

DETERMINAÇÃO DA CURVA APROXIMADORA UTILIZANDO TRECHOS DE CURVAS DE BÉZIER E O RECOZIMENTO SIMULADO

EDSON KENJI UEDA*, MARCOS DE SALES GUERRA TSUZUKI*, ROGÉRIO YUGO TAKIMOTO*,
ANDRÉ KUBAGAWA SATO*, THIAGO DE CASTRO MARTINS*,
ROBERTO SÍLVIO UBERTINO ROSSO JR[†]

**Laboratório de Geometria Computacional
Departamento de Engenharia Mecatrônica e de Sistemas Mecânicos
Escola Politécnica da Universidade de São Paulo
Avenida Prof. Mello Moraes, 2231, CEP 05508-030 - São Paulo, SP, Brasil*

*[†]Universidade do Estado de Santa Catarina
Joinville, SC, Brasil*

Emails: ueda.edson@gmail.com, mtsuzuki@usp.br, takimotoyugo@gmail.com,
andre.kubagawa@gmail.com, thiago@usp.br, rosso@joinville.udesc.br

Abstract— In CAD (computer aided design), one important task is the determination of an approximation curve from a given sequence of points. This work proposes an algorithm to determine a piecewise Bézier curve that approximates a sequence of points. The optimization is done using the simulated annealing with adaptive neighborhood aiming at minimizing the discrepancy between the given sequence of points and the curve. The curve length is used as a regularization factor. The discrepancy between the given sequence of points and the curve is determined by the sum of the distance between each point from the sequence and the approximation curve. To calculate the distance from a point to the curve, the curve is discretized.

Keywords— Bézier curve, simulated annealing, curve fitting.

Resumo— Em CAD, uma tarefa muito importante é a determinação de curvas aproximadoras a partir de uma sequência fornecida de pontos. Este trabalho propõe um algoritmo que determina uma curva de Bézier por trechos que aproxima uma sequência fornecida de pontos. A otimização utiliza o recozimento simulado com vizinhança adaptativa com o intuito de minimizar a discrepância entre a sequência fornecida de pontos e a curva aproximadora. O comprimento da curva é utilizado como um fator de regularização. A discrepância entre a sequência fornecida de pontos e a curva é determinada pela somatória da distância de cada ponto da sequência até a curva aproximadora. Para o cálculo da distância de um ponto à curva, a curva é discretizada.

Palavras-chave— Curva de Bézier, recozimento simulado, ajuste de curvas.

1 Introdução

A determinação de curvas a partir de uma sequência de pontos é muito importante na engenharia. Ela é útil na engenharia reversa, na reconstrução de objetos, ao determinar curvas que auxiliam na determinação de superfícies (Piegl and Tiller, 1996). Outra aplicação da determinação de curvas a partir de uma sequência de pontos é no planejamento de trajetórias, definindo pontos onde o veículo deve passar e, conseqüentemente, a sua trajetória.

A tarefa de determinar uma curva a partir de uma sequência de pontos pode ser feita por duas diferentes abordagens, uma por interpolação (Maekawa et al., 2007) e outra por aproximação (Pandunata and Shamsuddin, 2010; Hasegawa et al., 2013). A primeira determina uma curva que passe por todos os pontos, enquanto a segunda determina uma curva que melhor aproxima uma sequência de pontos.

Este trabalho apresenta uma nova técnica de aproximação de curvas que utiliza trechos de curvas de Bézier para aproximar uma sequência de pontos com uma nova função objetivo.

Este trabalho é estruturado da seguinte

forma. A seção 2 possui uma breve revisão sobre curvas de Bézier e curvas de Bézier por trechos. A seção 3 explica o problema de ajuste de curvas utilizando somente uma curva. A seção 4 apresenta a função custo a ser minimizada, assim como a modelagem do problema. A seção 5 mostra os resultados, e as conclusões são feitas na seção 6.

