Hindawi Publishing Corporation
Mathematical Problems in Engineering
Volume 2015, Article ID 417597, 11 pages
http://dx.doi.org/10.1155/2015/417597

Research Article

Mathematical Approach to Security Risk Assessment

Robert Vrabel, Marcel Abas, Pavol Tanuska, Pavel Vazan, Michal Kebisek, Michal Elias, Zuzana Sutova, and Dusan Pavliak

Faculty of Materials Science and Technology, Slovak University of Technology in Bratislava, Hajdoczyho 1, 917 01 Trnava, Slovakia

Correspondence should be addressed to Robert Vrabel; robert.vrabel@stuba.sk

Received 3 December 2014; Accepted 27 January 2015

Academic Editor: Evangelos J. Sapountzakis

The goal of this paper is to provide a mathematical threat modeling methodology and a threat risk assessment tool that may assist security consultants at assessing the security risks in their protected systems/plants, nuclear power plants and stores of hazardous substances: explosive atmospheres and flammable and combustible gases and liquids, and so forth, and at building an appropriate risk mitigation policy. The probability of a penetration into the protected objects is estimated by combining the probability of the penetration by overcoming the security barriers with a vulnerability model. On the basis of the topographical placement of the protected objects, their security features, and the probability of the penetration, we propose a model of risk mitigation and effective decision making.

1. Introduction

The term physical protection of safety-critical objects represents a set of technical regime actions or organizational actions necessary to prevent the unauthorized actions performed with or in the objects (intrusion and sabotage) of critical infrastructure, such as nuclear facilities, power plants, transmission grids, drinking water supplies, storages of chemicals, oil pipelines and related facilities, and roads.

The infrastructure of developed countries is highly vulnerable and also highly interconnected. As the critical infrastructure is an international phenomenon, an attack on any state may result in the infrastructure failure at the regional level as well as at a broader international geographic level. Thus, various countries seek to harmonize their legal procedures in this paper, for example, H.R.3696: National Cybersecurity and Critical Infrastructure Protection Act of 2014 (USA) [1], Council Directive 2008/114/EC on the identification and designation of European critical infrastructures and the assessment of the need to improve their protection [2] and the associated legal acts of member states, and so forth.

Currently, an increased attention is being paid to the safety of important objects. In the literature, we can find many different approaches to analyze and to solve the problem of assessing the threat for critical infrastructure.

For example, in the paper [3], the author presents new methodology and develops the strategy and solutions for vulnerability assessment to identify and understand the threats to and vulnerabilities of critical infrastructure.

In the work of Hromada and Lukas [4], the conceptual approach and the possible ways of how to develop relevant framework for critical infrastructure protection to increase the resilience of its functional continuity are discussed.

Oyeyinka et al. [5] develop an analytical methodology for physical protection systems evaluation and their effectiveness.

The paper of Woo [6] serves as a dynamical quantification of the detection and action against the incidents using the Vensim simulation software.

As the testing and validating in real conditions are feasible only to a limited extent, the computer technique allows simulating different types of attempts to violate the protected area and thus revealing the hidden security vulnerabilities. A carefully designed model of the real examined environment filled with the correct data is inevitable.

The aim of the study is to propose algorithms enabling the users to analyze the probability of an intruder penetration to the protected object located in the area bounded by multilevel barriers with transition gates (Figure 1).

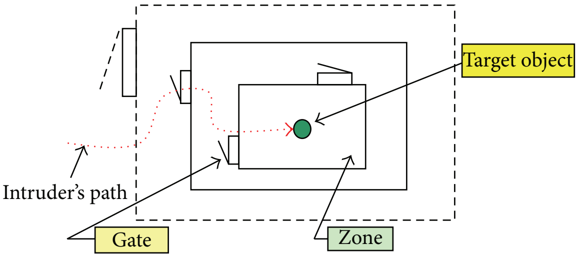


Figure 1: Sketch of the protected area with security features showing one of the possible intruder's paths to the target object.

The probability that the physical protection system prevents a hostile attack to finish an unwanted event is in the literature generally calculated as

$$\mathrm{PE} = \mathrm{PI} * \mathrm{PN}, \quad (1)$$

where PE is probability of total system effectiveness, PI is probability of interruption: the overall probability of the attack detection during its duration including the critical detection point (CDP) based on the principle of early detection and the concept of critical point detection, and PN is probability of neutralization: the probability that the corresponding force can prevent the completion of the malicious act, such as the theft of nuclear material or nuclear facility sabotage [7].

The term to neutralize means that the corresponding force stops the invader, occupies the object, or eliminates the hostile attack in another way (by causing the escape of the invader).

The principle of early detection is as follows: to interrupt the enemy attack before the requirement for the sabotage or theft is terminated. From the time of detecting the event, the reaction of the defense forces must be shorter in time than the time remaining for the completion of the enemy attack.

CDP is the last chance to detect the enemy attack. The time for the action is shorter than the time remaining for terminating the invader requirement [7].

In order to perform the intervention effectively, the early attack detection must be achieved at all possible paths to the target object.

On the basis of an extensive use of the principles of probability and the graph theory, the paper deals with the proposal of a mathematical model suitable for further computer processing. The mathematical model describes all the aspects of a real situation and creates an abstract view on the issue.

Based on the customer requirements, three scenarios have been developed:

(α) how far does the intruder penetrate into the object until the desired level of detection is reached?

(β) What is the distance between the intruder and the target when only the given time to the target is left?

(γ) Does the intruder get into the target and out of it until the desired level of detection is reached?

All three scenarios allow for detecting the supposed position of the intruder if the input condition of the given scenario is met. The algorithms, called Alpha, Beta, and Gamma, were developed for individual scenarios and are listed in other parts of the paper, in Section 6.

An application with a graphical user interface was developed for the purposes of verifying the mathematical model and algorithms.

At the end of the study, one of the series of tests carried out for random models with different parameters such as the number of barriers, gates, detection probability, is chosen.

According to our best knowledge, no paper focused on solving the specific tasks mentioned above, the scenarios Alpha, Beta, and Gamma.

2. Definitions of Security Features and Nomenclature

In this section, we introduce the definitions of principal concepts used in this paper as follows:

(1) *target object*: TO (e.g., nuclear reactor);

(2) barriers: the continuous obstacles to penetrate into the protected object (e.g., a fence): Bar_i, $i := 0, 1, \ldots, k$, $\mathrm{Bar}_k :=$ an outer barrier, and $\mathrm{Bar}_0 := \mathrm{TO}$. The number of barriers $\#\mathrm{Bar}_i = k$; with TO, we have $(k + 1)$ barriers;

(3) Zone: the area between two consecutive barriers: Z_i, $i := 1, 2, \ldots, k$. The total number of the zones $\#\mathrm{Z}_i = k$;

(4) Gates: inputs on the barriers: $\mathrm{G}_{i,j}$, $i := 1, 2, \ldots, k$, $j := 1, 2, \ldots, l_i$ $\mathrm{G}_{i,1}, \mathrm{G}_{i,2}, \ldots, \mathrm{G}_{i,l_i}$ fences on the barrier Bar_i. The total number of the gates on barrier Bar_i is $\#\mathrm{G}_i = l_i$, and the total number of the gates $\#\mathrm{G}_{i,j} = \sum_{i=1}^{k} l_i$;

(5) Rays: half-lines $\rho_{i,j}$ connecting the target object with the barriers; more precisely, $\rho_{i,j}$ is the half-line $\mathrm{TO} - \mathrm{G}_{i,j}$. The total number of the rays is equal to the total number of the gates; that is, $\#\rho_{i,j} = \sum_{i=1}^{k} l_i$;

(6) R-gates: physical places on the barriers lying on the rays $\rho_{i,j}$: $\mathrm{R}_{i,j,n}$, $i := 1, 2, \ldots, k$, $j := 1, 2, \ldots, l_i$, and $n := 1, 2, \ldots, k$, which is a physical place on the half-line $\rho_{i,j}$ lying on the barrier Bar_n. Thus, if $\rho_{i,j} = \mathrm{TO} - \mathrm{G}_{i,j}$, it is the connection of the target object TO and the gate $\mathrm{G}_{i,j}$, and then the R-gate $\mathrm{R}_{i,j,n}$ lies on the intersection of $\rho_{i,j}$ and Bar_n. The total number of the R-gates is $\#\mathrm{R}_{i,j,n} = k \cdot \#\rho_{i,j}$.

