Article

# Degree Reduction of $S-\lambda$ Curves Using a Genetic Simulated Annealing Algorithm 

Jing Lu ${ }^{1, *}$ and Xinqiang Qin ${ }^{2}$<br>1 College of Mathematics and Computer Application, Shangluo University, Shangluo 726000, China;<br>2 Department of Applied Mathematics, Xi'an University of Technology, Xi'an 710054, China; 229020@slxy.edu.cn<br>* Correspondence: 15991251163@163.com

Received: 8 November 2018; Accepted: 20 December 2018; Published: 25 December 2018


#### Abstract

The $S-\lambda$ Curves have become an important research subject in computer aided geometric design (CAGD), which owes to its good geometric properties (such as affine invariance, symmetry, and locality). This paper presents a new method to approximate an $S-\lambda$ curve of degree $n$ by using an $S-\lambda$ curve of degree $n-1$. We transform this degree reduction problem into the function optimization problem first, and then using a new genetic simulated annealing algorithm to determine the global optimal solution of the optimization problem. The method can be used to approximate $S-\lambda$ curves with fixed or unconstrained endpoints. Examples are given to verify the effectiveness of the presented algorithm; and these numeric examples show that the algorithm is not only easy to implement, but also offers high precision, which makes it valuable in practical applications.


Keywords: constraints; degree reduction of curve; genetic simulated annealing algorithm; optimization; $S-\lambda$ curve

## 1. Introduction

Many methods used in Computer Aided Geometric Design (CAGD) are closely associated with probability distributions, particularly discrete distributions (Johnson et al., 2005 [1]). For example, the Bernstein basis functions used in Bézier curves are related to the binomial distribution, the B-spline basis functions used in B-spline curves are connected with some stochastic processes (Dahmen et al., 1986 [2], Goldman et al., 2000 [3], Hu et al., 2018 [4-7]), the Poisson basis functions introduced by Goldman and Morin (2000) were developed from the Poisson distribution, and the Bernstein basis functions of negative degree presented by Goldman (1999) are taken from the negative binomial distribution [8]. These basis functions and other basis functions generated from discrete distributions, are important in the approximation process based on approximation theory (Hahn 1980 [9], Khan 1980 [10], Omey 1996 [11], De La Cal J. 1993 [12], Zeng 2005 [13], Zeng and Cheng 2001 [14]). On the basis of these points, Fan and Zeng [15] have explored a novel class of discrete distributions known as $S-\lambda$ distributions. The curves generated from $S-\lambda$ distributions are called $S-\lambda$ curves, which include Bézier curves, Poisson curves, rational Bézier curves, and many others.

Degree reduction of curves and surfaces has important practical significance because the approximation of curves and surfaces by other curves and surfaces of different degree is crucial to data conversion, transmission and compression (Watkins \& Worsey 1988 [16], Lutterkort 1999 [17]). However, the efficiency and stability of the existing degree reduction algorithms still have much room for improvement. Two classes of methods for this purpose have been studied extensively. The first class is geometric method based on control points, which is an inverse process of degree elevation and has been used to determine the control points of degree-reduced curves (Ren et al., 2007 [18]). This method utilizes the geometric properties of the Bézier curve itself and some mathematical theories
including generalized inverse matrix method, the best uniform approximation and the least squares approximation theory to achieve the degree reduction of Bézier curve (Cai et al., 2010 [19], Ahn et al., 2004 [20]). Moreover, some other scholars realize degree reduction of Bézier curves and surfaces by perturbing control vertices and using constrained optimization method (Lu et al., 2009 [21], Lu \& Wang 2006 [22], Delgado et al., 2007 [23], Ait-Haddou \& Bartoň 2016 [24]). The second class is algebraic method based on basis function transformation. In one example that utilized this approach, degree reduction was achieved by constraining Legendre polynomials (Liang et al., 2006 [25]). Similarly, based on Chebyshev polynomials (Rababah et al., 2006 [26], Lu et al., 2008 [27]) or Jacobi polynomials (Xu et al., 2008 [28]), the degree reduction of Bézier curves can be come true. A final interesting group of studies have focused on the problem of the approximate degree reduction of C-Bézier curves (Wang 2009 [29], Qin et al., 2013 [30]). Here we present a new method for the degree reduction of $S-\lambda$ curves approximately based on a genetic simulated annealing algorithm.

The outline of this paper is as follows. The definition of $S-\lambda$ curves is described in Section 2. In Section 3, we investigate the degree reduction of $S-\lambda$ curves using an intelligent algorithm. Finally, a short conclusion is given in Section 4.

