

# Otimização de código-g para Manufatura Aditiva: fabricação múltipla de scaffolds em série

**Alan Mateus Palos Losano<sup>1,2</sup>, José Luis Dávila<sup>2</sup>, Marcelo Fernandes de Oliveira<sup>2</sup>,  
Marcília Valéria Guimarães<sup>2</sup>**

{alan.losano, jose.davila, marcelo.oliveira, mvguimaraes}@cti.gov.br}

**<sup>1</sup>Faculdade de Engenharia Mecânica  
Universidade Estadual de Campinas (UNICAMP)**

**<sup>2</sup>Laboratório Aberto de Impressão 3D – Laprint**

**CTI/MCTI Renato Archer – Campinas/SP**

**Abstract.** Additive manufacturing (AM), popularly known as 3D printing, is a set of processes that allow the manufacture of three-dimensional objects by applying successive layers of material. G-code is one of the most used languages for controlling 3D printers, especially desktop printers. Since processes can be optimized for more efficient production, studies to modify the manufacturing trajectory implicit in the g-code are necessary. Thus, this article aims to implement an algorithm that solves the traveling salesman problem (TSP), for the travel moves of the print trajectory; in this case, it was applied to find the best movement path for material deposition.

**Key Words:** Additive manufacturing, g-code, 3D printing, travel salesman problem (TSP).

**Resumo.** A manufatura aditiva (MA), popularmente conhecida como impressão 3D, é um conjunto de processos que permitem a fabricação de objetos tridimensionais por meio da aplicação de camadas sucessivas de material. O código-g é uma das linguagens mais utilizadas para o controle de impressoras 3D, principalmente em impressoras desktop. Considerando que os processos podem ser otimizados para uma produção mais eficiente, estudos para modificar a trajetória de fabricação implícita no código-g são necessários. Assim, o objetivo do presente artigo é implementar um algoritmo que resolva o problema do caixeiro viajante (PCV), que nesse caso, foi aplicado para encontrar a melhor trajetória de movimentação para deposição de material.

**Palavras chave:** Manufatura aditiva (MA), código-g, impressão 3D, problema do caixeiro viajante (PCV).

## 1. Introdução

A manufatura aditiva (MA) é um conjunto de processos em que peças podem ser fabricadas por meio da superposição de camadas de um determinado material [1]. Com a característica de produzir geometrias complexas, as mesmas que podem ser customizadas, a MA configura-se como uma tecnologia transformativa, cujas aplicações permeiam diversas áreas do conhecimento, o que inclui aplicações na indústria automotiva, aeroespacial, farmacêutica, na bioengenharia, medicina, engenharia de tecidos, entre outras [2–6].

A larga aplicabilidade da MA permite uma flexibilidade considerável para o desenvolvimento de protótipos e peças funcionais, o que é associado a sua capacidade de diminuir significativamente as barreiras relacionadas à possibilidade de produção de peças complexas. O projeto de peças personalizadas e/ou otimizadas requer de ferramentas de *software CAD* (*Computer-Aided Design*) especializadas; quando analisamos a necessidade do mercado por processos automatizados, e que possam ser facilmente controlados pelo usuário, encontramos uma grande desvantagem nas ferramentas CAD tradicionais. Nesse aspecto, ferramentas de *software AAD* (*Algorithms-Aided Design*) contribuem de maneira significativa, já que combinam ferramentas CAD com programação, assim, objetos podem ser projetados em função de parâmetros definidos pelo usuário [7].

A utilização de *software* do gênero AAD alinha o ambiente CAD com diferentes linguagens de programação por meio de uma plataforma visual [8]. Esse *software* vem tomando cada vez mais espaço em diferentes tipos de aplicações; a possibilidade de programação em função dos mais diversos parâmetros geométricos permite controlar, modificar e customizar um objeto. Uma aplicação de interesse para a área de MA é o desenvolvimento e implementação de algoritmos que permitam compilar código-g para o controle de impressoras 3D [9,10]. Uma aplicação específica na área de engenharia tecidual é a compilação de código-g para fabricar *scaffolds*.