Remark 1. The reason for introducing the concept of R-gate is the need of implementing the calculations in real time by reducing the number of less probable paths of the intruder.

Remark 2. Obviously, $\mathrm{R}_{s,j,s} = \mathrm{G}_{s,j}$, $s = 1, \ldots, k$. Therefore, in the process of implementing the algorithms (Section 6), we will denote the gates and the R-gates on the same barrier (say, s) consecutively and with two indexes only.

3. Required Data

3.1. Location of Objects. Let $A = [Ax, Ay]$ denote the coordinates of the object A. Thus, one can see the following:

(1) the coordinates of the target object $\mathrm{TO} : [\mathrm{TO}x, \mathrm{TO}y]$, after translation $[\mathrm{TO}x, \mathrm{TO}y] = [0, 0]$;

(2) the coordinates of $\mathrm{G}_{i,j} : [\mathrm{G}_{i,j}x, \mathrm{G}_{i,j}y]$;

(3) the coordinates of $\mathrm{R}_{i,j,n} : [\mathrm{R}_{i,j,n}x, \mathrm{R}_{i,j,n}y]$;

(4) the rays $\rho_{i,j} : X = \mathrm{G}_{i,j}x \cdot t, Y = \mathrm{G}_{i,j}y \cdot t, t > 0$ $(\mathrm{TO} = [0, 0])$.

3.2. Probabilities of Detection during Penetration. Let $P_d^+ A$ denote the probability of detection of the subjects penetrating through the object A in the direction *to* the target object TO and $P_d^- A$ in the direction *from* the target object TO. Then, the probability of the penetration through the object A will be $P_p^+ A = 1 - P_d^+ A$ and $P_p^- A = 1 - P_d^- A$:

(1) $P_p\mathrm{TO} : (P_p^+\mathrm{TO}, P_p^-\mathrm{TO})$;

(2) $P_p\mathrm{Bar}_i : (P_p^+\mathrm{Bar}_i, P_p^-\mathrm{Bar}_i)$;

(3) $P_p\mathrm{G}_{i,j} : (P_p^+\mathrm{G}_{i,j}, P_p^-\mathrm{G}_{i,j})$;

(4) $P_p\mathrm{R}_{i,j} : (P_p^+\mathrm{R}_{i,j}, P_p^-\mathrm{R}_{i,j})$;

(5) $P_d\mathrm{Z}_i : (P_d^+\mathrm{Z}_i, P_d^-\mathrm{Z}_i)$ the probability of detection per second of the stay in the zone Z_i.

3.3. The Assumed Times Needed to Overcome the Security Features. Let T^+A denote the assumed time of the penetration through the object A (in the direction *to* the target object TO) and T^-A (in the direction *from* the target object TO). Then, we denote the following:

(1) $T\mathrm{TO} : (T^+\mathrm{TO}, T^-\mathrm{TO})$;

(2) $T\mathrm{Bar}_i : (T^+\mathrm{Bar}_i, T^-\mathrm{Bar}_i)$;

(3) $T\mathrm{G}_{i,j} : (T^+\mathrm{G}_{i,j}, T^-\mathrm{G}_{i,j})$;

(4) $T\mathrm{R}_{i,j} : (T^+\mathrm{R}_{i,j}, T^-\mathrm{R}_{i,j})$.

4. Required Inputs and Outputs

4.1. Inputs into the Mathematical Model. The necessary inputs into mathematical model are the following:

(1) scenario selection: α, β or γ;

(2) specifying v_i^+ and v_i^-, $i = 1, 2, \ldots, k$, the speed of the penetrating subject through the zone Z_i towards/from the target object TO, respectively;

(3) the probabilities and times of the penetration through the protection elements;

(4) the possibility to switch off the selected security features:

(a) if the barrier Bar_s is switched off when moving inwards, then $P_p^+\mathrm{Bar}_s = 1$, $P_p^+\mathrm{G}_{s,j} = 1$, $P_p^+\mathrm{R}_{i,j,s} = 1$, $P_p^+\mathrm{Z}_s = 1$, $T^+\mathrm{Bar}_s = 0$, and $T^+\mathrm{G}_{s,j} = 0$, $T^+\mathrm{R}_{i,j,s} = 0, \forall i, j$, analogously when moving outwards $P_p^-\mathrm{Bar}_s = 1$, $P_p^-\mathrm{G}_{s,j} = 1$, $P_p^-\mathrm{R}_{i,j,s} = 1P_p^-\mathrm{Z}_s = 1$, $T^-\mathrm{Bar}_s = 0$, and $T^-\mathrm{G}_{s,j} = 0$, $T^-\mathrm{R}_{i,j,s} = 0, \forall i, j$;

(b) if the gate $\mathrm{G}_{s,r}$ is switched off when moving inwards, then $P_p^+\mathrm{G}_{s,r} = 1$, $P_p^+\mathrm{Z}_s = 1$, $T^+\mathrm{G}_{s,r} = 0$, analogously when moving outwards $P_p^-\mathrm{G}_{s,r} = 1$, $P_p^-\mathrm{Z}_s = 1$, $T^-\mathrm{G}_{s,r} = 0$;

(c) if the zone Z_s is switched off when moving inwards, then $P_p^+\mathrm{Z}_s = 1$, analogously when moving outwards $P_p^-\mathrm{Z}_s = 1$.

4.2. Outputs from Mathematical Model. The required outputs from mathematical model are the following:

(α) For the given probability of the detection $\widetilde{P}_d$, determine the set of points (and paths belonging to them) in which the probability level of the detection $\widetilde{P}_d$ is reached exactly.

(β) For the given time T, determine the set of points (and paths belonging to them) by which the time for achieving the target object TO is equal to the time T.

(γ) For the given probability of the detection $\widetilde{P}_d$, find the return paths (if any) with the probability of the detection lower than required (the return path is defined as the path starting at some point on the outer barrier Bar_k, passing through TO, and ending on the outer barrier Bar_k).

5. Preliminary Calculations

Using the data specified in Sections 3.1, 3.2, and 3.3, we put together a mathematical model of the whole protected object. Obviously, these sensitive data require a high degree of confidentiality. In addition to these data, it is necessary to determine and calculate the following values.

(1) The target object TO is being translated into the origin $[0, 0]$ of the coordinate system.

