12th International Conference on Information Fusion
Seattle, WA, USA, July 6-9, 2009

The Optimal Searcher Path Problem with a Visibility Criterion in Discrete Time and Space

Michael Morin
Department of Computer Science and Software Engineering
Université Laval
Québec, QC, Canada.
Michael.Morin.3@ulaval.ca

Luc Lamontagne
Department of Computer Science and Software Engineering
Université Laval
Québec, QC, Canada.
Luc.Lamontagne@ift.ulaval.ca

Irène Abi-Zeid, Pascal Lang
Department of Operations and Decision Systems
Université Laval
Québec, QC, Canada.
Irene.Abi-Zeid, Pascal.Lang@osd.ulaval.ca

Patrick Maupin
Defence Research and Development Canada
Valcartier, QC, Canada.
Patrick.Maupin@drdc-rddc.gc.ca

Abstract – *In this paper, the problem of path planning for a ground search unit looking for an object of unknown location is considered. As in the classical optimal searcher path problem, the probability of finding the search object is the main criterion of optimality and the search unit is constrained by the environment topology that influences its choices for a navigable path as well as its detection capabilities. This paper proposes an extension to the classical optimal searcher path problem in discrete time and space by integrating inter-region visibility as an additional criterion. This new formulation allows a refinement in the discretization of the space in which a ground search unit evolves. A general mixed-integer programming model is proposed, and experimental results with a moving object in grid environments are discussed.*

Keywords: Optimal searcher path problem, visibility, path planning, search theory, mixed integer programming.

1 Introduction

The problem of searching for an object of unknown location is pervasive in everyday life and in many real world operations such as surveillance, oil exploration, etc. In particular, the importance of planning efficiently a search mission is obvious in search and rescue (SAR) operations where a shorter mean time to detection may result in a higher probability of finding survivors. The optimal searcher path (OSP) problem and its variations, as applied in classical search theory and in the robotics literature, provide a framework for developing optimal search plans in such critical situations.

We address the OSP problem in discrete time and space with indivisible search effort, where a search plan is determined by the path of the search unit in the environment. In other words, the path of the search unit determines where the search effort is allocated. In search theory, *search effort* is a general notion to represent the resources available for searching. In the indivisible case, the search effort represents discrete resources like discrete time periods or the number of search units allocated to a given area. In the infinitely divisible case, search effort represents continuous resources such as time, trajectory length or fuel used over a search area. In the classical OSP problem with indivisible search effort context, a search plan P is a finite ordered sequence of regions that is considered optimal if it maximizes the probability of finding the object at the end of the search operation, *i.e.*, the cumulative overall probability of success (COS). The probability of success in a given region r at a given time step t ($pos_t(r)$) is the product of the probability that the search object is contained in this region at this time step ($poc_t(r)$) and of the conditional probability of detecting the search object that is present in this region at this time step ($pod_t(r)$). The pod function varies according to the detection capabilities of the search unit in its surrounding environment and to the search effort allocated. Also, it may vary according to the search object characteristics and according to time. [8]

The main characteristic of an OSP problem is the constraint of region adjacency: in one time step, a search unit can only travel between adjacent regions. Without loss of generality, we can consider that a search unit evolves on a *navigability* graph where nodes represent regions and arcs represent the adjacency of two regions. One qualitative characteristic of the problems that comes from search theory is their large discretization scale, *i.e.*, each region represents a large part of the original continuous environment. In this large scale outlook, a search unit searching a given region is assumed to follow a specific search pattern that influences the *pod* function. [2]

This paper extends the classical OSP problem by taking into account a visibility criterion. It introduces the optimal searcher path problem with *visibility* (OSPV) in which we consider a finer discretization scale that formalizes, as a *visibility* graph, the inter-region visibility found in environments such as buildings (with rooms, footbridges, windows…), urban areas (with roads, buildings, bridges…) and natural sites (with rivers, trails, cliffs…). The main idea behind this concept is that a navigable path between two regions may be long, treacherous or obstructed and that it might be faster or safer to simply search from a distance, as opposed to the classical OSP where a search unit searches the region in which it is located. As in OSPs, the probability of detection is a measure of search effectiveness over a given region at a given time step. However, the search unit's position is added as a parameter of the *pod* function to consider inter-region visibility. Thus, a search plan in an OSPV is determined by the path of the search unit and by the object of its focus, *i.e.*, the region it searches at a given time step. We assume that the search unit is constrained to search only one region per time step; thus, the search effort is indivisible. Nonetheless, our model can be extended to divisible search efforts. Since this paper is concerned with modelling issues in the OSPV context, its emphasis is on the introduction of new *visibility* features providing additional realism to conventional OSP formulations.

The rest of the paper is organized as follows. The second section briefly reviews the OSP problem's history. The third section presents a general problem statement. The fourth section proposes a more specific mixed-integer programming formulation. The fifth section lays out the experimental plan used to test the performances of the mixed integer program (MIP) model. The experimental results are presented and discussed in sections 6 and 7. The paper concludes with potentially interesting avenues for further research.

