AJC
Academic Journal on Computing, Engineering and Applied Mathematics
EAM

2020
Volume 1 Issue 2

# Academic Journal on Computing, Engineering and Applied Mathematics

Image Designed by piksuperstar / Freepik

UFT
UNIVERSIDADE FEDERAL
DO TOCANTINS

ISSN: 2675-3588

# Universidade Federal do Tocantins

**Reitor**
Prof. Dr. Luís Eduardo Bovolato

**Vice-Reitor**
Prof. Dr. Marcelo Leineker Costa

**Pró-Reitoria de Graduação**
Prof. Dr. Eduardo José Cezari

**Pró-Reitoria de Pesquisa e Pós-Graduação**
Prof. Dr. Raphael Sanzio Pimenta

**Pró-Reitoria de Extensão e Cultura**
Profa. Dra. Maria Santana Ferreira dos Santos

**Pró-Reitoria de Administração e Finanças**
Me. Carlos Alberto Moreira de Araújo Júnior

**Pró-Reitoria de Assuntos Estudantis e Comunitários**
Prof. Dr. Kherlley Caxias Batista Barbosa

**Pró-Reitoria de Avaliação e Planejamento**
Prof. Dr. Eduardo Andrea Lemus Erasmo

**Pró-reitoria de Gestão e Desenvolvimento de Pessoas**
Profa. Dra. Vânia Maria de Araújo Passos

**Pró-Reitoria de Tecnologia da Informação e Comunicação**
Prof. Dr. Ary Henrique Morais Oliveira

**Direção do Campus de Palmas**
Prof. Dr. Moisés de Souza Arantes Neto

**Coordenação do Curso de Ciência da Computação**
Prof. Dr. Alexandre Tadeu Rossini da Silva

**Dados Internacionais de Catalogação na Publicação (CIP)**

Academic Journal on Computing, Engineering and Applied Mathematics (AJCEAM) [recurso eletrônico] / Universidade Federal do Tocantins, Curso de Ciência da Computação. – vol. 01, n. 02 ([maio/junho], 2020) – Palmas - TO, UFT, 2020. ISSN nº 2675-3588.

Quadrimestral no primeiro ano de publicação 2020
Semestral.
Disponível em:
https://sistemas.uft.edu.br/periodicos/index.php/AJCEAM/index

1. Ciência da Computação - periódico. 2. Matemática Aplicada. 3. Computação Aplicada. 4. Engenharias. 5. Ciências Exatas. I. Universidade Federal do Tocantins.

**CDD 22.ed. 004**

**Ficha Catalográfica elaborada por Edson de Sousa Oliveira – CRB/2 – 1069.**

# Expediente



**Realização**
Fundação Universidade Federal do Tocantins (UFT)
Quadra 109 Norte, Avenida NS-15, ALCNO-14 | Bloco III | sala 214 |Plano Diretor Norte | 77001-090 | Palmas / TO | Brasil

**Periodicidade**
Este periódico possui periodicidade semestral e utiliza a Licença Creative Commons 4.0 - CC BY-NC 4.0. Contudo, a publicação dos artigos em modalidade avançada ou ahead of print, ou seja, tão logo os manuscritos aprovados sejam editados para publicação, é possível. O AJCEAM não possui taxas de publicação, tanto pouco de submissão de manuscritos, sendo totalmente gratuita para autores e leitores.

**Indexadores**
Google Acadêmico, desde 9 de maio de 2020
International Standard Serial Number – ISSN, desde 28 de maio de 2020
Crossref, desde 7 de junho de 2020
Revistas de Livre Acesso – LivRe, desde 24 de junho de 2020

# Sumário

# Editorial (Português): Academic Journal on Computing, Engineering and Applied Mathematics

Tiago da Silva Almeida[1]

[1] *Universidade Federal do Tocantins, Palmas / TO, Brasil*

Em sua segunda edição, o AJCEAM traz algumas conquistas importantes. Em primeiro lugar, ao-longo do trimestre conseguimos obter nosso registro ISSN (*International Standard Serial Number*). O que nos permitiu alcançar a segunda conquista, o registro das publicações com o DOI (*Digital Object Identifier*) junto a CrossRef. Evidentemente não conseguiríamos sem a ajuda institucional da UFT. Nesse sentido, deixamos registrado nosso agradecimento ao professor Guilherme Nobre L. Nascimento, responsável pela diretoria de pesquisa e ao bibliotecário Edson de Sousa Oliveira, responsável pelo sistema de bibliotecas da UFT. Por fim, um agradecimento especial ao professor Rafael Lima de Carvalho, por sua enorme contribuição na arte da capa desta edição.

Esta edição traz cinco obras que consideramos importantes para construção do conhecimento e da pesquisa em Computação. O primeiro trabalho, intitulado "*Evaluation of Clause-Column Table method to design of MIC hazard-free in asynchronous finite state machines*", escrito por Almeida e Freitas [1], apresenta um estudo avaliativo de algoritmos heurísticos para a otimização de máquinas de estados finitosassíncronos, para obter o menor circuito possível. Encontrar os recursos mínimos na modelagem lógica é um problema NP-Difícil, pois o espaço de busca aumenta exponencialmente. Embora esse problema seja estudado há algum tempo, ainda há espaço para pesquisas mais recentes. Principalmente, em novos métodos computacionais para melhorar o desempenho na solução da otimização lógica em máquinas de estados finitos. Testes e resultados mostram que é possível sintetizar circuitos em um tempo razoável,mas com alguns erros lógicos.

O segundo trabalho, intitulado "*Computational experiment of error diffusion dithering for depth reduction in images*", escrito por Costa [2], descreve um estudo experimental com a utilização de pontilhamento com difusão de erros e uma análise de diferentes técnicas de meio-tom para alterar a profundidade de imagem. Essas técnicas são importantes para melhorar o poder de processamento de imagens em diferentes aplicações. A técnica de meio-tom é um processo que emprega padrões formados por pontos em preto e branco para reduzir o número de níveis de cinza em uma imagem. Devido à tendênciado sistema visual humano de suavizar a distinção entre pontos com tonalidades diferentes, os padrões de pontos em preto e branco produzem um efeito visual como se a imagem fosse composta por tons de cinza e escuro. Os resultados mostraram que a profundidade da imagem muda 1/8 por canal. Essa técnica de meio-tom pode ser usada para reduzir o peso da imagem, perdendo informações, mas obtendo bons resultados, dependendo do contexto.

