cryptography

MDPI

Article

Optimized AKS Primality Testing: A Fluctuation Theory Perspective

Bhupendra Nath Tiwari [1,2,*], Jude Kibinde Kuipo [1], Joshua M. Adeegbe [3] and Ninoslav Marina [3]

[1] Faculty of Computer Science and Engineering, University of Information Science and Technology, St. Paul the Apostle, Partizanska bb, 6000 Ohrid, Republic of Macedonia; kuipo.kibinde.jude@cns.uist.edu.mk

[2] INFN-Laboratori Nazionali di Frascati, Via E. Fermi, 40-I-00044 Rome, Italy

[3] Faculty of Communication Networks and Security, University of Information Science and Technology, St. Paul the Apostle, Partizanska bb, 6000 Ohrid, Republic of Macedonia; joshua.adeegbe@cse.uist.edu.mk (J.M.A.); ninoslav.marina@uist.edu.mk (N.M.)

* Correspondence: bhupendray2.tiwari.phd@iitkalumni.org

Received: 19 December 2018; Accepted: 19 April 2019; Published: 29 April 2019

Abstract: The AKS algorithm is an important breakthrough in showing that primality testing of an integer can be done in polynomial time. In this paper, we study the optimization of its runtime. Namely, given a finite cardinality set of alphabets of a deterministic polynomial runtime Turing machine and the number of strings of an arbitrary input integer whose primality is to be tested as the system parameters, we consider the randomized AKS primality testing function as the objective function. Under randomization of the system parameters, we have shown that there are definite signatures of the local and global instabilities in the AKS algorithm. We observe that instabilities occur at the extreme limits of the parameters. It is worth mentioning that Fermat's little theorem and Chinese remaindering help with the determination of the underlying stability domains. On the other hand, in the realm of the randomization theory, our study offers fluctuation theory structures of the AKS primality testing of an integer through its maximum number of irreducible factors. Finally, our optimization theory analysis anticipates a class of real-world applications for future research and developments, including optimal online security, system optimization and its performance improvements, (de)randomization techniques, and beyond, e.g., polynomial time primality testing, identity testing, machine learning, scientific computing, coding theory, and other stimulating optimization problems in a random environment.

Keywords: primality testing; AKS algorithm; fluctuation theory; system optimization; stability analysis; *P* vs. *NP* problems; Turing machine

1. Introduction

The Agrawal-Kayal-Saxena (AKS) algorithm plays an important role in determining the primality of an integer [1]. In the realm of modern computer science, it finds significance in cryptography, viz. the block ciphers, banking system, and associated online security [2,3]. Namely, from the perspective of the formal theory of computation, an apt design of protocols is achieved via a suitable algorithm to efficiently perform a given computational task [4,5]. The AKS primality testing finds further importance, as well. For example, see [6] concerning its elementary description, correctness, and asymptotic analysis towards the primality testing of an integer. From the inception of the AKS algorithm, the prime factoring problem, presumed to be an *NP*-type problem, has become a *P*-type problem in the realm of the randomization theory. Here, *P* stands for the class of algorithms running in polynomial time and that of *NP* for nondeterministic polynomial runtime problems [7]. In the light of computational complexity [8], it is known that the above *NP* and *P* type problems are polynomial time verifiable and solvable classes, respectively. Before the discovery of the AKS algorithm, there were numerous questions about testing of an integer on a particular machine that executes the algorithm in the shortest time.

Time is considered in a discrete dynamical sense, i.e., the number of steps required to achieve the final state of an algorithm. On the other hand, there have been various attempts to determine the equivalence between *P*- and *NP*-type complexity classes [9]. Indeed, the complexity classification finds discrete mathematical perspectives in connection with satisfiability problems, Boolean constraint circuits, and weighed optimization problems [10]. Furthermore, the AKS algorithm finds an important role in computer science via the randomization hypothesis [11]. There are various consequences of primality testing in counting problems, complexity theory, and cryptography [12–14]. In the realm of probabilistic models, the underlying goal of testing the primality of an integer is achieved via Fermat's little theorem and the Chinese remainder theorem; see [15] for aspects of the mathematical background such as the modular equivalence, encryption, decryption, cryptographic protocol, digital signature and designing of safety measures, and related security issues towards safe multi-party computation. The AKS algorithm is an important breakthrough in showing that primality testing can be done in polynomial time. Hence, optimizing runtime for the implementation of this algorithm is an important problem. As per the above consideration, it is worth mentioning that the mathematics involved appears to be interesting, see Section 2 for an overview. The respective results are discussed qualitatively and quantitatively in Sections 3 and 4.

The aim of this paper is to study the case of the corresponding ring of integers when it is prolonged to the field of real numbers. The AKS algorithm is used for testing the integer primality that follows via the fundamentals of number theory on the optimization of AKS primality testing in the light of an absolute deterministic polynomial runtime randomized algorithm that can help with the determination of the prime factors of an arbitrary integer. Given the algorithm, we optimize the AKS primality testing of an arbitrary integer by determining its stability structures. In particular, in order to test the optimize the AKS algorithm, we concentrate on the case when the corresponding ring of integers is prolonged to the field of real numbers. Under the randomization hypothesis of Agrawal and Biswas [11], it is worth emphasizing that the aforementioned AKS algorithm [1] brings the primality testing of an arbitrary integer into the category of *P*-type problems [8]. The AKS primality testing receives distinct roles in both (i) pure mathematics, e.g., probabilistic algorithms [11] and (ii) its applications to modern cryptography—see [15] for an introduction to information security, commitments, and Oblivious transfer functions. Generically, the role of the AKS algorithm in testing the primality of an integer follows the fundamentals of number theory [16], which is particularly rooted in the domain of the algorithmic number theory [17].

We focus on the optimization of AKS primality testing in the light of an absolute deterministic polynomial runtime randomized algorithm that can help one to know whether a given integer is prime or not. In other words, we concentrate on the determination of the prime factors of an arbitrary integer. In particular, for the case of large integers, the realization of such a goal is obtained on a specific Turing machine, whereby the cardinality of the set of its alphabets plays a vital role in computing the corresponding primality; see [18] concerning the fundamentals of the computable functions, (un)decidability, and unsolvable problems. In this case, we concentrate on a given integer whose primality is to be tested on a definite machine of finite cardinality. In order to optimize the AKS primality testing, we apply the notion of the fluctuation theory [19] by randomizing the system parameters, viz. (i) the input integer whose primality is to be tested as a string and (ii) the cardinality of the set of alphabets of the machine.

We find optimal domains of the AKS primality testing of an arbitrary integer with the inverse of the number of steps required to execute the algorithm as the order of the uncertainty. As the input integer tends to a large value, the determination of its primality becomes accurate. Our proposal generically enables one to address the issue of the stability of the AKS primality testing algorithm.

This is realized by considering the maximum value of the cardinality of the group of alphabets as the objective function. Note that such a function arises as a map from the space constituted by the input integer and the size of the group of residues from the set of all numbers that are introspective [1] to the polynomials in a set of the field of real numbers. Namely, when both the input integer and machine parameter are allowed to vary, the fluctuation theory analysis [19] offers an apt platform to optimize the AKS primality testing function as a real-valued map from the space of algorithm parameters to the field of real numbers. Following the notion of the intrinsic geometry and embedding theory [20], we determine the critical points, stability domains, and its correlation with a chosen machine performing the AKS algorithm. We give its global stability structures in the proximity of the critical points of the AKS primality testing function. In this setup, we compute fluctuation vectors in order to determine an intrinsic length of the AKS primality testing of the input integer string and machine parameters.

Following the above notion, we determine the regions where the ASK algorithm remains stable or unstable. Considering the fact that it is among the best primality testing methods of an arbitrary integer, we examine how the AKS algorithm behaves under variations of the input integer and the machine parameters that are used in its primality testing. Next, we tested its stability at the asymptotic limits. The qualitative discussion also shows that instabilities arise only in extreme regions, namely, near the initiation or halting stages of the algorithm. This supports the claim of AKS [1] that the AKS algorithm becomes almost exact for large integers. Indeed, there are various models that could be interesting to investigate further in the light of primality testing and optimization theory. See Section 2 for an overview.

At this juncture, the framework of randomization theory turns outs to be promising because of its probabilistic nature. We have considered the optimal Cunningham factorization of an arbitrary integer in the light of randomization theory [21]. To that end, we provide a brief account of previously known primality testing algorithms such as Fermat's little theorem, the Miller-Rabin algorithm, the Solovay-Strassen models, and others in the Appendix A. Following the assessment of the AKS primality testing and the above models, a natural research direction would be to compare the stability domains of the AKS testing with the above models. Bearing in mind that the AKS algorithm is the most efficient primality test of an integer to date, we focus on examining its optimality properties under fluctuations of the system parameters.

Following the above models and related fundamentals, we summarize the highlights of our analysis as follows. Our contribution is to classify the parameter space regions of the ASK primality testing algorithm, where it remains optimal. Namely, we focus on the optimization of the AKS primality testing of an arbitrary integer. Therefore, we examine the behavior of the peaks concerning the AKS primality testing on an input integer. This is realized by executing a qualitative analysis of the fluctuation quantities in the space of parameters in their specified ranges. For this, we use the asymptotic properties of the AKS algorithm. Interestingly, we observe that the qualitative behavior of the fluctuations remains the same when the input ranges are increased by 10 times. Following the AKS scaling structures, we anticipate that the asymptotic behavior of peaks holds with the incorporation of fluctuations in the model parameters. This offers enlightening insight in the areas of stability analysis, fluctuation theory, randomization theories, integer primality testing and identity testing.

The rest of the paper is organized as follows. In Section 2, we give a brief evolution of the AKS algorithm and its relation to Fermat's little theorem, AKS primality testing, and the randomization hypothesis in the light of ring theories and their localizations. In Section 3, we examine the stability structures of the AKS primality testing by randomizing the input integer string and cardinality of the set of alphabets of the machine executing the algorithm. In Section 4, we provide a qualitative discussion of the results and their implications towards the optimization of the AKS algorithm. Finally, in Section 5, we conclude the paper with prospective directions for future research and developments.

2. Randomized AKS Primality Testing

In this section, we offer a brief review of the AKS algorithm in light of Fermat's little theorem and Chinese remainder theorem in the light of randomization theory [21], also see [22] for basics of the subject matter under the consideration. From the perspective of the P versus NP problem [7], we illustrate how the AKS primality testing falls in the domain of P-type problems. Concerning the geometric and algebraic perspectives, see [20] for embedding, submersion and convexity theory, and [23] for the related commutative algebra background such as modules, fields, and rings.

