mathematics

MDPI

Article

Advanced Drone Swarm Security by Using Blockchain Governance Game

Song-Kyoo (Amang) Kim

Faculty of Applied Sciences, Macao Polytechnic University, R. de Luis Gonzaga Gomes, Macao SAR, China; amang@mpu.edu.mo; Tel.: +853-8599-6455

Abstract: This research contributes to the security design of an advanced smart drone swarm network based on a variant of the Blockchain Governance Game (BGG), which is the theoretical game model to predict the moments of security actions before attacks, and the Strategic Alliance for Blockchain Governance Game (SABGG), which is one of the BGG variants which has been adapted to construct the best strategies to take preliminary actions based on strategic alliance for protecting smart drones in a blockchain-based swarm network. Smart drones are artificial intelligence (AI)-enabled drones which are capable of being operated autonomously without having any command center. Analytically tractable solutions from the SABGG allow us to estimate the moments of taking preliminary actions by delivering the optimal accountability of drones for preventing attacks. This advanced secured swarm network within AI-enabled drones is designed by adapting the SABGG model. This research helps users to develop a new network-architecture-level security of a smart drone swarm which is based on a decentralized network.

Keywords: drone swarm; cybersecurity; Blockchain Governance Game; strategic alliance; artificial intelligence; network architecture; Internet of Things; fluctuation theory; 51 percent attack

MSC: 60C55; 60K10; 90B15; 90B50; 91A35; 91A55; 93A30

Citation: Kim, S.-K. Advanced Drone Swarm Security by Using Blockchain Governance Game. *Mathematics* **2022**, *10*, 3338. https://doi.org/10.3390/math10183338

Academic Editor: Tuan Phung-Duc

Received: 12 August 2022
Accepted: 12 September 2022
Published: 15 September 2022

1. Introduction

Drones have been widely adapted for both the military and civilians in ways which include criminal investigations, public safety, and surveillance forces [1]. Their dynamic mobility, rapid reaction, and simple development offer new possibilities for applications with affordable expenses [2–4]. A drone swarm is a coordination of multiple drones in which they communicate to make decisions for collective actions. One application of a drone swarm is a drone light show (see Figure 1) and this swarm has a central computer on the ground that tracks all individual drones and controls their swarm behavior [5].

(*Source: 2015 Magiya Fauna Light Show; Photo by Preetam Choudhury*)

Figure 1. Drone swarm art performance [5].

The movements of the drones are designed to achieve a flight path that minimizes the collisions between the individual ones, while each individual drone does not contribute in the decision making process [6]. Drone swarms are beneficial for performing casualty strikes. Particularly in a militarized drone swarm, a smart drone swarm which is intelligently guided without a control center has been considered as a single or integrated AI-enabled weapon system [7]. One example is the 103 Perdix drones launched by the Department of Defense (DOD) in 2016 [8]. The smart drones are operated by using artificial intelligence (AI) that allows the drones to transform in several formations, fly across sample battlefields, and update configuration factors [7–9]. A drone swarm could be adapted for states lacking nuclear weapons and for assassination weapons. It could be also extended for enhanced delivery systems for biochemical weapons [10].

Cybersecurity threats on drones have increased in sophistication, which increases high risks for attacks in drone communications. Their deployments face huge difficulties and criticism as a result. Recently, the vulnerability of DJI drones has been discovered through drone hijacking [11]. The absence of suitable drone security mechanisms made the attack possible [12]. The security treats in a military environment setting have brought harmful effects that damage classified military information. The session hijacking attack in a drone swarm is one of the examples in which an attacker is enabled to extract previously exchanged information for various malfunctioned activities [3]. For defending these cybersecurity threats, blockchain technology has been adapted after being implemented for cryptocurrency [13,14]. Blockchain-adapted security models have been widely studied [15–19] because blockchain is highly secured by design and exemplifies a federated computing system, although its records are alterable. Hence, blockchain-based network security is even more extended to the drones [20,21].

The recent studies are targeted on the presence of Byzantine robots which allow for logging events in a zero-knowledge proof to analyze the behavior of the robots in the swarm without incurring the risk that some malicious agent has modified them [13,15,16,22,23]. Blockchain-based smart contracts in autonomous robots allow decentralized systems with equally distrusting nodes to agree on the outcome of the programs [15,22]. The Autonomous Robots Go Swarming (ARGoS) model has been developed for analyzing a collective decision scenario in a robot swarm [23]. Despite the presence of Byzantine robots, blockchain technology allows a robot swarm to achieve consensus in a collective decision problem [15,16].

The Blockchain Governance Game (BGG) and its variants have been designed for preventing blockchain-based attacks and keeping the network decentralized [24,25]. The BGG is the stochastic game model to predict the moments of taking preliminary security actions before attacks [24]. One of the BGG variants is the Strategic Alliance for Blockchain Governance Game (SABGG), which introduces a strategic alliance within the nodes [25] instead of keeping the backups [24] (see Figure 2). This innovative game model is applied for improving drone securities in this research. The analytical function of the smart drone swarm network architecture for enhancing securities and protecting from attackers is constructed for avoiding a conventional drone attack in a swarm.

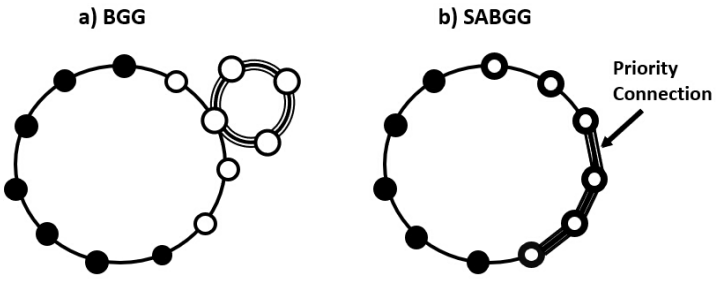

Figure 2. BGG vs. SABGG [24,25].

The advanced blockchain-based secured smart drone swarm network is a decentralized network for drones in a swarm to defend attacks by adapting the SABGG. The SABGG could secure the smart drones in a swarm on the network architecture level. The network architecture level of security by using the BGG has been studied recently [26] and this research is a successor of the BGG application. The objective of this research is adapting an innovative mathematical model into a practical application. It is the first practical BGG adaptation into a military domain, particularly, a network architecture security design in a smart drone swarm communication. The mathematically proven SABGG model provides a guideline for actual implementations into real-world security situations, including healthcare, Internet-of-Things, and federated machine learning, which is one of the main advantages of this research.