2 OSP history in brief

Two mains characteristics of OSP-like problems are the constraints on search units' movements and the unknown search object's location. In the case of a one-sided search for a moving object of unknown location, an important assumption is that the object's motion is independent of the search units' actions. In other words, the search object does not take action to meet or to evade its searcher.

The OSP problem is much more general than what we present in this section, where we limit our brief literature review to single search unit problems in discrete time and space. A proof of NP-Completeness for the OSP problem formulation in discrete time and space with a stationary search object can be found in [10].

Stewart [9] considered a variation with a moving object and infinitely divisible (continuous) search effort and solved it to optimality using a network flow formulation under the assumption of an exponential detection (*pod*) function. In the case of indivisible search effort, Stewart [9] considered a depth-first Branch and Bound algorithm with a bound without guarantee of optimality: a relaxed problem in which regions adjacency constraints were replaced by accessibility constraints solved by Brown's algorithm [1]. For the arbitrarily divisible search effort case (*e.g.*, where search effort is an integer number), Stewart [9] suggested using a sequential allocation or relaxing the indivisibility constraint. Eagle [3] considered the problem with indivisible search effort with a Markovian search object motion model and proposed a dynamic programming approach. Eagle and Yee [4] presented an optimal bound for the Branch and Bound algorithm of Stewart. Assuming a Markovian search object motion model and an exponential detection function, the approach gives an optimal bound by relaxing search effort indivisibility constraints over a specific set of regions that maintains path constraints. In [11], Washburn reviewed the Branch and Bound procedures and heuristics for OSPs. Among the possible OSP variations, Lau *et al.* [6] introduced the OSP with non-uniform travel times between connected regions (OSPT) and proposed the DMEAN bound for OSP or OSPT problems which was derived from the MEAN bound proposed in [7].

3 The OSP problem with a visibility criterion

This section introduces a general model for the single search unit OSPV problem in discrete time and space with indivisible search effort.

3.1 The OSPV with indivisible search effort

Consider a search unit that evolves in a discretized environment defined by a set R of N regions numbered from 1 to N (1). The time allocated to the search operation is discretized and defined by a set I of T time steps numbered from 1 to T (2). The search unit's position is known for each time step $t \in I$ and is denoted by the region where it stands. Its effort allocation for a time step $t \in I$ is denoted by the region it looks at.

$$R = \{1,...,N\} \tag{1}$$

$$I = \{1,...,T\} \tag{2}$$

The search object's unknown position is characterized by a prior probability distribution poc_0 such that for each region $r \in R$, $poc_0(r)$ is the initial *poc* of region r. Assuming that the object is located somewhere in the environment, the sum of the initial *poc* over all the regions must be equal to 1.0. If the object can be located outside the environment, this sum could be less than 1.0. (3)

$$\sum_{r\in R} poc_0(r) \le 1.0 \tag{3}$$

At each time step t, the object moves according to its motion model where $d(s,r)$ is the probability of moving from a region s to a region r in one time step. As in one-sided search OSP-like problems, a search object's move does not depend on the search unit's previous actions. Assuming that the object cannot move outside the environment, the sum of all the probabilities of movement for any given region s must be equal to 1.0. If the object can move outside the environment from a given region s, the sum can be less than 1.0. (4)

$$\forall s \in R : \sum_{r \in R} d(s,r) \leq 1.0 \tag{4}$$

In the case of a stationary object, the probability of a move from one region to the same region is 1.0.

In most OSP-like problems, the search can start in any region. We assume that the search unit has an initial position $y_0 \in R$. This is equivalent to saying that the search begins as soon as the search unit enters the environment. For each time step t, the search unit's position is denoted $y_t \in R$. The search unit's movements are constrained by the topology of the environment via a known navigability graph. The latter induces an *adjacency* map $A : R \rightarrow 2^R$, where $A(r)$ is the set of regions that are directly accessible from region r. The search unit's motion must thus obey:

$$y_t \in A(y_{t-1}) \tag{5}$$

At each time step t, the search unit's effort allocation is noted $e_t \in R$. The search unit's observations are again constrained by the environment topology via a given visibility graph, which defines a *visibility* map $V : R \rightarrow 2^R$, where $V(r)$ is the set of regions that are visible from region r. The search unit's observation effort must therefore obey:

$$e_t \in V(y_t) \tag{6}$$

Considering the described problem's environment, we need to define the *poc* distribution, the *pod* function and the *pos* in terms of the OSPV problem:

(i) $poc_t(r)$ is the probability that the search object is in region r at time step t given that it has not been found in earlier search plan's stages. There is no false target and the data are assumed accurate.