2 Curva de Bézier

A curva de Bézier $\mathbf{P}(u)$ é dada por

$$\mathbf{P}(u) = \sum_{i=0}^n \mathbf{p}_i B_{i,n}(u) \quad (1)$$

onde $u \in [0, 1]$ e $\mathbf{p}_i$ são os pontos de controle, $n + 1$ é o número de pontos de controle e $B_{i,n}(u)$ são as funções bases calculadas com o polinômio de Bernstein definidas por

$$B_{i,n}(u) = \binom{n}{i} u^i (1-u)^{n-i} \quad (2)$$

onde $i = 0, \dots, n$ e $\binom{n}{i}$ é o binomial dado por

$$\binom{n}{i} = \frac{n!}{i!(n-i)!} \quad \binom{0}{0} \equiv 1. \quad (3)$$

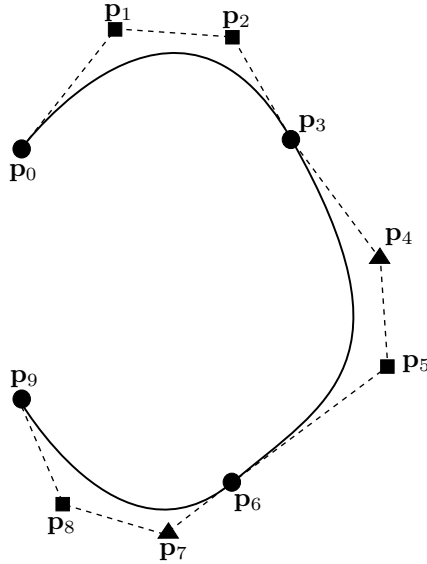


Figura 1: Sequência de três curvas cúbicas de Bézier. Os pontos p_3 e p_6 são pontos de transição entre curvas de Bézier.

2.1 Curva de Bézier por Trechos

Os pontos de controle de uma curva de Bézier modificam toda a curva. Uma forma de amenizar esta característica é utilizar sequências de curva de Bézier. Esta sequência é feita de modo que possua continuidade C^1 -fraca, que relaxa a condição de continuidade, onde as derivadas possuem a mesma direção, mas não necessariamente o mesmo módulo. Para criar estas curvas, o último ponto de controle de uma curva é o primeiro ponto de controle da curva seguinte, e para determinar o segundo ponto de controle de um segmento, considera-se que

$$p_{3i+4} = p_{3i+3} - (p_{3i+2} - p_{3i+3}) * \beta_{3i+4} \quad (4)$$

onde $i = 0, \dots, j-1$ e $j-1$ é o número de segmentos e β_i é um fator de continuidade. Um exemplo é mostrado na Fig. 1, que é a composição de três segmentos de curvas cúbicas de Bézier.

3 Ajuste com uma Única Curva de Bézier

O objetivo do problema de ajuste de curvas é aproximar uma sequência de pontos fornecida por uma curva pela determinação de seus pontos de controle ($p_0, \dots, p_n$). Para determinar estes pontos de controle, deve-se minimizar a discrepância entre a curva e a sequência de pontos fornecida. Isto corresponde a minimizar a somatória da distância de cada ponto da sequência até a curva. A função que descreve a função a ser minimizada é

$$f(p_0, \dots, p_n) = \sum_{k=1}^{m-1} |d_k - P(u_k)|^2 \quad (5)$$

onde $0 \leq k \leq m$ e $n+1$ é o número de pontos de controle, d é uma sequência de pontos com $m+1$