(2) Location of an arbitrary object A is $[Ax, Ay] := [Ax, Ay] - [\mathrm{TO}x, \mathrm{TO}y]$.

(3) The real position $[Ax, Ay]$ for each object A is being calculated using the map scale.

(4) Let A be the object from the set $\{\mathrm{G}_{i,j}, \mathrm{R}_{r,j,i}\}$, $i = 1, \ldots, k$, fixed and let B be the object from the set $\{\mathrm{G}_{i-1,j'}, \mathrm{R}_{r',j',i-1}\}$. The distance d, calculated using the classical Euclidean norm, is calculated for every such pair (A, B).

(5) Let $D = d(A, B)$. Then, the time that the subject passes from the object A to the object B through the zone Z_i at the rate v^+ is equal to $T^+D = D/v_i^+$. Similarly, the time that the subject passes from the object B to the object A through the zone Z_i at the rate v^- is equal to $T^-D = D/v_i^-$.

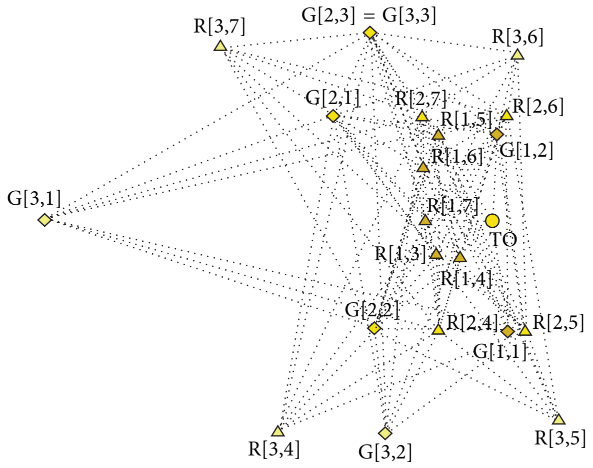


Figure 2: Topographical placement of the target object and its security features.

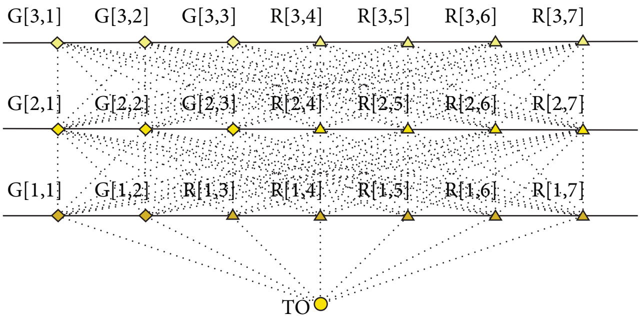


Figure 3: Schematic placement of the target object and its security features.

(6) The probability of the detection of the subject moving through the zone Z towards TO is $P_d^+D = 1 - (1 - P_d^+Z)^{T^+D}$ and that in the direction away from TO is $P_d^-D = 1 - (1 - P_d^-Z)^{T^-D}$.

6. Algorithms

Based on the three algorithms, there are three cases of the intrusion by intruding into the protected object proposed and analyzed in this section. The Alpha analysis represents the evaluation of the possibility of the intruder penetration algorithm based on a set of detection probability level. The Beta analysis evaluates the distance from the penetration spot to the target with respect to time. The Gamma analysis examines the possibilities of the intruder penetration into the target and out of the protected object based on the desired detection level.

6.1. Recursive Procedure, Path Alpha (Figure 10). This subsection introduces the flowchart [8] for the Alpha analysis used for implementing the mathematical model into the software environment.

Path characterization is as follows: *How far does the intruder penetrate into the object until the desired level of detection is reached*?

6.2. Recursive Procedure, Path Beta (Figure 11). In this subsection, we propose the flowchart implementing the mathematical model to the software environment with the purpose of examining the Beta path.

Path characterization is as follows: *What is the distance between the intruder and the target, when only the given time to the target is left*?

6.3. Recursive Procedures, Path Gamma (Figures 12 and 13). The flowchart presented in this subsection was designed for the Gamma path and is supposed to examine the probability of the intruder penetration into and out of the object successfully.

Path characterization is as follows: *Does the intruder get into the target and out of it until the desired level of detection is reached*?

7. Application of Mathematical Model

In this section, we apply the proposed methodology to the fictive model of the protected area.

Figures 2 and 3 show the topographical and schematic placement of the target object and its security features, respectively. The symbols used in Figures 2–9 are explained in Table 1.

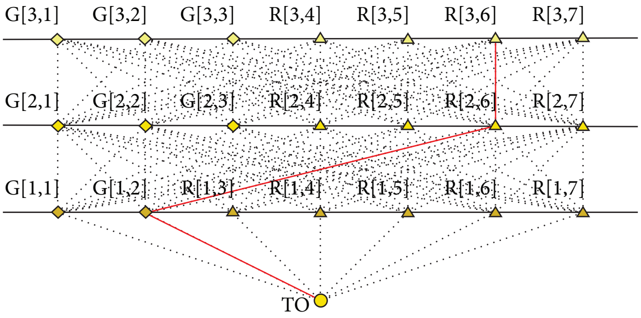


Figure 4: Alpha analysis, Path 1 (an example).

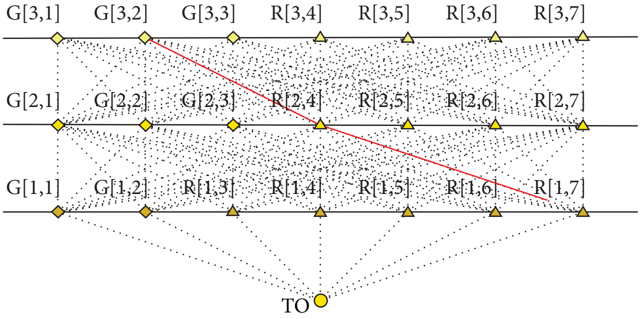


Figure 5: Alpha analysis, Path 2 (an example).

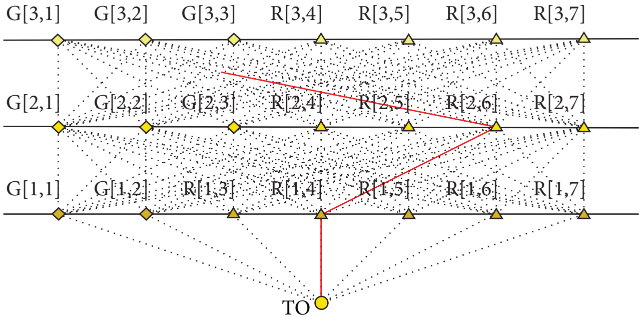


Figure 6: Beta analysis, Path 1 (an example).

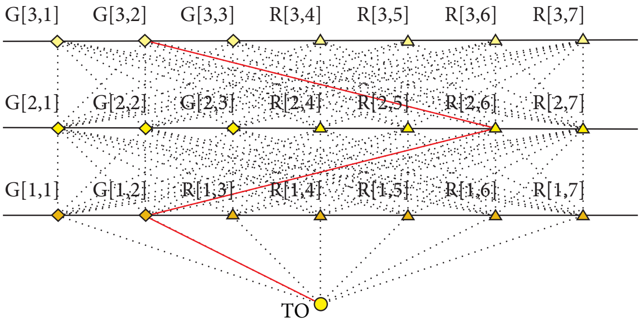


Figure 7: Beta analysis, Path 2 (an example).