## 2. Definition of $S-\lambda$ Curves

## 2.1. $S-\lambda$ Distributions and Basis Functions

Let $\left\{A_{j}\right\}_{j=0}^{m}$ be a given sequence of positive numbers, where $m$ may be equal to $+\infty$. The generating function ( $g f$ ) of the sequence is $S(x)=\sum_{j=0}^{m} A_{j} x^{j}$, which satisfies $S(0)=1$. Furthermore, suppose the convergence radius of $S(x)$ is $R$ and, let $\lambda(x)$ be a strictly continuous monotone increasing function satisfying $\lambda(0)=0$ and mapping $\left[0, R_{1}\right)$ onto $[0, R)$, where $R$ and $R_{1}$ may be equal to $+\infty$. $\lambda(x)$ is called the transformation factor ( $t f$ ). For $x \in\left[0, R_{1}\right)$, setting $P_{1, j}(x)=\frac{A_{j}[\lambda(x)]^{j}}{S[\lambda(x)]}$, and then $P_{1, j}(x) \in C\left[0, R_{1}\right)$ and $\sum_{j=0}^{m} P_{1, j}(x)=1$. Now we give the following definition (Fan and Zeng 2012 [15]).

Definition 1. Let $\xi$ be an integer random variable satisfying:

$$
\begin{equation*}
P(\xi=j)=P_{1, j}(x)=\frac{A_{j}[\lambda(x)]^{j}}{S[\lambda(x)]}(j=0,1, \ldots, m) \tag{1}
\end{equation*}
$$

$\left\{A_{j}^{(n)}\right\}_{j=0}^{m n}$ is an n-convolution of $\left\{A_{j}\right\}_{j=0}^{m}$ with the generating function:

$$
\begin{equation*}
S_{n}(x)=[S(x)]^{n}=\sum_{j=0}^{m n} A_{j}^{(n)} x^{j}, x \in\left[0, R_{1}\right) \tag{2}
\end{equation*}
$$

Let $\left\{\tilde{\zeta}_{i}\right\}_{i=1}^{\infty}$ be a sequence of independent random variables with the same distribution as $\xi$. Set $\eta_{n}=$ $\sum_{i=1}^{n} \xi_{i}$. Then by the convolution formula of probability, we can obtain:

$$
\begin{equation*}
P\left(\eta_{n}=j\right)=\frac{A_{j}^{(n)}[\lambda(x)]^{j}}{S_{n}[\lambda(x)]}(j=0,1, \ldots, m n) \tag{3}
\end{equation*}
$$

We call the probability distributions (3) $S-\lambda$ distributions. If we write:

$$
\begin{equation*}
P_{n, j}(x)=\frac{A_{j}^{(n)}[\lambda(x)]^{j}}{S_{n}[\lambda(x)]},\left(x \in\left[0, R_{1}\right), j=0,1, \ldots, m n\right), \tag{4}
\end{equation*}
$$

and then $P_{n, j}(x)$ are called the $S-\lambda$ basis functions corresponding to $S-\lambda$ distributions.

From Definition 1, two properties of $P_{n, j}(x) \geq 0$ and $\sum_{j=0}^{m n} P_{n, j}(x)=1$ are true obviously. According to Definition 1, we can construct different $S-\lambda$ distributions by selecting appropriate generating functions $S(x)=\sum_{j=0}^{m} A_{j} x^{j}$ and transformation factors $\lambda(x), x \in\left[0, R_{1}\right)$. We give two interesting instances as follows.

Example 1. Let $S(x)=1+$ xand $\lambda(x)=\frac{x}{1-x},(x \in[0,1))$; then $\left\{A_{j}\right\}_{j=0}^{1}=\{1,1\}$. By convolving $n$-times we get:

$$
\begin{equation*}
A_{j}^{(n)}=\sum_{k=0}^{j} A_{k}^{(n-1)} A_{j-k}^{(1)}=\binom{n}{j}, j=0,1, \ldots, n . \tag{5}
\end{equation*}
$$

By recombining the transformation factor and generating function, we have:

$$
S_{n}[\lambda(x)]=[1+\lambda(x)]^{n}=(1-x)^{-n} .
$$

Therefore, according to Definition 1 we obtain the classical Bernstein basis of degree n:

$$
\begin{equation*}
P_{n, j}(x)=\frac{A_{j}^{(n)}[\lambda(x)]^{j}}{S_{n}[\lambda(x)]}=\binom{n}{j} x^{j}(1-x)^{n-j}, \quad x \in[0,1), j=0,1, \ldots, n \tag{6}
\end{equation*}
$$

and the basis functions in (6) are shown in Figure 1a [15].


Figure 1. $S-\lambda$ basis functions in interval $[0,1)$ : Bernstein-Bézier and Szász-Mirakjan, where $n=3$.