O terceiro trabalho, intitulado "*5G energy efficiency for Internet of Things: a survey*", escrito por Barbosa, Kanehisa e Barros Filho [3], apresenta um estudo teórico sobre quinta geração da tecnologia de telecomunicações (5G), ainda em concepção em vários países, como o Brasil, e sua aplicação em Internet das Coisas (IoT – *Internet of Things*), dado o aumento crescente na demando de mercado. Dispositivos IoT podem atingir um trilhão de nós e, portanto, requerem conexões de rede capazes de lidar com um grande número de nós conectados e com transmissão de baixa energia. A 5G é um conceito-chave para atender a esses requisitos, pois novos aplicativos e modelos de negócios exigem novos critérios, como segurança confiável, latência ultra baixa, ultra confiabilidade e eficiência energética. Este artigo

descreve uma revisão das principais tecnologias, como Cloud, Software Defined Network, Device-to-Device Communication, Evolved Package Core e Network Virtual Function Orchestration, planejadas para serem aplicadas nos campos 5G e IoT.

O quarto trabalho, intitulado "*A Systematic Review About Use of TensorFlow for Image Classification and Word Embedding in the Brazilian Context*" escrito por Padilha e de Lucena [4], descreve um estudo, também teórico, sobre as principais pesquisas no Brasil sobre o uso do framework TensorFlow no processamento de imagem e incorporação de palavras em aplicações de Inteligência Artificial (IA). Os autores utilizaram o Google Scholar como mecanismo de pesquisa acadêmica e 90 trabalhos foram recuperados inicialmente. No entanto, restaram apenas 12 para leitura e obtenção das informações principais. Título, ano de publicação, domínio usado, tipo de aplicação e escopo coberto foram coletados de artigos recuperados para auxiliar nos estudos na área de IA e disseminar o potencial dessa estrutura para novos desafios na área.

Por fim, o quinto trabalho, intitulado "*Multilayer Perceptron Perceptron optimization through Simulated Annealing and Fast Simulated Annealing*", escrito por Camelo e Carvalho [5], descreve um estudo experimental usando as metaheurísticas Simulated Annealing e Fast Simulated Annealing para otimização dos hiperparâmetros de redes neurais artificiais do tipo MLP (Multi Layer Perceptron). O MLP é ummodelo de rede neural clássico e amplamente utilizado em aplicações de aprendizado de máquina. Como a maioria dos classificadores, os MLPs precisam de parâmetros bem definidos para produzir resultados otimizados. Os MLPs foram otimizados para produzir um classificador para o banco de dados MNIST. O experimento mostrou que o Fast Simulated Annealing produziu um MLP melhor, usando menos tempo computacional e menos avaliações da função de fitness.

A segunda edição já apresenta mais publicações que a primeira e com uma diversidade maior na origem dos trabalhos. São todos trabalhos sérios e com grande potencial. A partir de agora, caberá aos leitores avaliá-los e se julgá-los coerentes, citá-los. Boa leitura.

## REFERÊNCIAS

[1] T. d. S. Almeida and A. Luiz Gomes de Freitas, "Evaluation of clause-column table method to design of mic hazard-free in asynchronous finite state machines," *Academic Journal on Computing, Engineering and Applied Mathematics*, vol. 1, no. 2, p. 1–8, jun. 2020. [Online]. Available: https://sistemas.uft.edu.br/periodicos/index.php/AJCEAM/article/view/9304

[2] L. Rezende Costa, "Computational experiment of error diffusion dithering for depth reduction in images," *Academic Journal on Computing, Engineering and Applied Mathematics*, vol. 1, no. 2, p. 9–13, jun. 2020. [Online]. Available: https://sistemas.uft.edu.br/periodicos/index.php/AJCEAM/article/view/9533

[3] R. Kanehisa, F. Barbosa, and A. de Castro, "5g energy efficiency for internet of things : a survey," *Academic Journal on Computing, Engineering and Applied Mathematics*, vol. 1, no. 2, p. 14–23, jun. 2020. [Online]. Available: https://sistemas.uft.edu.br/periodicos/index.php/AJCEAM/article/view/9545

[4] T. P. Pereira Padilha and L. E. Alves de Lucena, "A systematic review about use of tensorflow for image classification and word embedding in the brazilian context," *Academic Journal on Computing, Engineering and Applied Mathematics*, vol. 1, no. 2, p. 24–27, jun. 2020. [Online]. Available: https://sistemas.uft.edu.br/periodicos/index.php/AJCEAM/article/view/9466

[5] P. H. C. Camelo and R. L. de Carvalho, "Multilayer perceptron optimization through simulated annealing and fast simulated annealing," *Academic Journal on Computing, Engineering and Applied Mathematics*, vol. 1, no. 2, p. 28–31, jun. 2020. [Online]. Available: https://sistemas.uft.edu.br/periodicos/index.php/AJCEAM/article/view/9474

# Evaluation of Clause-Column Table method to design of MIC hazard-free in asynchronous finite state machines

Tiago da Silva Almeida[1] e André Luiz Gomes de Freitas[1]

[1] *Universidade Federal do Tocantins, Curso de Ciência da Computação, Tocantins, Brasil*

Data de recebimento do manuscrito: 18/05/2020
Data de aceitação do manuscrito: 08/06/2020
Data de publicação: 12/06/2020

**Abstract**— Asynchronous finite state machines are of great interest because they use fewer transistors to manufacture them. However, find the minimum resources in logic modeling is an NP-Problem, since the search space increase exponentially. Although this problem is studied for some time, there is still space for newer researches. Mostly, in new computational methods to improve performance in solving logical optimization in finite state machines. Thus, this paper presents a study and evaluation of heuristic algorithms for the optimization of asynchronous finite state machines, to obtain the smallest possible circuit. Thus, the Clause-Column Table and Quine-McCluskey algorithms are combined in order to propose an algorithm capable of minimizing asynchronous sequential circuits. Tests and results show that it is possible to synthesize circuits in a reasonable time, but with some logical errors. It may be concluded that it still needs research, even though it is not such a recent line of research.