Os *scaffolds* são estruturas fundamentais na engenharia tecidual, eles têm a função temporária de reproduzir com fidelidade a matriz extracelular de tecidos e órgãos. Dessa maneira, permitem a proliferação, migração e diferenciação celular, além do transporte eficiente de nutrientes [11,12]. A construção de um ambiente adequado, que permita a regeneração tecidual é influenciada pela qualidade morfológica dos *scaffolds*. Assim, o controle dos parâmetros de fabricação é fundamental. No caso da impressão 3D, a compilação de códigos-g planejados de acordo com o processo de deposição de material, permitirá gerar estruturas com maior qualidade e morfologicamente adaptadas para cada material e finalidade [13].

Na literatura é reportado o *software* BioScaffolds PG [13,14], o mesmo que permite gerar códigos-g para a fabricação de *scaffolds*. Esse *software*, baseado na parametrização de geometrias e de trajetórias de deposição, permite um controle preciso dos diferentes aspectos estruturais e de impressão de *scaffolds*. Também, permite a deposição de amostras múltiplas, o que é principalmente aplicado na deposição de amostras diretamente em placas de cultura com diferentes números de poços. Nesse aspecto, é importante citar que o *software* realiza uma deposição sequencial de amostras, em que a trajetória de movimentação segue um padrão predefinido. Assim, foi identificada a necessidade de otimizar o percurso percorrido pelo cabeçote da impressora 3D, isso com o objetivo de diminuir o tempo e a distância para deposição de amostras entre poços. Nesse aspecto é fundamental ressaltar que diversas pesquisas estão sendo conduzidas para diminuir o tempo total de manufatura, o que é diretamente associado ao cálculo de trajetórias mais curtas durante a movimentação.

O planejamento das trajetórias de movimentação permite uma otimização mais eficiente do processo. Isto considerando que parâmetros como a velocidade de deposição de

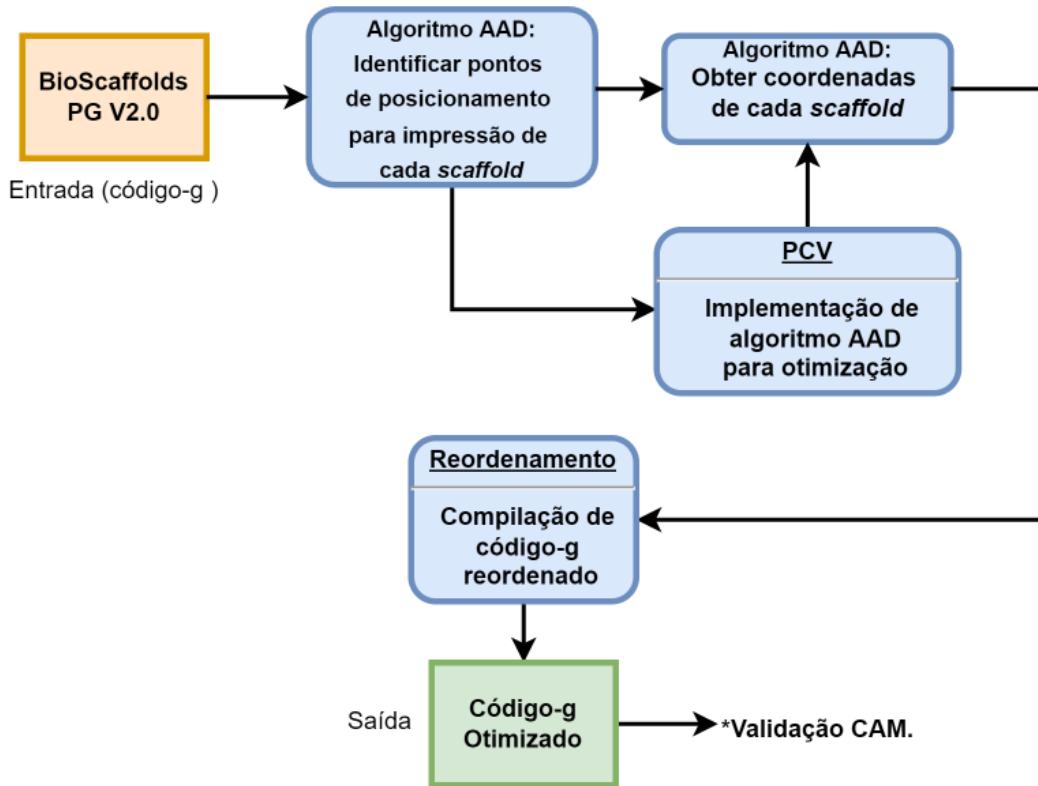
material, ou aumentar a velocidade de impressão, pode reduzir o tempo de processamento, no entanto, velocidades muito altas dificultam fixação do material na plataforma de construção, além de gerar instabilidades na deposição, o que afeta principalmente a morfologia do objeto fabricado [15]. Segundo Wang *et al.* [16], altas velocidades de deposição nos processos de extrusão de material podem afetar a microestrutura da peça formada, danificando suas propriedades mecânicas.

