the search tree no matter without considering any time constraints.

We show, in the third part, how the bounds perform when paired with the total detection (TD) heuristic used as a value-selection heuristic [10]. The TD heuristic is on many account similar to the bound derived in this paper which gives its theoretical justification. However, rather than using the cop and robber games theory to bound the objective value, TD is a heuristic that assigns values to the variables $PATH_t$. Even in this context, the Copwin bound performs and stays competitive.

4.1 Results and Discussion

Figure 1 shows the best incumbent objective value (COS) for all OSP problem instances. Hence, COS is the estimation of $\cos_{\pi_0}^*$ at the end of the solving process. Each dot represents the solution to a single OSP instance. The figure compares the objective value obtained when the solver uses the DMEAN bound against the objective value obtained when the solver uses the Copwin bound for all instances. The closer a dot is to the diagonal line, the closer it is to being the same value for both bounds which is viewed as both bounds having had the same effectiveness on this instance. It does appear the Copwin bound tends to lead the solver to better objective values in the total allowed time. Thus, Copwin is not only faster than DMEAN, it is sharp enough to help the solver to exploit more promising branches.

Figure 2 shows the number of times each bound empties the domain of the objective value variable in the search tree for each type of graph. We chose to plot the score dependant on the time step (or level in the search tree) to better analyse where branches were being pruned. It appears that on average

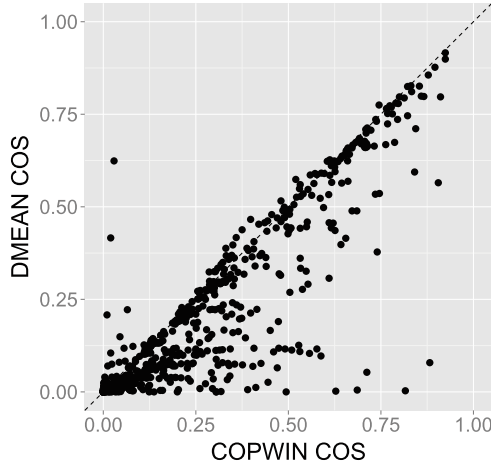


Fig. 1. Two by two comparison of DMEAN best incumbent COS value against Copwin best incumbent COS value for all 594 OSP instances; the incumbent solution of each instance is represented by a dot on the graph.