First of all, in order to determine whether a given input number is prime or not, an efficient primality testing algorithm requires guaranteeing its polynomial runtime complexity [1]. See the Appendix A.1 for a brief evolution of primality testing algorithms. In contrast to the probabilistic primality testing, the AKS primality testing [1] of an integer is based on a cyclotomic generalization of the Fermat's little theorem over a finite ring. This offers both a deterministic and polynomial time complexity of a reduced order. In particular, as highlighted in the Appendix A.2, the AKS algorithm essentially overcomes the exponential runtime complexity of Fermat's little theorem by comparing the coefficients of a polynomial $(x+a)^n$ with modular operation in terms of another polynomial of the form $(x^n + a)$. This settles a long-standing problem in primality testing: whether it falls in the domain of P-type or NP-type problems.

Following the above breakthrough, the deterministic characteristic of the AKS primality testing algorithm is brought down to the table by the value of r, which is bounded in polynomial runtime of its order $logn$, as in the Algorithm 1 below. The associated details are relegated to the Appendix A.2. At this point, it is worth mentioning that the AKS primality testing [1] satisfies all four requirements for an efficient prime testing algorithm in comparison to the previously mentioned algorithms. As a matter of fact, it possesses a reduced polynomial time complexity compared to the associated primality testing algorithms as in Appendix A.1. There, whereby we have summarized the associated concepts such as the computational complexity and its role in the analytic number theory [24] and others such as the Miller-Rabin test [25] and Solovay-Strassen primality testing [26] of an integer in the light of the algorithmic number theory.

This accounts for the wide usage of the AKS primality testing algorithm in the realm of applied cryptography and related subject matters. Indeed, there have been various reductions in the time complexity of the cyclotomic AKS primality testing of an integer.

In this paper, we offer the undermining experimental, mathematical, and computational perspectives. Namely, we focus on the optimization of the AKS algorithm in order to determine the regions of its input parameters that yield the optimal primality testing of an arbitrary integer. In short, the AKS primality testing [1] of an integer can be summarized as per the below Algorithm 1.

Algorithm 1: The AKS algorithm (the AKS primality testing [1] of an integer)

```
1- An integer $1 < n \in \mathbb{N}$ is said to be a composite number if there exists a pair $(a, b)$ such that $n = a^b$ for
   some $a \in \mathbb{N}$ and $b > 1$.
2- Given a triple $(a, b, r)$ with $gcd(a, r) = 1$, find the smallest $r$ such that $a^b = 1(mod\ r)$ holds. Then, the
   order $o_r(n)$ of $a$ modulo $r$ must satisfy the inequality $o_r(n) > log^2 n$.
3- For an integer $n$ with its factor $a \leq r$, $n$ is said to be composite if $1 < \gcd(a, n) < n$.
4- The input integer $n$ returns a prime if we have $n \leq r$.
5- For $a = 1, 2, \ldots, l$, an integer $n$ is said to be composite, if the Equation (A3) as in Appendix A.2 is not
   satisfied over $(mod\ X^r - 1,\ n)$, where $l = \sqrt{\phi(r)} \log n$. Here, $\phi(r)$ denotes the Euler totient function,
   which counts the relatively prime numbers less than $r$.
6- Otherwise the input integer $n$ is a prime.
```

It is worth mentioning that the AKS algorithm [1] arises via the randomization of Fermat's little theorem; see the Appendix A.2 for an overview. From Algorithm 1, we observe that if the input integer n returns the algorithm in steps 1 and 3, it is a composite number. On the other hand, the input integer

n turns out to be a prime if the algorithm returns in steps 4 and 6. In determining the primality of an integer via the AKS algorithm, steps 2 and 5 emerge as the key ingredients. Notice that step 2 plays an equally important role in testing the primality of an integer n as it determines a suitable r such that steps 2 and 3 are not satisfied whenever n is a prime. The above primality testing of an integer is examined locally by introducing the notion of a modular function in a given quotient ring with the maximum value of a as l as above in step 5. The proof of the AKS algorithm as depicted above in Algorithm 1 involves the introspective properties of a modular function; see [1] for the concerning details in the light of the algebraic closure, addition, multiplication, and quotient ring in a given basis of cyclotomic polynomials over a finite field.

In general, there are various higher-dimensional extensions of such a ring that consist of finitely many local rings of the form $\mathbb{Z}^n[x]$, where $n \in \mathbb{N}$. On the other hand, the randomization depicted in Algorithm 1 designates an extension from the integral valued inputs to their real counterparts [11]. However, the converse problem, termed de-randomization [1], is anticipated to yield the inverse procedure as a restriction map; see [20] in connection with the embedding and submersion theories over real and complex spaces. It is worth stressing that such a map carried out in n number of steps accompanies an effective error $1/n$. Thus, for a sufficiently large n, it follows that the randomization error concerning the primality testing of an integer turns out to be negligible or less than the precision of the machine.

In the light of the P versus NP problem [7], the primality testing of an integer that was believed to be an NP-type problem for several decades is now reduced to a P-type problem via the AKS algorithm [1]. Furthermore, in the realm of algorithmic number theory and computer science, Cook [7] finds that the SAT problem is NP-complete; see also [27] for parameterized and exact computations in the light of sub-exponential runtime Turing reductions. This leads to a P-type solvable problem, whenever there exist various algorithms that positively answer the equivalence between the complexity classes of P- and NP-type problems [7]. In light of the primality testing of an arbitrary integer, we anticipate that in terms of the computational capacity of an algorithm, whatever is achievable on a modern computer is equally achievable on a Turing machine; see [18] for an original account of the computable numbers. In the next section, we offer an intrinsic stability analysis of the AKS algorithm towards the primality testing of an arbitrary integer on a given Turing machine.

With the aforementioned motivations, in this paper, we study the case of the corresponding ring of integers when it is prolonged to the field of real numbers. Herewith, the AKS algorithm is used in testing the integer primality that follows via the fundamentals of number theory. In the light of an absolute deterministic polynomial runtime randomized algorithm, our analysis relies on the optimization of AKS primality testing that can help on the determination of the prime factors of an arbitrary integer. In particular, we have obtained the algorithm that optimizes the AKS primality testing of an arbitrary integer by determining its stability structures. From the fluctuation theory perspective, an essential rudiment of related works on prime factoring is addressed in Appendix A.1. Following the above insightful background, we focus on the AKS algorithm and its applications in the realm of fluctuation theory. The concerned stability analysis is considered in Section 3.

In Section 4, upon the multivariable analysis that is performed on an arbitrary randomized set of input parameters, we purpose in this paper to study the signatures of quantities concerning the runtime stability. This includes the eigenvalues of the fluctuation matrix that have algorithmic significance towards the global stability of the randomized AKS primality testing of an integer. Namely, Section 4 vividly portrays the same via plotted 3-D graphs of the mathematical findings with reported features. In this regard, we acknowledge the implication of asymptotic analysis, time complexities and runtime complications in such cases as interesting research. We suggest considering the former among future research directions. Pertinently, it is worth mentioning that a study of the randomized AKS algorithm at higher dimensions may equally be discussed; however, due to limited time constraints, we anticipate examining such investigations and their feasibility structures in future research and developments.

3. Fluctuation Theory Perspective

In this section, we explore the stability of the AKS primality testing algorithm for an arbitrary input integer by a deterministic Turing machine having a finite cardinality set of its alphabets. It is known that the AKS primality testing arises as an exponential execution of the algorithm [1]. We consider the maximum number of irreducible factors of an input integer as the objective function. In the setup of a randomized algorithm, we examine the corresponding local and global stability structures by determining the signatures of the heat capacities and determinant of the fluctuation matrix at its critical points. This enables us to classify the domains where we have the optimized AKS primality testing of an integer.

3.1. Stability Analysis

In the above setup, we concentrate to determine an optimized AKS primality testing algorithm with the objective function as the number of the prime factors of an arbitrary integer on a given machine. Namely, by varying the input integer x whose primality is to be tested on a certain Turing machine of finite cardinality y, we examine the efficiency of the AKS algorithm via the randomization hypothesis of Agrawal and Biswas (see [11]). In light of the P versus NP problem, the AKS algorithm [1] arises as the maximum number of possible factors of an arbitrary integer x determined by a deterministic polynomial runtime Turing machine M having cardinality y of the set of its alphabets. Thus, the AKS primality testing of an integer x on the Turing machine M can be viewed as a finite integral valued map from $\mathbb{Z}^2$ to $\mathbb{Z}$ as per the assignment $(x, y) \longmapsto f(x, y)$.

In the limit of randomized input parameters $\{x, y\}$, it follows [1] that the asymptotic AKS primality testing function is simplified as

$$f(x, y) = Ax^{\sqrt{y}} \tag{1}$$

where $A \leq 1$ signifies the efficiency of the algorithm. In a domain of varying (x, y), the qualitative behavior of the AKS function $f(x, y)$ is shown in Figure 1. In order to optimize the AKS primality testing algorithm, we consider the randomized setting [11], whereby x and y vary over the set of real numbers $\mathbb{R}$. We consider the AKS primality testing function as a real valued map $f : \mathbb{R}^2 \longrightarrow \mathbb{R}$ that assigns a given pair of real numbers (x, y) to a real number $f(x, y)$. We perform an optimization analysis of the randomized AKS primality testing function $f(x, y)$ to determine its stability domains. Namely, in the setup of fluctuation theory [19–21], we offer a stability analysis of an arbitrary polynomial factoring towards the primality testing of an integer x by a machine M of the cardinality y of the set of its alphabets. By differentiating $f(x, y)$ as in Equation (1) with respect to x, we find the following flow component

$$\frac{\partial f}{\partial x} = A\sqrt{y}x^{\sqrt{y}-1}. \tag{2}$$

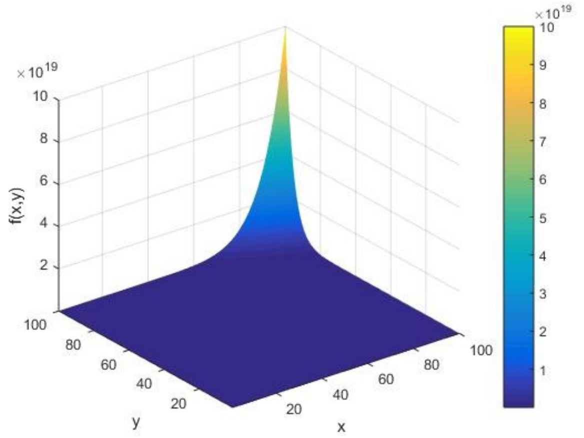

Figure 1. The AKS primality testing function as a function of the input integer x and cardinality y of the set of alphabets of the machine executing the AKS algorithm plotted in the range $x, y \in (1,\ 100)$.