No contexto apresentado, uma solução adequada para otimizar o tempo de fabricação é: definir os parâmetros de impressão de maneira que não influenciem na qualidade da peça, além de aplicar algoritmos que determinem estratégias mais curtas de movimentação para posicionamento e reposicionamento durante o processo de impressão [17]. Nessa abordagem, ocorreria uma redução de movimentos desnecessários do bico, bem como na distância total percorrida, acarretando numa diminuição do tempo total de impressão. Tratando-se de um problema conhecido de otimização de rotas, em que é necessário percorrer diversos pontos pelo menor caminho possível, a logística usualmente utiliza um algoritmo que procura a solução do problema do caixeleiro viajante (PCV), que por meio de métodos matemáticos de organização de dados, consegue encontrar o melhor caminho possível, sem ter que percorrer a totalidade de caminhos prevista pela combinatória.

Nesse artigo apresenta-se a implementação de um método de solução para o Problema do Caixeleiro Viajante, o mesmo que utiliza como dado de entrada o código-g do *software* BioScaffolds PG V2.0. Dessa maneira, o objetivo principal é otimizar o processo de deposição de múltiplas amostras em um processo de impressão 3D em série. Para isso, a programação visual AAD foi utilizada conjuntamente com *scripts* desenvolvidos na linguagem de programação *Python*. Como resultado, a trajetória otimizada é calculada e aplicada no código-g de entrada. Assim a rota de impressão prevista é reorganizada, encontrando o caminho otimizado de deslocamento. Nessa abordagem é possível, além de garantir uma impressão mais eficiente, otimizar os deslocamentos do cabeçote, processo que pode ser replicado para peças de maior complexidade.