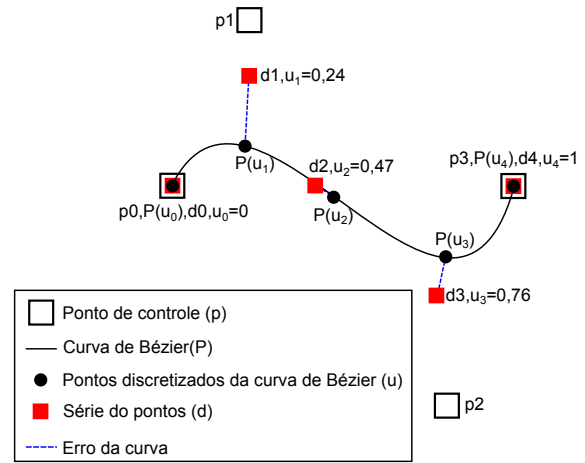


Figura 2: Distância do ponto à curva determinada por métodos aproximados.

elementos. $|d_k - P(u_k)|$ é a distância de um ponto d_k até a curva de Bézier. u_k é o valor do parâmetro que define o ponto da curva mais próximo ao ponto d_k , isto também é conhecido como projeção do ponto d_k sobre a curva. Os pontos d_0 e d_m não são considerados na somatória, uma vez que a curva de Bézier sempre passa pelo primeiro e último pontos de controle.

Existem vários métodos para determinar o parâmetro u_k , que define a projeção do ponto d_k sobre a curva. Hasegawa et al. (2013) e Pandunata and Shamsuddin (2010) calcularam este parâmetro de forma aproximada, utilizando a parametrização pela corda que aproxima u_k como sendo uma fração entre a somatória de cordas até o respectivo ponto d_k e a somatória de todas as cordas. Maekawa et al. (2007) determinaram este parâmetro considerando que o segmento que une $P(u_k)$ e d_k é ortogonal à tangente em $P(u_k)$, e isto pode ser resolvido pelo método de Newton-Raphson. Tavares et al. (2011) utiliza uma proposta distinta onde a curva é discretizada por uma sequência de pontos e todos estes pontos são comparados ao ponto fornecido d_k e determina-se o ponto da curva discretizada que mais se aproxima de d_k .

Os métodos propostos por Hasegawa et al. (2013) e Pandunata and Shamsuddin (2010) podem determinar pontos incorretos como mostra a Fig. 2. O método proposto por Maekawa et al. (2007) é mais vantajoso, mas demanda um tempo de processamento mais elevado. O método proposto por Tavares et al. (2011) é o mais vantajoso considerando que existe um balanço entre precisão e tempo de processamento. Quanto maior a discretização da curva, melhora a qualidade e aumenta o tempo de processamento.

Este trabalho procura aprimorar a proposta feita por Tavares et al. (2011) melhorando a qualidade sem aumentar o tempo computacional.

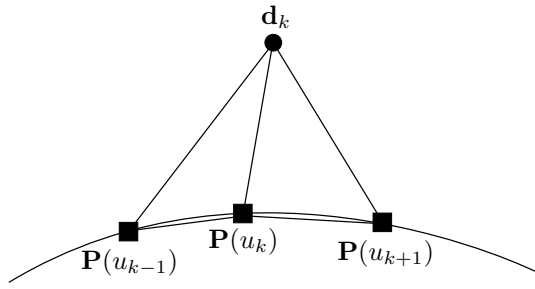


Figura 3: Encontrado o ponto $P(u_k)$ da curva discretizada que mais se aproxima de um ponto d_k , são definidos dois triângulos.

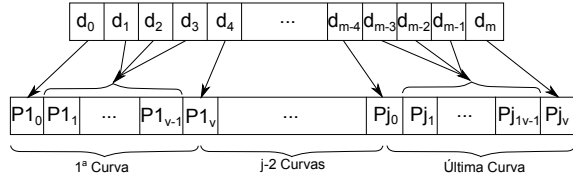


Figura 4: Mapeamento entre os pontos da sequência fornecida d_k e os pontos da curva discretizada Pj_v . Alguns pontos da curva discretizada são associados diretamente a pontos da sequência.