Similarly, under variations of the cardinality y of the set of alphabets of a Turing machine M, the corresponding rate of the AKS primality testing function of the integer x satisfies

$$\frac{\partial f}{\partial y} = \frac{Ax^{\sqrt{y}} lnx}{2\sqrt{y}}. \tag{3}$$

The critical points of $f(x,y)$ are computed as the zeros of the flow equations $\frac{\partial f}{\partial x} = 0$ and $\frac{\partial f}{\partial y} = 0$. In this case, the limiting critical points of $f(x,y)$ arise as the pairs $(1,0)$ and $(0,\infty)$. In order to discuss the nature of the critical points of $f(x,y)$, we need to calculate the heat capacities $\left\{a\frac{\partial^2 f}{\partial x^2}, d\frac{\partial^2 f}{\partial y^2}\right\}$ and the local correlation factor $c\frac{\partial^2 f}{\partial y \partial x}$. The local stability of the AKS primality testing computation is determined by the signature of one of the capacities $\{a, d\}$. Under variations of the input integer x, the heat capacity a as the pure fluctuation component is given by

$$a = A\sqrt{y}\big(\sqrt{y} - 1\big)x^{\sqrt{y}-2}. \tag{4}$$

Similarly, under variations of the cardinality y, the pure fluctuation component d reads as

$$d = \frac{Ax^{\sqrt{y}} lnx}{4y\sqrt{y}}\big(\sqrt{y}\, lnx - 1\big). \tag{5}$$

Moreover, it is not difficult to see that the mixed correlation component c simplifies as

$$c = \frac{Ax^{\sqrt{y}-1}}{2\sqrt{y}}\big(\sqrt{y}\, lnx + 1\big). \tag{6}$$

Herewith, at the critical point $(1,0)$, we see that both the pure fluctuation components $\{a, d\}$ vanish identically while the correlation factor c diverges. Namely, we have the following limiting behavior $a = 0 = d$ and $c \rightarrow \infty$. On the other hand, at the critical point $(0,\infty)$, we find that all the pure and mixed correlation components become ill-defined, i.e., we have an undefined triple $\{a, c, d\}$. It is worth mentioning that the pure correlation components d and a signify the heat capacity of the input integer x whose primality is to be tested on a machine M of the cardinality y of the set of its alphabets. Physically, $\{a, d\}$ can be viewed as factors for investigating the overheating of a given computation state of the machine M while testing the primality of an integer x, whereby we may design an apt cooling system of a Turing machine.

In order to examine the global stability of the AKS algorithm, given the objective function $f(x,y)$ as in Equation (1), we define its fluctuation matrix H as a 2×2 symmetric matrix

$$H = \begin{pmatrix} a & c \\ c & d \end{pmatrix}, \tag{7}$$

where $\{a, d\}$ signify the heat capacities of the system as defined in Equations (4) and (5) respectively. The cross-component c denotes the local correlation of the system, see Equation (6). Substituting the values of $\{a, c, d\}$ from Equations (4)–(6), we have the following Hessian matrix:

$$H = \begin{pmatrix} A\sqrt{y}\big(\sqrt{y} - 1\big)x^{\sqrt{y}-2} & \frac{Ax^{\sqrt{y}-1}}{2\sqrt{y}}\big(\sqrt{y}\, lnx + 1\big) \\ \frac{Ax^{\sqrt{y}-1}}{2\sqrt{y}}\big(\sqrt{y}\, lnx + 1\big) & \frac{Ax^{\sqrt{y}} lnx}{4y\sqrt{y}}\big(lnx\sqrt{y} - 1\big) \end{pmatrix}. \tag{8}$$

In order to achieve a stable domain of the AKS algorithm in testing the primality of a given input integer x, the randomized $f(x,y)$ as the objective function of the optimization problem must yield a positive definite Hessian determinant:

$$\Delta := ad - c^2 > 0. \tag{9}$$

By substituting the values of the heat capacities $\{a, d\}$ from Equations (4) and (5) and the correlation factor c from Equation (6), it follows that the above determinant Δ simplifies as

$$\Delta = -\frac{A^2 x^{2\sqrt{y}-2}}{4y}\Big(\sqrt{y} ln^2 x + \Big(3\sqrt{y} - 1\Big) lnx + 1\Big). \tag{10}$$

To determine the signature of $\Delta(x, y)$, we need to factorize the quadratic polynomial $\delta(x, y)\Delta \sqrt{y} ln^2 x + \Big(3\sqrt{y} - 1\Big) lnx + 1$ as a function of lnx. It follows that the corresponding quadratic equation $\delta(x, y) = 0$ leads to the following roots:

$$lnx = \frac{3\sqrt{y} - 1 \mp \sqrt{9y + 1 - 10\sqrt{y}}}{-2\sqrt{y}}. \tag{11}$$

In Equation (11), let α and β be the respective positive and negative roots in lnx as follows:

$$\alpha := \frac{3\sqrt{y} - 1 + \sqrt{9y + 1 - 10\sqrt{y}}}{-2\sqrt{y}} \tag{12}$$

$$\beta := \frac{3\sqrt{y} - 1 - \sqrt{9y + 1 - 10\sqrt{y}}}{-2\sqrt{y}}. \tag{13}$$

Thus, the overall stability of the AKS algorithm depends on the range of the input integer x and the cardinality y of the set of alphabets of the machine. In particular, when lnx lies between α and β, as above in Equations (12) and (13), there is a globally stable AKS algorithm with a positive Hessian determinant $\Delta(x, y)$ for $y > 0$. In the other case, for a negative y corresponding to damped oscillations as the algorithm proceeds, it follows that lnx must either be larger than the root β as in Equation (12) above or less than the root α as above in Equation (13), viz. we have the optimal solution whenever x satisfies the inequality $e^{\alpha} < x < e^{\beta}$ for $y < 0$. Therefore, we can achieve an optimal AKS primality testing algorithm for different values of input parameters (x, y). Namely, our analysis shows that the AKS algorithm renders the optimal prime testing of an arbitrary integer x on a suitable Turing machine M of the cardinality y of the set of its alphabets.

3.2. Limiting Stability Analysis

In this subsection, we provide limiting behaviors of the local and global stability components as the model parameters (x, y) approach the specific critical points $(1, 0)$ and $(0, \infty)$ of $f(x, y)$. In doing so, we examine specific values of the local heat capacities $\{a, d\}$ and correlation c. Furthermore, we see that the global stability component $\Delta(x, y)$ tends to infinity when the point (x, y) approaches the root $(1, 0)$ of the flow Equations (2) and (3). On the other hand, it follows that $\Delta(x, y)$ becomes ill-defined when it is evaluated at the critical point $(0, \infty)$. Thus, for any physical model, we deem such an outcome algorithmically undesirable as the cardinality of the set of alphabets is assumed to be a finite positive number.

In order to examine the limiting behavior, we randomize the roots $(1, 0)$ and $(0, \infty)$ to their corresponding values $(1, \epsilon)$ and (ϵ, N) with $\epsilon \to 0$ and $N \to \infty$. Physically, N could represent the possible size of the hard disk of a given machine and ϵ the step size, which could be the least value of the randomized integer whose primality is to be tested by the AKS algorithm [1]. For the randomized root $(1, \epsilon)$, from Equation (4) we find that the input integer heat capacity a simplifies as

$$a = A\Big(\epsilon - \sqrt{\epsilon}\Big). \tag{14}$$

From Equation (5), we see that the limiting local capacity d corresponding to the cardinality of the set of alphabets of the machine vanishes identically, namely, as we approach the critical point $(1, \epsilon)$, we have $d \rightarrow 0$. Furthermore, from Equation (6), we find that the cross-correlation c solely depends on the efficiency A of the chosen machine M executing the algorithm. Namely, the correlation c satisfies

$$c = \frac{A}{2}. \tag{15}$$

On the other hand, at the critical point $(1, \epsilon)$, we see that the determinant Δ of the Hessian matrix H takes a negative value for all positive ϵ and $A \neq 0$. From Equation (10) it follows that we have

$$\Delta = -\frac{A}{4\epsilon}^{2}. \tag{16}$$

Thus, in the randomized limit, it follows that the system remains stable for a negative value of ϵ. Furthermore, from Equation (16), we find a definite signature of global instabilities in the limit of $\epsilon \rightarrow 0$. Similarly, in the limit of large N, the stability components corresponding to the randomized root (ϵ, N) of the flow Equations (2) and (3) lead to a positive value of the heat capacities $\{a, d\}$. In particular, for a given machine M being used for the primality testing of an integer x, it turns out that the AKS algorithm yields the following local heat capacities:

$$a = A\sqrt{N}\Big(\sqrt{N} - 1\Big)\epsilon^{\sqrt{N}-2} \tag{17}$$

$$d = \frac{A ln\epsilon}{4N\sqrt{N}}\Big(ln\epsilon\sqrt{N} - 1\Big). \tag{18}$$

For a positive value of the determinant Δ, the system remains stable as long as either of the heat capacities $\{a, d\}$ remains positive, that is, we have either $N > 1$ or $N > 1/ln^2\epsilon$. Furthermore, at the limit of $\epsilon \rightarrow 0$, we find that the heat capacity $a \rightarrow 0$ for $N \neq 4$. However, the other heat capacity d becomes ill-defined by entailing fluctuations of large negative amplitudes. This follows from the fact that the logarithm of a small number approaching zero is a large negative number. On the other hand, we find that the corresponding correlation c reads as

$$c = \frac{A}{2\epsilon}\Big(1 - \epsilon^{\sqrt{N}} ln\epsilon\Big). \tag{19}$$

In this case, there exists a positive correlation c whenever the randomized root (ϵ, N) satisfies the inequality $ln\epsilon < \epsilon^{-\sqrt{N}}$. From Equation (19), we see that the correlation c largely modulates as per the first term. This is because the second term $\epsilon^{\sqrt{N}} ln\epsilon$ takes a small negative value in comparison to the first term whenever $\epsilon \rightarrow 0$, viz. the sample parameters N and ϵ satisfy the constraint $-\epsilon^{\sqrt{N}} ln\epsilon \ll 1$. Therefore, we find that the AKS algorithm becomes highly correlated in the limit of $\epsilon \rightarrow 0$. Qualitatively, as a function of ϵ, we observe that the correlation $c(\epsilon)$, as in Equation (19), displays identical behavior under different values of N.