(ii) $pod_t(s,r)$ is the conditional probability that the search unit catches a glimpse of the search object (given that it is in region r) when it searches region r from region s at time step t. The *pod* function is assumed accurate and is known *a priori*.

(iii) $pos_t(r)$ is the probability of finding the search object in region r at time step t given that it has not been found in earlier search plan's stages.

As in the classical OSP problem, the primary criterion of optimality is the cumulative overall probability of success (*COS*) which depends on the probability of finding the search object at each time step t in each region r ($pos_t(r)$). (7) As a function of the environment, the search unit's *pod* function varies according to where the search unit stands and to where it directs its focus. Since we consider one unit of indivisible search effort, the *pod* function does not vary with search effort. Here, the *pod* function is a form of *glimpse function* as described in [6].

$$\forall r \in R, t \in I : \quad pos_t(r) = \begin{cases} poc_t(r) \times pod_t(y_t, r) & \text{if } r = e_t \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

At each time step $t \in I$, the *poc* distribution is updated according to the search unit actions at time step t - 1 and to the motion model of the object. (8) Note that $pos_0(r) = 0$ for all regions r since the search operation begins in t = 1.

$$\forall r \in R, t \in I : \quad poc_t(r) = \sum_{s' \in R} d(s', r) \left[poc_{t-1}(s') - pos_{t-1}(s') \right] \tag{8}$$

Finally, the search unit visit a navigable sequence of regions and search in a sequence of visible regions which define a search plan *P*. *P* is optimal if it maximize the cumulative overall probability of success *COS* as defined by the sum of the $pos_t(r)$ over all the regions r and all the time steps t.

4 A MIP model for the OSPV problem

This section describes a possible mixed-integer programming formulation to solve the OSPV problem. For this section, all variables are written in *italic* font and all data known *a priori* are written in standard font. Since the problem formulation deals with indivisible search effort, the *pod* function (*i.e.*, the *glimpse* function) values can be computed before solving the MIP. Thus, the available information are the initial probability of containment ($\mathrm{poc}_0(.)$), the probability of detection for each pair of regions (pod(.,.)), the motion model of the search object (d(.,.)), the initial search unit's position (y_0) and finally the visibility graph and the navigability graph defined by two adjacency matrices V and A (respectively). In addition, the set of regions (R) and the set of time steps (I) are known. The objective function is defined by equation (9). Note that at any time period t, $pos_t(\mathrm{R}) = \sum_{r \in \mathrm{R}} pos_t(r)$ since successes across regions are mutually exclusive. Furthermore, note from (7) and (8) that $pos_t(\mathrm{R})$ is the probability of finding the target at time step t (given that it has not been found before time t). Hence the cumulative overall probability of success over T periods is $\sum_{t \in \mathrm{I}} pos_t(\mathrm{R})$.

$$\max \sum_{t=1}^{\mathrm{T}} \sum_{r=1}^{\mathrm{N}} pos_t(r) \tag{9}$$

subject to constraints (10) to (23). Constraint (10), where M is a sufficiently large integer, specifies that a region can be searched at a given time step only if it is visible from the search unit's current position and ensures that that the *pos* is computed according to equation (7). Constraint (11) ensures that the *pos* of any given region is less than or equal to the search effort allocation in that region (since it can be at most 1). Constraint (12) specifies that the search operation does not begin in the initialization step. Constraint (13) ensures that the *poc* for each region and for each time step is computed according to equation (8). Constraints (14) and (16) are navigability constraints. The first one ensures that for each time step, the position of the search unit is accessible from its previous position. The second one ensures that for each time step, the search unit is located in only one region. Constraints (15) and (17) are visibility constraints and ensure that for each time step, the search unit focuses on one and only one region (17) that is visible from its current position (15). Constraints (18) and (19) specify the search unit's initial position for each region ($\mathrm{y}_0(.)$) and the initial probability of containment ($\mathrm{poc}_0(.)$). Finally, constraints (20) and (21) specify the bounds for probability variables and constraints (22) and (23) for binary variables.

$$pos_t(r) - \mathrm{pod}_t(s,r)\,poc_t(r) \le \mathrm{M}(1 - y_t(s)), \quad \forall s, r \in \mathrm{R}, \forall t \in \mathrm{I} \tag{10}$$

$$pos_t(r) \le e_t(r), \forall r \in \mathrm{R}, \forall t \in \mathrm{I} \tag{11}$$

$$pos_0(r) = 0, \forall r \in \mathrm{R} \tag{12}$$

$$poc_t(r) - \sum_{s'=1}^{\mathrm{N}} \mathrm{d}(s',r)\,poc_{t-1}(s') + \sum_{s'=1}^{\mathrm{N}} \mathrm{d}(s',r)\,pos_{t-1}(s') = 0, \quad \forall r \in \mathrm{R}, \forall t \in \mathrm{I} \tag{13}$$