The paper begins with the introduction of an advanced blockchain-based secured smart drone swarm network and the smart drone swarm network is described by using the BGG variant in Section 2. This stochastic game model predicts how many blocks are generated and finds the moment of taking the security actions in advance. The mixed strategic game for constructing a cost function of the model is also provided in this section. The optimization of the drone swarm network for a special case is analytically calculated and numerically simulated in Section 3. The paper finally ends with the conclusion in Section 4.

2. Stochastic Game for Smart Drone Network Framework

The Strategic Alliance for Blockchain Governance Game (SABGG) [25] is applied into the drone swarm network architecture to improve the communication network security. Two players are involved in this game. One player is an attacker who intends to fork a private chain and the other player is a defender who honestly mines blocks. The explicit function from the SABGG predicts the moment of one step prior to 51 percent attack [24,25] and this function is applied to a drone swarm network architecture.

2.1. Advanced Blockchain-Based Secured Smart Drone Swarm Network Structure

The advanced secured drone swarm network structure is considered when the drones in a swarm are connected to one another and the swarm is hooked up as a single blockchain network (see Figure 3). Although drones in a swarm are fully connected, they may not be connected with a command center (or a control center). Such a drone swarm could execute their commands artificially and independently despite their disconnection with a command center. Each drone randomly generates unique data (e.g., GPS coordinates or motor RPM values) and broadcasts these data to others. These activities are equivalent with transactions in a typical blockchain network.

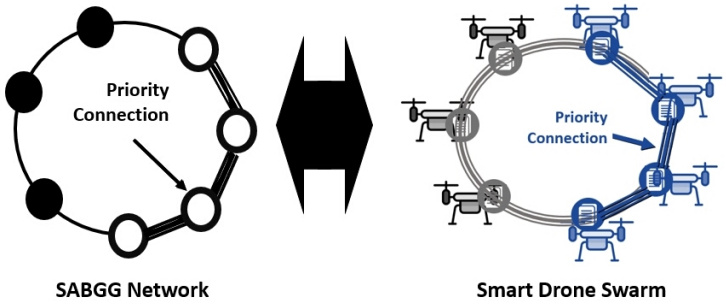

Figure 3. Adapting SABGG for the drone swarm network architecture.

Unlike other conventional blockchain networks, BGG-based networks do not have a reward system but use the verifiable random function (VRF) for generating new blocks [27,28]. Alternatively, the weighted-mean-subsequence-reduced (WMSR) algorithm is designed for achieving resilient consensus in decentralized sensor networks and smart robot swarms [15,29–32]. By applying the VRF or WMSR on the EVM (Ethereum virtual ma-

chine), all drones in the swarm network shall have nearly equal chances to generate the blocks without or only with minimal computational power. The mechanism for protecting a smart drone swarm network is identical to the SABGG. The governance in a swarm network is driven by the decision making parameters that include a prior time before catching more than half of the total drones by an attacker.

2.2. SABGG Models for Advanced Blockchain-Based Secured Smart Drone Swarm Network

The SABGG which is a antagonistic game of two players describes the smart drone swarm network structure. Each player becomes either a defender or an attacker (called "A" and "B"). They compete each other to win the game by building blocks either for true or false ones in a drone swarm network.

The probability space $(\Omega, \mathcal{F}(\Omega), P)$ is considered with independent σ-subalgebras $\mathcal{F}_A$, $\mathcal{F}_B$, $\mathcal{F}_\tau \subseteq \mathcal{F}(\Omega)$. Two processes

$$\mathcal{A} := \sum_{k\geq 0} X_k \varepsilon_{s_k}, \;\; s_0(=0) < s_1 < s_2 < \cdots, \text{a.s.} \tag{1}$$

$$\mathcal{B} := \sum_{j\geq 0} Y_j \varepsilon_{t_j}, \;\; t_0(=0) < t_1 < t_2 < \cdots, \text{a.s.} \tag{2}$$

are $\mathcal{F}_A$-measurable and $\mathcal{F}_B$-measurable marked Poisson processes with respective intensities λ_a and λ_b. These two values λ_a and λ_b represent the computing performance for generating the blocks of an attacker and a defender in a blockchain network. The above processes indicate the actions of players A (an attacker) and B (a defender). An attacker generates blocks with fake transactions and builds blocks of magnitudes X_1, $X_{2,}$ ... formalized by this process. The processes $\mathcal{A}$ and $\mathcal{B}$ are transformed as follows:

$$\mathbb{E}\left[g^{\mathcal{A}(s)}\right] = e^{\lambda_a(s)(g-1)}, \mathbb{E}\left[z^{\mathcal{B}(t)}\right] = e^{\lambda_b(t)(z-1)}. \tag{3}$$

This game system is monitored at random times in accordance with the point process and this observation process is equivalent with the PoW (Proof-of-Work) completion duration in a typical blockchain-based network:

$$\mathcal{T} := \sum_{i\geq 0} \varepsilon_{\tau_i}, \tau_0(>0), \tau_1, \ldots, \tag{4}$$

which is a delayed renewal process, and the formalization of the observation process is as follows:

$$\mathcal{A}_\tau \otimes \mathcal{B}_\tau := \sum_{k\geq 0} (X_k, Y_k)\varepsilon_{\tau_k}, \tag{5}$$

and it is with position-dependent marking and with X_k and Y_k being dependent with the notation:

$$\Delta_k := \tau_k - \tau_{k-1}, \; k = 0, 1, \ldots, \; \tau_{-1} = 0, \tag{6}$$

and

$$\gamma(g, z) = \mathbb{E}\left[g^{X_k} \cdot z^{Y_k}\right], \; g > 0, \; z > 0. \tag{7}$$