4 Definição da Função Custo

A expressão (5) considera que a curva aproximadora é exclusivamente definida pela discrepância entre a curva e os pontos fornecidos. Esta restrição não é suficiente para definir uma única solução e cria uma família de soluções. Neste trabalho, acrescenta-se um fator regularizador para que a curva de menor comprimento que minimize a expressão (5) seja encontrada.

A nova função custo é definida por

$$f(\mathbf{p}_0, \dots, \mathbf{p}_n) = W_1 \cdot \sum_{k=1}^{m-1} |d_k - P(u_k)|^2 + W_2 \cdot L \quad (6)$$

onde W_1 e W_2 são pesos que controlam a influência de cada parte da função de forma que $W_1 + W_2 = 1$, e L é a diferença o comprimento da curva e o comprimento da sequência de pontos fornecida. Ou seja, a nova função custo é a composição de dois termos: discrepância entre a curva e a sequência de pontos fornecida, e diferença entre o comprimento da curva e o comprimento da sequência de pontos fornecida.

O cálculo da distância entre o ponto e a curva foi baseado no método proposto por Tavares et al. (2011), onde a curva é discretizada. Ao se determinar o ponto da curva discretizada que mais se aproxima de um ponto dado d_k , são considerados dois triângulos como mostra a Fig. 3. Os pontos mostrados por quadrados são pontos da curva discretizada e o ponto mostrado por um círculo é um ponto da sequência fornecida de pontos. Em um primeiro passo, é feita uma busca para determinar qual ponto da curva discretizada é mais

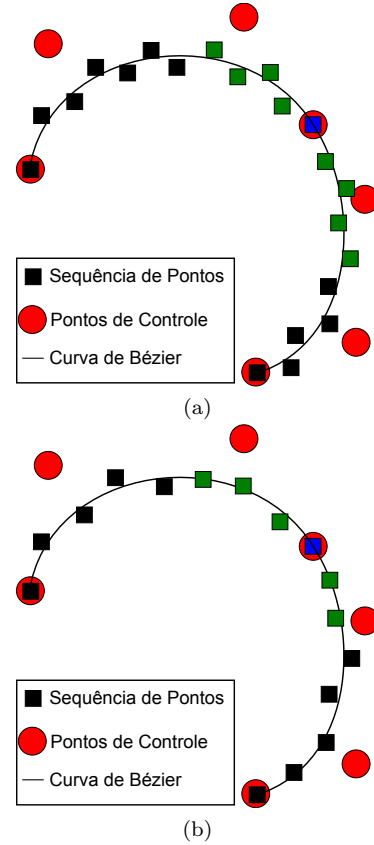


Figura 5: (a) Primeiro limite de deslocamento da junção de curvas; (b) Segundo limite de deslocamento da junção de curvas.

próximo do ponto da sequência d_k , este ponto será denominado por $P(u_k)$. Considerando a Fig. 3, dois triângulos são definidos, o primeiro triângulo é definido pelos pontos d_k , $P(u_k)$ e $P(u_{k-1})$; e o segundo triângulo é definido pelos pontos d_k , $P(u_k)$ e $P(u_{k+1})$.

Em seguida, é avaliado qual dos triângulos possuem a menor área. A seguir é feita uma verificação se o triângulo com menor área é agudo ou obtuso, especificamente nos ângulos entre os segmentos $d_k P(u_k)$ e $P(u_k) P(u_{k-1})$ para o primeiro triângulo, e $d_k P(u_k)$ e $P(u_k) P(u_{k+1})$ para o segundo triângulo. Se o triângulo de menor área possuir este ângulo como agudo, é calculada a altura do triângulo referente ao segmento definidos pelos dois pontos da curva discretizada e, esta altura é a distância entre ponto e a curva discretizada. Se o triângulo de menor área possuir este ângulo como obtuso será utilizado o outro triângulo para o cálculo da altura, e caso os dois triângulos possuírem ângulos obtusos, ambos os triângulos são descartados e é considerado que a distância entre o ponto e a curva discretizada é a distância entre d_k e $P(u_k)$.