$$y_t(r) - \sum_{s=1}^{\mathrm{N}} \mathrm{a}(s,r)\,y_{t-1}(s) \le 0, \quad \forall r \in \mathrm{R}, \forall t \in \mathrm{I} \tag{14}$$

$$e_t(r) - \sum_{s=1}^{\mathrm{N}} \mathrm{v}(s,r)\,y_t(s) \le 0, \quad \forall r \in \mathrm{R}, \forall t \in \mathrm{I} \tag{15}$$

$$\sum_{s=1}^{\mathrm{N}} y_t(s) = 1, \forall t \in \mathrm{I} \tag{16}$$

$$\sum_{r=1}^{N} \mathrm{e}_t(r) = 1, \forall t \in \mathrm{I} \tag{17}$$

$$y_0(s) = \mathrm{y}_0(s), \forall s \in \mathrm{R} \tag{18}$$

$$poc_0(r) = \mathrm{poc}_0(r), \forall r \in \mathrm{R} \tag{19}$$

$$0 \le poc_t(r) \le 1, \forall r \in \mathrm{R}, \forall t \in \mathrm{I} \tag{20}$$

$$0 \le pos_t(r) \le 1, \forall r \in \mathrm{R}, \forall t \in \mathrm{I} \tag{21}$$

$$y_t(s) \in \{0,1\}, \forall s \in \mathrm{R}, \forall t \in \mathrm{I} \tag{22}$$

$$e_t(r) \in \{0,1\}, \forall r \in \mathrm{R}, \forall t \in \mathrm{I} \tag{23}$$

This model is a mixed integer linear program. General purpose algorithms to solve this class of problems to global optimality are in the public domain (*e.g.*, in [5]). Several commercial MILP software based on these algorithms are available, one of which is Ilog Cplex. The unavoidable cost of guaranteeing global optimality is complexity. Overcoming complexity requires the development of ad-hoc heuristics forfeiting global optimality. This line of research is not pursued here since our main concern is with modelling rather than with algorithmic development. As expressed later, further research avenues include addressing complexity management and performance issues.

5 Experimentation

The main purpose of the experimentation is to evaluate the feasibility of the MIP formulation of the new OSPV problem with Ilog Cplex 11.2 and Concert Technology 2.7. All the tests were run on the following hardware: an Intel Core2 Quad Q6600 (2.4 GHz) processor with 3 Go of RAM. The operating system used for the experimentation was Microsoft Windows XP Professional with SP3. All OSPV problem instances were generated using a framework designed especially for the experimentation. The experimentation framework was implemented using the C++ programming language and the sources were compiled with the Microsoft Visual Studio 2005 compiler.

To stay close to the classical OSP problem instances presented in literature, all the tests were conducted under the following assumptions:

(i) All the discretized environments used for the tests are rectangular grids of equal area cells where each cell defines a region.

(ii) The search object is moving. Moreover, the object's motion model probability distribution is uniform on rows for all accessible regions and the sum of all rows is equal to 1.0, *i.e.*, for a given region the probability of movement is uniformly distributed over the accessible regions.

(iii) If the search unit searches the region where it is located, it obtains a *pod* of 0.8 whereas if it searches a visible region different from where it is located, it obtains a *pod* of 0.6.

Furthermore, the initial position of the search unit and the initial *poc* distribution were randomly generated and the sum of the initial *poc* over all the regions is equal to 1.0.

The experimentation is structured in two parts. The first part is over a gridded environment where the search unit can move and look from its current position to any adjacent region (including its current position). The second part is over a gridded environment where the search unit is constrained to stay in its current position or to move to the North, South, West or East from its current position, whereas it can search any adjacent region (including its current position). These types of constraints are summarized in Table 1.

For each of the two parts of the experiment, four environments with different characteristics were randomly generated using the experimentation framework. Each experimentation group is the instance of one of the four environments coupled with a navigability and visibility constraint type. The characteristics of each test group are given in Table 2 where W and H are the number of rows and the number of columns of the gridded environment and where T is the number of time steps allowed during the search operation.

Table 1. The navigability and visibility constraint types.

Constraint type	Description of the constraint type for a given search unit
Star	Move (or look) from its current position to the same region or to any adjacent region.
Plus	Move (or look) from its current position to the same region or to an adjacent region to its north, south, west or east.

Table 2. The test environments.