Table 1: Explanatory notes to the schemes.

TO ●	Target object
G[1,1] ◆	Gate
R[1,1] ▲	R-gate
—	Fence
—	Path in the direction to target object
—	Path in the direction from target object

Tables 2 and 3 refer to the parameters of gates and zones, respectively.

7.1. Analysis Alpha: The Selected Paths

7.1.1. Path 1. One can see the following:

desired probability of detection $\widetilde{P}_d = 0.97015$;

path: R[3,6] → R[2,6] (segment length 24.33 m) → G[1,2] (segment length 8.06 m) → TO (segment length 34.06 m) (Figure 4);

total time of penetration = 123 s.

7.1.2. Path 2. One can see the following:

desired probability of detection $\widetilde{P}_d = 0.96000$;

path: G[3,2] → R[2,4] (segment length 45.18 m) → R[1,7] (segment length 37.40 m of 43.29 m) (Figure 5);

total time of penetration = 73 s.

Table 2: The gate parameters (an example).

Zone number	Gate number	Gate type	Probability of detection inwards	Probability of detection outwards	Inward penetration time [s]	Outward penetration time [s]	X [m]	Y [m]
TO			0.410	0.400	26	25	207	114
1	1	G	0.350	0.340	29	27	213	71
1	2	G	0.350	0.340	29	27	209	148
1	3	R	0.280	0.300	33	31	185	101
1	4	R	0.360	0.340	31	31	194	100
1	5	R	0.390	0.370	41	43	186	148
1	6	R	0.410	0.450	35	38	180	135
1	7	R	0.360	0.340	36	38	181	114
2	1	G	0.340	0.360	35	32	145	155
2	2	G	0.350	0.360	31	34	161	72
2	3	G	0.340	0.360	35	32	159	188
2	4	R	0.340	0.360	29	27	186	71
2	5	R	0.420	0.390	31	32	220	71
2	6	R	0.380	0.340	28	24	213	155
2	7	R	0.390	0.340	40	38	179	155
3	1	G	0.410	0.380	35	34	32	115
3	2	G	0.380	0.350	29	28	165	31
3	3	G	0.410	0.380	35	34	159	188
3	4	R	0.400	0.390	34	36	123	32
3	5	R	0.370	0.360	28	26	233	36
3	6	R	0.390	0.340	28	23	217	179
3	7	R	0.280	0.290	26	28	101	182

Table 3: Zone parameters (an example).

Zone number	Probability of detection inwards	Probability of detection outwards	Inward penetration speed [m/s]	Outward penetration speed [m/s]
1	0.100	0.080	5.600	5.500
2	0.130	0.110	5.400	5.500
3	0.150	0.120	5.400	5.500

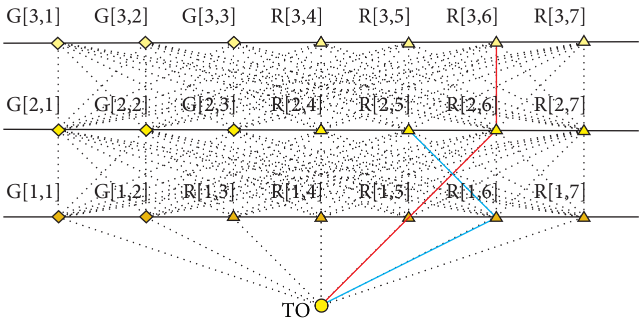


Figure 8: Gamma analysis, Path 1 (an example).

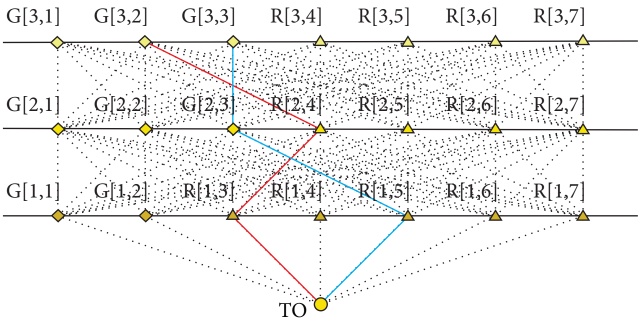


Figure 9: Gamma analysis, Path 2 (an example).