Example 2. Letting $S(x)=e^{x}$ and $\lambda(x)=x, x \in[0,+\infty)$, we can get $S_{n}[\lambda(x)]=e^{n x}, m=\infty$ and $\left\{A_{j}\right\}_{j=0}^{\infty}=\left\{\frac{1}{j!}\right\}_{j=0}^{\infty}$. By taking $n$-times convolution operation of the infinite sequence $\left\{A_{j}\right\}_{j=0}^{\infty}$, we have:

$$
\begin{equation*}
A_{j}^{(n)}=\frac{n^{j}}{j!} \tag{7}
\end{equation*}
$$

Thus, we can get the Poisson basis functions:

$$
\begin{equation*}
P_{1, j}(x)=\frac{A_{j}[\lambda(x)]^{j}}{S[\lambda(x)]}=\frac{e^{-x} x^{j}}{j!} \tag{8}
\end{equation*}
$$

and the Szász-Mirakjan basis functions (see Figure 1b)

$$
\begin{equation*}
P_{n, j}(x)=\frac{A_{j}^{(n)}[\lambda(x)]^{j}}{S_{n}[\lambda(x)]}=\frac{e^{-n x}(n x)^{j}}{j!} \tag{9}
\end{equation*}
$$

## 2.2. $S-\lambda$ Curves

Definition 2. Let $\mathrm{V}=\left\{\boldsymbol{V}_{j}\right\}_{j=0}^{m n}$ be a given set of the control points, $\boldsymbol{V}_{j} \in R^{3}(j=0,1, \ldots, m n)$, then an $S-\lambda$ curve of degree $n$ can be defined as follows [15]:

$$
\begin{equation*}
\boldsymbol{C}(t)=\sum_{j=0}^{m n} P_{n, j}(t) \boldsymbol{V}_{j}, t \in\left[0, R_{1}\right), \tag{10}
\end{equation*}
$$

where $n \geq 1$, the number of m may be finite or infinite. However, since this paper focuses on practical applications, only the case of finite $m$ is considered here. Some illustrative $S-\lambda$ curves are discussed below.

Example 3. Suppose that $\left\{\boldsymbol{V}_{j}\right\}_{j=0}^{4}=\{(0,0) ;(-5,6) ;(5,12) ;(15,6) ;(10,0)\}, S(t)=\sum_{j=1}^{4} A_{j} t^{j}$ and $\lambda(t)=$ $\frac{t}{1-t}$ where $m=4, n=1$ and $R_{1}=1$. The corresponding $S-\lambda$ curve is $\boldsymbol{C}(t)=\sum_{j=0}^{4} P_{1, j}(t) \boldsymbol{V}_{j}$. Figure 2 shows the curves $\boldsymbol{C}(t)$ with different generating functions $S(t)$. From Figure 2, we can see clearly that if the $j$-th coefficient $A_{j}$ of the generating functions is even bigger, then the corresponding $S-\lambda$ curve is even closer to the control point $\boldsymbol{V}_{j}$ [15].


Figure 2. The $S-\lambda$ curves with same transformation factor and different generating functions.

## 3. Degree Reduction of S- $\lambda$ Curves Using an Intelligent Algorithm

### 3.1. The Degree Reduction Problem

Degree reduction for $S-\lambda$ curves of degree $n$ means finding a collection of control points on the $S-\lambda$ curve of degree $n-1$, that is:

$$
\begin{equation*}
\overline{\boldsymbol{C}}(t)=\sum_{j=0}^{m(n-1)} P_{n-1, j}(t) \overline{\boldsymbol{V}}_{j}, t \in\left[0, R_{1}\right) \tag{11}
\end{equation*}
$$

and the initial curve $\boldsymbol{C}(t)$ and the degree-reduced curve $\overline{\boldsymbol{C}}(t)$ satisfy the following condition:

$$
\begin{equation*}
\max _{0 \leq t<R_{1}} d[\boldsymbol{C}(t), \overline{\boldsymbol{C}}(t)]=\max _{0 \leq t<R_{1}}\|\boldsymbol{C}(t)-\overline{\boldsymbol{C}}(t)\|_{2} \rightarrow \min , \tag{12}
\end{equation*}
$$

where the operator $d[\cdot, \cdot]$ represents the Euclidean distance of the vector.
This problem can be translated into the following constrained optimization problem:

$$
\begin{gather*}
f(\mathrm{~V}, \overline{\mathrm{~V}}, t)=\min \max _{0 \leq t<R_{1}} d(\boldsymbol{C}(t), \overline{\boldsymbol{C}}(t))=\min _{0 \leq t<R_{1}} \max _{0}\|\boldsymbol{C}(t)-\overline{\boldsymbol{C}}(t)\|_{2}  \tag{13}\\
\text { s.t. } \boldsymbol{V}_{j}, \overline{\boldsymbol{V}_{j}} \in \mathrm{R}^{3}, t \in\left[0, R_{1}\right),
\end{gather*}
$$

where $\mathrm{V}=\left\{\boldsymbol{V}_{j}\right\}_{j=0}^{m n}$ and $\overline{\mathrm{V}}=\left\{\overline{\boldsymbol{V}}_{j}\right\}_{j=0}^{m(n-1)}$ represent the control point vectors of the initial $S-\lambda$ curve and the degree-reduced $S-\lambda$ curve, respectively.