By using the double expectation,

$$\gamma(g, z) = \delta(\lambda_a(1-g) + \lambda_b(1-z)), \tag{8}$$

and

$$\gamma_0(g, z) = \mathbb{E}\left[g^{A_0} z^{B_0}\right] = \delta_0(\lambda_A(1-g) + \lambda_b(1-z)), \tag{9}$$

where

$$\delta(\theta) = \mathbb{E}\left[e^{-\theta\Delta_1}\right], \; \delta_0(\theta) = \mathbb{E}\left[e^{-\theta\tau_0}\right], \tag{10}$$

are the magical transforms of increments $\tau_1, \tau_2, \ldots$. The game is ended when the total number of attacked drones $\mathcal{A}_j$ in the swarm network becomes more than the half of the total drones in a swarm by player A or when player B keeps more than the half of the total drones respectively in advance (i.e., $\mathcal{B}_l > \frac{M}{2}$). To further formalize the game, the exit indexes are defined as follows:

$$\nu := min\left\{j : A_j\,(= A_0 + X_1 + \cdots + X_j) \geq \left(\frac{M}{2}\right)\right\}, \tag{11}$$

$$\nu_1 := min\left\{j : A_j\,(= A_0 + X_1 + \cdots + X_j) - C \geq \left(\frac{M}{2}\right)\right\}, \tag{12}$$

$$\mu := min\left\{l : B_l(= B_0 + Y_1 + \cdots + Y_l) \geq \left(\frac{M}{2}\right)\right\}, \tag{13}$$

where C is the random number of available intact nodes (i.e., drones) and the capability of the allies is less than the half of the total nodes in the network system (i.e., $C < \frac{M}{2}$). The defender (player B) might still win the game even without requesting an alliance but the chance of winning is lower than with a strategic alliance because an attacker should govern both the allied nodes and half of the total nodes at t_ν. Contrarily, the attacker (player A) could win the game at τ_μ when it takes the place beforehand. The game is ended at $min\{\nu, \nu_1, \mu\}$ and the σ-subalgebra of the process $(\mathcal{A}, \mathcal{B})$ can be analogously denoted as $\mathcal{F}(\Omega) \cap \{\nu < \nu_1 < \mu\}$. The confined game of player A is targeted. The first passage time τ_ν is the associated exit time from this confined game and formula (5) is modified as

$$\overline{\mathcal{A}_t} \otimes \overline{\mathcal{B}_t} := \sum_{n\geq 0}^{\nu} (X_n, Y_n)\varepsilon_{\Delta_n}, \tag{14}$$

which provides an explicit definition of the basic model observed until t_ν and the joint functional of the swarm network model is as follows:

$$\Phi_{\lceil \frac{M}{2} \rceil}(\xi, g_0, g_1, b, z_0, z_1) = \mathbb{E}\left[\xi^\nu \cdot g_0^{A_{\nu-1}} \cdot g_1^{A_\nu} \cdot b^{A_\nu - C} \cdot z_0^{B_{\mu-1}} \cdot z_1^{B_\mu} \mathbf{1}_{\{\nu<\nu_2<\mu\}}\right], \tag{15}$$

$$\|\xi\| \leq 1, \ \ \|g_0\| \leq 1, \ \ \|g_1\| \leq 1, \ \ \|b\| \leq 1, \|z_0\| \leq 1, \ \ \|z_1\| \leq 1, \tag{16}$$

where M indicates the total number of drones in the swarm network (see Figure 3). The SABGG Theorem establishes an explicit formula $\Phi_{\lceil \frac{M}{2} \rceil}$ from (7)–(10). According to the theorem in [25], the functional $\Phi_{\lceil \frac{M}{2} \rceil}$ of the process of (15) satisfies the following expression:

$$\Phi_{\lceil \frac{M}{2} \rceil}(\xi, g_0, g_1, z_0, z_1) = \mathfrak{D}_{(q,r,s)}^{(\lceil \frac{M}{2} \rceil, \lceil \frac{M}{2} \rceil, \lceil \frac{M}{2} \rceil)} \Lambda, \tag{17}$$

where

$$\Lambda = \sigma \cdot \Gamma\left(\frac{1-\Gamma^1}{1-\Gamma}\right)\left(\gamma_0^1 - \gamma_0 + \frac{\zeta\Theta_0}{1-\zeta\Theta}\left(\gamma^1 - \gamma\right)\right), \tag{18}$$

and

$$\Theta := \gamma(g_0 g_1 bqr, z_0 z_1 s), \tag{19}$$

$$\Theta_0 := \gamma_0(g_0 g_1 bqr, z_0 z_1 s), \tag{20}$$

$$\gamma := \gamma(g_1 bq, z_1), \tag{21}$$

$$\gamma_0 := \gamma_0(g_1 bq, z_1), \tag{22}$$

$$\gamma^1 := \gamma(g_1 b, z_1), \tag{23}$$

$$\gamma_0^1 := \gamma_0(g_1 b, z_1), \tag{24}$$

$$\Gamma := \gamma(br, s), \tag{25}$$

$$\Gamma^1 := \gamma(r, 1), \tag{26}$$

$$\sigma := \mathbb{E}\left[b^{-C}\right]. \tag{27}$$

Additionally, the operator $\mathfrak{D}_{(q,r,s)}^{(a,b,c)}$ in (17) is defined as follows [24,25]:

$$\mathfrak{D}_{(q,r,s)}^{(a,b,c)}(\bullet) = \begin{cases} \left(\frac{1}{a!\cdot b!\cdot c!}\right) & \lim_{(q,r,s)\to 0} \frac{\partial^a \partial^b \partial^c}{\partial q^a \partial r^b \partial s^c} \frac{1}{(1-q)(1-r)(1-s)}(\bullet), \quad a, b, c \geq 0 \\ 0, & \text{otherwise.} \end{cases} \tag{28}$$

and then we can find

$$h(a, b, c) = \mathfrak{D}_{(q,r,s)}^{(a,b,c)}\left[\mathcal{D}_{(a,b,c)}\left\{h(a, b, c)\right\}(q, r, s)\right] \tag{29}$$

where

$$\mathcal{D}_{(a,b,c)}^{(q,r,s)}\left[h(a, b, c)\right] := (1-q)(1-r)(1-s)\left\{\sum_{a\geq 0}\sum_{b\geq 0}\sum_{c\geq 0} h(a, b, c) q^a r^b s^c\right\}, \tag{30}$$

$$||q|| < 1, \;\; ||r|| < 1, \;\; ||s|| < 1. \tag{31}$$

From (11)–(13), we can find the PGFs (probability-generating functions) of the exit index ν:

$$\mathbb{E}[\xi^\nu] = \Phi_{\lceil \frac{M}{2} \rceil}(\xi, 1, 1, 1, 1), \tag{32}$$