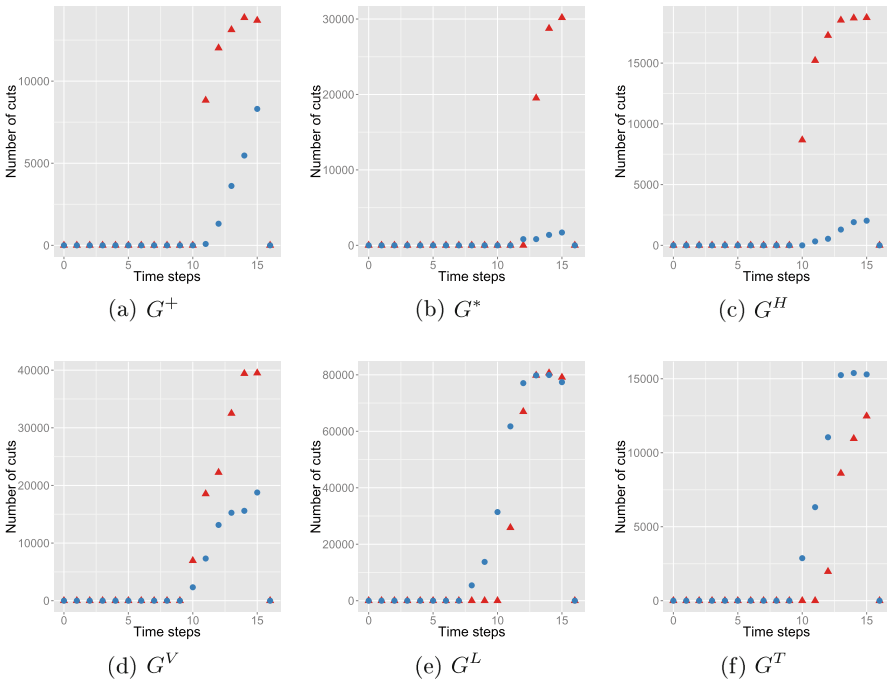


Fig. 2. Total number of cuts in the solver’s search tree for both bounds against searcher’s path length with $T = 17$; results for each instance are aggregated on motion models ρ and on detection models $\text{pod}(r)$. \blacktriangle refers to the use of Copwin whereas \bullet refers to DMEAN.

Copwin empties the objective value variable's domain more often than DMEAN independently of the graph, even though on some instances such as G^L both lead to a great number of prunings. This is a strong indication that Copwin performs well on all accounts on most instances: for a fixed resolution time, it leads to better objective values while pruning more branches than the DMEAN bound.

We present on Figure 3, the objective value of the best so far solution against computational time for both bounding strategies. We present these comparisons on three representative instances: one in which Copwin is better than DMEAN,

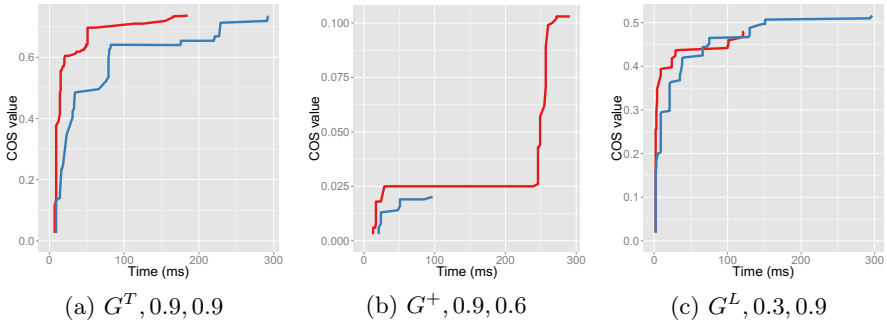


Fig. 3. The COS value achieved and the time it was found for each bound. Red lines are associated with Copwin and blue ones with DMEAN. Each title refers to the graph type, the instance $\text{pod}(\tau)$, the instance ρ and maximal time steps of 17.

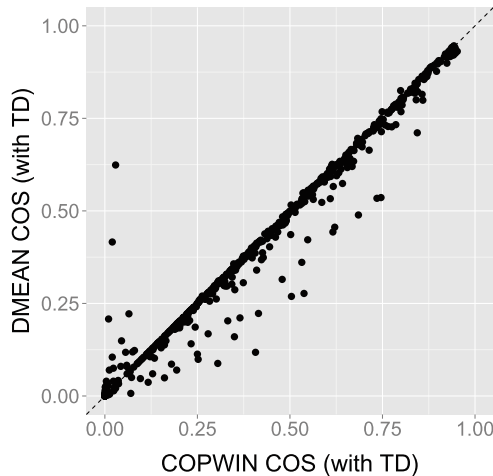


Fig. 4. Comparison of the best incumbent COS value obtained with the TD heuristic for the Copwin bound against the best COS value obtained with the TD heuristic for the DMEAN bound for all OSP problem instances; the incumbent solution of each instance is represented by a dot on the graph.