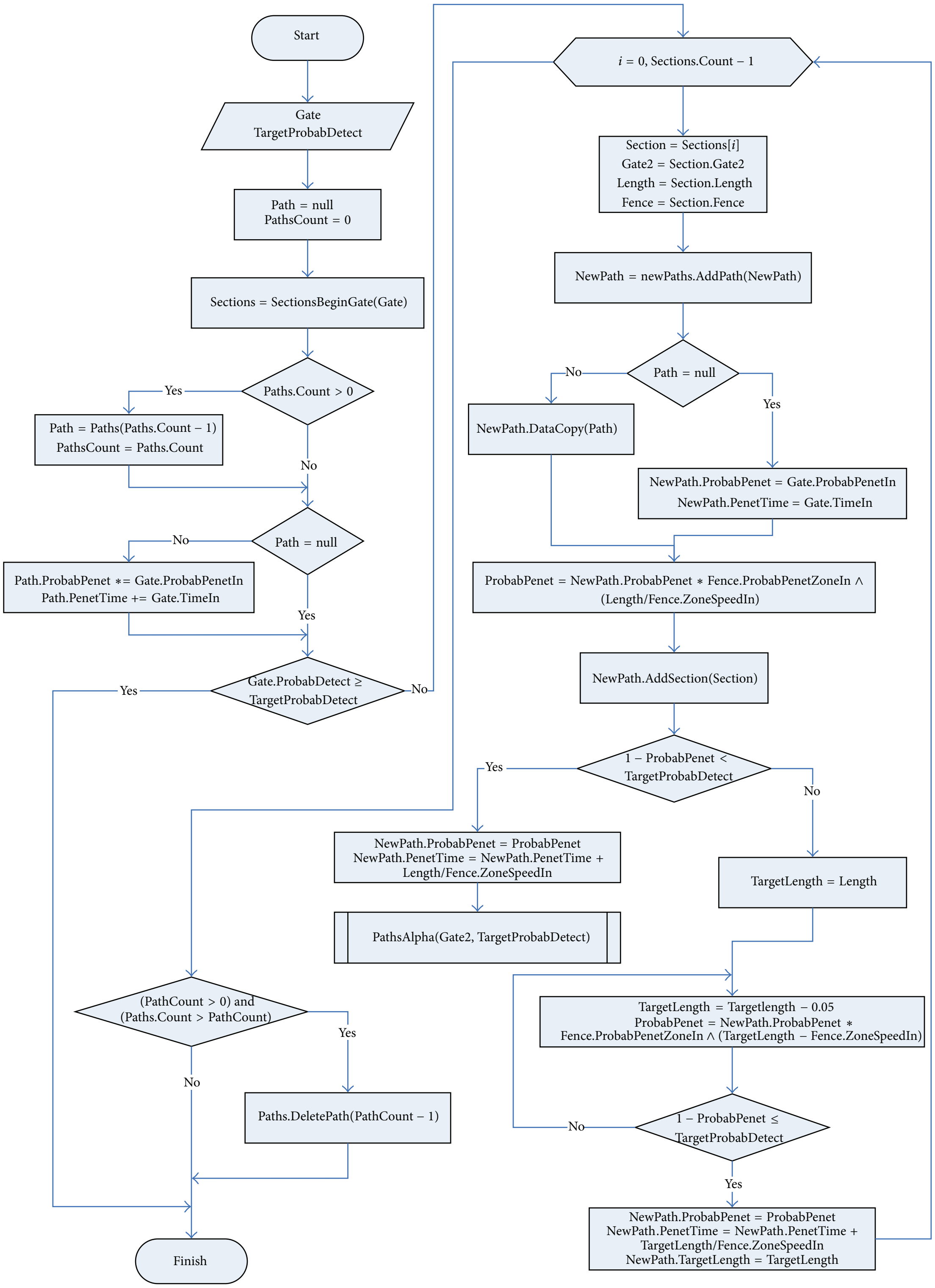

Start
Gate
TargetProbabDetect
Path = null
PathsCount = 0
Sections = SectionsBeginGate(Gate)
Paths.Count > 0
Yes
Path = Paths(Paths.Count − 1)
PathsCount = Paths.Count
No
Path = null
No
Path.ProbabPenet ∗= Gate.ProbabPenetIn
Path.PenetTime += Gate.TimeIn
Yes
Gate.ProbabDetect ≥
TargetProbabDetect
Yes
No
i = 0, Sections.Count − 1
Section = Sections[i]
Gate2 = Section.Gate2
Length = Section.Length
Fence = Section.Fence
NewPath = newPaths.AddPath(NewPath)
Path = null
No
Yes
NewPath.DataCopy(Path)
NewPath.ProbabPenet = Gate.ProbabPenetIn
NewPath.PenetTime = Gate.TimeIn
ProbabPenet = NewPath.ProbabPenet ∗ Fence.ProbabPenetZoneIn ∧
(Length/Fence.ZoneSpeedIn)
NewPath.AddSection(Section)
1 − ProbabPenet <
TargetProbabDetect
Yes
No
NewPath.ProbabPenet = ProbabPenet
NewPath.PenetTime = NewPath.PenetTime +
Length/Fence.ZoneSpeedIn
PathsAlpha(Gate2, TargetProbabDetect)
TargetLength = Length
TargetLength = Targetlength − 0.05
ProbabPenet = NewPath.ProbabPenet ∗
Fence.ProbabPenetZoneIn ∧ (TargetLength − Fence.ZoneSpeedIn)
1 − ProbabPenet ≤
TargetProbabDetect
No
Yes
NewPath.ProbabPenet = ProbabPenet
NewPath.PenetTime = NewPath.PenetTime +
TargetLength/Fence.ZoneSpeedIn
NewPath.TargetLength = TargetLength
(PathCount > 0) and
(Paths.Count > PathCount)
Yes
No
Paths.DeletePath(PathCount − 1)
Finish

Figure 10: The flowchart of recursive procedure, path Alpha.