Group	*W*	*H*	*T*	Navigability constraints	Visibility constraints
1	4	3	6	Star	Star
2	4	3	12	Star	Star
3	4	7	14	Star	Star
4	4	7	28	Star	Star
5	4	3	6	Plus	Star
6	4	3	12	Plus	Star
7	4	7	14	Plus	Star
8	4	7	28	Plus	Star

To evaluate the performance obtained with the MIP formulation and Ilog Cplex, specific solver configurations were tested for each test group. For the sake of comparison, the first configuration is the default configuration of the Cplex solver. The second configuration involves a major scale up of all the probabilities of the model. Since each probability is multiplied by 100 000 and all constraints are consequently modified, this configuration can help the solver to avoid rounding or decimal error due to the computation of small probability values for *poc* and *pos* variables and constraints. The third configuration emphasizes feasibility. This configuration may increase the overall performance by finding a feasible solution sooner in the solving process. The fourth configuration is the counterpart of feasibility, where the solver was tuned to put more emphasis on optimality in order to rapidly diminish the duality gap, defined as the difference between the primal and dual objectives. The duality gap gives an idea of how far the incumbent solution is from an optimal solution at the end of the allowed solving time. Sometimes, the solver is able to find an optimal solution rapidly, but if the duality gap is not sufficiently low the solver may take much more time to prove the solution's optimality. The fourth configuration aims at accelerating the proof of optimality. The fifth configuration uses a specification of the logical order of the variables in branching. The order specified to the solver is given by the following decreasing priority sequence: [$y_1(.)$, $e_1(.)$, $pos_1(.)$, $poc_1(.)$, $y_2(.)$, $e_2(.)$, $pos_2(.)$, $poc_2(.)$, …, $y_T(.)$, $e_T(.)$, $pos_T(.)$, $poc_T(.)$] where a lower time step implies a higher priority and where all regions, for a given time step, have the same priority. In the sixth configuration, all the cuts are disabled. For MIP models, cuts are supplementary constraints added by the solver to restrict non-integer solutions to feasible solutions. Usually, cuts reduce the total space and time needed to solve the problem. However, cuts generation costs resolution time. The idea here is to diminish the time allowed to cuts generation and to increase the time allowed to search for feasible and optimal solutions. Table 3 summarizes the Ilog Cplex tested configurations for the MIP formulation.

Table 3. Ilog Cplex tested configurations.

Configuration number	Configuration name and description
1	CplexDefault: Default Ilog Cplex configuration.
2	ScaleUp: All the probabilities are multiplied by 100 000 and the constraints are modified accordingly.
3	Feasibility: MIP emphasis on feasibility.
4	Optimality: MIP emphasis on optimality.
5	Order: A logical priority order is specified to Ilog Cplex.
6	NoCuts: All the cuts are disabled.

6 Results

This section presents the test results of the experimentation where the allowed resolution time is 7200 seconds. The first two subsections (6.1 and 6.2) present the test results for groups 1 to 4 (part 1) and groups 5 to 8 (part 2). Subsection 6.3 presents the comparative results for the two parts.

A problem instance was randomly generated and each configuration was tested on the same problem instance rescaled for each test group. When the problem was not solved to optimality, the duality gap gives an idea of how far Ilog Cplex is from an optimal solution at the end of the 7200 seconds. However, in some cases, a gap different from 0.00% may only mean that Ilog Cplex needs more time to prove that the solution is optimal. When the optimal objective value is known from another configuration result, it is easy to identify those cases. But, for larger problem instances where all the configurations failed to prove optimality before the end of the test, the duality gap remains the only basis on which to compare the results of the different groups. For this reason, the two next subsections present the time needed to solve the problem for each configuration and the duality gap obtained in an increasing problem's size order. The problem's size is based on the environment size (H by W) and on the total number of time steps allowed for the search operation (T).

For part 1, the increasing problem's size order is [group 1, group 2, group 3 and group 4]. For part 2, the increasing problem's size order is [group 5, group 6, group 7 and group 8].

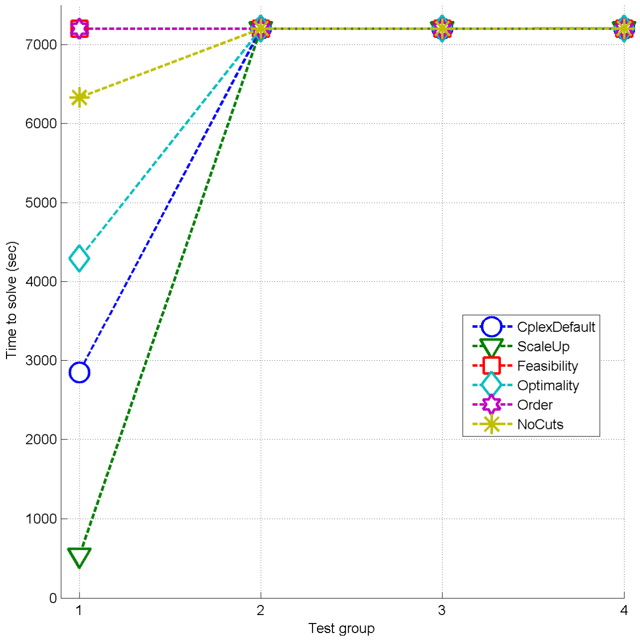

Figure 1. Time needed for groups 1 to 4.