The global stability is determined by the signature of the determinant of the Hessian matrix as in Equation (10). At the critical point (ϵ, N), the value of the pre-factor $A^2\epsilon^{2\sqrt{N}-2}/4N$ of the determinant Δ depends on two competing terms A^2 and $\epsilon^{2\sqrt{N}-2}/4N$. Namely, the factor $\epsilon^{2\sqrt{N}-2}/4N$ takes a small positive value in the limit of $\epsilon \rightarrow 0$ and $N \rightarrow \infty$. Furthermore, the factor $\delta(\epsilon, N)$ can take both the positive and negative values depending upon the values of ϵ and N. For example, for $N = 100$ and $\epsilon = 0.1$, it follows that $\delta(\epsilon, N)$ takes an approximate value of -12. Thus, from Equation (10), we find a positive signature of the determinant Δ. In such cases, the AKS algorithm remains stable while testing the primality of an integer.

3.3. Eigenvalues and Eigenvectors of H

For a given AKS primality testing function $f(x,y)$ with the fluctuation matrix H as in Equation (8), the corresponding global (in)stability can be examined via the relative signature of the eigenvalues of H. Namely, given the heat capacities $\{a,d\}$ as in Equations (4) and (5) and the system correlation factor c as in Equation (6), we evaluate the eigenvalues and associated eigenvectors of H as a function of $\{a,c,d\}$ for the AKS primality testing function $f(x,y)$ as in Equation (1) of an arbitrary integer x on a machine M of the cardinality y of the set of its alphabets.

3.3.1. Evaluation of Eigenvalues

As per the aforementioned representation of the Hessian matrix H as in Equation (7), there exists an eigenvalue λ if we have a non-null vector $v \in \mathbb{R}^2$ satisfying the following eigenvalue equation:

$$Hv = \lambda v. \tag{20}$$

Algebraically, for a given triple $\{H,\lambda,v\}$, the characteristic equation arises as a condition of the vanishing determinant of the matrix $(H-\lambda I)$ for all nonzero $v \in \mathbb{R}^2$. In other words, we have

$$\begin{vmatrix} a-\lambda & c \\ c & d-\lambda \end{vmatrix} = 0. \tag{21}$$

It follows that the eigenvalue λ of H satisfies the quadratic equation

$$\lambda^2 - (a+d)\lambda + ad - \ c^2 = 0. \tag{22}$$

From Equation (22), we find the following eigenvalues:

$$\lambda_{1,2} = \frac{(a+d) \pm \sqrt{(a+d)^2 - 4(ad-c^2)}}{2}. \tag{23}$$

To study the linear transformation properties of the fluctuation vector $v \in \mathbb{R}^2$ for a given eigenvalue λ as above in Equation (23), we consider two invariants of H as the trace $tr(H) = a+d$ and determinant $\Delta = ad - c^2$. In this setup, it follows that the eigenvalues $\lambda_{1,2}$ can be expressed as

$$\lambda_{1,2} = \frac{tr(H) \pm \sqrt{tr^2(H) - 4\Delta}}{2}. \tag{24}$$

We find a unique real eigenvalue λ of H whenever the above linear class operators $\{tr(H),\Delta\}$ satisfy the equality $tr^2(H) = 4\Delta$. The qualitative description of the global stability component and eigenvalues $\lambda_{1,2}$ is relegated to the next section.

3.3.2. Evaluation of Eigenvectors

Next we compute the corresponding fluctuation vectors underlying the AKS primality testing function $f(x,y)$ as in Equation (1). Namely, the fluctuation vectors are defined as the eigenvectors of the fluctuation matrix H as in Equation (20). In this case, there are two eigenvalues, whereby we have two corresponding eigenvectors. For a given pair of eigenvalues $\{\lambda_1,\lambda_2\}$, the eigenvectors $\{v_1,v_2\}$ are evaluated by the eigenvalue equation, viz. Equation (20). For $\lambda = \lambda_1$, the eigenvector v_1 is obtained as the following two-dimensional vector:

$$v_1 = \begin{pmatrix} x_1 \\ y_1 \end{pmatrix}. \tag{25}$$

For the fluctuation matrix H as in Equation (7), the corresponding eigenvalue equation reads as

$$\begin{pmatrix} a & c \\ c & d \end{pmatrix}\begin{pmatrix} x_1 \\ y_1 \end{pmatrix} = \lambda_1\begin{pmatrix} x_1 \\ y_1 \end{pmatrix}. \tag{26}$$

In other words, we have the following pair of simultaneous linear equations:

$$(\lambda_1 - a)\, x_1 = cy_1 \tag{27}$$

$$(\lambda_1 - d)y_1 = cx_1. \tag{28}$$

In order to solve the above pair of linear equations as in Equations (27) and (28), we may choose $y_1 = k \in \mathbb{R}$. Thus, from Equation (28), it follows that we have

$$x_1 = \frac{k(\lambda_1 - d)}{c}. \tag{29}$$

With these values of x_1 and y_1, the eigenvector v_1 corresponding to the eigenvalue λ_1 is given by

$$v_1 = \begin{pmatrix} \frac{k(\lambda_1 - d)}{c} \\ k \end{pmatrix}. \tag{30}$$

Therefore, the corresponding norm $||v_1||$ of the fluctuation vector v_1 is given by

$$||v_1|| = \sqrt{\left(\frac{k(\lambda_1 - d)}{c}\right)^2 + k^2}. \tag{31}$$

In this case, the normalized eigenvector $\hat{v}_1 = v_1/v_1$ associated to $\lambda = \lambda_1$ reads as

$$\hat{v}_1 = \frac{1}{\sqrt{\left(\frac{(\lambda_1 - d)}{c}\right)^2 + 1}}\begin{pmatrix} \frac{(\lambda_1 - d)}{c} \\ 1 \end{pmatrix}, \tag{32}$$

where λ_1 is given as above in Equation (23) with the choice of the positive signature. By following the aforementioned methodology, the fluctuation vector v_2 corresponding to the eigenvalue λ_2 of H can be obtained in a similar manner. Namely, by defining the eigenvector

$$v_2 = \begin{pmatrix} x_2 \\ y_2 \end{pmatrix} \tag{33}$$

with the choice of $y_2 = k \in \mathbb{R}$, we obtain the first component of v_2 as

$$x_2 = \frac{kc}{(\lambda_2 - a)}. \tag{34}$$

For $\lambda = \lambda_2$ with the above $\{x_2, y_2\}$, the eigenvector v_2 has the following norm:

$$||v_2|| = \sqrt{\left(\frac{kc}{(\lambda_2 - a)}\right)^2 + k^2}. \tag{35}$$

Finally, it follows that the normalized eigenvector $\hat{v}_2 = v_2/||v_2||$ reduces as

$$\hat{v}_2 = \frac{1}{\sqrt{\left(\frac{c}{(\lambda_2 - a)}\right)^2 + 1}}\begin{pmatrix} \frac{c}{(\lambda_2 - a)} \\ 1 \end{pmatrix}, \tag{36}$$

where λ_2 reads as above in Equation (23) with the choice of the negative signature. The above pair of normalized eigenvectors $\{\hat{v}_1, \hat{v}_2\}$ as depicted in Equations (32) and (36) gives the direction of fluctuations under the randomized AKS primality testing of an arbitrary integer x on a particular machine M of the cardinality y of the set of its alphabets. The corresponding norms $\{||v_1||, ||v_2||\}$, as in Equations (31) and (34), respectively, signify the intrinsic errors in determining the primality of the integer x on the machine M. In this concern, the total error e is defined as the maximum of the norms of eigenvectors $\{v_1, v_2\}$ of the fluctuation matrix H, viz. we have $e = max\{||v_1||, ||v_2||\}$ as the effective uncertainty in testing the primality of an arbitrary integer string x on M.

4. Discussion of the Results

In this section, we provide the qualitative description of the results concerning the optimized primality testing of an arbitrary integer via the AKS algorithm [1]. For an illustration, we take an efficient AKS algorithm with a varying input integer x whose primality is to be tested on a machine M of the cardinality y of the set of its alphabets. Henceforth, without loss of generality, we may choose a randomized AKS algorithm with its pre-factor $A = 1$. For the purpose of qualitative discussion, we set the parameters $x, y \in (0, 100)$ as below.

We depict the qualitative behavior of the randomized AKS primality testing function $f(x, y)$, associated flow components, heat capacities $\{a, d\}$, local correlation c, and the determinant Δ of its fluctuation matrix H under fluctuations of the system parameters $\{x,\ y\}$. The positivity of $\{a, d\}$ and Δ determines the underlying stability domains of the AKS primality testing of an integer x with the maximum number of its prime factors obtained by a given machine M of the cardinality y of the set of its alphabets as the objective function.

For a given AKS primality testing function $f(x, y)$, as in Equation (1) as the objective function, we perform our analysis by increasing the input integer x whose primality is to be tested on a machine of the cardinality y of the set of its alphabets. Namely, from Figure 1, we see that the objective function $f(x, y)$ has an amplitude of the order 10^{20}. In due course of execution of the algorithm, we observe that $f(x, y)$ behaves smoothly for all values of the system parameters $x, y \in (1,\ 100)$, except at their extreme values, where it blows up. Furthermore, we find that there are no fluctuations in $f(x, y)$ for small values of $\{x, y\}$. In this case, it follows that $f(x, y)$ increases as we augment the algorithm parameters $\{x, y\}$ of the AKS primality testing of an integer.

We assess the qualitative behavior of the input integer rate $f_x(x, y)$ of the AKS primality testing function $f(x, y)$ under variations in the input parameters $\{x, y\}$. From Figure 2, we find that the rate $f_x(x, y)$ concerning the input integer x takes an amplitude on the order of 10^{20} as x and y vary. Moreover, for a large input x, we see that $f_x(x, y)$ forms an increasing arc with respect to the cardinality y of the set of alphabets of the machine executing the AKS algorithm.

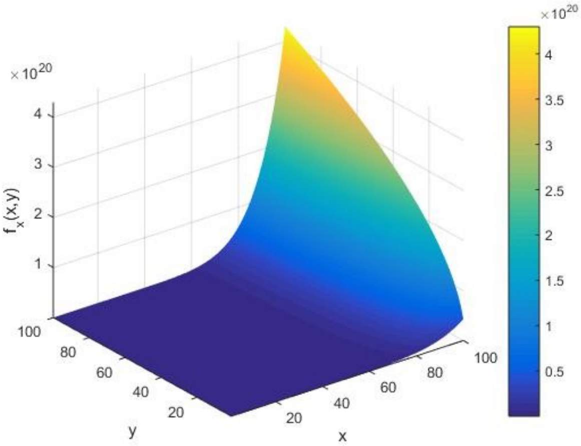

Figure 2. The input integer rate $f_x(x, y)$ as a function of the input integer x and cardinality y of the set of alphabets of the machine executing the AKS algorithm plotted in the range $x, y \in (1,\ 100)$.