**Keywords**—Assincronous Finite State Machine, Clause-Column Table, Heuristic, Quine-McCluskey

## I. Introdução

Com o avanço da tecnologia, o uso de sistemas digitais tem sido amplamente difundido, principalmente dos circuitos eletrônicos digitais. Como resultado desse avanço é necessário lidar com circuitos digitais cada vez mais complexos. Consequentemente ainda é necessário que os custos e o tempo de implementação sejam cada vez menores [1]. Além disso, os softwares mais robustos, como os que são baseados em aprendizagem profunda ou que processam grandes quantidades de dados, necessitam de hardwares cada vez mais robustos, às vezes até hardwares especializados como os ASICs (*Application Specific Integrated Circuits*), para que possam obter um desempenho melhor. Por causa disso, a área de otimização é cada vez mais utilizada para obter circuitos menores e mais eficientes sem a necessidade de utilizar componentes menores, ou seja, diminuir a escala de integração.

A área de minimização de lógica não é nova, mas é cada vez mais necessária uma lógica eficiente e de baixo custo em diversas áreas, como por exemplo, na minimização de circuitos e sistemas VLSI (*Very Large Scale Integration*), ULSI (*Ultra Large Scale Integration*), além de outros [2].

Na maioria desses sistemas são necessários circuitos que sejam capazes de salvar dados e realizar operações aritméticas, lógicas sobre esses dados, ou mesmo algum controle especializado dentro do sistema. Esse tipo de circuito é chamado de circuito sequenciais, porque a sequência de operações define o comportamento do sistema. As Máquinas de Estados Finitos (MEF) é um modelo abstrato que descreve o comportamento dos circuitos sequenciais [3].

Com as MEFs podemos abstrair determinados circuitos, e pensar em seu comportamento como um algoritmo. Onde o algoritmo pode ser representado como uma máquina, ou grafo, que possui vários estados, e entre cada estado possui transições que ocorrem de acordo com as entradas recebidas. A partir dessa representação é possível chegar ao modelo físico do circuito.

O modelo físico obtido através de uma MEF pode ser internamente ou externamente síncronos em relação ao sistema ou assíncrono. O modelo abstrato que descreve o circuito assíncrono também é chamado de Máquina de Estado Finito Assíncrona (MEFA), ou também chamada como Máquina de Estados Finitos em Modo Fundamental.

Os circuitos assíncronos são uma alternativa para resolver alguns dos problemas que os sistemas síncronos complexos possuem, que são geralmente relacionados ao sinal do relógio (*clock*) global, entre outros. Além de possuir outras vantagens em relação aos síncronos, como sem distorções no relógio, sem distribuição de relógio, menor consumo de energia, maior modularização e mais robustez a variações de temperatura e menor interação com efeitos eletromagnéticos [4].

Como normalmente todo hardware, seja ele VLSI ou ASIC ou um processador convencional, possui partes ou são inteiramente feitos utilizando MEFA, obtê-los com circuitos

Dados de contato: Tiago da Silva Almeida, tiagoalmeida@uft.edu.br

de tamanho ótimo ou o menor possível pode levar a estruturas mais eficientes e mais baratas. Porém, minimização de MEF ou MEFA é complexo e possui um alto custo computacional, por causa disso o estudo de novos métodos para realização dessa tarefa está cada vez mais presente, para obter ferramentas com um custo computacional menor e que geram resultados satisfatório.

Uma MEFA pode ser expressa somente por funções booleanas, pois, é composta apenas por componentes com estruturas simples como as portas lógicas. O que torna possível aperfeiçoar tanto o espaço, utilizando minimização lógica combinatória, como também o custo, visto que os componentes lógicos possuem um custo inferior ao de estruturas mais complexas utilizadas em MEF (como elementos de memória).

Na literatura existem algoritmos de minimização de funções booleanas considerados exatos, que geram implicantes primos para então obter uma solução ótima [1]. Com base nisso, esse projeto tem como objetivo desenvolver um algoritmo baseado no método *Clause-Column Table* e no método Quine-McCluskey de forma a utiliza-los para otimizar uma MEFA.

O número de variáveis e termos de uma expressão booleana estão relacionados com o tamanho do circuito combinacional da MEFA, o que está relacionado com custo e o desempenho. Assim, objetivo geral deste trabalho é apresentar um algoritmo baseado em minimização de funções booleanas de *Clause-Column Table* e no algoritmo de Quine-McCluskey, com o propósito de realizar sínteses das MEFA e realizar uma avaliação comparativa dos resultados produzidos com os apresentados na literatura.

## II. CIRCUITOS SEQUENCIAIS ASSÍNCRONOS

A MEF em modo fundamental, também chamada de circuito sequencial assíncrono ou MEF Assíncrona (MEFA), utiliza-se da ideia que os *flip-flops* apenas proporciona um atraso entre as variações nos níveis lógicos por um período de relógio. Portanto, na MEFA os *flip-flops* foram substituídos por elementos que introduzem um atraso. O tipo de atraso proporcionado é o atraso decorrente da propagação do sinal elétrico transmitido através de um fio, ou uma cascata de portas lógicas [5].

Existe diversas formas de representar uma MEFA. O modelo da Figura 1 é o mais simples para entender como o circuito funciona. Também chamado de máquina de Huffman, o circuito é caracterizado pelo fato que as entradas podem mudar a qualquer momento e pelo uso de elementos de atraso como dispositivos de memória [3].