and the general decision making parameters are ν, $\tau_{\nu-1}$, A_ν, and $A_{\nu-1}$. Calculating a marginal mean of certain parameters is occasionally more efficient than finding an explicit PGF of each parameter, and the particular decision making parameters of the smart drone swarm network can be found as follows:

$$\mathbb{E}[\tau_{\nu-1}] = \mathbb{E}[\tau_0] + \mathbb{E}[\Delta_1](\mathbb{E}[\nu] - 1), \tag{33}$$

$$\mathbb{E}[A_\nu] = \mathbb{E}[\mathbb{E}[A_\nu|\nu]] = \mathbb{E}[A_0] + \mathbb{E}[\nu - 1]\mathbb{E}[X_k], \tag{34}$$

$$\mathbb{E}[A_\nu - C] = \mathbb{E}[A_\nu] - \mathbb{E}[C], \tag{35}$$

$$\mathbb{E}[A_{\nu-1}] = \mathbb{E}[\mathbb{E}[A_\nu|\nu - 1]] = \mathbb{E}[A_0] + \mathbb{E}[\nu - 2]\mathbb{E}[X_k]. \tag{36}$$

2.3. Mixed Strategy Game Design for SABGG

A two-person mixed strategy game is considered for drone swarm security and player B is a defender who has two strategies at each monitoring moment, one step before an attacker completes generating alternative blocks with false transactions. Player B has the following two strategies: (1) *Regular*—regular operations that the smart drone swarm network is running as usual and (2) *Safety*—the network is operating under the safety mode by executing a preliminary action in the drone swarm. In the view of player A (an attacker), he might either succeed or fail to catch the intact drones. Therefore, the responses of player A would be either *NotBurst* or *Burst*. Let us assume that the cost for maintaining the alliance is c_b, where b is the number of drones in a swarm. If the attacks succeed at generating alternative blocks within the smart drones, the drone network bursts and the whole value of the drone swarm V shall be lost. The drone swarm network might still be busted although the smart drones are fully connected with its allies before catching blocks by an attacker. In this case, the cost shall be counted by all drones in the swarm on top of the alliance costs. The normal form of this game is as follows:

$$\textbf{.Players:} \qquad \boldsymbol{N} = \{\mathbf{A}, \mathbf{B}\}. \tag{37}$$

.Strategy sets: s

$$\boldsymbol{s_a} = \{\text{“}NotBurst\text{”}, \text{“}Burst\text{”}\}.$$

$$\boldsymbol{s_b} = \{\text{“}Regular\text{”}, \text{“}Safety\text{”}\}.$$

The conventional cost matrix at the prior time to burst at $\tau_{\nu-1}$ could be constructed as follows (see Table 1).

Table 1. Cost matrix.

	NotBurst $(\mathbf{1} - q(s_b))$	***Burst*** $(q(s_b))$
Regular	0	V
Safety	c_b	$c_b + V$

Here, $q(s_b)$ is the bursting probability of a swarm network, which depends on the strategic decision of player B:

$$q(s_b) = \begin{cases} \mathbb{E}\left[\mathbf{1}_{\{A_\nu \geq \frac{M}{2}\}}\right], & s_b = \{\text{Regular}\}, \\ \mathbb{E}\left[\mathbf{1}_{\{A_\nu - C \geq \frac{M}{2}\}}\right], & s_b = \{\text{Safety}\}, \end{cases} \tag{38}$$

and the alliance cost should be less than the cost of other strategies. Otherwise, player B spends a greater cost for the strategic alliance than for the cost of a whole drone swarm. Recalling from (38), the probability of bursting a blockchain network becomes a Poisson compound process when the observation process is memoryless (see Section 3.1 in detail):

$$q(s_b) = \begin{cases} \sum_{k > \frac{N}{2}} \mathbb{E}\left[\mathbf{1}_{\{A_\nu = k\}}\right], & s_b = \{\text{Regular}\}, \\ \mathbb{E}\left[\mathbb{E}\left[\sum_{k > \frac{N}{2} + C} \mathbb{E}\left[\mathbf{1}_{\{A_\nu = k\}}\right] \middle| C\right]\right], & s_b = \{\text{Safety}\}, \end{cases} \tag{39}$$

where

$$\mathbb{E}\left[\mathbf{1}_{\{A_\nu = k\}}\right] = \mathbb{E}\left[\mathbb{E}\left[\frac{\lambda_a \tau_\nu}{k!} \cdot e^{-\lambda_a \tau_\nu} \middle| \tau_\nu\right]\right]. \tag{40}$$

Although the total number of the drones in a swarm is determined, the actual alliance C remains uncertain until an alliance request is actually accepted from others. Let us assume that the number of available allied drones C follows the binomial distribution with the average acceptance rate for each strategic ally as ϱ. The Laplace transform of C (i.e., σ from (27)) is as follows:

$$\sigma = \mathbb{E}\left[b^{-C}\right] = \left(\frac{\varrho}{b} - (1 - \varrho)\right)^{\left(\frac{M}{2} - 1\right)}. \tag{41}$$

The number of allies C at the moment $\tau_{\nu-1}$ might be arbitrary with satisfying the number of allies $C \in \left\{0, 1, \ldots, \frac{M}{2} - 1\right\}$. Even though any discrete random variable might be considered for the strategic alliance, the binomial random variable well describes these kinds of binary sum decision making situations. The optimal value ϱ^* is the mean of the Bernoulli distribution when the intact drones are reserved. The rate of accountability ϱ could be defined as follows:

$$\varrho^* = \inf\left\{\eta \geq 0 : \mathfrak{S}_0\left(q^0\right) \geq \mathfrak{S}_1(\varrho)\right\}, \tag{42}$$

where (at the moment $\tau_{\nu-1}$),

$$\mathfrak{S}_0\left(q^0\right) = V \cdot q^0, \tag{43}$$

$$\mathfrak{S}_1(\varrho) = c(\varrho)\left(1 - q^1(\varrho)\right) + (c(\varrho) + V)q^1(\varrho), \tag{44}$$

$$q^0 = \mathbb{E}\left[\mathbf{1}_{\left\{A_\nu \geq \lceil \frac{M}{2} \rceil\right\}}\right],\ q^1_{(n,\rho)} = \mathbb{E}\left[\mathbb{E}\left[\mathbf{1}_{\left\{A_\nu \geq \lceil \frac{M}{2} \rceil + C\right\}}\middle| C\right]\right]. \tag{45}$$

Therefore,

$$\mathfrak{S}(\varrho) = \mathfrak{S}_1 \cdot p_{A_{-1}} + \mathfrak{S}_0 \cdot \left(1 - p_{A_{-1}}\right), \tag{46}$$

where

$$p_{A_{-1}} = \boldsymbol{P}\left\{A_{\nu-1} < \frac{M}{2}\right\} = \sum_{k=0}^{\lfloor \frac{M}{2} \rfloor} \boldsymbol{P}\{A_{\nu-1} = k\}. \tag{47}$$

3. The Optimization Practice for SABGG-Based Drone Security

The drone security optimization practice in a swarm network is introduced in this section. The best strategy for defending a swarm network is providing the strategic alliance to give the least chance for governing drones by an attacker with false command requests. This practice case is targeting to optimize the accountability of the strategic alliance with other drones in a swarm.