### 3.2. Basic Principles of the Degree Reduction for $S-\lambda$ Curve

### 3.2.1. Elementary Algorithms

The simulated annealing algorithm is inspired by the annealing of hot physical objects, proposed by Metropolis et al. in 1953. It is an efficient way for finding a global optimal solution of a given problem. The process begins by generating an initial solution, we take it as the current solution; a nonlocal optimal solution in the vicinity of the current solution is then chosen and the algorithm decides whether to move to the new solution or not using a probability $p$. This algorithm not only has no prematurity, but also avoid trapping the optimal solution into local optimization. The temperature is gradually reduced (similar to the annealing of a real object), which makes the activity of the system decrease until it stabilizes in the vicinity of the global minimum with a probability of 1.

The concepts of genetic algorithms are based on Darwin's theory of biological evolution and Mendelian genetics [30-33]. They combine Darwin's theory of survival of the fittest with random exchange theory to create a statistical model that is applicable to the evolution of natural systems and, more generally, as a universal adaptive search method for solving optimization problems. In a genetic algorithm, the parameters of the problem need to be solved are coded as chromosomes, and then exchanging their information through iterative selection, crossover and mutation, until the 'ideal' chromosomes are obtained. However, during initial iterations of the genetic algorithm, it is possible for some superior chromosomes to rapidly propagate throughout the population, this phenomenon is prematurity.

In this paper, we attempt to overcome the prematurity problem of genetic algorithms by combining them with complementary simulated annealing methods. In addition, we present a method for selecting a fitness function, which makes the algorithm more efficient and increases the convergence speed to global optimal solution.

### 3.2.2. Initialization of the Control Parameters and the Group

To use a genetic algorithm or simulated annealing method, we must define the values of several control parameters including the size of the population, the maximum number of evolutionary generations, the probability of crossover, the probability of mutation, the initial temperature, the cooling factor, and the final temperature. Generally, since the parameters of a problem cannot convert into the terms of a genetic algorithm directly, it is necessary to translate the feasible solutions of the problem into chromosomes or individuals within genetic space first. Various coding techniques can be used for this purpose containing binary coding, Greg coding, real-valued coding, and multi-level parameter coding. We adopt real-valued coding in this work because most real-valued optimization
problems used this method do not require numerical conversion, which makes they can utilize genetic algorithms directly.

The populations of interest in this work are the feasible solutions of the control points for the degree-reduced $S-\lambda$ curve. The first step in the initialization process is to generate a random number $\alpha_{j} \in[0,1], j=0,1, \ldots, m(n-1)$, and then applying the following equation to compute a set of control points in the interval $\left[V_{\min }, V_{\max }\right]$.

$$
\begin{equation*}
\bar{V}_{j}=V_{\min }+\alpha_{j}\left(V_{\max }-V_{\min }\right), \quad j=0,1, \ldots, m(n-1), \tag{14}
\end{equation*}
$$

where $\bar{V}_{j}$ represents a coordinate component of the control point vector $\bar{V}_{j}=\left(\bar{x}_{j}, \bar{y}_{j}, \bar{z}_{j}\right)$ for the degree-reduced $S-\lambda$ curve $\overline{\boldsymbol{C}}(t)$; that is to say, $\bar{V}_{j}$ can be taken as the coordinate component $\bar{x}_{j}$, or $\bar{y}_{j}$, or $\bar{z}_{j}$. The symbol $V_{\min }$ and $V_{\max }$ is a coordinate component of the vector $V_{\min }=\left(x_{\min }, y_{\min }, z_{\min }\right)$ and $\boldsymbol{V}_{\max }=\left(x_{\max }, y_{\max }, z_{\max }\right)$, respectively; the coordinate components of the vectors $\boldsymbol{V}_{\min }$ and $\boldsymbol{V}_{\max }$ are calculated as follows:

$$
\left\{\begin{array}{l}
x_{\min }=\min \left\{x_{j}\right\}_{j=0^{\prime}}^{m n} y_{\min }=\min \left\{y_{j}\right\}_{j=0^{\prime}}^{m n} z_{\min }=\min \left\{z_{j}\right\}_{j=0^{\prime}}^{m n}  \tag{15}\\
x_{\max }=\max \left\{x_{j}\right\}_{j=0^{\prime}}^{m n} y_{\max }=\max \left\{y_{j}\right\}_{j=0^{\prime}}^{m n} z_{\max }=\max \left\{z_{j}\right\}_{j=0}^{m n}
\end{array}\right.
$$

### 3.2.3. Selection of a Fitness Function

The fitness function is a criterion to assess the fitness of an individual in the group; that is, measuring an individual whether can carried forward into the next iteration or not. It is generally obtained by transforming the objective function. To satisfy the approximation conditions of the $S-\lambda$ curve, we choose the fitness function of the following form:

$$
\begin{equation*}
f(t)=-\max \|\boldsymbol{C}(t)-\overline{\boldsymbol{C}}(t)\|=-\left[\sum_{j=0}^{s} d^{2}\left[\boldsymbol{C}\left(t_{j}\right), \overline{\boldsymbol{C}}\left(t_{j}\right)\right]\right]^{1 / 2} \tag{16}
\end{equation*}
$$

where $d\left[\boldsymbol{C}\left(t_{j}\right), \overline{\boldsymbol{C}}\left(t_{j}\right)\right]$ are Euclidean distances between $\boldsymbol{C}\left(t_{j}\right)$ and $\boldsymbol{C}\left(t_{j}\right)(j=0,1, \ldots, s) ; \boldsymbol{C}\left(t_{j}\right)$ and $\overline{\mathcal{C}}\left(t_{j}\right)(j=0,1, \ldots, s)$ are the different points on the initial $S-\lambda$ curve and the degree-reduced $S-\lambda$ curve, respectively.