A combinação dos sinais de entrada $x_1, x_2, ..., x_l$ são chamados de estado de entrada. Na saída dos elementos de atraso D (*delay*) existem as variáveis $y_1, y_2, ..., y_k$, que são chamadas de variáveis internas ou secundárias. Da mesma forma o conjunto dessas variáveis é chamado de estado secundário ou interno, e as variáveis $Y_1, Y_2, ..., Y_k$ são chamadas de variáveis de excitação. As saídas geradas pela lógica combinacional determina as variáveis de saída como também o estado secundário, que o sistema irá assumir em seguida. Um estado de entrada é dito estar em um estado estável, se e somente se $y_i = Y_i$ para todo $i = 1, 2, ..., k$ [3].

Quando o estado de entrada muda, ou seja, alguma de

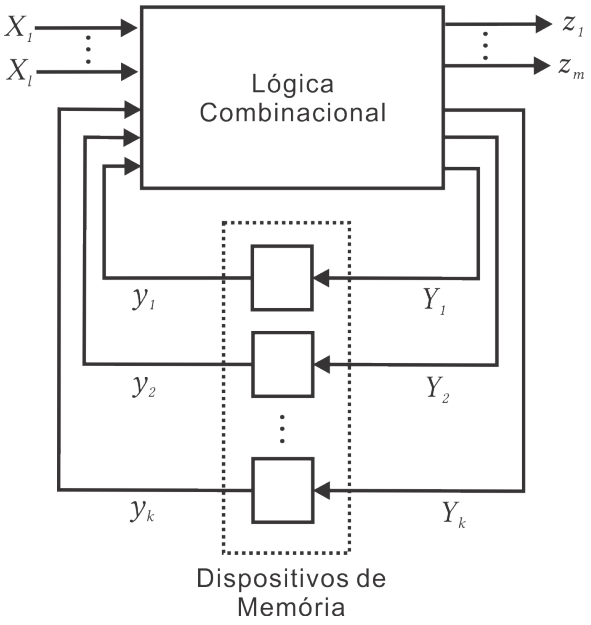



**Figura 1:** Modelo básico para uma MEF em modo Fundamental [3].

suas variáveis são alteradas, a lógica combinacional produz um novo conjunto de valores para as variáveis de excitação, como consequência o circuito entra no que é chamado de estado instável. Após um instante de tempo de atraso as variáveis secundárias assumirão o novo valor das variáveis de excitação, como resultado o circuito vai para o próximo estado estável [5]. Portanto, uma transição de um estado estável para outro só irá ocorrer quando o estado de entrada sofre alguma alteração.

A ideia é que, depois que um valor de entrada muda, não ocorra nenhuma outra mudança nas entradas enquanto o circuito não entra no estado estável. Tal forma de operação é chamada de modo fundamental. Se somente uma variável de entrada puder mudar, é chamado de modo fundamental SIC (*Single-Input-Change*), no entanto se mais de uma entrada puder mudar, é chamado de circuito em modo fundamental MIC (*Mutiple-Input-Change*). Uma generalização da MIC é chamada de circuito *Burst-Mode* [3].

Assim como no caso da MEF, existe algumas maneiras convenientes de demonstrar o comportamento desejado de um circuito. Para as MEFA, o método para descrever o comportamento é chamado de tabela de fluxo. A Tabela 1 é um exemplo de tabela de fluxo de um circuito que conta em módulo 4 (de 0 a 3). A Tabela 1 apresenta o número de inversões de nível lógico sobre a entrada $X$. Logo, quando $X$ passar por $X = 0 \rightarrow 1 \rightarrow 0 \rightarrow 1 \rightarrow 0$ etc.. a saída esperada será $z_1 z_0 = 00 \rightarrow 01 \rightarrow 10 \rightarrow 11 \rightarrow 00$ assim por diante [5].

Logo, a partir da Tabela **??** nota-se que possui 4 estados estáveis, que são representados pelos estados em negrito. Os demais estado são instáveis. Portanto, quando $Xmudapara1$ e o circuito está no estado 00, ou seja, $y_1 y_0 = 00$, $Y_1 Y_0$ mudara para 00 , neste momento $y_1 y_0 \neq Y_1 Y_0$. Portanto, a MEFA irá para um estado instável, mas após o atraso $y_1 y_0 = Y_1 Y_0 = 01$, levando o circuito para o próximo estado estável. A saída não está representada, pois nesta atribuição $z_1 z_0 = Y_1 Y_0$. De forma semelhante a MEF, pode se obter as equações de excitação a partir da Tabela 1 e construir o circuito lógico.

**TABELA 1:** TABELA DE FLUXO DE ESTADOS.

| Estado,Saida | | |
|---|---|---|
| | $X$ | |
| $y_1y_0$ | 0 | 1 |
| 00 | **00** | 01 |
| 01 | 10 | **01** |
| 10 | **10** | 11 |
| 11 | 00 | **11** |

## *1. Circuito Bust-Mode*

As máquinas MIC possuem uma maior flexibilidade, pois permite que diversas entradas mudem de uma vez antes que a máquina entre no estado estável. Uma generalização é chamado de circuito de *Bust-Mode* (BM). o circuito BM permite que múltiplas entradas mudem simultaneamente, mas diferente da MIC que todas as entradas devem chegar em um período de tempo limitado, pode chegar em qualquer ordem e a qualquer momento no chamado *input-bust* [6].

Ela é representada por meio de um diagrama de estado chamado de especificação *bust-mode*. O diagrama possui um número finito de estados, cada transição é representada por um arco conectando a um par de estados e possui um estado inicial distinguível. O arco é rotulado com possíveis transições, levando o sistema de um estado para outro. Cada transição consiste de um conjunto não vazio de entradas *input-bust* e um de saída *output-bust*, se nenhuma entrada muda então o sistema está estável. Em um dado estado quando todas as entradas mudaram de valor, a máquina gera o conjunto de saída *output-bust* correspondente e então muda de estado. O conjunto de entrada *input bust* muda somente uma vez por transição de estado [7, 4].