3.1. Special Case for Advanced Drone Swarm Security

The memoryless observation process has been adapted into various stochastic models because the model does not need to spend any additional cost for the past information. This special condition is practical particularly for the actual implementations (i.e., numerical simulations or coding) of our theoretical model. It is noted that the operator $\mathfrak{D}$ from (28) is defined on the space of all analytic functions at 0. Under the memoryless condition, we can explicitly find the solutions of q^0 and $p_{A_{-1}}$ and calculate the exit index ν, the decision making moments (i.e., τ_ν and $\tau_{\nu-1}$), and the probabilities of the number of blocks from (17). To construct the cost function of the advanced drone swarm, Formulas (7)–(10) shall be reformulated as follows:

$$\gamma(g,z) = \delta(\lambda_a(1-g) + \lambda_b(1-z)) = \gamma_a(g) \cdot \gamma_b(z), \tag{48}$$

$$\gamma_a(g) = \delta(\lambda_a(1-g)), \tag{49}$$

$$\gamma_b(z) = \delta(\lambda_b(1-z)), \tag{50}$$

and

$$\gamma_0(g,z) = \delta_0(\lambda_a(1-g) + \lambda_b(1-z)) = \gamma^0_a(g) \cdot \gamma^0_b(z), \tag{51}$$

$$\gamma^0_a(g) = \mathbb{E}\left[g^{A_0}\right] = \delta_0(\lambda_a(1-g)), \tag{52}$$

$$\gamma^0_b(z) = \mathbb{E}\left[z^{B_0}\right] = \delta_0(\lambda_b(1-z)), \tag{53}$$

From (20)–(28) and (48)–(53),

$$\Theta = \Theta_a \cdot \Theta_b := \gamma_a(g_0 g_1 bqr) \cdot \gamma_b(z_0 z_1 s), \tag{54}$$

$$\Theta_0 = \Theta^0_a \cdot \Theta^0_b := \gamma_0(g_0 g_1 bqr) \cdot \gamma_0(z_0 z_1 s), \tag{55}$$

$$\gamma = \gamma_a \gamma_b := \gamma_a(g_1 bq)\gamma_b(z_1), \tag{56}$$

$$\gamma_0 = \gamma^0_a \cdot \gamma^0_b := \gamma^0_a(g_1 bq)\gamma^0_b(z_1), \tag{57}$$

$$\gamma^1 := \gamma(g_1 b, z_1) = \gamma_a(g_1 b)\gamma_b(z_1), \tag{58}$$

$$\gamma_0^1 := \gamma_0(g_1 b, z_1) = \gamma_a^0(g_1 b)\alpha_b^0(z_1), \tag{59}$$

$$\Gamma := \gamma(br, s) = \gamma_a(br)\gamma_b(s), \tag{60}$$

$$\Gamma^1 := \gamma(r, 1) = \gamma_a(r) \tag{61}$$

The memoryless observation process allows the process to be exponentially distributed and the functionals from (48)–(61) are transformed as follows:

$$\gamma_a^0(q) = \frac{1}{(1+\widetilde{\alpha_0}\cdot\lambda_a) - \widetilde{\alpha_0}\cdot\lambda_a q} = \frac{b_a^0}{1 - a_a^0 \cdot q}, \tag{62}$$

$$\gamma_a(q) = \frac{1}{(1+\widetilde{\alpha}\cdot\lambda_a) - \widetilde{\alpha}\cdot\lambda_a q} = \frac{b_a}{1 - a_a \cdot q}, \tag{63}$$

$$\gamma_b^0(s) = \frac{1}{(1+\widetilde{\alpha_0}\cdot\lambda_b) - \widetilde{\alpha_0}\cdot\lambda_b s} = \frac{b_b^0}{1 - a_b^0 \cdot s}, \tag{64}$$

$$\gamma_b(s) = \frac{1}{(1+\widetilde{\alpha}\cdot\lambda_b) - \widetilde{\alpha}\cdot\lambda_b s} = \frac{b_b}{1 - a_b \cdot s}, \tag{65}$$

$$b_a^0 = \frac{1}{(1+\widetilde{\gamma_0}\cdot\lambda_a)}, a_a^0 = \frac{\widetilde{\gamma_0}\cdot\lambda_a}{(1+\widetilde{\gamma_0}\cdot\lambda_a)}, \tag{66}$$

$$b_a = \frac{1}{(1+\widetilde{\gamma}\cdot\lambda_a)}, a_a^0 = \frac{\widetilde{\gamma}\cdot\lambda_a}{(1+\widetilde{\gamma}\cdot\lambda_a)}, \tag{67}$$

$$b_b^0 = \frac{1}{(1+\widetilde{\gamma_0}\cdot\lambda_b)}, a_b^0 = \frac{\widetilde{\gamma_0}\cdot\lambda_b}{(1+\widetilde{\gamma_0}\cdot\lambda_b)}, \tag{68}$$

$$b_b = \frac{1}{(1+\widetilde{\gamma}\cdot\lambda_b)}, a_b^0 = \frac{\widetilde{\gamma}\cdot\lambda_b}{(1+\widetilde{\gamma}\cdot\lambda_b)}, \tag{69}$$

$$\widetilde{\gamma_0} = \mathbb{E}[\tau_0], \widetilde{\gamma} = \mathbb{E}[\Delta_k]. \tag{70}$$