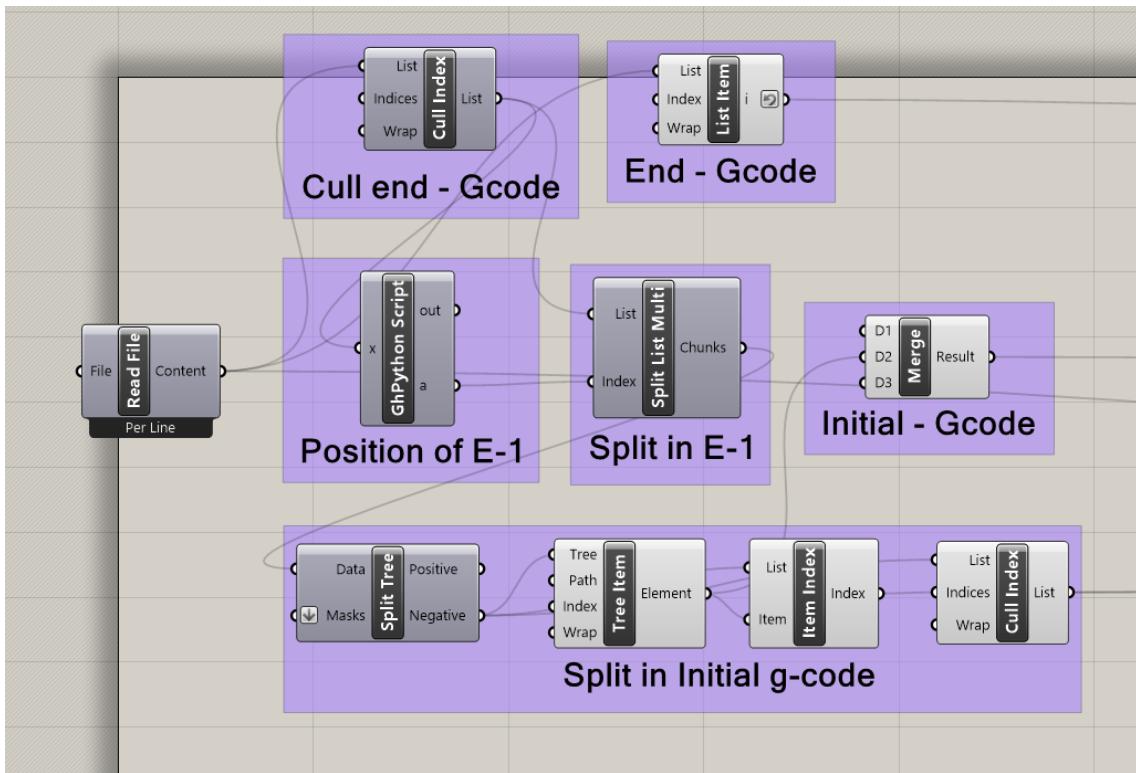
## 2. Metodologia

Programação visual AAD foi utilizada em conjunto com a linguagem *Python* na plataforma Grasshopper®, *plug-in* do *software* Rhinoceros 7. Para a geração de código-g foi utilizado o *software* livre BioScaffolds PG V2.0 [14]. Como ilustrado na Figura 1, o código-g contendo a trajetória de impressão de *scaffolds* em deposição múltipla, é o dado de entrada no algoritmo AAD implementado. Inicialmente, os dados são lidos e reorganizados, de maneira que os pontos de movimentação entre *scaffolds* são identificados. Em seguida, foi implementado o algoritmo que procura soluções do PCV. Após isso, o código-g otimizado é reorganizado e validado utilizando o *software* aberto CAMotics V1.2.0.



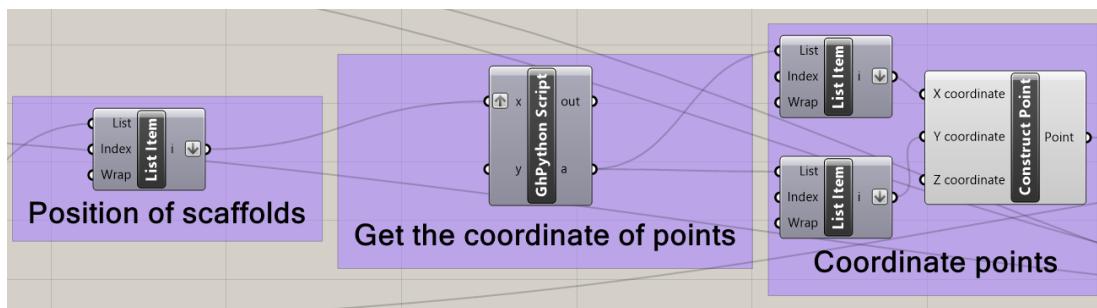
**Figura 1. Diagrama ilustrando a metodologia utilizada para otimização de código-g.**

Na Figura 2 são apresentadas as operações iniciais do algoritmo desenvolvido para a leitura e identificação de cada *scaffold* dentro do código-g de entrada. Nessas operações são detectados e removidos o protocolo inicial do código-g (relacionado com os parâmetros de referência e controle da impressora 3D) e o protocolo final (relacionado com a finalização do processo de impressão). A organização e armazenamento de dados no *Grasshopper* é realizada por meio de árvores; cada árvore é formada por “galhos”, os mesmos que podem conter listas e/ou sub listas de dados. Nesse caso, o código-g intermediário (protocolo do processo) é armazenado em uma estrutura de árvore, sendo que, o código para cada *scaffold* foi armazenado em um galho. Dessa maneira, é possível identificar cada lista de dados, o que permitirá a posterior reorganização do código após a otimização da trajetória de movimentação.