O circuito em BM possui duas restrições na sua especificação. Primeiro, nenhum conjunto de entrada de um determinado estado pode ser subconjunto de outro ou então o comportamento será ambíguo. Esta restrição é chamada de propriedade do conjunto máximo. Segundo, em um dado estado sempre entrará com o mesmo conjunto de entrada, ou seja, cada estado possui apenas um ponto de entrada. Chamada de propriedade do ponto de entrada único. No caso da propriedade do ponto de entrada único, se um diagrama não a satisfaz, pode ser transformado em um diagrama equivalente que o satisfaz dividindo os estados [7]. Um exemplo da MEFA em BM pode ser visto na Figura 2, onde cada transição é rotulada com um conjunto de entrada e um de saída divido por um $/$. Uma transição de ascendente $0->1$ é representada por "$+$" e uma transição de descendente $1->0$ por "$-$".

## *2. Corridas*

Qualquer transição alternativa realizada do estado de origem ao estado de destino, que envolva mudança de duas ou mais variáveis é chamada de corrida. Por exemplo se quiser ir do estado 00 para o 11, terá que passar pelo 01 ou 10. Esse evento acontece porque as variações nos $Y's$ ocorrem através de tempos de atraso aleatórios (devido a natureza física dos componentes) e acabam chegando nos $y's$ em tempos diferentes. Pois, os circuitos para cada um possuem tamanhos diferentes e quantidade de portas lógicas distintas. Logo, o

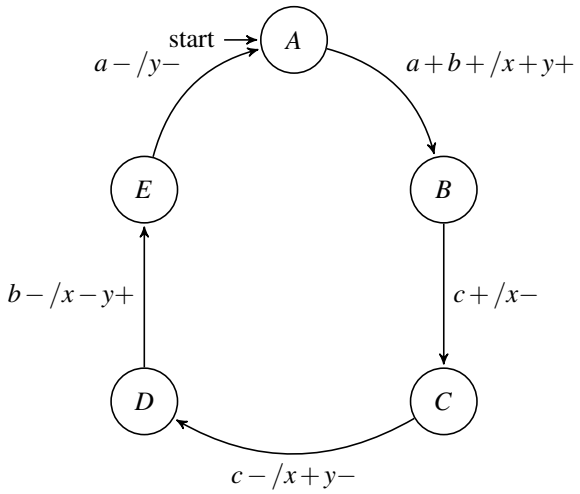



**Figura 2:** Um exemplo de circuito *Bust-Mode* [7].

tempo necessário para cada uma das variáveis percorrer o seu circuito é diferente.

Em alguns casos essas corridas são permitidas, pois pode fazer parte ou ser essencial para a lógica da MEFA projetada. Uma corrida que possa resultar em uma transição que se estabilize em um estado errado é chamado de corrida crítica. Esse defeito deve ser corrigido ou eliminado, visto que não pode ser permitido tal defeito no circuito [5, 8].

## *3. Hazard de Funções*

A presença de *hazard* coloca um desafio a mais na hora de desenvolver circuitos sequencias assíncronos. Os *hazard* são as falhas (*glitches* ou *spikes*) que podem se manifestar na saída por algum instante de tempo. Uma função $f$ que não muda monotonicamente durante uma transição de entradas é dito ter um *hazard* de função na transição. As seguinte definições são de [9, 7].

**Definição II.1.** Uma função booleana $f$ contém um *hazard* de função estático para a transição de entrada de $A$ para $C$, se e somente se:

1. $f(A) = f(C)$, e;
2. Existe algum estado de entrada $B \in [A,C]$ tal que $f(A) \neq f(B)$.

**Definição II.2.** Uma função booleana $f$ contém um *hazard* de função dinâmico para a transição de entrada $A$ para $D$, se e somente se:

1. $f(A) \neq f(D)$.
2. Existe um par de estado de entrada $B$ e $C$, onde $A \neq B$ e $C \neq D$ tal que
   (a) $B \in [A,D]$ e $C \in [B,D]$ e;
   (b) $f(B) = f(D)$ e $f(A) = f(C)$.

Se uma transição possui um *hazard* de função, nenhuma implementação da função é garantida ser livre de *glitches* durante uma transição, assumindo um atraso arbitrário nos fios e portas logicas.

### *4. Hazards Lógicos*

Se $f$ é uma função é livre de *hazard* para uma transição do estado de entrada $A$ para $B$, ele ainda pode conter *hazards* por causa do atraso na lógica realizada. As seguintes definições foram tiradas do [7].

**Definição II.3.** Um circuito combinacional para uma função $f$ contém um *hazard* lógico estático para a transição de entradas de um mintermo $A$ para o mintermo $B$ se e somente se:

1. $f(A) = f(B)$;
2. Não existe nenhum *hazard* de função estático na transição de $A$ para $B$;
3. Para alguma alocação de atraso, a saída do circuito não é monotônica durante o intervalo de transição.

**Definição II.4.** Um circuito combinacional para uma função $f$ contém *hazard* lógico dinâmico para a transição de estado de entrada $A$ para o $B$, somente se:

1. $f(A) \neq f(B)$;
2. Não existe *hazard* dinâmico de função na transição de $A$ para $B$.
3. Para uma certa alocação de atraso, a saída do circuito não é monotônica durante o intervalo de transição.

As definições formalizam que o *hazard* lógico ocorre para alguma combinação particular de portas e fios de atraso no circuito, devido aos fatos das alterações dos estados de entradas não serem instantâneo. Portanto, ocorrerá um *glich* ou *spike* na saída durante a transição.

## III. TRABALHOS RELACIONADOS

Um dos principais problemas da minimização é achar um algoritmo adequado e eficiente. Para resolver esse problema o [2] utiliza de uma abordagem em duas fases distintas. Na primeira fase todos os implicantes primos são gerados, enquanto na segunda gera-se um subconjunto de implicantes primos do conjunto original onde a sua união forma a função booleana minimizada. Sendo o algoritmo proposto o *Clause-Column Table* que é computacionalmente muito eficiente em funções dadas na forma soma de produto, produto da soma, na sua forma canônica ou não.