We assess the qualitative behavior of the AKS primality testing rate $f_y(x, y)$ with respect to the cardinality y of the set of alphabets of the machine as in Figure 3. This is realized by varying the first partial derivative $f_y(x, y)$ of the AKS function $f(x, y)$ with respect to the input parameters $\{x, y\}$, as represented in Equation (3). In this case, we find that there is an increasing arc with respect to the input integer x in the limit of a large cardinality y of the set of alphabets of the machine. In particular, from Figure 3, we notice that the appearance of arc shifts along the x-axis for the rate $f_y(x, y)$ in contrast to the input integer rate $f_x(x, y)$ when they vary with respect to the parameters $\{x, y\}$ of the algorithm.

From Figure 4, we find that the local input integer capacity $f_{xx}(x, y)$ takes the amplitude of the order 10^{20} as $\{x, y\}$ vary in the interval (1, 100). For a given input integer x, from Figure 4, we see that the input string capacity $f_{xx}(x, y)$ increases in an arc with respect to the cardinality y of the set of alphabets of the machine executing the AKS primality testing algorithm. Furthermore, as x increases, we observe that f_{xx} remains constant up to a certain value of x; however, it starts showing a sharply increasing amplitude of the order of 10^{21} at an extreme value of x. This implies that the primality testing of an arbitrary integer x in its randomized limit possesses a nonzero input string heat capacity f_{xx} for all values of the cardinality y of the machine executing the algorithm.

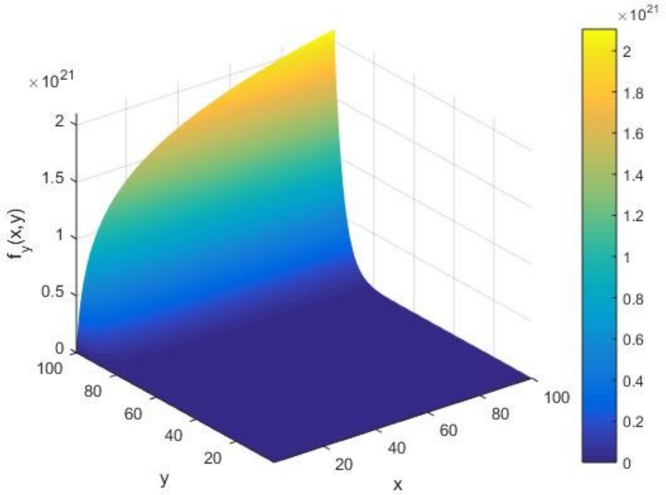

Figure 3. The rate $f_y(x, y)$ as a function of the input integer x and cardinality y of the set of alphabets of the machine executing the AKS algorithm plotted in the range $x, y \in (1,\ 100)$.

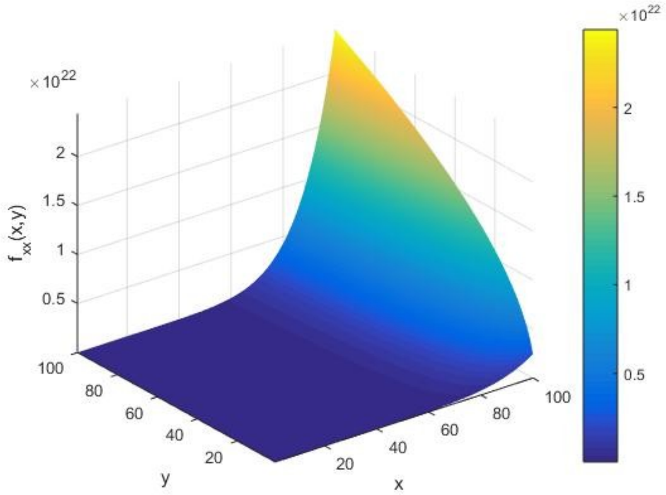

Figure 4. The input integer capacity $f_{xx}(x, y)$ as a function of the input integer x and cardinality y of the set of alphabets of the machine executing the AKS algorithm plotted in the range $x, y \in (1, 100)$.

From Figure 5, we see that the runtime heat capacity $f_{yy}(x, y)$ has a large amplitude of fluctuations on the order of 10^{24} at an initial stage of the execution. However, as the algorithm proceeds, we find a vanishingly small value of the machine heat capacity $f_{yy}(x, y)$, that is, the primality testing of an integer x is approximately granted for all values of $y > 80$ as in Figure 5. Subsequently, for an arbitrary input integer x, we observe that $f_{yy}(x, y)$ is a well-behaved function of (x, y), with an amplitude on the order of 10^{23} in the limit of small y.

In Figure 6, we show the qualitative behavior of the local correlation $f_{xy}(x, y)$ underlying the AKS primality testing function $f(x, y)$ under variations of the input integer x and cardinality y of the set of alphabets of the machine. In this case, apart from the amplitude of fluctuations, we see that $f_{xy}(x, y)$ possesses an identical character to the execution rate $f_y(x, y)$ as far as variations in the input integer x and execution of the algorithm are concerned. In the limit of a large y, we find that $f_{xy}(x, y)$ modulates with the amplitude on the order of 10^{23} in an increasing arc upon an increase of the input integer x whose primality is to be tested. In the course of executing the AKS primality testing algorithm, the above arc shifts along the x-axis in contrast to the rate f_x under variations of the system parameters $\{x, y\}$.

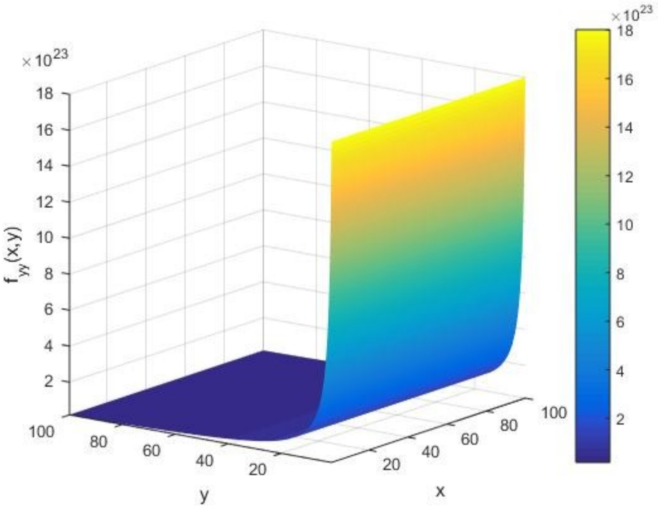

Figure 5. The local runtime capacity $f_{yy}(x, y)$ as a function of the input integer x and cardinality y of the set of alphabets of the machine executing the AKS algorithm plotted in the interval $x, y \in (1, 100)$.

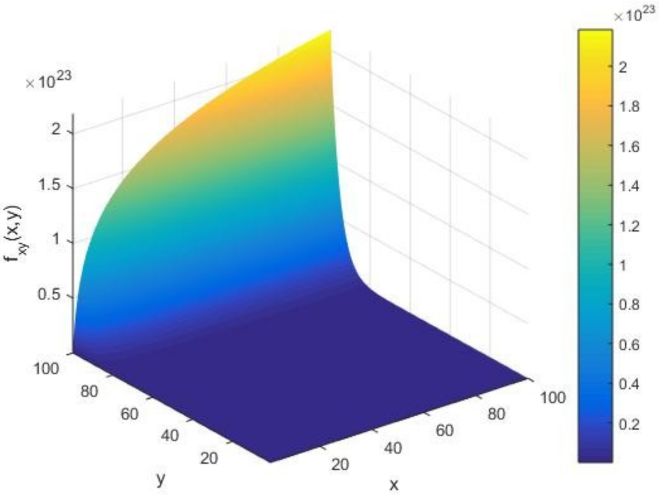

Figure 6. The correlation $f_{xy}(x, y)$ as a function of the input integer x and the cardinality y of the set of alphabets of the machine executing the AKS algorithm plotted in the range $x, y \in (1,\ 100)$.

In Figure 7, we provide a qualitative depiction of the global stabilities underlying the AKS primality testing algorithm of an arbitrary integer x on a Turing machine M of the cardinality y of its set of alphabets as the positivity of the determinant $\Delta(x, y)$ as in Equation (7) for $x, y \in (1,\ 100)$. For a given input integer $x \in (1,\ 100)$, we see that $\Delta(x, y)$ takes a large negative value with its amplitude of the order 10^{48} in the limit of large cardinality y. On the other hand, for relatively smaller values of the system parameters $\{x, y\}$, we find that $\Delta(x, y)$ has vanishingly small fluctuations. This implies that the AKS primality testing could be globally unstable under certain values of $\{x, y\}$. In addition, we find that the above qualitative behavior of $\Delta(x, y)$ remains the same for large values of $\{x, y\}$, e.g., $x, y \rightarrow 1000$. In this case, the corresponding fluctuations in $\Delta(x, y)$ however grows to a large amplitude of the order 10^{194}.

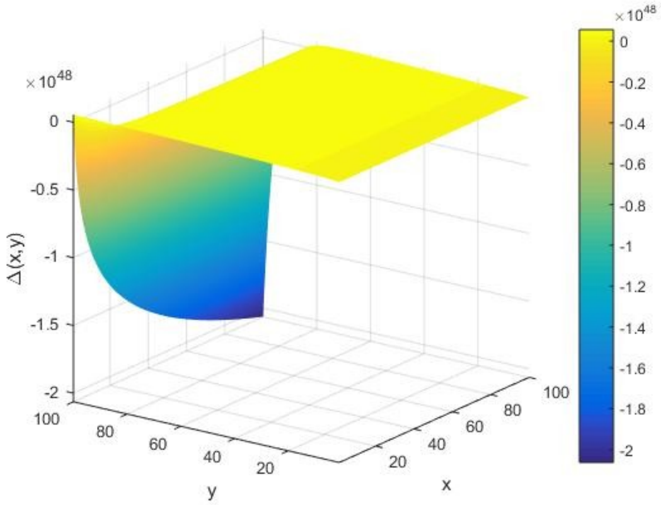

Figure 7. The determinant $\Delta(x, y)$ of the Hessian matrix H as a function of the input integer x and cardinality y of the set of alphabets of the machine executing the AKS algorithm plotted in the interval $x, y \in (1,\ 100)$.