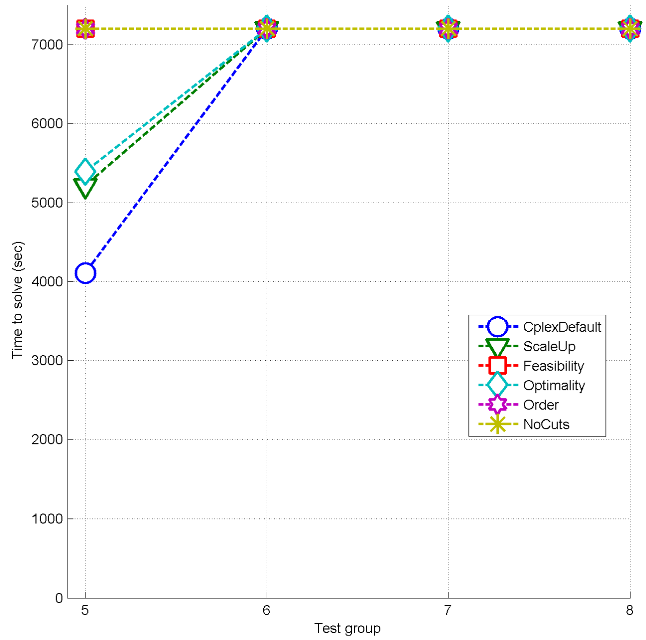

Figure 3. Time needed for groups 5 to 8.

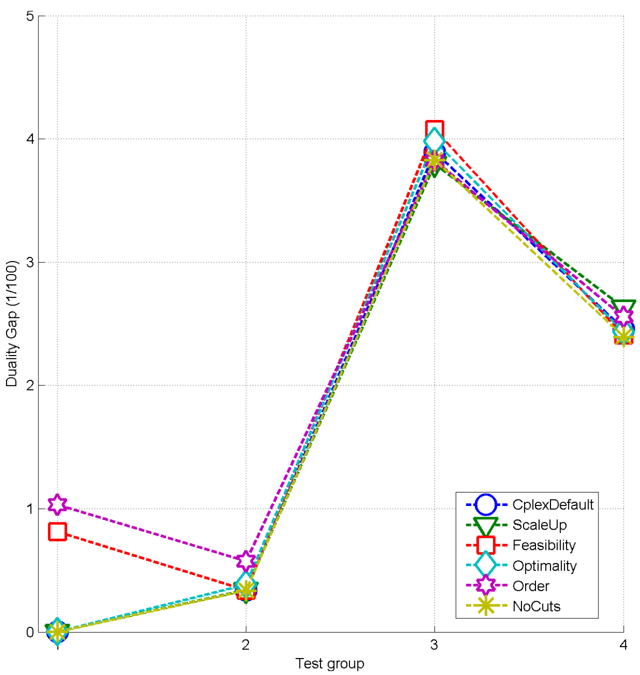

Figure 2. Duality gap obtained for groups 1 to 4.

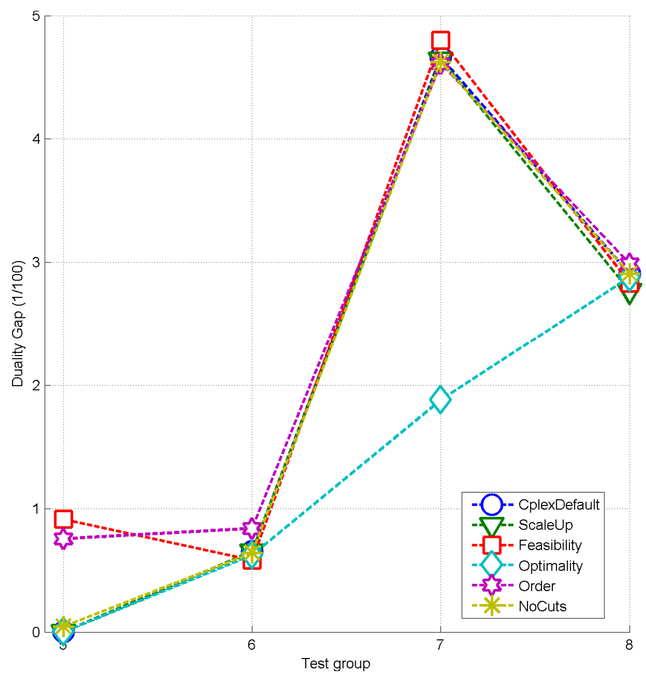

Figure 4. Duality gap obtained for groups 5 to 8.

6.1 Results for part 1

This subsection presents the results for part 1 where the search unit is confronted to "star" visibility and to "star" navigability constraints (see Table 1). Figure 1 shows the time needed to solve the instances of groups 1 to 4 for each configuration. Figure 2 shows the duality gap obtained for the instances of groups 1 to 4 for each configuration.