### 3.2.4. Selection

The purpose of selection is to select superior individuals from the old group to form a new population, so that we can obtain a new generation. The likelihood of an individual be selected depends on its fitness value: the greater the fitness value, the higher the probability of selection. Various selection methods can be utilized including roulette wheel selection, random competition, best reserved selection and random selection without replacement. To ensure the optimal solutions are retained during search process, we use the standard random competition process for selection.

### 3.2.5. Crossover

Crossover involves selecting two individuals from the population at random and recombining the corresponding chromosomes such that the 'offspring' can inherit the superior characteristics of both parents, thereby creating even fitter individuals. An arithmetic hybridization system is adopted for crossover in this work. This approach works as follows. Suppose there are two parental vectors: $f_{1}=\left(m_{0}^{I}, \ldots, m_{n-1}^{I}\right), f_{2}=\left(m_{0}^{I I}, \ldots, m_{n-1}^{I I}\right)$. Using arithmetic hybridization system to create two offspring vectors $s_{1}=\left(v_{0}^{1} \ldots v_{n-1}^{1}\right)$ and $s_{2}=\left(v_{0}^{2} \ldots v_{n-1}^{2}\right)$ that satisfy:

$$
\left\{\begin{array}{l}
v_{i}^{1}=\beta_{i} m_{i}^{I}+\left(1-\beta_{i}\right) m_{i}^{I I}  \tag{17}\\
v_{i}^{2}=\gamma_{i} m_{i}^{I}+\left(1-\gamma_{i}\right) m_{i}^{I I}
\end{array}\right.
$$

where $\beta_{i}, \gamma_{i} \in[0,1]$ are random numbers.

### 3.2.6. Mutation

The mutation aim is to maintain the diversity of the population. Some of the genes in the chromosomes will undergo mutation and generate new genes. The mutation process gives genetic algorithms a local random search capacity and retains the diversity of the population, which lends to the absence of prematurity. In this work, mutation is implemented by using the usual method of substitution with random points, i.e. we mutate the point selected in the sub-solution vector and replace it with a point chosen randomly from the interval [ $p_{\min }, p_{\max }$ ]. More specifically, if $r_{1}<p_{m}$, then we proceed as follows:

$$
p_{i j}= \begin{cases}p_{i j}+\left(p_{i j}-p_{\max }\right) * r *(1-\text { gen } / \text { maxgen })^{2}, & r_{2} \geq 0.5  \tag{18}\\ p_{i j}+\left(p_{\min }-p_{i j}\right) * r *(1-\text { gen } / \text { maxgen })^{2}, & r_{2}<0.5\end{cases}
$$

where $p_{\min }$ and $p_{\max }$ are the lower boundary and upper boundary of the value for gene $p_{i j}$, respectively; $p_{m}$ is the probability of the mutation, "gen" denotes the current number of evolutionary generations, "maxgen" is the maximum permitted number of evolutionary generations, and $r, r_{1}, r_{2}$ are random numbers in the interval $[0,1]$.

### 3.2.7. Termination Conditions

The algorithm can be terminated if either of the two conditions is satisfied: (1) if the current temperature is below the termination temperature and and (2) if there is no difference between successive generations (indicating that further iteration will not yield a better result).

### 3.2.8. Setting Parameters

The genetic simulated annealing algorithm requires seven parameters to be set in advance: the size of the population (psize), the maximum number of evolutionary generations (maxgen), the probability of crossover $p_{c}$, the probability of mutation $p_{m}$, the initial temperature $T_{0}$, the cooling factor $q$, and the final temperature $T_{\text {end }}$. A large number of experiments show that these parameters can be set to: psize $=100$, maxgen $\in[10,30], p_{m}=0.1, T_{0}=100, q=0.8$ and $T_{\text {end }}=1$.

### 3.3. The Procedure of the Degree Reduction for $S-\lambda$ Curve

Degree reduction using the genetic simulated annealing method is performed via the following steps:

Step 1: Input the dimension of the initial $S-\lambda$ curve and the corresponding control points $\left\{\boldsymbol{V}_{0}, \boldsymbol{V}_{1}, \cdots \boldsymbol{V}_{m n}\right\}$, then draw the $\mathrm{S}-\lambda$ curve and its control polygon.