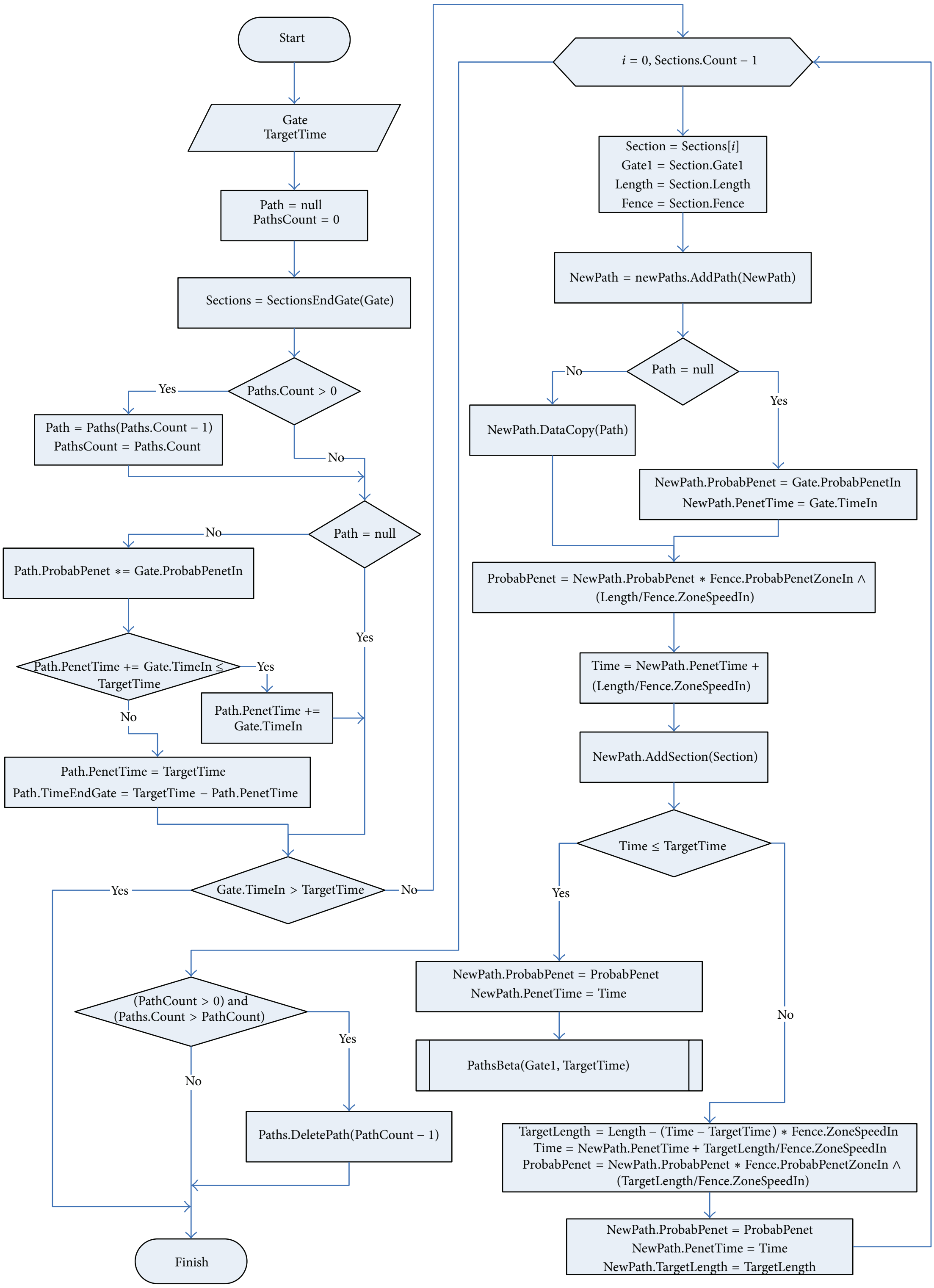

Start
Gate
TargetTime
Path = null
PathsCount = 0
Sections = SectionsEndGate(Gate)
Paths.Count > 0
Yes
Path = Paths(Paths.Count − 1)
PathsCount = Paths.Count
No
Path = null
No
Path.ProbabPenet ∗= Gate.ProbabPenetIn
Path.PenetTime += Gate.TimeIn ≤ TargetTime
Yes
Path.PenetTime += Gate.TimeIn
No
Path.PenetTime = TargetTime
Path.TimeEndGate = TargetTime − Path.PenetTime
Yes
Gate.TimeIn > TargetTime
Yes
No
(PathCount > 0) and (Paths.Count > PathCount)
Yes
No
Paths.DeletePath(PathCount − 1)
Finish
i = 0, Sections.Count − 1
Section = Sections[i]
Gate1 = Section.Gate1
Length = Section.Length
Fence = Section.Fence
NewPath = newPaths.AddPath(NewPath)
Path = null
No
Yes
NewPath.DataCopy(Path)
NewPath.ProbabPenet = Gate.ProbabPenetIn
NewPath.PenetTime = Gate.TimeIn
ProbabPenet = NewPath.ProbabPenet ∗ Fence.ProbabPenetZoneIn ∧ (Length/Fence.ZoneSpeedIn)
Time = NewPath.PenetTime + (Length/Fence.ZoneSpeedIn)
NewPath.AddSection(Section)
Time ≤ TargetTime
Yes
No
NewPath.ProbabPenet = ProbabPenet
NewPath.PenetTime = Time
PathsBeta(Gate1, TargetTime)
TargetLength = Length − (Time − TargetTime) ∗ Fence.ZoneSpeedIn
Time = NewPath.PenetTime + TargetLength/Fence.ZoneSpeedIn
ProbabPenet = NewPath.ProbabPenet ∗ Fence.ProbabPenetZoneIn ∧ (TargetLength/Fence.ZoneSpeedIn)
NewPath.ProbabPenet = ProbabPenet
NewPath.PenetTime = Time
NewPath.TargetLength = TargetLength

FIGURE 11: The flowchart of recursive procedure, path Beta.

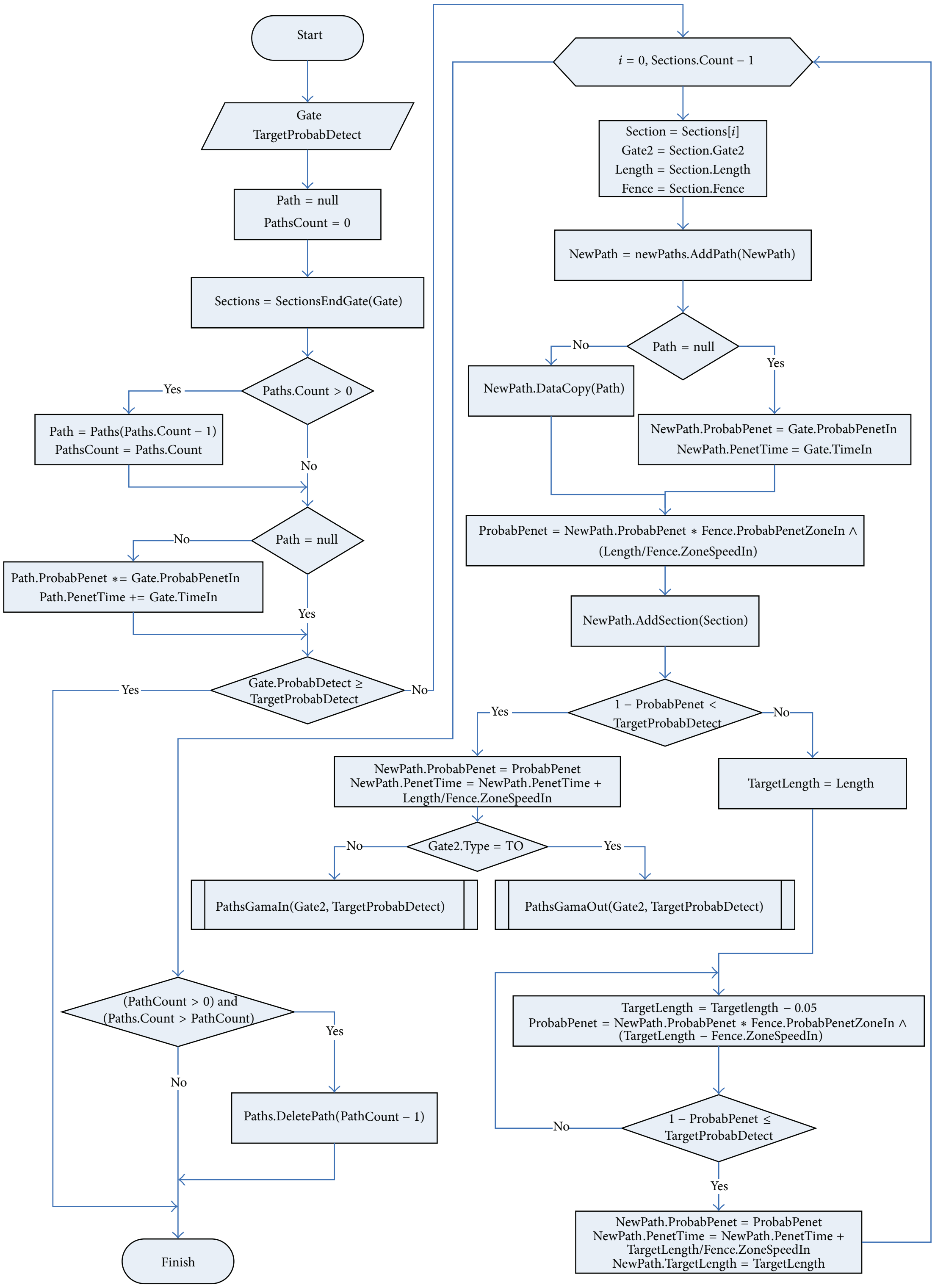

Start
Gate
TargetProbabDetect
Path = null
PathsCount = 0
Sections = SectionsEndGate(Gate)
Paths.Count > 0
Yes
Path = Paths(Paths.Count − 1)
PathsCount = Paths.Count
No
Path = null
No
Path.ProbabPenet ∗= Gate.ProbabPenetIn
Path.PenetTime += Gate.TimeIn
Yes
Gate.ProbabDetect ≥
TargetProbabDetect
Yes
No
(PathCount > 0) and
(Paths.Count > PathCount)
Yes
No
Paths.DeletePath(PathCount − 1)
Finish
i = 0, Sections.Count − 1
Section = Sections[i]
Gate2 = Section.Gate2
Length = Section.Length
Fence = Section.Fence
NewPath = newPaths.AddPath(NewPath)
Path = null
No
Yes
NewPath.DataCopy(Path)
NewPath.ProbabPenet = Gate.ProbabPenetIn
NewPath.PenetTime = Gate.TimeIn
ProbabPenet = NewPath.ProbabPenet ∗ Fence.ProbabPenetZoneIn ∧
(Length/Fence.ZoneSpeedIn)
NewPath.AddSection(Section)
1 − ProbabPenet <
TargetProbabDetect
Yes
No
NewPath.ProbabPenet = ProbabPenet
NewPath.PenetTime = NewPath.PenetTime +
Length/Fence.ZoneSpeedIn
TargetLength = Length
Gate2.Type = TO
No
Yes
PathsGamaIn(Gate2, TargetProbabDetect)
PathsGamaOut(Gate2, TargetProbabDetect)
TargetLength = Targetlength − 0.05
ProbabPenet = NewPath.ProbabPenet ∗ Fence.ProbabPenetZoneIn ∧
(TargetLength − Fence.ZoneSpeedIn)
1 − ProbabPenet ≤
TargetProbabDetect
No
Yes
NewPath.ProbabPenet = ProbabPenet
NewPath.PenetTime = NewPath.PenetTime +
TargetLength/Fence.ZoneSpeedIn
NewPath.TargetLength = TargetLength

Figure 12: The flowchart of recursive procedure, path Gamma (inwards).