As the group number of a given part increases, the problem's size increases and the solver fails to find an optimal solution or to prove the optimality of the incumbent solution in the allowed time (7200 seconds). An interesting observation is that the duality gap results (Figure 2) does not increase in a constant manner for all configurations. The fact is that the only difference between groups 1 and 2 and between groups 3 and 4 is the number of time steps allowed for the search (T). It seems easier for Ilog Cplex (with the given configurations) to decrease the duality gap when T is larger. We must add that for group 2 and 4, the value of T allows the search unit to visit all the cells. Since this may have an impact on the duality gap, we cannot conclude that it is not influenced by the problem's size.

6.2 Results for part 2

This subsection presents the results for part 2 where the search unit is confronted to "star" visibility and to "plus" navigability constraints (see Table 1). Figure 3 shows the time needed to solve the instances of groups 5 to 8 for each configuration. Figure 4 shows the duality gap obtained for the instances of groups 5 to 8 for each configuration.

As in section 6.1, the problem's size and complexity increase rapidly and the duality gap does not increase in a constant manner. However, we notice by examining Figure 1 and Figure 3 that the difference between the minimal and maximal solving times is tighter for group 5 than for group 1. The only difference between groups 1 and 5 are the navigability constraints ("star" and "plus" constraints respectively). The solver has more difficulties with group 5 even though the search unit has fewer choices for the path it may follow. We therefore deduce that the complexity of an OSPV is not only tied to the total number of possible search plans or to the size of the environment. In fact, it seems that the complexity of an OSPV problem instance varies according to the environment's structure which is expressed by navigability and visibility constraints. However, further tests are needed to evaluate all the complexity factors (including the initial *poc* distribution, the movement model and other problem's data) and to infer stronger conclusions.

6.3 Comparative results for parts 1 and 2

This subsection presents comparative results for parts 1 and 2. Table 4 shows the results for test groups 1 and 5.

Table 4. Comparative results for test groups 1 and 5.

Configuration	Part 1 Test group 1		Part 2 Test group 5	
	Objective (*COS*)	Time (sec)	Objective (*COS*)	Time (sec)
CplexDefault	0.50619	**2851.05**	0.40400	4110.52
ScaleUp	0.50619	**537.55**	0.40400	5215.13
Feasibility	0.50619	7200.11	0.40400	7200.11
Optimality	0.50619	**4292.39**	0.40400	5391.91
Order	0.50619	7200.02	0.40400	7200.02
NoCuts	0.50619	**6329.33**	0.40400	7200.02

Recalling that the only difference between test groups 1 and 5 are the navigability constraints (the same environment, randomly generated initial *poc* distribution and search unit's position are used), the results comparison illustrates the impacts of the navigability constraints on the maximization of the *COS*: for the given problem instances (group 1 and 5), adding more navigability constraints induces smaller *COS* and thus a less effective search plan. For a ground search unit which is constrained to the environment's structure, a way to do an effective search is to look from a distance. However, since the grid environment of this experiment is randomly generated, the complexity of the initial *poc* distribution and the search object motion model make it hard to analyze the impact of the visibility criterion as we add more navigability constraints.

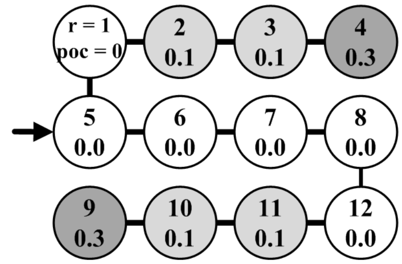

Figure 5. A grid environment represented by its navigability graph and its *poc* distribution.

We illustrate the importance of the visibility criterion on the example of Figure 5 that represents the navigability graph and the initial *poc* distribution of a given grid environment where the search unit initial position is region 5.

Considering a stationary object, we can compute the optimal search plan with "star" visibility constraints and with "OSP" visibility constraints (the search unit is constrained to search in its current region). The first search plan has a *COS* of 0.664 and the second one a *COS* of 0.464. Considering a search object motion model uniform on rows for all accessible regions leads to a *COS* of 0.520 for the first case and to a *COS* of 0.391 for the second case. Finally, if we consider an environment with the same initial *poc* distribution, stationary object and search unit initial position but with "plus" navigability constraints, the "star" visibility optimal search plan leads to a *COS* of 0.78 and the "OSP" visibility optimal search plan leads to a *COS* of 0.72.

7 Discussion

This section discusses the results of parts 1 and 2 of the experimentation.

As shown by the graphs (Figure 1 and Figure 3), the problem's complexity grows rapidly and the solver fails to prove the optimality in two hours for relatively small environments. The fact is that a 4 by 7 grid is not a realistic discretization scale for large continuous outdoor environments of SAR operations which involves constrained search units like manned patrol. However, since the OSPV problem is more general than the OSP, the latter inherits the complexity of the former. Therefore, the development of specific heuristics and the investigation of possibly faster solving methods are capital for practical applications of the OSPV problem.