Neste método, é considerada uma sequência de curvas de Bézier. Assim, uma curva de Bézier é associada a um subconjunto de pontos da sequência fornecida de pontos. Esta propriedade

é exibida na Fig. 4. Este fato permite definir uma segunda modificação na direção de acelerar a busca pelo ponto mais próximo. Dado um ponto da sequência $\mathbf{d}_k$, o ponto da curva discretizada mais próximo deve ser procurado entre os pontos inicial e final da curva de Bézier discretizada associada. No exemplo da Fig. 4, os pontos $\mathbf{d}_5$ e $\mathbf{d}_{m-4}$ são os pontos onde ocorrem transição de curvas de Bézier. Neste caso, cada curva de Bézier possui v pontos e existem j curvas. Assim, para os pontos $\mathbf{d}_1, \mathbf{d}_2, \mathbf{d}_3$ e $\mathbf{d}_4$, a busca pelo ponto mais próximo será realizada somente na primeira curva de Bézier definida pelos pontos $\mathbf{P1}_1, \dots, \mathbf{P1}_{v-1}$. E para os pontos $\mathbf{d}_{m-4}, \mathbf{d}_{m-3}, \mathbf{d}_{m-2}$ e $\mathbf{d}_{m-1}$, a busca pelo ponto mais próximo será realizada apenas na última curva de Bézier definida pelos pontos $\mathbf{Pj}_1, \dots, \mathbf{Pj}_{v-1}$.

Para o cálculo do comprimento uma vez que a curva está discretizada, é feita a somatória de todos os segmentos que compõem a curva discretizada.

4.1 Parametrização do problema

Para resolver o problema de otimização foi utilizado o recozimento simulado com vizinhança adaptativa proposto por Martins et al. (2012). O recozimento simulado com vizinhança adaptativa processa muito bem variáveis contínuas como coordenadas de pontos de controle ou dimensão de vetores. A dimensão do vetor está associada ao parâmetro β definido na expressão (4). Mas, ainda existem alguns parâmetros inteiros que correspondem a índices associados aos pontos da sequência que indicam os pontos de transição entre as curvas de Bézier. Utilizando como exemplo a curva com três curvas de Bézier cúbicas da Fig. 1, os parâmetros contínuos serão os seguintes:

- Coordenada x do ponto **p1** (p1x)
- Coordenada y do ponto **p1** (p1y)
- Coordenada x do ponto **p2** (p2x)
- Coordenada y do ponto **p2** (p2y)
- Fator de continuidade β_4 do ponto **p4**
- Coordenada x do ponto **p5** (p5x)
- Coordenada y do ponto **p5** (p5y)
- Fator de continuidade β_7 do ponto **p7**
- Coordenada x do ponto **p8** (p8x)
- Coordenada y do ponto **p8** (p8y)

e os parâmetros discretos são índices que correspondem a algum ponto da sequência fornecida de pontos, de modo que estes pontos da sequência serão considerados como pontos de transição entre curvas de Bézier.

4.2 Geração do novo candidato

Como já foi explicado, o recozimento simulado com vizinhança adaptativa proposto por Martins et al. (2012) processa muito bem variáveis contínuas. E para as variáveis inteiras foram definidas novas regras. Considera-se que inicialmente, os pontos de transição são selecionados segundo a divisão do número de pontos da sequência e o número de curvas de Bézier. Assim, a seleção de qual ponto da sequência será um ponto de transição não é aleatória. Para movimentar o índice associado ao ponto de transição, é gerado um valor aleatório entre $[0, 1]$, se o valor for menor que 0.5 diminui-se uma unidade o valor deste índice, caso contrário acrescenta-se uma unidade ao índice. Esta operação permite modificar o ponto da sequência fornecida de pontos que será utilizado como ponto de transição entre duas curvas de Bézier. Esta regra obedece as seguintes restrições:

- a mudança não pode ser maior que quatro unidades em relação à posição original, tanto em incremento e decremento. Isto foi adotado para evitar que surjam índices que modifiquem demasiadamente as posições iniciais e finais de cada curva de Bézier. Este limite é mostrado na Fig. 5(a), onde o quadrado em azul é a posição inicial de transição de duas curvas de Bézier e esta posição pode ser modificada para qualquer um dos quadrados em verde.
- cada parâmetro precisa ter uma distância mínima em relação ao parâmetro anterior e posterior, pois desta forma evitará a criação de curvas de Bézier demasiadamente curtas que aproximem poucos pontos da sequência fornecida de pontos. Assim foi adotada que a distância mínima é de 4 unidades. Este limite é mostrado na Fig. 5(b), onde o quadrado em azul é a posição inicial de transição de duas curvas de Bézier e os quadrados em verde mostram as posições possíveis dessa transição, note que de acordo com a primeira restrição, o limite de deslocamento poderia ser maior, mas como o início e o fim das curvas de Bézier são próximos este deslocamento é menor.

5 Resultados

O método desenvolvido foi utilizado em dois exemplos. Cada exemplo foi gerado ao criar uma curva no formato desejado e discretizando-os com 50 pontos, e um ruído foi adicionado nas coordenadas de cada ponto. Os formatos escolhidos para teste foram uma parábola e uma curva com auto-intersecção. Cada exemplo foi testado com diversos valores para W_1 e W_2 , e com três distintos números de curvas de Bézier para cada exemplo,

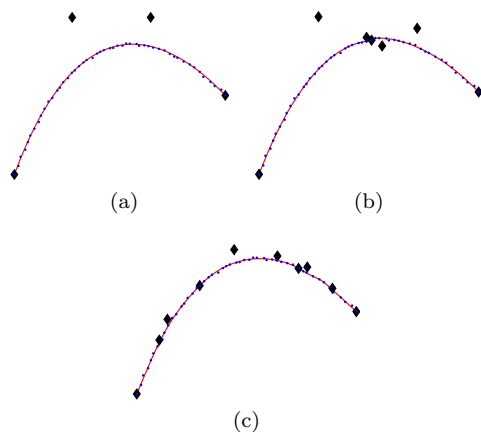


Figura 6: Exemplo de uma parábola com: (a) 1 curva de Bézier; (b) 2 curvas de Bézier; (c) 3 curvas de Bézier.

sendo que cada curva de Bézier foi discretizada com 100 pontos. Para o recozimento simulado com vizinhança adaptativa, foi adotado como critério de parada local um máximo de 1.000 candidatos gerados ou 500 candidatos aceitos, e como parada global um mínimo 1% de candidatos aceitos (ou condição de congelamento). As Figs. 6(a), (b) e (c) mostram a determinação da curva parabólica, respectivamente, com 1, 2 e 3 trechos de curvas de Bézier. As Figs. 7(a), (b) e (c) mostram a determinação da curva com auto-intersecção, respectivamente, com 5, 6 e 7 trechos de curvas de Bézier.

A Fig. 8(a) mostra um gráfico entre os valores da discrepância entre a sequência de pontos e a curva aproximadora com a variação do peso W_1 para o exemplo da curva parabólica. A discrepância é definida pela expressão (5). É possível observar que o valor da discrepância pouco se altera, mas com o aumento do número de curvas, a discrepância diminui. Inicialmente essa diminuição pode ser incorretamente associada ao maior número de pontos que a curva foi discretizada. Mas testes com 1 curva que possuía 300 pontos de discretização não foram obtidos resultados melhores que os testes com 3 curvas com 100 pontos cada. Assim, de fato, o que resulta em um menor valor de discrepância é o maior número de pontos de controle que permite controlar localmente regiões da curva. A Fig. 8(b) mostra a diferença entre os comprimentos com a variação do peso W_1 para o exemplo da curva parabólica, que pouco se variou com diversas configurações. A Fig. 8(c) mostra a variação do custo com a variação do peso W_1 para o exemplo da curva parabólica, e é possível observar que com o aumento de W_1 o custo também aumenta.