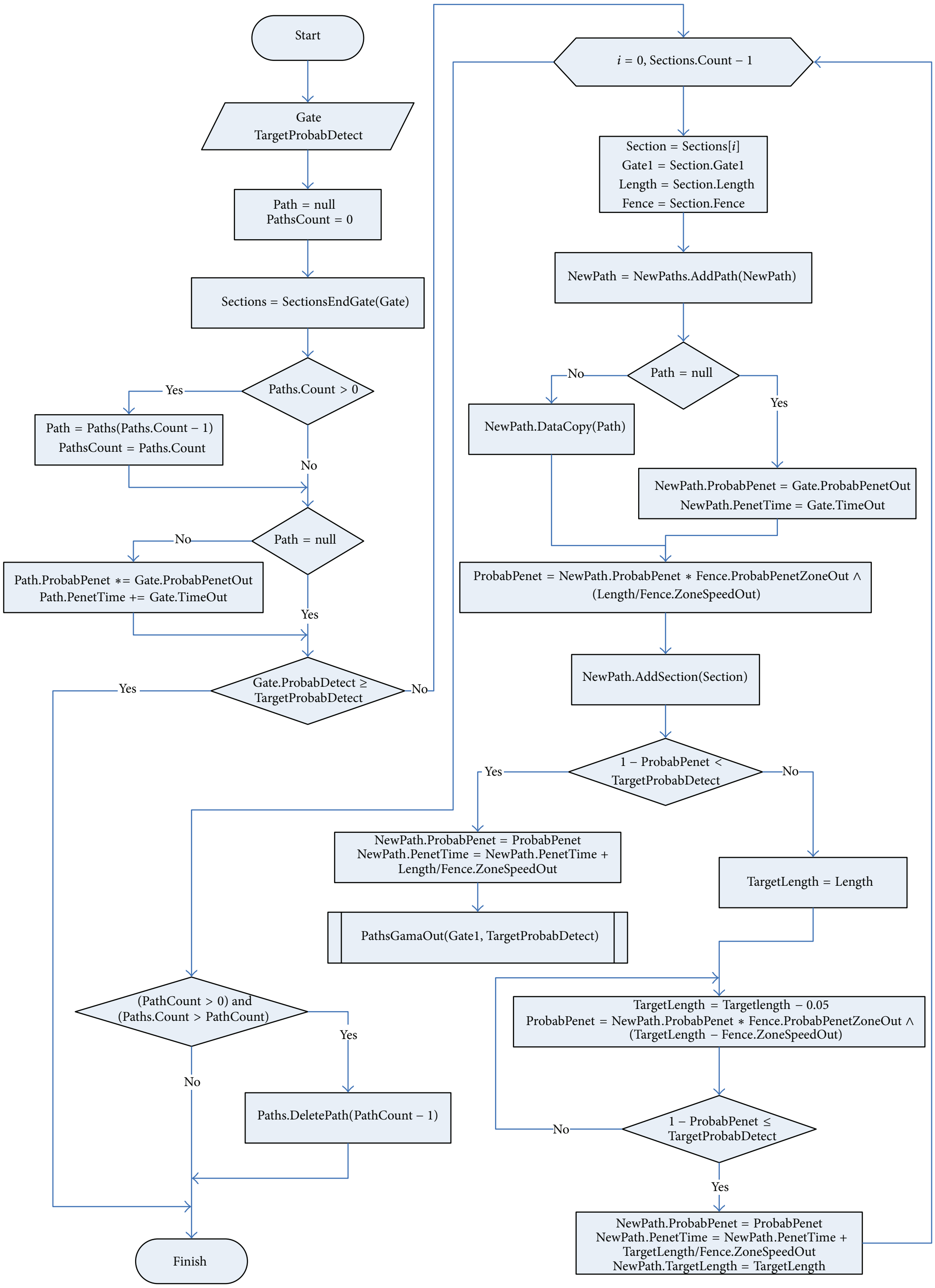

Start
Gate
TargetProbabDetect
Path = null
PathsCount = 0
Sections = SectionsEndGate(Gate)
Paths.Count > 0
Yes
Path = Paths(Paths.Count – 1)
PathsCount = Paths.Count
No
Path = null
No
Path.ProbabPenet *= Gate.ProbabPenetOut
Path.PenetTime += Gate.TimeOut
Yes
Gate.ProbabDetect ≥
TargetProbabDetect
Yes
No
i = 0, Sections.Count – 1
Section = Sections[i]
Gate1 = Section.Gate1
Length = Section.Length
Fence = Section.Fence
NewPath = NewPaths.AddPath(NewPath)
Path = null
No
Yes
NewPath.DataCopy(Path)
NewPath.ProbabPenet = Gate.ProbabPenetOut
NewPath.PenetTime = Gate.TimeOut
ProbabPenet = NewPath.ProbabPenet ∗ Fence.ProbabPenetZoneOut ∧
(Length/Fence.ZoneSpeedOut)
NewPath.AddSection(Section)
1 – ProbabPenet <
TargetProbabDetect
Yes
No
NewPath.ProbabPenet = ProbabPenet
NewPath.PenetTime = NewPath.PenetTime +
Length/Fence.ZoneSpeedOut
PathsGamaOut(Gate1, TargetProbabDetect)
TargetLength = Length
TargetLength = Targetlength – 0.05
ProbabPenet = NewPath.ProbabPenet ∗ Fence.ProbabPenetZoneOut ∧
(TargetLength – Fence.ZoneSpeedOut)
(PathCount > 0) and
(Paths.Count > PathCount)
Yes
No
Paths.DeletePath(PathCount – 1)
1 – ProbabPenet ≤
TargetProbabDetect
No
Yes
NewPath.ProbabPenet = ProbabPenet
NewPath.PenetTime = NewPath.PenetTime +
TargetLength/Fence.ZoneSpeedOut
NewPath.TargetLength = TargetLength
Finish

FIGURE 13: The flowchart of recursive procedure, path Gamma (outwards).

7.2. *Analysis Beta: The Selected Paths*

7.2.1. Path 1. One can see the following:

time needed to reach TO: $T = 120$ s-CDP;

path: G[3,1] → R[2,6] (segment length 115.17 m of 185.37 m) → R[1,4] (segment length 58.19 m) → TO (segment length 19.10 m) (Figure 6);

probability of detection = 0.99886.

7.2.2. Path 2. One can see the following:

time needed to reach TO: $T = 120$ s-CDP;

path: G[3,2] → R[2,6] (segment length 132.97 m) → G[1,2] (segment length 8.06 m) → TO (segment length 34.06 m) (Figure 7);

probability of detection = 0.99885.

7.3. *Analysis Gamma: The Selected Paths*

7.3.1. Path 1. One can see the following:

desired probability of detection $\widetilde{P}_d = 0.99938$;

path: R[3,6] → R[2,6] (segment length 24.33 m) → R[1,5] (segment length 27.89 m) → TO (segment length 39.96 m) → R[1,6] (segment length 34.21 m) → R[2,5] (segment length 75.47 m) (Figure 8);

total time of penetration = 229 s;

probability of detection = 0.99938.