In section 6.3, we asserted that as the search unit's path becomes more constrained by the environment's structure or by the search unit's own limits, the inter-region visibility importance increases: looking from a distance may be the only way of searching an inaccessible region. At the end of that section, we presented examples where the visibility criterion leads to higher *COS*. The purpose of this experiment was not to prove that the results obtained with the OSPV formulation are better than the results obtained with the classical OSP formulation: the two problems are different. We wanted to show the importance of considering inter-region visibility in modeling search operations.

Among the possible extensions to the OSPV problem, the infinitely divisible search effort case is particularly interesting because it drastically increases its expressivity. As an example, we could imagine a manned patrol that is looking in a subset of visible regions instead of being constrained to focus at only one particular region at each time step. In this case the *pod* function must vary according to the effort allocation. For a specific search unit and for a given time step, the sum of all allocated search effort must be equal to the total effort available for this search unit and this time step. Furthermore, as in the OSP problem, we could consider a total amount of available effort by search unit for the entire search operation. In all these cases, the search unit itself is constrained to stand in only one region at each time step. This statement is a major difference from the OSP problem with infinitely divisible search effort as stated by Stewart [9]. In the OSP problem with infinitely divisible search effort, at each time step, the search effort is reallocated over a subset of accessible regions. Even if a specific search pattern is assumed in searched regions, there is no precise global search path in the environment.

8 Conclusion

In conclusion, the new OSPV problem generalizes the classical OSP problem from search theory by adding the inter-region visibility which can express the constraints of real environments. In section 3, a general model for the indivisible search effort case has been presented and discussed. In sections 4 and 5, a proof of concept involving a MIP formulation of the OSPV problem has been presented. And finally, the sections 6 and 7 presented and discussed the results of the experimentation and some interesting extensions and observations.

Further work and experiments on the OSPV problem with a single search unit will include the introduction of a formulation for infinitely divisible search effort. In addition, the multiple search unit case will be considered in further formulations.

Due to the NP-Completeness of the original OSP problem [10], the potential of the OSPV problem formulation has been shown on relatively small instances. However, in practice, SAR operations are conducted over very large areas and as a result, the heuristics defined for the classic OSP must be adapted and new heuristics must be developed for the OSPV in order to manage this complexity. The evaluation of other resolution approaches such as local search and evolutionary algorithms may also be helpful in complexity management.

In order to make our model closer to the reality of SAR operations, we will consider a multi-criteria approach to incorporate more aspects of real indoor and outdoor environments topology and of real life searches. For instance, we may wish to minimize the total time steps spent in the environment, the total number of search units involved and the risk to which a search unit may be exposed. Criteria such as risk introduce the notion of safety, *i.e.*, a safer path is preferred over a perilous path among rivers or collapsed buildings or in a hostile environment. Criteria such as the total time steps spent introduce the urgency to detect in order to increase survivability of those involved in a SAR incident. Criteria such as the total search units involved deal with the logistic aspects and costs of the SAR operations.

References

[1] Brown, S. S., "Optimal search for a moving target in discrete time and space." *Operations Research.* 1980, Vol. 28, 6, pp. 1275-1289.

[2] Cooper, D. C., J. R. Frost and R. Quincy Robe, *Compatibility of land SAR procedures with search theory.* U.S. Department of Homeland Security, United States Coast Guard Operations. Potomac Management Group Inc., Washington, 2003.

[3] Eagle, J. N., "The optimal search for a moving target when the search path is constrained." *Operations Research.* 1984, Vol. 32, 5, pp. 1107-1115.

[4] Eagle, J. N., and J. R. Yee, "An optimal branch-and-bound procedure for the constrained path, moving target search problem." *Operations Research.* 1990, Vol. 38, 1, pp. 110-114.

[5] Garfinkel, R. S., and G. L. Nemhauser, *Integer Programming*. John Wiley & Sons, New York, 1972.

[6] Lau, H., S. Huang and G. Dissanayake, "Discounted MEAN bound for the optimal searcher path problem with non-uniform travel times." *European journal of operational research.* 2008, Vol. 190, 2, pp. 383-397.

[7] Martins, G., *A new branch-and-bound procedure for computing optimal search paths*. Master's Thesis. Naval Postgraduate School. 1993.

[8] Souris, G., and J. P. Le Cadre, "Un panorama des méthodes d'optimisation de l'effort de recherche en détection." *Traitement du signal.* 1999, Vol. 16, 6, pp. 403-424.

[9] Stewart, T. J., "Search for a moving target when the searcher motion is restricted." *Computers and Operations Research.* 1979, Vol. 6, pp. 129-140.

[10] Trummel, K. E., and J. R. Weisinger, "The complexity of the optimal searcher path problem." *Operations Research.* 1986, Vol. 34, 2, pp. 324-327.

[11] Washburn, A. R., "Branch and bound methods for a search problem." *Naval Research Logistics.* 1998, Vol. 45, pp. 243-257.