Figure 8 displays the corresponding qualitative behavior of the discriminant defining the eigenvalues $\{\gimel_1,\ \gimel_2\}$ as in Equation (23) of the fluctuation matrix H of the AKS primality testing function $f(x, y)$. For all values of the system parameters $\{x, y\}$, we see that the discriminant always remains positive with the amplitude of the order 10^{48}. Furthermore, in an intermediate range of y, we find that it has a vanishingly small value. This implies that both the eigenvalues $\gimel_1$ and $\gimel_2$ of the Hessian matrix of $f(x, y)$ take an approximately identical value.

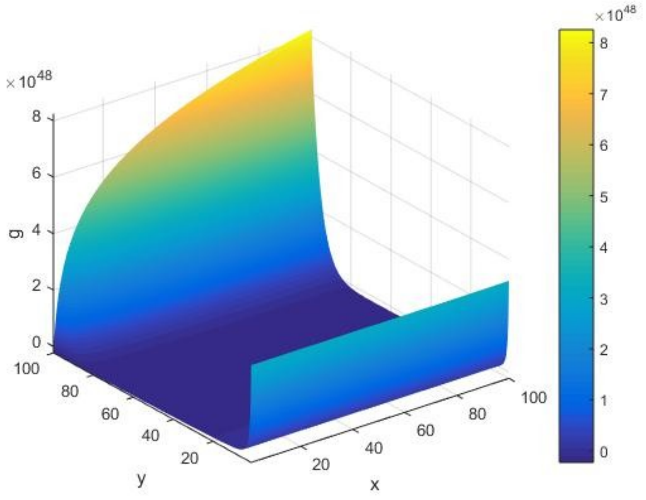

Figure 8. The discriminant as a function of the input integer x and cardinality y of the set of alphabets of the machine executing the AKS algorithm plotted in the interval $x, y \in (1,\ 100)$.

From Figure 9, we see that the eigenvalue $\gimel_1$, as a global stability component, always remains positive for various values of system parameters $\{x, y\}$. It is observed that $\gimel_1$ grows to large amplitudes of the order 10^{24} at an initial execution of the algorithm. Furthermore, in the limit of small y, it follows that the amplitude of fluctuations in $\gimel_1$ takes a relatively higher numerical value than its corresponding value at a large y. A smaller value of $\gimel_1$ shows that the AKS primality testing algorithm of an integer gets stabilized upon its execution with increasing values of y.

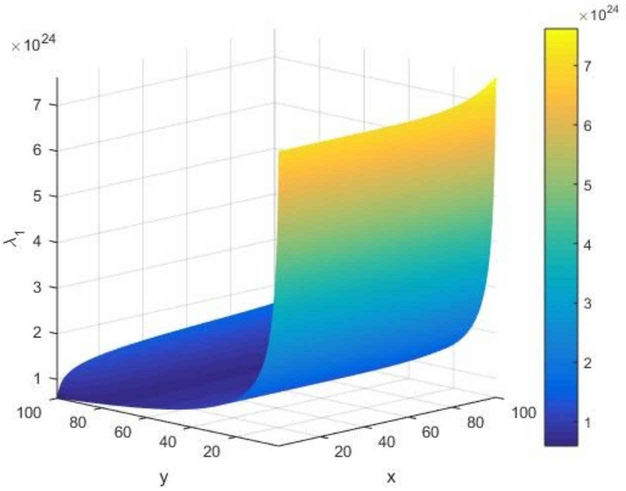

Figure 9. The eigenvalue $\gimel_1$ of the Hessian matrix H as a function of the input integer x and cardinality y of the set of alphabets of the machine executing the AKS algorithm plotted in the range $x, y \in (1,\ 100)$.

In Figure 10, we display the qualitative behavior of the eigenvalue $\gimel_2$ as the other global stability component under variations of the system parameters $\{x, y\}$. In this case, we find that $\gimel_2$ always takes a large negative value of the amplitude of the order 10^{24}. Namely, in the limit of large x and small y, we find the signature of possible instabilities in execution of the AKS primality testing algorithm of an integer on a given Turing machine. On the other hand, it is noted that stability rises in the limit of increasing values of y. However, towards its extreme values, we see that the stability of the AKS primality testing algorithm increases for an arbitrary input integer x and $y \rightarrow 100$.

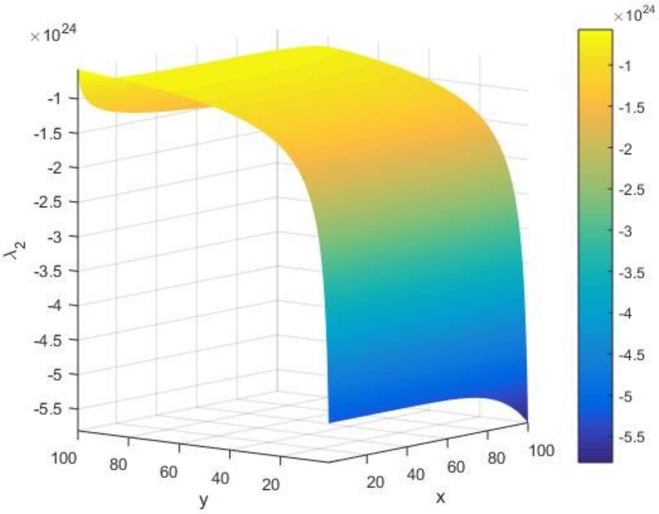

Figure 10. The eigenvalue $\gimel_2$ of the Hessian matrix H as a function of the input integer x and cardinality y of the set of alphabets of the machine executing the AKS algorithm plotted in the range $x, y \in (1,\ 100)$.

In Figure 11, we offer a qualitative description of the trace $tr(H)$ of the fluctuation matrix H as in Equation (7) under variations of the system parameters $\{x, y\}$. In this case, it follows that $tr(H)$ has a nonzero value of the amplitude of the order 10^{24} at an initial execution of the algorithm. However, as the algorithm runs, we find that it attains a vanishingly small value in the limit of large y. The AKS primality testing of an integer x is approximately granted, viz. we can determine whether it is prime or not. Furthermore, for a given input integer x, it is observed that $tr(H)$ is a well-behaved decreasing function of y. It is worth mentioning that $tr(H)$ equally has large amplitude of the order of 10^{23} in the limit of a large y.

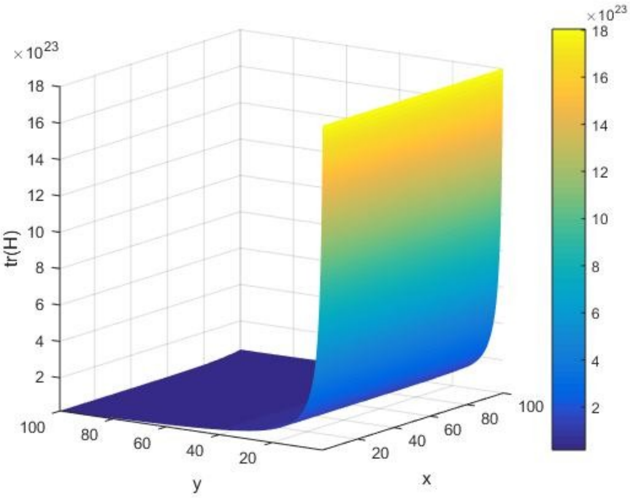

Figure 11. The trace $tr(H)$ of the Hessian matrix H as a function of the input integer x and cardinality y of the set of alphabets of the machine executing the AKS algorithm plotted in the interval $x, y \in (1,\ 100)$.

From Figure 12, we see the qualitative behavior of the norm $|v_1|$ of the fluctuation vector v_1 corresponding to the eigenvalue $\gimel_1$ of H under variations of the input integer x and cardinality y of the set of alphabets of the machine. In this case, in the limit of small values of the system parameters $\{x, y\}$, we find that there exists a large peak of the order of 10^{21}. Furthermore, we notice that the AKS primality testing exists smoothly, without variations in the norm $|v_1|$ of the fluctuation vector v_1 for various values of system parameters $\{x, y\}$ governing the algorithm.

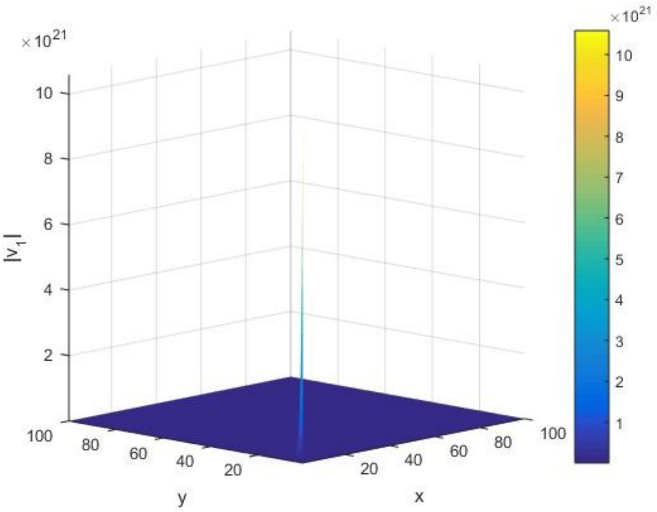

Figure 12. The norm $|v_1|$ corresponding to the eigenvalue $\gimel_1$ of the Hessian matrix H as a function of the input integer x and cardinality y of the set of alphabets of the machine executing the AKS algorithm plotted in the interval $x, y \in\ (1,\ 100)$.

Under variations of the input integer x and cardinality y of the set of alphabets of a Turing machine, in Figure 13, we offer the corresponding qualitative behavior of the norm $|v_2|$ of the fluctuation vector v_2 concerning execution of the AKS algorithm. Note that our calculation is performed in their randomized limits. Given a Turing machine M with the set of its alphabets of a large cardinality y, we find that the norm $|v_2|$ increases on an arc for increasing values of the input integer x. On the other hand, for $y < 100$, we observe that the norm $|v_2|$ takes a unit value for all input integers x whose primality is to be tested by the machine with its set of alphabets of the cardinality y. Furthermore, we note that the appearance of an arc in $|v_2|$ shifts along the x-axis when the input integer x is varied with respect to a given machine with the set of its alphabets of the cardinality y executing the algorithm.