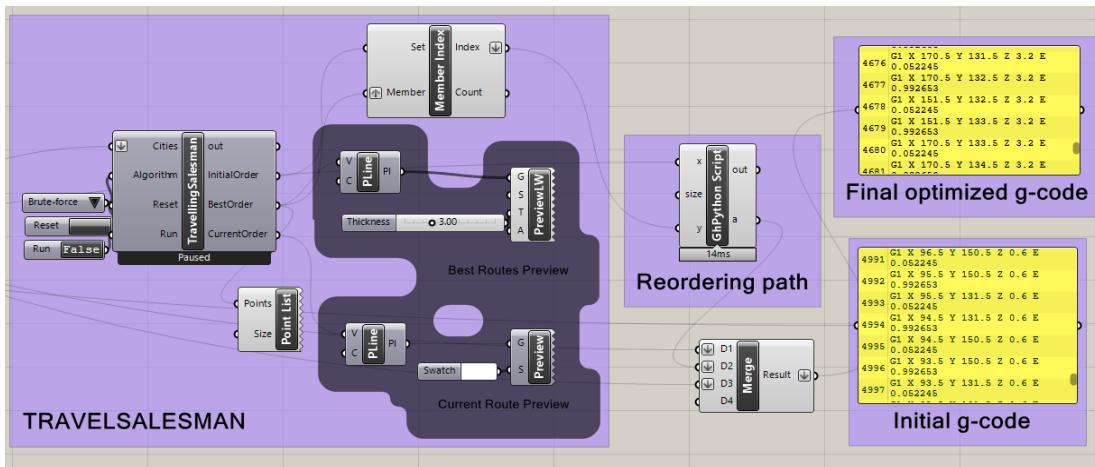
**Figura 2. Diagrama ilustrando a metodologia utilizada para otimização de código-g.**

Em seguida, o algoritmo reconhece as posições de cada *scaffold*, tomando como referência a deposição em placas de cultura. Para isso, as coordenadas ( $x, y$ ) são identificadas utilizando o algoritmo apresentado na Figura 3.



**Figura 3. Localização dos scaffolds na placa de cultura.**

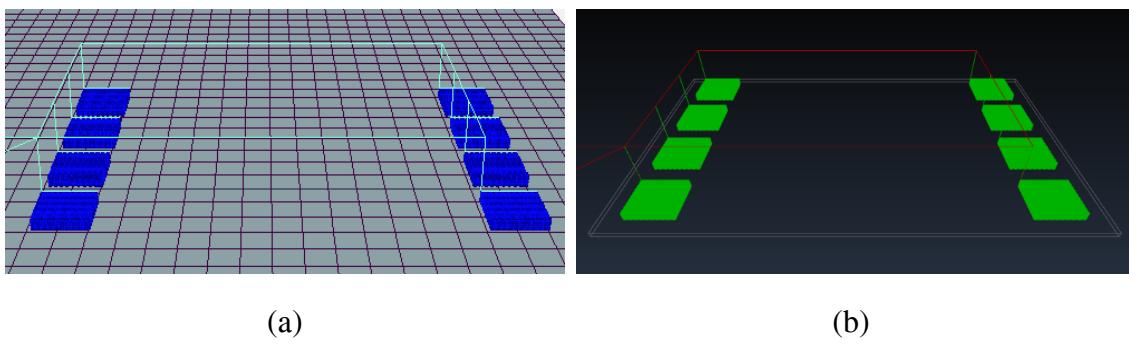
Finalmente, como apresentado na Figura 4, o algoritmo PCV [18] é integrado com o algoritmo que localiza as posições de cada *scaffold*; com essas coordenadas de entrada é calculada a melhor rota possível para percorrer todos os pontos. Assim, a trajetória inicial é reorganizada e o código-g otimizado é compilado. É importante ressaltar que o código-g final inclui: (i) Protocolo inicial, (ii) trajetória otimizada e (iii) protocolo final.