A Fig. 9(a) mostra um gráfico entre os valores da discrepância entre a sequência de pontos e a curva aproximadora com a variação do peso W_1

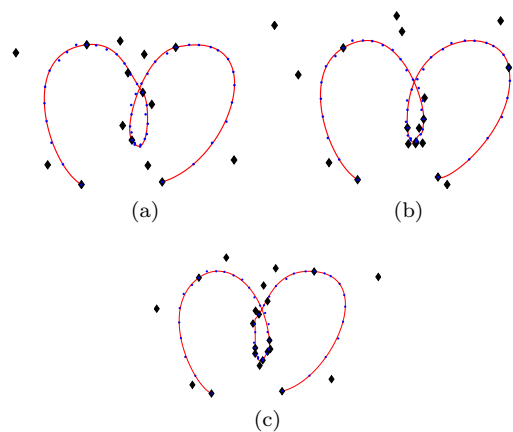


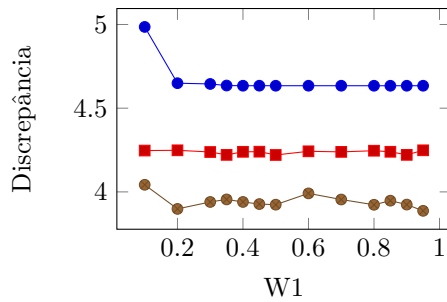
Figura 7: Exemplo de uma curva com auto-intersecção com: (a) 5 curvas de Bézier; (b) 6 curvas de Bézier; (c) 7 curvas de Bézier.

para o exemplo da curva com auto-intersecção. É possível observar que, novamente, com o aumento no número de curvas, ocorre uma diminuição na valor da discrepância, porém diferentemente do exemplo anterior, com o aumento do peso W_1 ocorreu uma diminuição da discrepância. A Fig. 9(b) mostra a diferença entre os comprimentos com a variação do peso W_1 e a Fig. 9(c) mostra a variação do custo com a variação do peso W_1 .

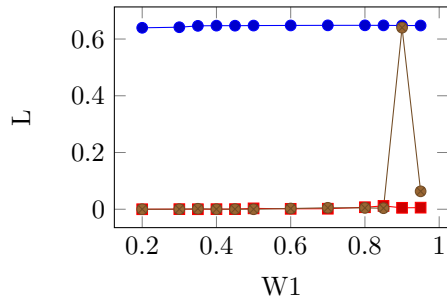
6 Conclusões e Trabalhos Futuros

Foi desenvolvido um algoritmo para se resolver o problema de ajuste de curvas, onde foi utilizado trechos de curvas de Bézier com continuidade C^1 -fraca junto com o recozimento simulado com vizinhança adaptativa. Também foi definido uma nova função objetivo, composta pela discrepância entre uma sequência de pontos e curva aproximadora e pela diferença de comprimentos da sequência de pontos e da curva. Este algoritmo, juntamente com o nova função objetivo foi aplicado com sucesso em dois exemplos, e pode-se esperar que com a utilização desta nova função objetivo o comprimento da curva aproximadora gerada funcione como regularização, evitando a criação de curvas com grande tortuosidade.

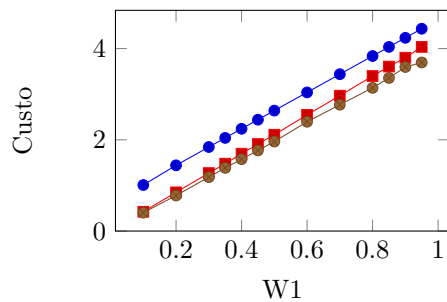
Como trabalhos futuros, é necessário fazer um estudo para determinar a quantidade de trechos de curvas ótima, e também a implementação deste algoritmo com outras curvas, como NURBS (*Non Uniform Rational Basis Spline*). Também será feita uma análise multi-objetivo do problema, assim será possível analisar o valor mais adequado de W_1 .