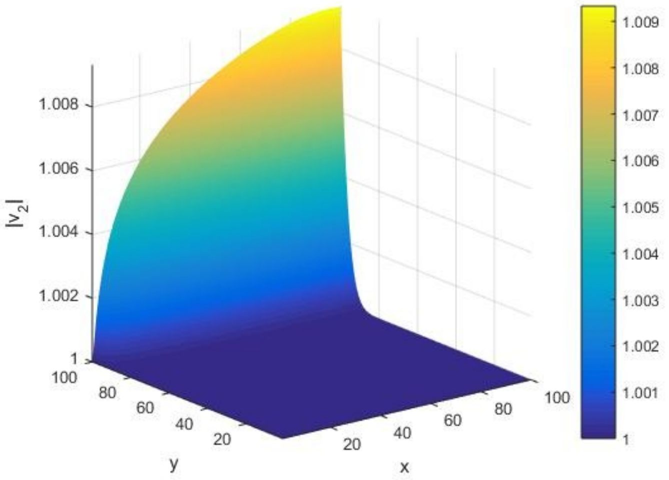

Figure 13. The norm $|v_2|$ corresponding to the eigenvalue $\gimel_2$ of the Hessian matrix H as a function of the input integer x and cardinality y of the set of alphabets of the machine executing the AKS algorithm plotted in the interval $x, y \in (1,\ 100)$.

5. Summary and Conclusions

In this paper, we study the AKS primality testing of an integer from the perspective of optimization theory. Namely, by concentrating on the randomization hypothesis, we optimize the AKS algorithm under fluctuations of the input integer and cardinality of the set of alphabets of the machine. We examine the (in)stability domains of the AKS algorithm under variations of input integers in testing their primality. This yields a deterministic finite time polynomial type optimized solution for testing the primality of an arbitrary integer. From the viewpoint of the randomization theory [11], our analysis leads to an optimal system with a pair of system parameters as the input integer and cardinality of the set of alphabets of the machine executing the algorithm. In order to do so, we choose the input integer in a given representation, e.g., a definite sequence of $\{0, 1\}$ in the binary representation, whereby allowing the positions of the numbers 0 and 1 to vary. As far as the AKS primality testing of an integer is concerned [1], there have been various research articles, books, and monographs in computer science, number theory, complexity theories, and others; see [27] for the complexity and satisfiability theories. Fermat's little and Chinese remainder theorems play an important role in extending the primality testing of integers from their modular representations to real counterparts.

In particular, we optimize the AKS primality testing algorithm of an arbitrary integer on a Turing machine by determining its stability structures. By invoking the rule of multivariable analysis, we offer the corresponding quantitative and qualitative depictions. Considering the maximum number of irreducible factors of a given integer, as determined by the deterministic polynomial time Turing machine as the AKS objective function, our investigation is realized by randomizing the cardinality of the set of alphabets of the machine and the input integer whose primality is to be tested. As the input integer becomes large, the number of its prime factors also probabilistically grows, whereby we offer an asymptotic stability analysis of the randomized AKS primality testing of an arbitrary integer. Furthermore, this is realized via multidimensional illustrations of the fluctuation components concerning the local and global execution stability of the AKS algorithm. Here, the local heat capacities with respect to the system parameters are defined as the pure second-order derivatives of the AKS primality testing function, whereas the associated correlation between them is taken as its mixed derivative. At the critical point corresponding to a unit input integer, we find that both the heat capacities vanish identically; however, the correlation turns out to be ill-defined. Moreover, we observe that both the heat capacities and the local correlation become ill-defined at the critical point corresponding to the set of alphabets of a large cardinality of the machine, performing a computation of the primality testing of an integer.

On the other hand, given the AKS primality testing function of an integer, the global stability regions are determined by the positivity of the determinant of its fluctuation matrix with a positive fluctuation capacity. We find both stable and unstable regions depending on the value of the input integer and the cardinality of the set of alphabets of the machine. Furthermore, we provide a limiting analysis of the stability components at the roots of the flow components. Indeed, our consideration follows from the randomization hypothesis [11] of an algorithm as well. Our analysis does not stop here, but continues further in the realm of local linear algebra. We investigate the behavior of randomized fluctuation vectors as the basis vectors of the fluctuation matrix. The corresponding fluctuations in their norms provide execution stability characteristics of the AKS primality testing algorithm. In addition, we offer the qualitative behavior of the AKS primality testing function as the objective function, with its flow rates, correlation, and the local and global stability regions revealed through the positivity of the heat capacities and that of the determinant of the fluctuation matrix, respectively. Our analysis confirms that the AKS primality testing falls in the realm of P-type problems. By taking up the randomization theory of Agrawal and Biswas [11], this paper gives the parameter space optimization of the AKS algorithm. Namely, we offer the optimization theory perspective of the AKS algorithm under variations of its parameters, viz. the number of alphabets of the input string and the cardinality of the set of alphabets of a machine performing the primality testing. Following this optimization theory initiative of the AKS algorithm, the associated execution time analysis and issues pertaining to the asymptote time complexity, potential extensions towards higher dimensions and comparisons with other primality testing and identity testing models are relegated to separate research publications.

From the perspective of stability theories, it is worth emphasizing that the optimization characteristics of the AKS primality testing are well determined for an arbitrary integer. This enables us to classify the local and global (in)stability domains of the AKS primality testing. Following the above classification, our proposal shows that the determinant has a large amplitude of fluctuations around one of its critical points (see Figure 7). Depending on the signature of the Hessian determinant of the AKS primality testing function as the objective function, we see that there are both concave and convex type domains of the AKS algorithm. Indeed, apart from the field of real numbers, an extension of our analysis is anticipated to be realized by localizing a finite ring of integers to different algebraic sets, e.g., the field of complex numbers, hyper-complex numbers, quaternions, octonions, and higher spin representations in the light of Clifford algebra and octonions [28]. Interesting perspectives are expected to arise via an extension of the input parameters to the set of rational numbers, algebraic sets, and varieties [20–23]. Industrial applications are sought via complexity classifications of the classical P, NP, coNP, PSPACE, NC, #P type classes and some modern ones such as PO and NPO, arising from the approximations of certain optimization problems [10,29,30] towards the complexity classifications of algorithms and their generalizations. Finally, we anticipate that the corresponding (de)randomization maps [1,11,20–22] could play an equally important role in understanding the notion of parametric fluctuations of the AKS algorithm in testing the primality of an arbitrary integer. We leave such investigations open for future research and developments.

Author Contributions: For research articles with several authors, a short paragraph specifying their individual contributions must be provided. The following statements should be used "conceptualization, B.N.T.; methodology, B.N.T., J.K.K. and J.M.A.; software, B.N.T., J.K.K. and J.M.A.; validation, B.N.T., J.K.K. and J.M.A.; formal analysis, B.N.T., J.K.K. and J.M.A.; investigation, B.N.T., J.K.K., J.M.A. and N.M.; resources, B.N.T., J.K.K., J.M.A. and N.M.; data curation, B.N.T., J.K.K. and J.M.A.; writing—original draft preparation, B.N.T., J.K.K. and J.M.A.; writing—review and editing, B.N.T., J.K.K., J.M.A. and N.M.; visualization, B.N.T., J.K.K., J.M.A. and N.M.; supervision, B.N.T. and N.M.; project administration, B.N.T. and N.M.; funding acquisition, B.N.T., J.K.K., J.M.A. and N.M.".

Funding: This research received no external funding.

Acknowledgments: The authors would like to thank the Ministries of Education and Science of the Republic of Macedonia and the Istituto Nazionale di Fisica Nucleare, Ministero dell'Istruzione, dell'Università e della Ricerca, Italy for providing a stimulating environment for the realization of this research.

Conflicts of Interest: The authors declare no conflict of interest.

Appendix A.

Below, we briefly depict the evolution of the AKS algorithm and its relationships to Fermat's little theorem and Chinese remaindering. In doing so, we provide a brief account of the evolution of the AKS algorithm of an arbitrary integer. In particular, we give an explicit presentation of the AKS primality testing algorithm and its relation to the randomization hypothesis in light of the ring theories and their localizations as the following.

Appendix A.1. A Brief Account of the Evolution of Primality Testing

Below, among the known facts for primality testing, we review essentials from the theory of numbers, viz. Fermat little theorem, Chinese remainder theorem, Sieve of Eratosthenes, Miller-Rabin test, etc. Scientifically, this provides essential rudiment of related works and insightful background that makes repeated references in the known literature on prime factoring as developed in Section 3. First of all, recall that in order to determine whether the given input number is a prime or not, an efficient primality testing algorithm requires having a polynomial runtime [1]. In general, this is required to possess the unconditional and deterministic properties of the algorithm. It is worth remembering that the Sieve of Eratosthenes [21,22] was the first prime testing algorithm, which may be referred to as an antiquity test of the primality of an integer. Despite it being able to categorize a given input as prime or not, it fell short of having an exponential runtime with its complexity $\Omega\left(\sqrt{n}\right)$ for an input string of size n; see [24] for an overview of the computational complexity and associated concepts in the realm of the analytic number theory.

Fermat's little theorem [21,22] was an immediate precursor of the former test. The Fermat test played a significant role in the prime determination of a given integral valued input. Despite the fact that it correctly determined the nature of a given input (whether it was a prime or not), it presented some technical difficulties. Namely, it occasionally classified some composite numbers as the primes, as well. Such integers include numbers such as the pseudo-primes, Carmichael primes, Mersenne's primes, and Cunningham numbers; see [21,22] for details. Thus, Fermat's test is extended towards the probabilistic primality testing algorithms.

With the above motivations, the Miller-Rabin test [25], which is founded on the principle of Fermat's little theorem, unconditionally and probabilistically certifies the prime characterization of a given input, that is, it enables us to know whether a given integer is prime or not. In terms of the runtime complexity, its overall time complexity is of a polynomial type of the order $O\left(k \times k\, log^3 n\right)$ for a given input of size n. Here, the factor k quantifies the number of bases a that are used in the algorithm; see [25] for the randomized Miller-Rabin test and related notions.

In the course of improving the prime testing algorithms, the Solovay-Strassen primality [26] encompasses an exciting working principle. This was jointly built as a consequence of Euler's Criterion and Fermat's little theorem [21,22]. However, this test presents the probability of failure as $1/2^k$, where k is the total number of different bases a that are used in the primality testing algorithm. Furthermore, the probability of failure makes the overall accuracy of the algorithm similar to that of Fermat testing.