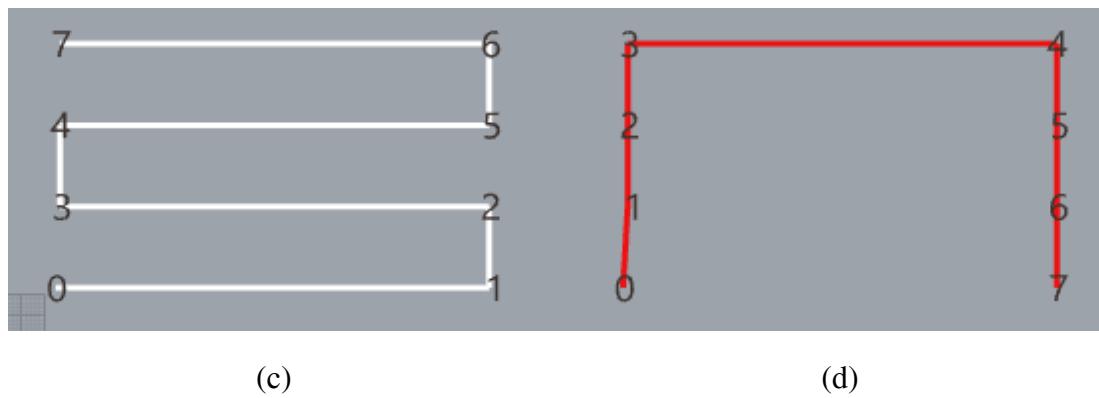


**Figura 4. Aplicação do algoritmo PCV e reorganização do código.**

### 3. Resultados e discussão

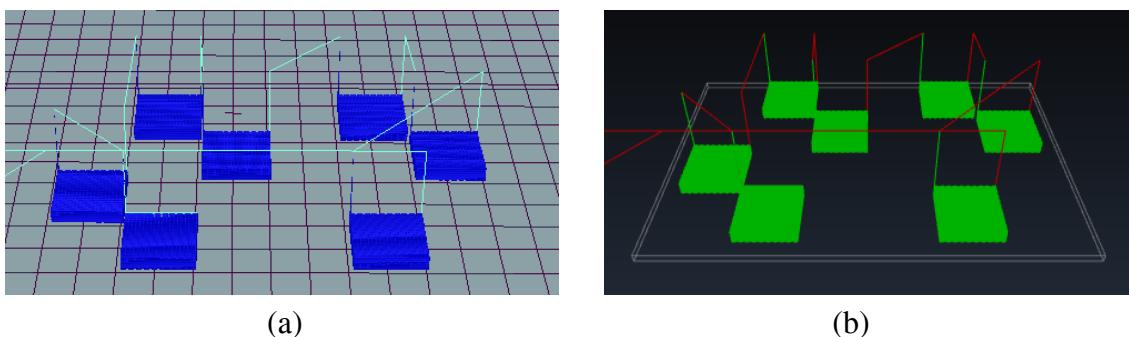
Na Figura 5 são apresentados os resultados obtidos para a fabricação de 8 *scaffolds* em uma placa de cultura de 24 poços. Os *scaffolds* foram posicionados apenas nas laterais da placa de cultura. Na Figura 5a é observado o código-g no *software* para impressão 3D Repetier-Host V2.2.2; como foi verificado, o código-g compilado é compatível, o que garante que poderá ser enviado à impressora 3D. A Figura 5b mostra o código-g no *software* CAMotics V1.2.0, no mesmo que foi verificada a sequência de movimentação para uma correta deposição de *scaffolds*. Na Figura 5(c,d) é mostrada a sequência de fabricação antes e depois da otimização da trajetória. Analisando o resultado obtido pela otimização, é possível verificar que é adotada uma trajetória mais curta em comparação com o código inicial. Nessa configuração, foi encontrado pelas estatísticas de impressão disponibilizadas pelo Repetier e pelo CAMotics, uma economia de aproximadamente 7 segundos no tempo total.