Step 2: Specify the control parameters including the population size, maximum number of evolutionary generations and so on.

Step 3: Generate the initial population and calculate the fitness value of each individual within it according to Equation (16).

Step 4: Initialize the next generation and set "gen" equals to 1.
Step 5: Use the genetic algorithm to obtain a new population through selection, crossover and mutation, and then calculate the fitness value of each individual in the new population, finally, utilize the simulated annealing algorithm to substitute the old individuals.

Step 6: If gen < maxgen, augment the value of gen by one and go to Step 5; otherwise, go to Step 7.
Step 7: Check the current temperature. If it is higher than the final temperature, let $T_{i+1}=q T_{i}$, and go to Step 4; if not, go to Step 8.

Step 8: Draw the degree-reduced $S-\lambda$ curve and its corresponding control polygon.

### 3.4. Results and Discussions

We assess the validity of the proposed algorithm by applying it to some numerical examples. The domain of the transformation factor is set to $R_{1}=1$, which is used in all cases of this paper. And the following expression is employed to measure the error between the original curve $C(t)$ and the degree-reduced curve $\overline{\boldsymbol{C}}(t)$ in this paper:

$$
\begin{equation*}
\varepsilon=\left[\int_{0}^{1}\|\overline{\boldsymbol{C}}(t)-\boldsymbol{C}(t)\|_{2}^{2} d t\right]^{1 / 2} \tag{19}
\end{equation*}
$$

A large number of numeric examples have been carried out on the proposed degree reduction method, and good degree reduction effects are achieved. In this section, we only give partial numeric example results. Note that the detailed information of simulation environment as follows: (1) the operating environment is personal computer with graphics card of i5/GTX1060 and processor frequency of 2.8 GHz ; (2) WINDOWS 7 operating system with memory of two gigabytes; and (3) using the programming language of Matlab 2010. Moreover, the run time of each example in this section is less than 300 ms (the reference time).

### 3.4.1. Degree Reduction of $S-\lambda$ Curves with Fixed Endpoints

Figures 3-5 show the result $S-\lambda$ curves after reducing the degree of the third, fourth and sixth-order $S-\lambda$ curves, respectively. Note that each group curves in these figures hold fixed endpoints. The control points for the third-order $S$ - $\lambda$ curve are $\{(1,1),(2,6),(4,12),(7,18),(11,22),(16,22),(20,18),(23,12),(25,6)$, $(26,1)\}$ and the generating function is $S(x)=1+2 x+5 x^{2}+3 x^{3}$. The control points for the fourth-order $S$ - $\lambda$ curve are $\{(1,1),(3,4),(5,3),(7,0),(9,2)\}$ and the generating function is $S(x)=1+5 x$. The control points for the sixth-order $S$ - $\lambda$ curve are $\{(1,1),(2,4),(3,5),(5,6),(7,5),(8,4),(9,1)\}$ and the generating function is $S(x)=1+3 x$.


Figure 3. Degree reduction of a third-order $S-\lambda$ curve with fixed endpoints. The control points of the degree-reduced curve are $\{(1,1),(2.6958,8.9393),(5.3362,18.5222),(13.7203,25.5980),(20.8985,21.6798)$, (24.6082, 7.1808), (26,1)\}.


Figure 4. Degree reduction of a fourth-order $S-\lambda$ curve with fixed endpoints. The control points of the degree-reduced curve are $\{(1,1),(3.7973,5.0336),(6.2139,0.0000),(9,2)\}$.


Figure 5. Degree reduction of a sixth-order $S-\lambda$ curve with fixed endpoints. The control points of the degree-reduced curve are $\{(1,1)$, ( $2.21112,4.5028),(3.2254,5.6096),(6.9537,5.6776),(7.5792,4.4878),(9,1)\}$.

### 3.4.2. Degree Reduction of $S-\lambda$ Curves with Unconstrained Endpoint

Figures $6-8$ show the result $S-\lambda$ curves obtained by reducing the degree of the third, fourth, and sixth-order $S-\lambda$ curves using proposed genetic simulated annealing algorithm, respectively. Note that each group curves in these figures possess unconstrained endpoints. The control points for the third-order $S$ - $\lambda$ curve are $\{(1,1),(2,6),(4,12),(7,18),(11,22),(16,22),(20,18),(23,12),(25,6),(26,1)\}$ and the generating function is $S(x)=1+2 x+5 x^{2}+3 x^{3}$. The control points for the fourth-order $S-\lambda$ curve are $\{(2,1),(1,2),(1,3),(2,4),(3,4.5),(4,4),(5,3),(5,2),(4,1)\}$ and the generating function is $S(x)=1+3 x+2 x^{2}$. The control points for the sixth-order $S$ - $\lambda$ curve are $\{(1,1),(2,4),(3,5),(5,6),(7,5),(8,4),(9,1)\}$ and the generating function is $S(x)=1+3 x$.