In addition, it is worth mentioning that the concerned accuracy in the prime characterization is less than that of the Miller-Rabin test [25,26]. The time complexity is of the order of $O\left(log^3 n\right)$, which is closer to that of the Miller-Rabin test. It is important to note that both the above tests are probabilistic in nature as they rely on the choice of a random base a; see [25,26] for an introduction to the Miller-Rabin and Solovay-Strassen primality testing of an integer.

Appendix A.2. The AKS Algorithm: An Overview

The deterministic characteristics of AKS primality testing are bounded as a polynomial runtime algorithm. Namely, in order to define the AKS primality testing of an integer [1], we begin by recalling Fermat's little theorem [21,22], which may be viewed as the modular equivalence:

$$a^n = a(mod\ n),\ \text{where } a, n \in \mathbb{N} \text{ and } n \nmid a. \tag{A1}$$

It is worth mentioning that the AKS algorithm locally extends as a primality testing of a given polynomial, whether it vanishes identically or not in a ring of local polynomials [1]. In this setup, the AKS algorithm [1] results as the following (generalized Fermat's little theorem):

$$(x-1)^n = x^n - 1\ mod(x^r - 1,\ n). \tag{A2}$$

It is worth mentioning that the above modular congruence as in Equation (A2) can be verified as a P-type problem, whenever r is polynomial to the digits of the given input integer n. The above localization statement [11] of Fermat little's theorem emerges from the identity

$$(a+z)^n = a + \ z^n(mod\ n). \tag{A3}$$

Thus, the primality testing of an integer n can be realized as a modular identity by choosing an integer a such that Equation (A3) holds. This follows via the binomial expansion of $(a+z)^n$ over $mod\ n$. Namely, the coefficients $\left(\begin{array}{c} n \\ i \end{array}\right)$ in the binomial expansion of the polynomial

$$p_n(z) := (a+z)^n - (a + z^n) \tag{A4}$$

vanish identically over $mod\ n$ for all $i = 1, 2, \ldots, n$. Notice further that the above congruence, as depicted in Equation (A2), can be viewed as a particular equality in the polynomial ring $\mathbb{Z}_n[x]$. By evaluating a quotient ring of $\mathbb{Z}_n[x]$, one finds an upper bound to the degree of the polynomials $p_n(x)$. Thus, the AKS primality testing algorithm evaluates the vanishing of $p_n(x)$ in the quotient ring $\mathbb{Z}_n[x]/(x_r - 1)$. This results in an explicit dependence of the computational complexity of the algorithm on the size of r. Namely, for a given pair of polynomials $\{f, g\}$, the modular equivalence in Equation (A2) can be expressed as the validation of the identity $p_n(x) = (x^r - 1)g + nf$. Consequently, by setting $g = 0$ and $x = z$, it follows that all the primes obeying the congruence relation $(a+z)^n - (a+z^n) = (z^r - 1)g + nf$ satisfy the aforementioned equivalence (A3), whenever n is a prime.

By localizing $z \in \mathbb{Z}[x]$ as in Equation (A3), for a given integer $n \in \mathbb{N}$, we may concentrate on modular valued polynomials $p_n(z)$; see [1,11] for associated details. In light of the standard modular equivalence [21,22], the above statement concerning the vanishing of $p_n(x)$ does not hold globally as in the case of the standard Fermat's little theorem, as depicted above in Equation (A1). This is because n lies in the ring $\mathbb{Z}_n$, which is strictly allowed to fluctuate over $\mathbb{N}$. With a suitable extension as $\mathbb{N} \ni n \longmapsto z \in \mathbb{R}$, the localized version of Fermat's little theorem arises as per the modular equivalence

$$p_n(z) = 0(mod\ n). \tag{A5}$$

This justifies the AKS primality testing algorithm of an integer, as stated above in Equation (A2). From the above-localized version of Fermat's little theorem, we observe that the primality testing of an integer reduces as a convenient way of obtaining the roots of the modular polynomial $p_n(z)$. Whenever Equation (A5) holds, the input integer n results as a prime. Hereby, we can obtain a definite value of r and a set of the values of a with the properties mentioned in Algorithm 1 in Section 2, such that n is a prime whenever the congruence (A2) holds.

Finally, it is worth mentioning that, unlike in the case of the standard Fermat's little theorem in Equation (A1), when n varies over the set of integers, i.e., we have $n - 1$ elements forming the ring $\mathbb{Z}_n$, we may concentrate on a randomized version of the AKS identity as in Equation (A2) for the primality testing of an arbitrary integer n. This can be performed via an extension of the Chinese remaindering in a local ring $\mathbb{Z}[x]$ consisting of a set of finite degree polynomials $p_n(z)$; see [1,11] for an extended introduction of the primality and identity testing algorithms.

References

1. Agrawal, M.; Kayal, N.; Saxena, N. PRIMES is in P. *Ann. Math.* **2004**, *160*, 781–793. [CrossRef]
2. Savu, L. Cryptography role in information security. In Proceedings of the 5th WSEAS International Conference on Communications and Information Technology (CIT11), Corfu Island, Greece, 14–17 July 2011; pp. 36–41.
3. Knudsen, L.R.; Matthew, J.B.R. *The Block Cipher Companion*; Springer: Berlin/Heidelberg, Germany, 2011.
4. Van Sinderen, M.; Pires, L.F.; Vissers, C.A. Protocol design and implementation using formal methods. *Comput. J.* **1992**, *35*, 478–491. [CrossRef]
5. Sharp, R. *Principles of Protocol Design*; Springer: Berlin/Heidelberg, Germany, 2008.
6. Clark, K. An Algorithm that Decides PRIMES in Polynomial Time. Available online: https://sites.math.washington.edu/~{}morrow/336_11/papers/kevin.pdf (accessed on 20 April 2019).
7. Cook, S. The P versus NP problem. In *The Millennium Prize Problems*; Carlson, J., Carlson, J.A., Jaffe, A., Wiles, A., Eds.; Clay Mathematics Institute: Cambridge, MA, USA; American Mathematical Society: Providence, RI, USA, 2006; pp. 87–104.
8. Fortnow, L.; Homer, S. *A Short History of Computational Complexity*; Computer Science: Technical Reports, 2003-10-02; Boston University Computer Science Department: Boston, MA, USA, 2003.
9. Sudan, M. The P vs. NP problem. Available online: http://madhu.seas.harvard.edu/papers/2010/pnp.pdf (accessed on 20 April 2019).
10. Creignou, N.; Khanna, S.; Sudan, M. *Complexity Classifications of Boolean Constraint Satisfaction Problems*; SIAM: Philadelphia, PA, USA, 2001.
11. Agrawal, M.; Biswas, S. Primality and identity testing via Chinese remaindering. *J. ACM (JACM)* **2003**, *50*, 429–443. [CrossRef]
12. Sudan, M. Algebra and Computation, Lecture 12. Available online: http://people.csail.mit.edu/madhu/ST12/scribe/lect12.pdf (accessed on 20 April 2019).
13. Kopparty, S. Primality Testing, Lecture 14, Algorithmic Number Theory. Available online: http://www.math.rutgers.edu/~{}sk1233/courses/ANT-F14/lec14.pdf (accessed on 20 April 2019).
14. Hansen, P.B. Primality Testing. Available online: http://surface.syr.edu/eecs_techreports/169/ (accessed on 20 April 2019).
15. Smart, N.P. *Cryptography: An Introduction*, 3rd ed.; Mcgraw-Hill: New York, NY, USA, 2003; pp. 1–22.
16. Hardy, G.H.; Wright, E.M. *Introduction to the Theory of Numbers*, 6th ed.; Oxford University Press: Oxford, UK, 2008; pp. 63–72.
17. Sudan, M. Primality Testing, Lecture 12, Algebra and Computation. Available online: http://people.csail.mit.edu/madhu/ST12/scribe/lect12.pdf (accessed on 20 April 2019).
18. Turing, A.M. On computable numbers, with an application to the Entscheidungsproblem. *Proc. Lond. Math. Soc.* **1937**, *2*, 230–265. [CrossRef]
19. Ruppeiner, G. Riemannian geometry in thermodynamic fluctuation theory. *Rev. Mod. Phys.* **1995**, *67*, 605–659. [CrossRef]
20. Tiwari, B.N. Geometric perspective of entropy function: Embedding, spectrum and convexity. *arXiv* **2011**, arXiv:1108.4654.
21. Tiwari, B.N.; Adeegbe, J.M.; Kibindé, J.K. *Randomized Cunningham Numbers in Cryptography: Randomization theory, Cryptanalysis, RSA cryptosystem, Primality testing, Cunningham numbers, Optimization theory*; LAP LAMBERT Academic Publishing: Riga, Latvia, 2018.

22. Rosen, K.H. *Discrete Mathematics and Its Applications*, 7th ed.; McGraw-Hill Pub: New York, NY, USA, 1998; pp. 237–310.
23. Atiyah, M.F.; MacDonald, I.G. *Introduction to Commutative Algebra*; Addison-Wesley Series in Mathematics; CRC Press: Boca Raton, FL, USA, 1994.
24. Borwein, J.; Borwein, P.B. *Pi and the AGM: A Study in Analytic Number Theory and Computational Complexity*; Wiley: New York, NY, USA, 1987; p. 6.
25. Conrad, K. The Miller–Rabin Test. Available online: http://www.math.uconn.edu/~{}kconrad/blurbs/ugradnumthy/millerrabin.pdf (accessed on 20 April 2019).
26. Conrad, K. The Solovay–Strassen Test. Available online: http://www.math.uconn.edu/~{}kconrad/blurbs/ugradnumthy/solovaystrassen.pdf (accessed on 20 April 2019).
27. Impagliazzo, R.; Paturi, R. Exact Complexity and Satisfiability. In *Parameterized and Exact Computation, Proceedings of the International Symposium on Parameterized and Exact Computation, IPEC 2013, Sophia Antipolis, France, 4–6 September 2013*; Gutin, G., Szeider, S., Eds.; Lecture Notes in Computer Science; Springer: Cham, Switzerland, 2013; Volume 8246, pp. 1–3.
28. Baez, J.C. The Octonions. *Bull. Amer. Math. Soc.* **2002**, *39*, 145–205. [CrossRef]
29. Creignou, N.; Schmidt, J.; Thomas, M. Complexity Classifications for Propositional Abduction in Post's Framework. *J. Logic Comput.* **2012**, *22*, 1145–1170. [CrossRef]
30. Wang, X.Z.; Wang, R.; Xu, C. Discovering the relationship between generalization and uncertainty by incorporating complexity of classification. *IEEE Trans. Cybern.* **2018**, *48*, 703–715. [CrossRef] [PubMed]