Outra abordagem ao mesmo algoritmo relevante para esse projeto foi a de [1] onde foi proposto uma melhoria no *Clause-Column Table*, adicionando o teorema da adjacência e um novo critério de parada. Além disso, também realizou a segunda fase por meio de programação linear inteira. Os resultados obtidos mostraram que o método *Clause-Column Table* aprimorado pode ser mais eficiente que o método original de [2].

Outro método proposto por [10] é o de minimização de expressões booleanas na forma de soma de produtos de ou exclusivo (XOR). É uma área bastante explorada na literatura por causa de suas propriedades que facilitam o processo de testes e o circuito final nessa forma é consideravelmente menor que na soma de produtos. Porém, ainda não foi encontrado uma solução eficiente para minimizar funções como mais de 6 variáveis nos casos gerais.

Seguindo essa mesma abordagem, [11] propuseram uma heurística para minimização de expressões booleanas na forma soma de produtos de ou exclusivo, para funções incompletamente especificadas de múltiplas saídas. Pois, utilizando essa abordagem reduz consideravelmente a complexidade do circuito.

[10] desenvolveram um algoritmo baseado no DCMIN (*Don't Care MINimization*) que usa decomposição funcional e lógica de múltiplos valores para produzir expressões quase mínimas para essas funções chamadas QuickDCMIN, que demonstrou ser mais eficiente.

Já no trabalho de [12] foi proposto um método de aprendizagem de máquina para minimização de funções booleanas chamada de MLBM (*Machine-Learning-Based Minimization*). Porém, o MLBM é diferente dos métodos de minimização comuns na literatura, como já apresentado. A abordagem possui uma grande habilidade de processamento, já que não possui limite de variável, e sempre produz um resultado excelente.

Diferentemente dos outros trabalhos o [4] propôs dois novos algoritmos baseados no algoritmo genético, para minimização e alocação de estado de uma MEFA. A arquitetura da MEFA utilizada foi da máquina no modo *Bust-mode*. Tal arquitetura não possui realimentação por parte do circuito de saída. O que possibilita um baixo tempo de resposta, além da capacidade de alteração de mais de uma variável de entrada no mesmo instante. O algoritmo apresentou uma alta eficiência em seus testes quando comparado com a ferramenta *Minimalist* que é o estado da arte para tal arquitetura.

[13] desenvolveram um algoritmo atrelado a um sistema CAD (*Computer-Aided Design*) para lidar com máquina de estados em nível lógico. O sistema foi desenvolvido para analisar a MEFA na linguagem de descrição de hardware VHDL. Assim, como o algoritmo realiza a minimização e retorna um modelo minimizado em VHDL. O sistema foi modelado tanto para computação com um único processador, quanto para computação de alto desempenho.

A eficiência da ferramenta CAD foi avaliada utilizando diferentes números de nós computacionais e MEFAs. Com base nisso, o aumento de eficiência com base no aumento de nós é diretamente ligado a complexidade da MEFA. Quanto mais complexa, mais significativa era o ganho de eficiência com o aumento de nós. Da mesma forma, quanto menos complexa menos significativa era o aumento.

## IV. METODOLOGIA

Como as MEFAs são compostas basicamente de lógica combinacional com realimentação, pode-se utilizar otimização lógica para minimizar tais máquinas. O processo de minimização de uma MEFA é uma síntese que possui três etapas, sendo a minimização lógica a última delas. Como as duas primeiras etapas, minimização de estados e alocação de estados estão fora do escopo desse trabalho, foram realizadas utilizando a ferramenta *minimilist*, uma ferramenta que possui uma coletânea de algoritmos, de todas as três etapas, ela possui o objetivo de auxiliar no desenvolvimento de novos algoritmos e síntese de MEFAs [6].

Nesse projeto é proposto um algoritmo para minimização lógica de MEFA baseados nos métodos *Clause-Column Table* aprimorado desenvolvido por [1] e no algoritmo Quine-

McCluskey [14].

A escolha do *Clause-Column Table* foi feita, pois ele é eficiente mesmo com um número grande de variáveis, além de aceitar expressões booleanas tanto na forma soma de produtos e produto de somas na forma canônica e não canônica.

### *a. Clause-Column Table*

O método *Clause-Column Table* é uma técnica tabular e iterativa que permite a geração de implicantes primos. O resultado converge rapidamente mesmo no caso de funções com muitos termos e variáveis [2]. Apesar disso o método possui algumas deficiências.

[1] desenvolveram um nova versão nomeada de *Clause-Column Table* aprimorado, em que foram feitos algumas alterações para evitar a geração de termos nulos e diminuir a quantidade de iterações utilizada para geração dos implicantes primos.

Foi adicionado, a propriedade algébrica $AB + A\overline{B} = A$. Quando aplicado a tabela inicial formada pelos maxtermos da função booleana muitos termos são eliminados. Outra alteração foi a adição de um critério de parada com o objetivo de evitar a geração de termos nulos. Se houver apenas uma coluna a partir da segunda iteração o algoritmo deve ser interrompido. O algoritmo *Clause-Column Table* aprimorado desenvolvido pela [1] é composto pelos seguintes passos:

1. Passo: Forme uma tabela em que cada coluna é composta por um maxtermo da função. Se a função booleana inicial estiver na forma de produto de soma, execute o próximo passo, se não obtenha o seu dual.
2. Passo: Execute as seguintes operações até que todas as colunas sejam excluídas, caso não for possível excluir todas vá para o passo 3:
   (a) Se algum critério do passo 6 for atendido pare a execução, senão continue;
   (b) Se houver colunas dominadas, elimine-as;
   (c) Se existir colunas redundantes elimine aleatoriamente uma delas;
   (d) Simplifique as colunas restantes com o teorema da adjacência (apenas uma vez por iteração);
   (e) Se houver literal essencial selecione-os. Se houver mais de um literal essencial selecione um deles aleatoriamente;
   (f) Verifica novamente o critério de para 6(c);
   (g) Elimine todas as colunas que contém literais essenciais;
   (h) Elimine o complemento do literal selecionado das colunas restantes;
3. Passo: Se algum literal foi selecionado no passo anterior execute o passo 5, senão execute o passo 4.
4. Passo: Verifique se na tabela reduzida a quantidade de incidência de cada um dos literais na forma complementada ou não:
   (a) Se houver literais com a mesma incidência selecione um deles aleatoriamente;
   (b) Se o literal com a maior incidência estiver contido em todas as colunas ele deve ser considerado implicante primo da iteração atual;
   (c) Execute o passo 6.
5. Passo:
   (a) Todas as colunas que contêm o literal selecionado devem ser eliminadas;
   (b) O complemento do literal selecionado também deve ser eliminado das demais colunas;
   (c) Forme a tabela reduzida com o restante das colunas;
   (d) Se na tabela reduzida tiver mais de uma coluna selecione o literal essencial, senão repita o passo 5(a) e 5(b).
   (e) Realize a operação lógica AND entre o literal selecionado e a tabela reduzida cujo o termo resultantes são implicantes primos.
6. Passo: Verifique os critérios de parada.
   (a) Todas as colunas sejam excluídas;
   (b) Dois literais, complementados ou não se tornem essenciais ao mesmo tempo;
   (c) Um literal torna-se essencial e na iteração seguinte o seu complemento também;
   (d) Reste apenas uma coluna a partir da segunda iteração.
7. Passo: Quando atingido algum critério de parada reúna os implicantes primos resultantes e forme o conjunto de solução.

O Algoritmo deve então ser executado iterativamente até que um dos critérios de parada seja atingido. Com os implicantes primos obtidos pode-se então usar de outros algoritmos para gerar a expressão mínima da função booleana.

### *b. Quine-McCluskey*

Usualmente a abordagem para o problema de minimização de expressões booleanas envolve duas fases distintas. Na primeira fase, o algoritmo encontra todas os implicantes primos. Já na segunda fase, a partir do conjunto de implicantes primos obtidos na primeira fase, um subconjunto mínimo é selecionado de uma forma que a sua união é equivalente a função original [2]. O Quine-McCluskey segue essa mesma lógica, possuindo duas fases: na primeira ele encontra os implicantes primos, na segunda ,ele gera o conjunto mínimo para minimizar a função.

O método começa com uma lista de todos os $n$ mintermos da função e continuamente deriva todos os implicantes primos com $n-1$, $n-2$ variáveis até que todos os implicantes tenham sido identificados. O algoritmo possui 4 passos, sendo o $1^\circ$ e $2^\circ$ para encontrar os implicantes primos e o $3^\circ$ e $4^\circ$ para realizar a simplificação.

Na primeira etapa todos os mintermos da função são listados em uma coluna com sua representação binária. Em seguida, são separados em grupos de acordo com o número de bits 1 em sua forma binária, pois isso simplifica a procura

de mintermos logicamente adjacentes. Já que para ser logicamente adjacentes os minitermos se diferenciam em apenas um literal, logo na forma binária ele tem que ter um bit 1 a mais ou um a menos.

No segundo passo é realizado uma busca por força bruta entre os grupos vizinhos procurando por mintermos adjacentes e combinando todos eles em uma coluna de $n-1$ implicantes primos, marcando todos os que foram possíveis de combinar. Após isso repita para todas as colunas, combinando as colunas de $n-1$, $n-2$ até quando não for possível combinar mais nenhum implicante, marcando todos eles. Após isso todos os implicantes não marcados são primos, tornando uma lista de implicantes primos.

Obtido a lista de implicantes primos, o passo 3 constrói se uma tabela que lista todos os mintermos na horizontal e implicantes primos na vertical, marque onde um certo implicante primo (linha) for capaz de cobrir um mintermos (coluna). No último passo selecione o menor número de implicantes que é capaz de representar todos os mintermos da função [9].

### *c. Síntese de MEFA*

Assim como as abordagens usuais, que divide a síntese de circuitos lógicos combinacionais em algumas etapas [2]. O método proposto tem como objetivo tentar melhorar o desempenho do processo de síntese utilizando de dois algoritmos conhecidos na literatura (CCT/QM –*Clause-Column Table* e Quine-McCluskey). Tal método possui três etapas como pode ser visto na Figura 3, uma etapa para realizar a minimização de estados e alocação dos mesmos. A segunda etapa utilizada para obter os implicantes primos de cada equação de excitação, pôr fim a última etapa selecionar o menor subconjunto que consiga representar a função lógica de cada equação, desta maneira fornecendo o circuito lógico da MEFA.

O circuito foi operado em modo fundamental MIC, mais especificamente na sua generalização BM (*Bust-mode*), por ter uma grande quantidade de estudos e ferramentas, além de ser comercialmente mais utilizado. O algoritmo recebe como entrada uma tabela de estados, o número de variáveis de entrada, número de variáveis de saída, número de estados usados e número de transições entre os estados. As etapas demostradas na Figura 3 funcionam da seguinte maneira:

- Etapa 1: Na primeira etapa recebe um arquivo com todas as especificações da MEFA, foi feito então a minimização de estados e alocação dos mesmos utilizando a ferramenta chamada *chasm* presente na *minimilist* [6]. O *chasm* realiza alocação de estado livre de corrida crítica próximo do ótimo. Após a alocação é construída a tabela de transição de estados final e dela gerado o conjunto de equações de excitação que representam o circuito da MEFA;
- Etapa 2: A partir do conjunto de equações de excitação obtido é encontrado os implicantes primos de cada equações, utilizando o método do *Clause-Column Table*, obtendo então o conjunto de implicantes primos;
- Etapa 3: Com o conjunto de implicantes primos, utiliza-se como base os passos 3 e 4 do algoritmo Quine-McCluskey para gerar expressão booleana mínima de cada equação de excitação da MEFA, assim retornando o circuito combinacional na forma de soma de produtos.