The first exceed level (also called the exit index) is the most vital factor that can be fully analyzed at first [33]. Other decision making parameters, including the marginal mean of $\tau_{\nu-1}$, A_ν, and $A_{\nu-1}$, can be easily calculated once the exit index is explicitly solved. From (32) and (54)–(61), the functional of the exit index is as follows:

$$\mathbb{E}[\xi^\nu] = \Phi_{\lceil \frac{M}{2} \rceil}(\xi, 1, 1, 1, 1) = R^1 + R^2 - R^3, \tag{71}$$

where

$$\begin{aligned} R^1 \quad &= \left\{\tfrac{a_b b_a b_b}{1-b_a b_b}\right\}\left\{\Xi_{\frac{N}{2}}(0)\right\}\left(1 - b_a + b_a\left(\textstyle\sum_{l\geq 0}^{\frac{M}{2}}\left(a_a^0\right)^l\right)\right) \\ &\quad -\left\{\Xi_{\frac{N}{2}}(0)\right\}\left(\tfrac{b_a^0}{a_b}\right)\left\{\textstyle\sum_{k\geq 0}^{\frac{M}{2}}\left(1 + \left(\tfrac{a_b b_a b_b}{1-b_a b_b}\right)\right)^{k+1}\right\}\left(\textstyle\sum_{l\geq 0}^{\frac{M}{2}}\left(a_a^0\right)^l\right), \end{aligned} \tag{72}$$

$$\begin{aligned} R^2 \quad &= \left(\tfrac{\xi b_a^0 b_a b_b^0}{a_a^0\{1-b_a b_b - a_a\} - a_b(a_a+1)}\right) \\ &\quad \cdot \textstyle\sum_{l\geq 0}\left[\left(a_b^0\right)^l\left\{\Xi^{\frac{M}{2}-l}(\xi) - (a_b)\Xi^{\frac{M}{2}-l-1}(\xi)\right\}\left\{\textstyle\sum_{j=l}\binom{j}{l}\left(\tfrac{1}{a_b^0}\right)^{j+1}\right\}\right], \end{aligned} \tag{73}$$

$$
\begin{aligned}
R^3 &= \left\{ \frac{\xi b_a \left(b_a^0\right)^2}{\{1-b_a b_b - a_a\} - a_b(a_a+1)} \right\} \\
&\cdot \sum_{j\geq 0} \binom{k}{j} \left(a_b^0\right)^{j-1} \left\{ \sum_{k\geq j} \left(\frac{a_a^0}{a_b^0} \right)^k \right\} \left[\Xi^{\frac{M}{2}-j}(\xi) - (a_b)\Xi^{\frac{M}{2}-j-1}(\xi) \right] \\
&+ \left(\frac{\xi b_a^0 b_b^0 (b_a)^2}{a_a^0(1-a_a)} \right) \left(\frac{1}{(1-b_a b_b) - a_a - a_b(1-a_a)} \right) \\
&\cdot \sum_{j\geq 0} \binom{k}{j} \left(a_b^0\right)^j \sum_{k\geq j} \left(\frac{a_a^0}{a_b^0} \right)^{k+1} \left[\Xi^{\frac{M}{2}-j}(\xi) - (a_b)^k \Xi^{\frac{M}{2}-j-1}(\xi) \right],
\end{aligned} \tag{74}
$$

and

$$
\Xi_j(0) = \Xi_{\frac{M}{2}}(0) = \left\{ \sum_{u=0}^{m} \left(\frac{\left(\frac{M}{2}\right)!}{\left(\left(\frac{M}{2}\right) - u\right)!} \right) \prod_{j=1}^{\left(\frac{M}{2}\right)} \left(\frac{j!}{1 - \left(\frac{a_a}{1-b_a b_b} \right)} \right) \right\}, \tag{75}
$$

$$
\Xi^m(\xi) := \Xi^m(\xi) = \left\{ \sum_{u=0}^{m} \left(\frac{m!}{(m-u)!} \right) \prod_{l=1}^{m} \left(\frac{l!}{1 - \left(\frac{a_b^0}{1-\xi b_a^0 b_b^0} \right)} \right) \right\}. \tag{76}
$$

The explicit solution of the first exceed index under the memoryless observation from (71)–(76) is exactly the same as the solution of the SABGG, and the functional has been analytically proven by Kim [25].

3.2. Linear Programming Practice

A drone swarm network security is considered for a linear programming (LP) practice. The strategy direction for protecting the drone swarm is for priority connection with neighbor drones to lower the change in an attacker catching blocks with false control requests. This case deals with the drone swarm which consists of 20 AI-enabled drones with the estimated cost of USD 1500 per each drone (see Table 2). It is noted that all values shown in the table shall be only for demonstration purposes and these numeric values shall be invested further for the real cases.

Table 2. Initial conditions for the smart drone swarm optimization.

Name	Value	Description
M	20 (drones)	Total number of the nodes in the drone swarm
V	$1500 \cdot M$ (USD)	Total value of a blockchain-enabled drone
$c(\varrho)$	$= 3 \cdot \left(\frac{M}{2} - 1 \right) \cdot \varrho$ (USD)	Cost for reserving nodes to avoid attacks per each car
$\mathbb{E}[\nu]$	3 (trial)	Average number of the observation until the attacker governs the smart drone swarm
C	$\leq \frac{M}{2}$ (drone)	Random number of accepted drones at $\tau_{\nu-1}$

Since the SABGG-based advanced drone swarm network has been analytically solved, calculating the values for the cost function and the required probability distributions is straightforward. The LP model based on the above conditions is as follows from (43)–(46):

Objective:

$$
MinG = \mathfrak{S}(\varrho) \tag{77}
$$

Subject to:

$$\varrho \leq \frac{V \cdot q^0}{c_\varrho \cdot \left(\frac{M}{2} - 1\right)}. \tag{78}$$