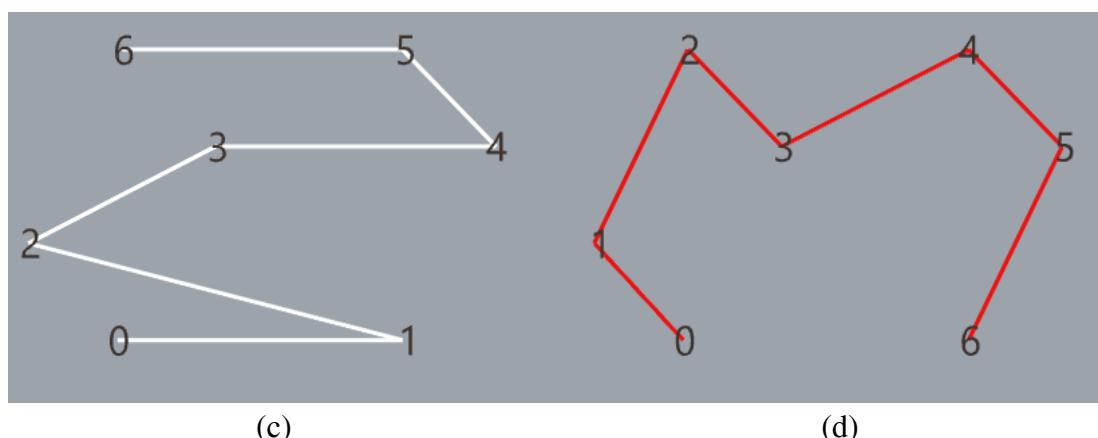




**Figura 5.** Preview do código-g otimizado para impressão de 8 scaffolds em placa de cultura de 24 poços: (a) Software Repetier-Host V2.2.2 e (b) CAMotics V1.2.0. (c) Trajetória de movimentação antes da otimização e (d) trajetória otimizada. A sequência de números mostra a ordem de fabricação de cada scaffold.

Na Figura 6 é apresentada uma configuração diferente para impressão de 7 scaffolds em uma placa de 24 poços. Nesse caso, o posicionamento foi realizado de maneira aleatória, com o objetivo de testar uma trajetória de movimentação com maior grau de complexidade. A Figura 6(a,b) apresenta a trajetória otimizada no teste de compatibilidade no Repetier e no teste de movimentação no CAMOTics. A Figura 6(c,d) apresenta a trajetória inicial e a trajetória otimizada, respectivamente. Apesar da otimização, é importante citar que o tempo requerido para achar uma solução ao PCV incrementa de forma fatorial para um maior número de pontos [19]. Nesse caso, a otimização de uma grande quantidade de amostras pode requerer um maior recurso computacional. Também, é importante citar que a aplicação dessa metodologia pode ser aplicada em amostras com fatiamento convencional; a otimização pode ser aplicada em cada camada de uma geometria, o que permitiria uma economia de tempo igual ao número de camadas multiplicado pelo tempo economizado em cada camada.





**Figura 6.** Preview do código-g otimizado para impressão de 7 scaffolds em placa de cultura de 24 poços: (a) Software Repetier-Host V2.2.2 e (b) CAMotics V1.2.0. (c) Trajetória de movimentação antes da otimização e (d) trajetória otimizada. A sequência de números mostra a ordem de fabricação de cada scaffold.

#### **4. Conclusão**

A integração de um algoritmo para achar a solução ao Problema do Caixeiro Viajante (PCV), com um *software* para compilação de código-g para impressão 3D de *scaffolds*, permitiu otimizar a rota de deposição sobre placas de cultura. Com uma redução de tempo e uma trajetória mais curta de movimentação, o algoritmo visual implementado é aplicável tanto para os usuários do programa BioScaffolds PG V2.0, como para aqueles que buscam alternativas para reduzir o tempo total de impressão. Com um estudo mais aprofundado dessa metodologia, é possível aplicar o código para organizar a deposição de material entre as camadas, tornando mais atrativo e eficiente o processo de manufatura aditiva para outros tipos de geometrias.

## 5. Referências

1. ISO/ASTM 52900:2015(E). Standard Terminology for Additive Manufacturing – General Principles – Terminology. 2015;
  2. Blakey-Milner B, Gradl P, Snedden G, Brooks M, Pitot J, Lopez E, et al. Metal additive manufacturing in aerospace: A review. Mater Des [Internet]. 2021;209:110008. Available from: <https://doi.org/10.1016/j.matdes.2021.110008>
  3. Haleem A, Javaid M. Additive Manufacturing Applications in Industry 4.0: A Review. J Ind Integr Manag. 2019;04(04):1930001.
  4. Li C, Pisignano D, Zhao Y, Xue J. Advances in Medical Applications of Additive Manufacturing. Engineering [Internet]. 2020;6(11):1222–31. Available from: <https://doi.org/10.1016/j.eng.2020.02.018>
  5. Moreno Madrid AP, Vrech SM, Sanchez MA, Rodriguez AP. Advances in additive manufacturing for bone tissue engineering scaffolds. Mater Sci Eng C [Internet]. 2019;100(March 2018):631–44. Available from:

<https://doi.org/10.1016/j.msec.2019.03.037>

6. Norman J, Madurawe RD, Moore CMV, Khan MA, Khairuzzaman A. A new chapter in pharmaceutical manufacturing: 3D-printed drug products. *Adv Drug Deliv Rev* [Internet]. 2017;108:39–50. Available from: <http://dx.doi.org/10.1016/j.addr.2016.03.001>
7. García-Dominguez A, Claver J, Sebastián MA. Optimization methodology for additive manufacturing of customized parts by fused deposition modeling (FDM). Application to a shoe heel. *Polymers (Basel)*. 2020;12(9).
8. Tedeschi A. *AAD Algorithms-Aided Design : Parametric Strategies Using Grasshopper* [Internet]. Le penseur publisher. 2014. 498 p. Available from: [http://encore.fama.us.es/iii/encore/record/C\\_\\_Rb2649316\\_\\_Sgrasshopper\\_\\_Orightresult\\_\\_U?lang=spi&suite=cobalt--](http://encore.fama.us.es/iii/encore/record/C__Rb2649316__Sgrasshopper__Orightresult__U?lang=spi&suite=cobalt--)
9. Zheng H, Darweesh B, Lee H, Yang L. Caterpillar - A Gcode translator in Grasshopper. *Proc 24th Conf Comput Aided Archit Des Res Asia* [Volume 2]. 2022;2:253–62.
10. Dávila JL, Inforçatti Neto P, Noritomi PY, Coelho RT, Lopes da Silva JV. Algorithms-aided design applied to the tool-paths generation for hybrid manufacturing. *Manuf Lett*. 2021;27:53–7.
11. Spicer CD. Hydrogel scaffolds for tissue engineering: The importance of polymer choice. *Polym Chem*. 2020;11(2):184–219.
12. Agrawal S, Srivastava R. Osteoinductive and osteoconductive biomaterials. *Racing for the Surface: Antimicrobial and Interface Tissue Engineering*. 2020. 355–395 p.
13. Dávila JL, Freitas MS, Neto PI, Silveira ZC, Silva JVL, d'Ávila MA. Software to generate 3-D continuous printing paths for the fabrication of tissue engineering scaffolds. *Int J Adv Manuf Technol*. 2016;84(5–8):1671–7.
14. Dávila JL, Manzini BM, Lopes da Fonsêca JH, Mancilla Corzo IJ, Neto PI, Aparecida de Lima Montalvão S, et al. A parameterized g-code compiler for scaffolds 3D bioprinting. *Bioprinting*. 2022;27(March):e00222.
15. Attalla R, Ling C, Selvaganapathy P. Fabrication and characterization of gels with integrated channels using 3D printing with microfluidic nozzle for tissue engineering applications. *Biomed Microdevices*. 2016;18(1):1–12.
16. WANG P, ZOU B, DING S, LI L, HUANG C. Effects of FDM-3D printing parameters on mechanical properties and microstructure of CF/PEEK and GF/PEEK. *Chinese J Aeronaut* [Internet]. 2021;34(9):236–46. Available from: <https://doi.org/10.1016/j.cja.2020.05.040>
17. Fok KY, Cheng CT, Ganganath N, Iu HHC, Tse CK. Accelerating 3D Printing Process Using an Extended Ant Colony Optimization Algorithm. *Proc - IEEE Int Symp Circuits Syst*. 2018;2018-May:13–7.
18. Travelling Salesman Problem [Internet]. Available from: <https://discourse.mcneel.com/t/genetic-algorithm-for-the-travelling-salesman-problem-i>

[n-python-completed/76627](#)

19. Kovacs, W. J., & Goodin, D. T. (1985). A statistical approach to the traveling salesman problem. *Transportation Research Part B: Methodological*, 19(3), 239–252.  
[https://doi.org/10.1016/0191-2615\(85\)90007-4](https://doi.org/10.1016/0191-2615(85)90007-4)