(a)



(b)



(c)

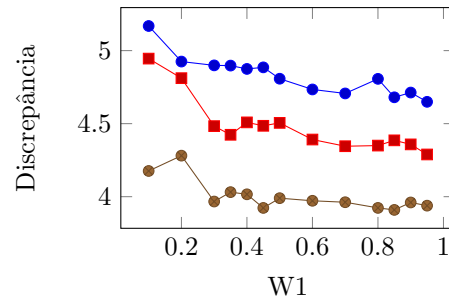
Figura 8: Resultados para o exemplo da parábola: (a) $W_1 \times$ Discrepância; (b) $W_1 \times$ Comprimento; (c) $W_1 \times$ Custo. Azul = 1 Curva, Vermelho = 2 Curvas, Marrom = 3 Curvas.

Agradecimentos

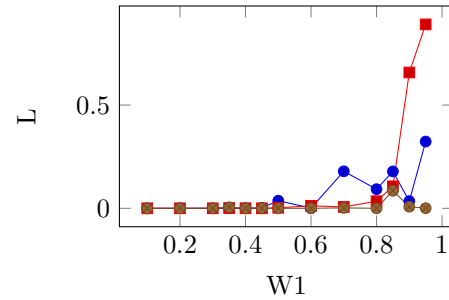
MSG Tsuzuki e TC Martins possuem suporte parcial do CNPq (respectivamente, processos 310.663/2013-0 e 306.415/2012-7). RY Takimoto e EK Ueda são suportados pelo CNPq (respectivamente, processos 150508/2015-8 e 140518/2015-0). AK Sato é suportado pela CAPES/PNPD.

Referências

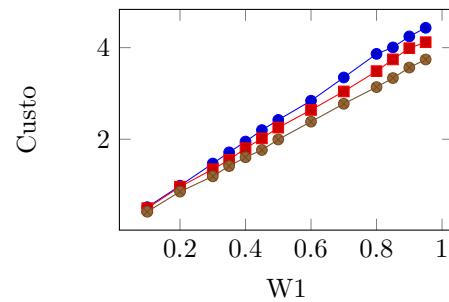
- Hasegawa, A. Y., Jr., R. S. U. R. and Tsuzuki, M. S. G. (2013). Differential evolution optimization for bezier curve fitting, *11th IFAC Workshop on Intelligent Manufacturing Systems*, pp. 233–238.
- Maekawa, T., Matsumoto, Y. and Namik, K. (2007). Interpolation by geometric algorithm, *Computer-Aided Design* **39**: 313–323.



(a)



(b)



(c)

Figura 9: Resultados para o exemplo da auto-intersecção: (a) $W_1 \times$ Discrepância; (b) $W_1 \times$ Comprimento; (c) $W_1 \times$ Custo. Azul = 5 Curvas, Vermelho = 6 Curvas, Marrom = 7 Curvas.

Martins, T., Sato, A. and Tsuzuki, M. (2012). *Adaptive Neighborhood Heuristics for Simulated Annealing over Continuous Variables*, InTech.

Pandunata, P. and Shamsuddin, S. (2010). Differential evolution optimization for bezier curve fitting, *Computer Graphics, Imaging and Visualization (CGIV), 2010 Seventh International Conference on*, pp. 68–72.

Piegl, L. and Tiller, W. (1996). Algorithm for approximate nurbs skinning, *Computer-Aided Design* **28**(9): 699 – 706.

Tavares, R., Martins, T. and Tsuzuki, M. (2011). Simulated annealing with adaptive neighborhood: A case study in off-line robot path planning, *Expert Systems with Applications* **38**(4): 2951 – 2965.