7.3.2. Path 2. One can see the following:

desired probability of detection $\widetilde{P}_d = 0.99938$;

path: G[3,2] → R[2,4] (segment length 45.18 m) → R[1,3] (segment length 30.02 m) → TO (segment length 25.55 m) → R[1,5] (segment length 39.96 m) → G[2,3] (segment length 48.26 m) → G[3,3] (segment length 0.00 m, G[2,3] = G[3,3]) (Figure 9);

total time of penetration = 260 s;

probability of detection = 0.99937.

8. Conclusions

The submitted study analyzes the alternatives of the intruder penetration into the protected area by processing the data describing the detection capabilities in overcoming the transition gates and barriers or moving through the area. The solution relevance is closely related to the accuracy of the input data.

A mathematical view of the studied issue created an abstraction serving as a basis for the model and algorithm proposal. The computer technology must be involved due to the number of combinations arising in the model transition. Therefore, the user interface, suggesting the design of application assisting in the processing of the issue, was proposed. The subsequent implementation was necessary in order to verify the correctness of the mathematical model, the functionality of the proposed algorithms, and the applicability and intuitiveness of the designed user interface.

Emerging from the performed tests, it can be concluded that the proposed algorithms are functional and are able to achieve the desired results. The tests also highlight the problem of an exponential increase of road alternatives after increasing the number of barriers and gates. It will be necessary to establish criteria, filtering out the uninteresting intrusive ways. A significant reduction in the total paths is required for the postprocessing of results by man.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

References

[1] H.R.3696—National Cybersecurity and Critical Infrastructure Protection Act of 2014, http://www.congress.gov/bill/113th-congress/house-bill/3696/text.

[2] Council Directive 2008/114/EC on the identification and designation of European critical infrastructures and the assessment of the need to improve their protection, http://eur-lex.europa.eu/LexUriServ/LexUriServ.do?uri=OJ:L:2008:345:0075:0082:EN:PDF .

[3] V. Kostadinov, "Developing new methodology for nuclear power plants vulnerability assessment," *Nuclear Engineering and Design*, vol. 241, no. 3, pp. 950–956, 2011.

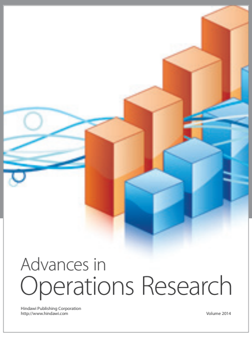
[4] M. Hromada and L. Lukas, "Critical infrastructure protection and the evaluation process," *International Journal of Disaster Recovery and Business Continuity*, vol. 3, pp. 37–46, 2012.

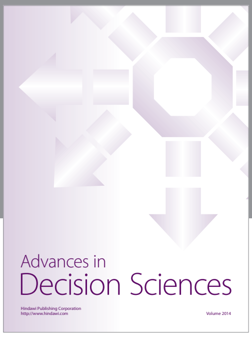
[5] O. D. Oyeyinka, L. A. Dim, M. C. Echeta, and A. O. Kuye, "Determination of system effectiveness for physical protection systems of a nuclear energy centre," *Science and Technology*, vol. 4, no. 2, pp. 9–16, 2014.

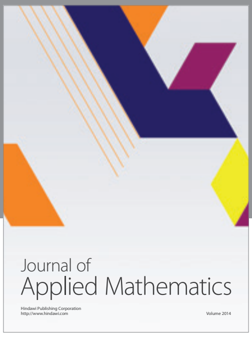
[6] T. H. Woo, "Systems thinking safety analysis: nuclear security assessment of physical protection system in nuclear power plants," *Science and Technology of Nuclear Installations*, vol. 2013, Article ID 473687, 5 pages, 2013.

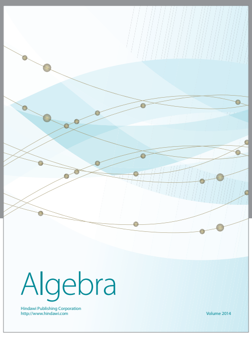
[7] ITC, *Physical Protection of Nuclear Facilities and Materials*, ITC, Albuquerque, NM, USA, 2010.

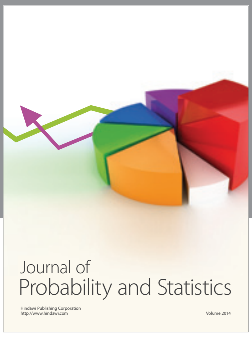
[8] D. Madison, *Process Mapping, Process Improvement and Process Management*, Paton Press, 2005.


Advances in
Operations Research

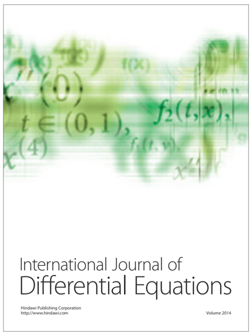

Advances in
Decision Sciences


Journal of
Applied Mathematics

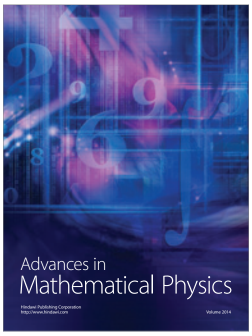

Algebra

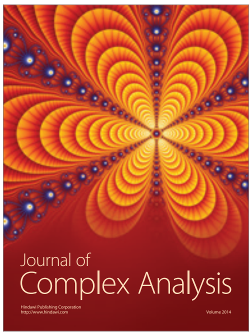

Journal of
Probability and Statistics

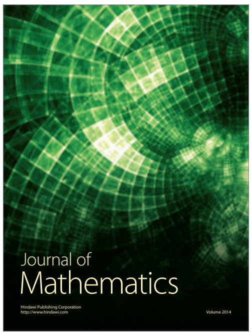
The Scientific
World Journal

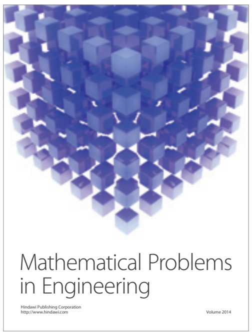

International Journal of
Differential Equations

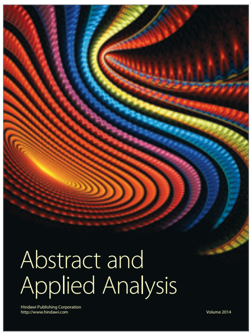
International Journal of
Combinatorics

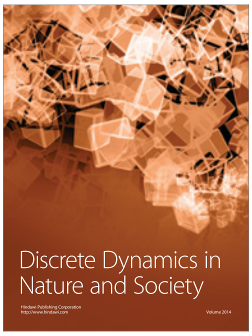

Advances in
Mathematical Physics

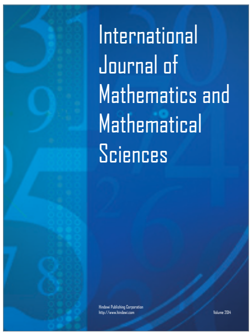

Journal of
Complex Analysis


Journal of
Mathematics


Mathematical Problems
in Engineering

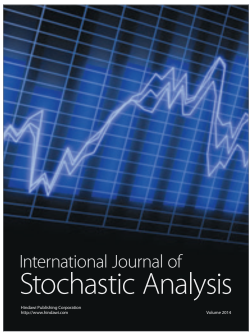

Abstract and
Applied Analysis


Discrete Dynamics in
Nature and Society


International
Journal of
Mathematics and
Mathematical
Sciences

Journal of
Discrete Mathematics

Journal of
Function Spaces


International Journal of
Stochastic Analysis

Journal of
Optimization