Figure 6. Degree reduction of a third-order $S-\lambda$ curve with unconstrained endpoints. The control points of the degree-reduced curve are $\{(0.9766,0.8736),(2.3848,10.1090),(6.2801,13.9343),(12.4859,34.1484)$, (21.7390, 16.0439), (24.2554, 8.6127), (25.9813, 1.0708)\}.


Figure 7. Degree reduction of a fourth-order $S-\lambda$ curve with unconstrained endpoints. The control points of the degree-reduced curve are $\{(1.9132,0.9329),(0.8636,2.5036),(1.0289,3.6672),(3.5579,4.4441)$, (3.9541, 4.5389), ( $5.7761,1.8075$ ), (3.8825, 1.0933)\}.


Figure 8. Degree reduction of a sixth-order $S-\lambda$ curve with unconstrained endpoints. The control points of the degree-reduced curve are $\{(1.0346,0.8772),(1.8500,4.3859),(3.8810,6.6002),(6.4763,4.1338)$, (7.6701, 5.4494), (9.0588, 0.7503)\}.

### 3.4.3. Quantitative Evaluation of the Degree Reduction Algorithm

The examples discussed in preceding section show that the presented algorithm is valid, which imply that it is a general tool to reduce degree of $S-\lambda$ curves approximately. In this section we attempt to evaluate its performance objectively based on the error between the degree-reduced curves and the corresponding originals.

Tables 1 and 2 show the degree reduction errors for the curves shown in Figures 2-4 and Figures 5-7, respectively. From the tables, we can see that the degree reduction errors of the genetic simulated annealing method are relatively small in all cases, which is consistent with the figures showed in Figures 2-7, and means the proposed algorithm is quite effective.

Table 1. Error values for degree reduction with fixed endpoints.

| Error | Figure 3 | Figure 4 | Figure 5 |
| :---: | :---: | :---: | :---: |
| $\varepsilon$ | 0.1402 | 0.9456 | 0.1914 |

Table 2. Error values for degree reduction with unconstrained endpoints.

| Error | Figure 6 | Figure 7 | Figure 8 |
| :---: | :---: | :---: | :---: |
| $\varepsilon$ | 0.0687 | 0.2347 | 0.1903 |

## 4. Conclusions

This paper presents a new method, which combines the geometric properties of the curves with a new genetic simulated annealing algorithm, to reduce degree of the $S-\lambda$ curves approximately. The performance of the algorithm is demonstrated by applying it to several examples; the results show that the proposed algorithm can achieve the degree reduction process efficiently and the generated degree-reduced curves are very close to the originals. It is worth noting that the proposed method in this paper is used for approximate degree reduction of $S-\lambda$ curves for the first time. And we will focus on studying the degree reduction of $S-\lambda$ surfaces in future work.

Author Contributions: J.L. contributed to the conception of the study and helped perform the analysis with constructive discussions. All authors of the manuscript have read and agreed to its content and are accountable for all aspects of the accuracy and integrity of the manuscript.
Funding: This work is supported by the National Natural Science Foundation of China (No. 11626185, and No. 11571275). This work is also supported by the Shaanxi Provincial Association of Science and Technology Young Talents Support Project in China (No. 20180508).

Acknowledgments: We thank to the anonymous reviewers for their insightful suggestions and recommendations, which led to the improvements of presentation and content of the paper.
Conflicts of Interest: The authors declare that they have no conflict of interest.