Table 1. Results on OSP problem instances; bold values are better.

pod(r)	ρ	Objective (COS)		Time [†] (s)		Backtracks	
		DMEAN	Copwin	DMEAN	Copwin	DMEAN	Copwin
<i>G^T with T = 17</i>							
0.3	0.3	0.111	0.113	298	197	3578	2791
	0.6	0.139	0.129	200	144	2045	2157
	0.9	0.362	0.354	275	208	2973	3304
0.9	0.3	0.252	0.258	131	287	1400	4613
	0.6	0.318	0.325	259	236	3082	3544
	0.9	0.736	0.736	292	185	3543	2706
<i>G⁺ with T = 17</i>							
0.3	0.3	0.011	0.014	96	37	283	118
	0.6	0.007	0.036	88	281	243	1307
	0.9	0.001	0.022	90	111	276	497
0.9	0.3	0.030	0.077	95	296	290	1394
	0.6	0.020	0.103	95	272	283	1229
	0.9	0.001	0.047	26	262	19	1174
<i>G^V with T = 17</i>							
0.3	0.3	0.035	0.079	154	296	2510	7240
	0.6	0.038	0.105	146	206	2510	4922
	0.9	0.036	0.229	158	196	2510	4898
0.9	0.3	0.099	0.215	159	240	2510	5539
	0.6	0.112	0.304	155	264	2510	6040
	0.9	0.109	0.531	161	219	2510	4996
<i>G^L with T = 17</i>							
0.3	0.3	0.344	0.318	273	145	9224	7749
	0.6	0.417	0.359	287	139	9666	7476
	0.9	0.516	0.480	297	122	9298	6462
0.9	0.3	0.638	0.651	278	119	11304	6249
	0.6	0.713	0.713	289	146	11603	7890
	0.9	0.811	0.833	249	162	9489	8727
<i>G[*] with T = 17</i>							
0.3	0.3	0.069	0.072	270	286	1264	2264
	0.6	0.107	0.117	273	183	1261	1482
	0.9	0.235	0.324	276	281	1254	2461
0.9	0.3	0.192	0.205	274	252	1264	2077
	0.6	0.304	0.333	264	219	1261	1842
	0.9	0.671	0.711	259	231	1253	1925
<i>G^H with T = 17</i>							
0.3	0.3	0.023	0.087	258	241	522	618
	0.6	0.015	0.122	255	277	519	742
	0.9	0.001	0.318	250	233	514	623
0.9	0.3	0.064	0.227	274	258	522	686
	0.6	0.043	0.342	260	286	520	719
	0.9	0.003	0.816	248	280	516	680

† The time to last incumbent solution.

one in which it is worse and another where both bounds lead to a similar performance. These graphs confirm the intuition that, most of the time, Copwin finds good solutions more quickly than DMEAN.

Figure 4 summarizes further experiments where Copwin and DMEAN were paired with the TD heuristic. Even though both bounds lead the solver to greater objective values, we observe that Copwin's objective values are on average slightly better. Thus, the behavior observed on Figure 1 is preserved when a good heuristic is added.

As an addition to the graphs of Figures 1 to 4, we include Table 1 which contains the results of the experiments of the Copwin versus the DMEAN bound for all graphs with $T = 17$.

5 Conclusion

We tackled the OSP problem from search theory using constraint programming. As a first contribution, we provided the first implementation of the DMEAN bound from the search theory literature in CP. As our main contribution, we developed the Copwin bound, a novel and competitive bound based on MDPs. This bound is derived from a simple and elegant relaxation of a search problem into a graph theory pursuit game. Involving a polynomial computational cost, the Copwin bound leads to an improved solver performance on the vast majority of OSP problem instances in our benchmark. Although we used the bound on a CP model of the OSP, it remains a general technique applicable to other OSP algorithms.

Acknowledgments. This research is funded by the Natural Sciences and Engineering Research Council of Canada (NSERC) and the Fonds de recherche du Québec - Nature et technologies (FRQ-NT).

Computations were made on the supercomputer Colosse from Université Laval, managed by Calcul Québec and Compute Canada. The operation of this supercomputer is funded by the Canada Foundation for Innovation (CFI), NanoQubec, RMGA and the Fonds de recherche du Québec - Nature et technologies (FRQ-NT).