From (46), the total cost function $\mathfrak{S}(\varrho)$ is as follows:

$$\mathfrak{S}(\varrho) = \left(c(\varrho)\left(1 - q_\eta^1\right) + (c(\varrho) + V)q^1(\varrho)\right)p_{A_{-1}} + V \cdot q^0\left(1 - p_{A_{-1}}\right) \tag{79}$$

where

$$p_{A_{-1}} = \sum_{k=0}^{\left\{\frac{M}{2} - \lambda_a\widetilde{\delta}\right\}} \left(\frac{\left\{\lambda_a\left(\widetilde{\delta_0} + \mathbb{E}[\nu - 1]\widetilde{\delta}\right)\right\}^k}{k!} \cdot e^{-\lambda_a\left(\widetilde{\delta_0} + \mathbb{E}[\nu-1]\widetilde{\delta}\right)} \right), \tag{80}$$

$$q^0 \simeq 1 - \sum_{k=0}^{\frac{M}{2}} \left(\frac{\{\lambda_a(\widetilde{\gamma_0} + \mathbb{E}[\nu - 1]\widetilde{\gamma})\}^k}{k!} \cdot e^{-\lambda_a(\widetilde{\gamma_0} + \mathbb{E}[\nu-1]\widetilde{\gamma})} \right) \tag{81}$$

$$q^1(\varrho) = \sum_{j=0}^{\frac{M}{2}-1} \sum_{\left\{k \geq \frac{M}{2} + j\right\}} \left(\frac{\lambda_a\left(\widetilde{\delta_0} + \mathbb{E}[\nu - 1]\widetilde{\delta}\right)}{k!} \cdot e^{-\lambda_a\left(\widetilde{\delta_0} + \mathbb{E}[\nu-1]\widetilde{\delta}\right)} \right) P_j, \tag{82}$$

$$P_j = \binom{\frac{M}{2} - 1}{j} \varrho^j (1 - \varrho)^{\frac{M}{2} - 1 - j}. \tag{83}$$

The total cost $\mathfrak{S}(\varrho)$ shall be minimized by the given ϱ, which is the optimal value of the alliance accountability (i.e., the acceptance rate). The illustration in Figure 4 visualizes an optimal result by using the SABGG-based drone swarm network with the given conditions. According to the initial setup in Table 2, the minimal cost for executing one smart drone swarm operation is USD 876 when the drone swarm keeps a 48 percent acceptance rate (i.e., $\varrho^* = 0.48$) for the alliance request to other drones. The moment of alliance request $\tau_{\nu-1}$ shall be one step prior to the time when an attacker catches more than half of the total drones. This practice is targeted only for visualizing cost optimization, not for simulating smart drone management. It is noted that the visualization (see Figure 4) is aimed for showing how the SABGG closed forms (e.g., (17) and (78)) could be applied for analyzing the operation costs of a secured smart drone swarm.

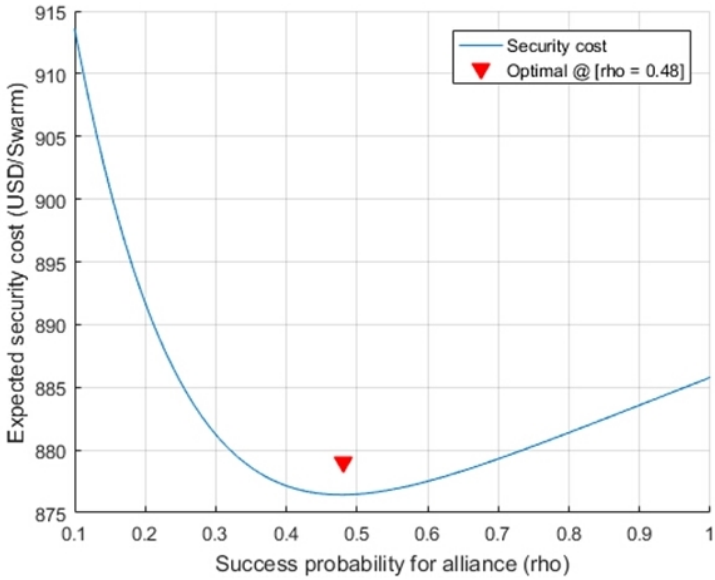

Figure 4. Optimization practice for the drone swarm security.

4. Conclusions

An advanced secure drone swarm network architecture protects a drone swarm from an attacker by adapting a blockchain governance game variant. The Strategic Alliance for Blockchain Governance Game (SABGG) which is an analytically proven game model has been applied as a blockchain governance game variant. The SABGG has been adapted for a decentralized network to improve drone swarm security. The special SABGG case demonstrates how the theoretical model is actually implemented for smart drone security. Although this research is still theoretical and there are several steps remaining for actual implementation into real drones, the practical case demonstrates how an SABGG network could be implemented for smart drone securities and its feasibility. This paper is the first piece of research that applies an SABGG model into a swarm network architecture security. The advanced smart drone swarm network is the successor of blockchain-governance-game-based IoT security applications, particularly in the intelligent military domain. The managerial aspects and actual implementations of smart drone operations could be the next step. Additionally, expending the domains for applying the BGG and its variants could definitely be another direction of future research.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: There are no available data to be stated.

Acknowledgments: Special thanks to the guest editor Tuan Phung-Duc who has guided the author to submit the proper topic of the journal. In addition, thanks to Jongwhi Kim who gives her meritorious efforts for proofreading and English editing on a voluntary basis.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Vergouw, B.; Nagel, H.; Bondt, G.; Custers, B. Drone technology: Types, payloads, applications, frequency spectrum issues and future developments. In *The future of Drone Use*; TMC Asser Press: Hague, The Netherlands, 2016; pp. 21–45.
2. Shahmoradi, J.; Talebi, E.; Roghanchi, P.; Hassanalian, M. A Comprehensive Review of Applications of Drone Technology in the Mining Industry. *Drones* **2020**, *4*, 34. [CrossRef]
3. Ko, Y.; Kim, J.; Duguma, D.G.; Astillo, P.V.; You, I.; Pau, G. Drone Secure Communication Protocol for Future Sensitive Applications in Military Zone. *Sensors* **2021**, *21*, 2057. [CrossRef] [PubMed]
4. Sliusar, N.; Filkin, T.; Huber-Humer, M.; Ritzkowski, M. Drone technology in municipal solid waste management and landfilling: A comprehensive review. *Waste Manag.* **2022**, *139*, 1–16. [CrossRef] [PubMed]
5. Intel. Drone100 Performed by Ars Electronica Futurelab. 2015. Available online: https://inteldronelightshows.com/ (accessed on 1 December 2021).
6. Hambling, D. What Are Drone Swarms and Why Does Every Military Suddenly Want One? *Forbes*. **2021**. Available online: https://www.forbes.com/sites/davidhambling/2021/03/01/what-are-drone-swarms-and-why-does-everyone-suddenly-want-one/ (accessed on 1 December 2021).
7. Kallenborn, Z. Israel's Drone Swarm over Gaza Should Worry Everyone. 2021. Available online: https://www.defenseone.com/ideas/2021/07/israels-drone-swarm-over-gaza-should-worry-everyone/183156/ (accessed on 1 December 2021).
8. DOD; Department of Defense Announces Successful Micro-Drone Demonstration. US Dep. of Defense. 9 January 2017. Available online: https://www.defense.gov/ (accessed on 1 December 2021).
9. Naqvi, S.A.; Hassan, S.A.; Pervaiz, H.; Ni, Q. Drone-Aided communication as a key enabler for 5G and resilient public safety networks. *IEEE Commun. Mag.* **2018**, *56*, 36–42. [CrossRef]
10. Kallenborn, Z. Meet the future weapon of mass destruction, the drone swarm. *Bull. At. Sci.* **2021**. Available online: https://thebulletin.org/2021/04/meet-the-future-weapon-of-mass-destruction-the-drone-swarm/ (accessed on 1 December 2021).
11. Choudhary, G.; Sharma, V.; You, I. Sustainable and secure trajectories for the military Internet of Drones (IoD) through an efficient Medium Access Control (MAC) protocol. *Comput. Electr. Eng.* **2019**, *74*, 59–73. [CrossRef]
12. He, D.; Chan, S.; Guizani, M. Communication security of unmanned aerial vehicles. *IEEE Wirel. Commun.* **2016**, *24*, 134–139. [CrossRef]
13. Nakamoto, S. Bitcoin: A Peer-to-Peer Electronic Cash System. 2008. Available online: https://bitcoin.org/bitcoin.pdf (accessed on 1 December 2021).