## References

1. Johnson, N.L.; Kemp, A.W.; Kotz, S. Univariate Discrete Distributions; John Wiley \& Sons: New York, NY, USA, 2005.
2. Dahmen, W.; Micchelli, C.A. Statistical encounters with B-splines. Contemp. Math. 1986, 59, 17-48.
3. Goldman, R.N. Urn models and B-splines. Constr. Approx. 2000, 4, 265-288. [CrossRef]
4. Hu, G.; Bo, C.; Wu, J.; Wei, G.; Hou, F. Modeling of Free-Form Complex Curves Using SG-Bézier Curves with Constraints of Geometric Continuities. Symmetry 2018, 10, 545. [CrossRef]
5. Hu, G.; Wu, J.L.; Qin, X.Q. A novel extension of the Bézier model and its applications to surface modeling. Adv. Eng. Softw. 2018, 125, 27-54. [CrossRef]
6. Hu, G.; Wu, J.L.; Qin, X.Q. A new approach in designing of local controlled developable H-Bézier surfaces. Adv. Eng. Softw. 2018, 121, 26-38. [CrossRef]
7. Hu, G.; Cao, H.X.; Zhang, S.X.; Guo, W. Developable Bézier-like surfaces with multiple shape parametersand its continuity conditions. Appl. Math. Model. 2017, 45, 728-747. [CrossRef]
8. Goldman, R.N. The rational Bernstein bases and the multirational blossoms. Comput. Aided Geom. Des. 1999, 16, 701-738. [CrossRef]
9. Hahn, L. Approximation by operators of probabilistic type. J. Approx. Theory 1980, 30, 1-10. [CrossRef]
10. Khan, R.A. Some probabilistic methods in the theory of approximation operators. Acta Math. Sci. Hung. 1980, 35, 193-203. [CrossRef]
11. Omey, E. Operators of probabilistic type. Theory Probab. Appl. 1996, 41, 219-225. [CrossRef]
12. De La Cal, J. On Stancu-Mühlbach operators and some connected problems concerning probability distributions. J. Approx. Theory 1993, 74, 59-68. [CrossRef]
13. Zeng, X.M. Approximation properties of Gamma operators. J. Math. Anal. Appl. 2005, 311, 389-401. [CrossRef]
14. Zeng, X.M.; Cheng, F. On the rate of approximation of Bernstein type operators. J. Approx. Theory 2001, 109, 242-256. [CrossRef]
15. Fan, F.L.; Zeng, X.M. S- $\lambda$ bases and $S-\lambda$ curves. Comput. Aided Des. 2012, 44, 1049-1055. [CrossRef]
16. Watkins, M.A.; Worsey, A.J. Degree reduction of Bézier curves. Comput. Aided Des. 1988, 20, 398-405. [CrossRef]
17. Lutterkort, D.; Peters, J.; Reif, U. Polynomial degree reduction in the L2-norm equals best Euclidean approximation of Bézier coefficients. Comput. Aided Geom. Des. 1999, 16, 607-612. [CrossRef]
18. Ren, S.L.; Zhang, K.Y.; Ye, Z.L. Matrix representation for degree reduction of Bézier curves. J. Eng. Math. 2007, 24, 1007-1014.
19. Cai, H.J.; Wang, G.J. Constrained approximation of rational Bézier curves based on a matrix expression of its end points continuity condition. Comput. Aided Des. 2010, 42, 495-504. [CrossRef]
20. Ahn, Y.J.; Lee, B.G.; Park, Y.; Yoo, J. Constrained polynomial degree reduction in the L2-norm equals best weighted Euclidean approximation of Bézier coefficients. Comput. Aided Geom. Des. 2004, 21, 181-191. [CrossRef]
21. Lu, L.Z.; Hu, Q.Q.; Wang, G.Z. An iterative algorithm for degree reduction of Bézier curves. J. Comput. Aided Des. Comput. Graph. 2009, 21, 1689-1693.
22. Lu, L.Z.; Wang, G.Z. Optimal multi-degree reduction of Bézier curves with $G^{2}$-continuity. Comput. Aided Geom. Des. 2006, 23, 673-683. [CrossRef]
23. Delgado, J.; Pena, J.M. Progressive iterative approximation and bases with the fastest convergence rates. Comput. Aided Geom. Des. 2007, 24, 10-18. [CrossRef]
24. Ait-Haddou, R.; Bartoň, M. Constrained multi-degree reduction with respect to Jacobi norms. Comput. Aided Geom. Des. 2016, 42, 23-30. [CrossRef]
25. Liang, X.X.; Zhang, C.M.; Lin, X. Constrained degree reduction of Bézier curve in $L_{\infty}$ Norm using basic curve and correction curve. J. Comput. Aided Des. Comput. Graph. 2006, 18, 401-405.
26. Rababah, A.; Lee, B.G.; Yoo, J. A simple matrix form for degree reduction of Bézier curves using Chebyshev-Bernstein basis transformations. Appl. Math. Comput. 2006, 181, 310-318. [CrossRef]
27. Lu, L.Z.; Wang, G.Z. Application of Chebyshev II-Bernstein basis transformations to degree reduction of Bézier curves. J. Comput. Appl. Math. 2008, 221, 52-65. [CrossRef]
28. Xu, S.P.; Bai, S.X.; Xiong, Y.H.; Zeng, W. Degree reduction of Bézier curves with constraints of asymmetry parametric endpoints continuity. J. Eng. Graph. 2008, 5, 91-97.
29. Wang, W.T. Degree reduction of C-Bézier curves. J. Zhejiang Univ. Sci. A 2009, 36, 396-400.
30. Qin, X.Q.; Wang, W.W.; Hu, G. Degree reduction of C-Bézier curve based on genetic algorithm. Comput. Eng. Appl. 2013, 49, 174-181.
31. Khuda Bux, N.; Lu, M.; Wang, J.; Hussain, S.; Aljeroudi, Y. Efficient Association Rules Hiding Using Genetic Algorithms. Symmetry 2018, 10, 576. [CrossRef]
32. Caponetto, R.; Fortuna, L.; Graziani, S.; Xibilia, M.G. Genetic algorithms and applications in system engineering: A survey. Trans. Inst. Meas. Control 1993, 15, 143-156. [CrossRef]
33. Lange, K. Poisson Approximation. In Geometric Modeling and Processing 2000. Theory and Applications. Proceedings; IEEE: Piscataway, NJ, USA, 2000; pp. 141-149.
© 2018 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (http:/ /creativecommons.org/licenses/by/4.0/).