References

1. Trummel, K., Weisinger, J.: The complexity of the optimal searcher path problem. *Operations Research* **34**(2), 324–327 (1986)
2. Stewart, T.: Search for a moving target when the searcher motion is restricted. *Computers and Operations Research* **6**(3), 129–140 (1979)
3. Stone, L.: *Theory of Optimal Search*. Academic Press, New York (2004)
4. Netsch, R.: The USCG search and rescue optimal planning system (SAROPS) via the commercial/joint mapping tool kit (c/jmtk). In: *Proceedings of the 24th Annual ESRI User Conference*, vol. 9, August 2004
5. Abi-Zeid, I., Frost, J.: A decision support system for canadian search and rescue operations. *European Journal of Operational Research* **162**(3), 636–653 (2005)
6. Washburn, A.R.: Branch and bound methods for a search problem. *Naval Research Logistics* **45**(3), 243–257 (1998)
7. Lau, H., Huang, S., Dissanayake, G.: Discounted MEAN bound for the optimal searcher path problem with non-uniform travel times. *European Journal of Operational Research* **190**(2), 383–397 (2008)

8. Morin, M., Lamontagne, L., Abi-Zeid, I., Lang, P., Maupin, P.: The optimal searcher path problem with a visibility criterion in discrete time and space. In: Proceedings of the 12th International Conference on Information Fusion, pp. 2217–2224 (2009)
9. Sato, H., Royset, J.O.: Path optimization for the resource-constrained searcher. *Naval Research Logistics*, 422–440 (2010)
10. Morin, M., Papillon, A.-P., Abi-Zeid, I., Laviolette, F., Quimper, C.-G.: Constraint programming for path planning with uncertainty. In: Milano, M. (ed.) CP 2012. LNCS, vol. 7514, pp. 988–1003. Springer, Heidelberg (2012)
11. Nowakowski, R., Winkler, P.: Vertex-to-vertex pursuit in a graph. *Discrete Mathematics* **43**(2), 235–239 (1983)
12. Quilliot, A.: Problème de jeux, de point fixe, de connectivité et de représentation sur des graphes, des ensembles ordonnés et des hypergraphes. Ph.D thesis, Université de Paris VI (1983)
13. Bonato, A., Nowakowski, R.: The game of cops and robbers on graphs. American Mathematical Soc. (2011)
14. Fomin, F.V., Thilikos, D.M.: An annotated bibliography on guaranteed graph searching. *Theoretical Computer Science* **399**(3), 236–245 (2008)
15. Beldiceanu, N., Demasse, S.: Global constraint catalog (2014). <http://sofdem.github.io/gccat/> (accessed 2015–04)
16. Martins, G.H.: A new branch-and-bound procedure for computing optimal search paths. Master’s thesis, Naval Postgraduate School (1993)
17. Kehagias, A., Mitsche, D., Prałat, P.: Cops and invisible robbers: The cost of drunkenness. *Theoretical Computer Science* **481**, 100–120 (2013)
18. Kehagias, A., Prałat, P.: Some remarks on cops and drunk robbers. *Theoretical Computer Science* **463**, 133–147 (2012)
19. Komarov, N., Winkler, P.: Capturing the Drunk Robber on a Graph. arXiv preprint [arXiv:1305.4559](https://arxiv.org/abs/1305.4559) (2013)
20. Bäuerle, N., Rieder, U.: Markov Decision Processes with Applications to Finance. Springer (2011)
21. Puterman, M.L.: Markov decision processes: discrete stochastic dynamic programming, vol. 414. John Wiley & Sons (2009)
22. Barto, A.G., Sutton, R.S.: Reinforcement learning: An introduction. MIT press (1998)
23. Laburthe, F., Jussien, N.: Choco solver documentation. École de Mines de Nantes (2012)
24. O’Madadhain, J., Fisher, D., Nelson, T., White, S., Boey, Y.B.: The JUNG (Java universal network/graph) framework (2010)