14. Wood, G. Ethereum: A Secure Decentralised Generalised Transaction Ledger. Ethereum Project Yellow Paper. 2014. Available online: http://gavwood.com/paper.pdf (accessed on 1 December 2021).
15. Strobel, V.; Ferrer, E.C.; Dorigo, M. Managing Byzantine Robots via Blockchain Technology in a Swarm Robotics Collective Decision Making Scenario. In Proceedings of the 17th International Conference on Autonomous Agents and Multi Agent Systems, Stockholm, Sweden, 10–15 July 2018; pp. 541–549.
16. Strobel, V.; Ferrer, E.C.; Dorigo, M. Blockchain Technology Secures Robot Swarms: A Comparison of Consensus Protocols and Their Resilience to Byzantine Robots. *Front. Robot. AI*. **2020**, *7*, 54. [CrossRef] [PubMed]
17. Millard, A.G.; Timmis, J.; Winfield, A.F.T. Towards Exogenous Fault Detection in Swarm Robotic Systems. In Proceedings of the 14th Annual Conference, TAROS 2013, Oxford, UK, 28–30 August 2013; pp. 429–430.
18. Restuccia, F. Blockchain for the Internet of Things: Present and Future. *arXiv* **2019**, arXiv:1903.07448.
19. Jesus, E.F.; Chicarino, V.R.L.; de Albuquerque, C.V.N.; Rocha, A.A.D.A. Survey of How to Use Blockchain to Secure Internet of Things and the Stalker Attack. *Secur. Commun. Netw.* **2018**, *2018*, 9675050. [CrossRef]
20. Wazid, M.; Bera, B.; Mitra, A.; Das, A.K.; Ali, R. Private blockchain-envisioned security framework for AI-enabled IoT-based drone-aided healthcare services. In Proceedings of the 2020 DroneCom, London, UK, 25 September 2020; pp. 37–42.
21. Cheema, M.A.; Shehzad, M.K.; Qureshi, H.K.; Hassan, S.A.; Jung, H. A Drone-Aided Blockchain-Based Smart Vehicular Network. *IEEE Trans. Intel. Trans. Sys.* **2021**, *22*, 4160–4170. [CrossRef]
22. Buterin, V. A Next-Generation Smart Contract and Decentralized Application Platform. Ethereum Project White Paper. 2014. Available online: https://github.com/ethereum/wiki/wiki/White-Paper (accessed on 1 December 2021).
23. Pinciroli, C.; Trianni, V.; O'Grady, R.; Pini, G.; Brutschy, A.; Brambilla, M.; Mathews, N.; Ferrante, E.; Di Caro, G.; Ducatelle, F.; et al. ARGoS: A modular, parallel, multi-engine simulator for multi-robot systems. *Swarm Intell.* **2012**, *6*, 271–295. [CrossRef]
24. Kim, S.-K. Blockchain Governance Game. *Comput. Ind. Eng.* **2019**, *136*, 373–380. [CrossRef]
25. Kim, S.-K. Strategic Alliance for Blockchain Governance Game. *Probab. Eng. Inf. Sci.* **2020**, *36*, 184–200. [CrossRef]
26. Kim, S.-K. Enhanced IoV Security Network by Using Blockchain Governance Game. *Mathematics* **2021**, *9*, 109. [CrossRef]
27. Micali, S.; Rabin, M.O.; Vadhan, S.P. Verifiable random functions. In Proceedings of the 40th IEEE Symposium on Foundations of Computer Science, New York, NY, USA, 17–18 October 1999; pp. 120–130.
28. Goldberg, S.; Reyzin, L.; Papadopoulos, D.; Vcelak, J. Verifiable Random Functions. *IETF* 2022. Available online: https://datatracker.ietf.org/doc/draft-irtf-cfrg-vrf/ (accessed on 1 September 2022).
29. Guerrero-Bonilla, L.; Prorok, A.; Kumar, V. Formations for resilient robot teams. *IEEE Robot. Autom. Lett.* **2017**, *2*, 741–848. [CrossRef]
30. Saldana, D.; Prorok, A.; Sundaram, S.; Campos, M.F.; Kumar, V. Resilient consensus for time-varying networks of dynamic agents. In Proceedings of the American Control Conference, Seattle, WA, USA, 24–26 May 2017; pp. 252–258.
31. LeBlanc, H.J.; Zhang, H.; Koutsoukos, X.; Sundaram, S. Resilient asymptotic consensus in robust networks. *IEEE J. Select. Areas Commun.* **2013**, *31*, 766–781. [CrossRef]
32. Saulnier, K.; Saldana, D.; Prorok, A.; Pappas, G.J.; Kumar, V. Resilien flocking for mobile robot teams. *IEEE Robot. Autom. Lett.* **2017**, *2*, 1039–1046. [CrossRef]
33. Dshalalow, J.H. *First Excess Level Process, Advances in Queueing*; CRC Press: Boca Raton, FL, USA, 1995; pp. 244–261.