### *d. Benchmark*

O algoritmo foi avaliado utilizando o *benchmark* da Tabela 2. O *benchmark* é um conjunto com 10 MEFAs amplamente utilizados em estudos na utilizado na literatura [4, 15, 6], sendo algumas das MEFAS especificadas são modelos reais utilizados na indústria.

**TABELA 2:** DESCRIÇÃO DOS CASOS CONTIDOS NO *benchmark*.

| Benchmark | Entradas | Saidas | Estados |
|---|---|---|---|
| dme-e | 3 | 3 | 8 |
| it-control | 5 | 7 | 10 |
| hp-if-rf-control | 6 | 5 | 12 |
| stetson-p2 | 8 | 12 | 25 |
| stetson-p3 | 4 | 2 | 8 |
| isend | 4 | 3 | 10 |
| scsi-tsend-bm | 5 | 4 | 11 |
| opt-token | 4 | 4 | 12 |
| hp-ir | 3 | 2 | 6 |
| isend-bm | 5 | 4 | 10 |

O algoritmo utilizado nesse projeto foi incorporado no processo de síntese de MEFA na sua última etapa, para geração do circuito combinacional reduzido. Para as partes de minimização de estado e alocação dos mesmos foi utilizado a ferramenta *minimalist*, já que essas etapas fogem do escopo do trabalho.

## V. RESULTADOS

As configurações utilizadas para minimização de estado e alocação dos mesmo realizado com o *minimalista* de forma heurística para as MEFAs não garantem possuir uma alocação ótima, apenas ser livre de corrida crítica. Para diminuir as variáveis de estado e conseguintemente gerar um circuito equivalente reduzido, foi utilizada uma variação da máquina de Huffman, onde só as variáveis de estados são realimentadas (modelo de Moore), como algumas das variáveis de saída realimenta o circuito como entrada de estado. Após a alocação é gerado um arquivo no formato de PLA (*Programmable Logic Array*), onde informa uma tabela verdade com os mintermos de onde pode ser obtida as funções de excitação do circuito, e assim realizar a minimização lógica.

De forma a avaliar o desempenho do método foi utilizado o *benchmark* com 10 máquinas com o circuito em modo BM. Sendo alguns projetos reais de circuitos assíncronos e por ser bem utilizado na literatura. Para mostrar a eficiência do algoritmo foi feito uma comparação com os resultados de [4] que propõe a ferramenta SAGAAs. A minimização e alocação de estados usando Algoritmos Genéticos, a minimização lógica foi realizada com o HFMIN que é um algoritmo exato para quantidades de literais [6].

Todos os *benchmark* foram executados em um processador AMD A12-9720p Quad-Core com até 3,60 GHz e 8GB de SDRAM DDR4. Quanto aos resultados obtidos do [4] não foi informado as especificações onde os *benchmark* foram executados. A comparação dos dois resultados pode ser vista na Tabela 3.

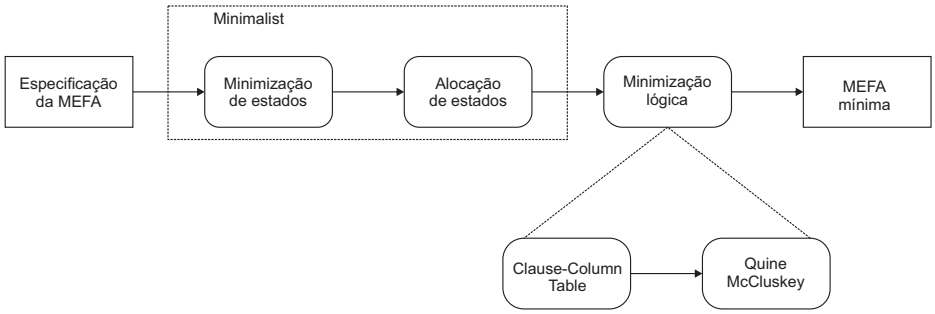



**Figura 3:** Fluxo de minimização de uma MEFA proposto neste trabalho.

As especificações E, S e EST da MEFA contida na Tabela 3 são respectivamente a quantidade de entrada, saída e estados, os resultados Pr, Li e T também são respectivamente, o número de produtos, de literais das funções finais do circuito é o tempo de execução do algoritmo em relação ao tempo de CPU. O tempo de execução do SAAGA's é o tempo de toda a síntese (minimização de estado, alocação de estado e minimização lógica) diferente do CCL/QM a onde o tempo do *minimalist* não foi considerado, apenas o do método proposto por esse projeto.

O CCT/QM em relação ao SAAGAS/HFMIN não obteve um desempenho tão bom, houve um aumento de $14,6\%$ no número de produtos gerado na solução final como pode se ver na Tabela 3 em que o total de produtos foram de 178 para o CCT/QM contra 152 do SAAGAs//HFMIN e um aumento de $12,6\%$ no número de literais também, com 650 contra 568 literais gerado no *benchmark*. Sendo as maiores diferenças no número de produtos e literais são respectivamente na máquina *steson-p2* com uma diferença de 10 produtos e na *scsi-tsend-bm*.

Em relação ao tempo de execução, incluindo o *benchmark steson-p2* houve uma diminuição no tempo de execução em $91,7\%$ e se não consideramos o mesmo será de 64% de redução. Porém, o tempo demostrado pelo SAAGAs é o tempo execução das três etapas do processo de síntese, em quanto o do CCT/QM é tempo somente da minimização lógica. No geral os tempos foram semelhantes, em relação as MEFs menores, já com as MEFs com mais variáveis de entradas e saída ambas houve um salto no tempo de execução, sendo CCT/QM com um tempo inferior ao do SAAGAS/HFMIN, no entanto este gerou um circuito final menor.

O desempenho do método em relação ao trabalho do [4], o circuito final gerado foi inferior, apesar de ter um tempo parcial melhor. Além disso o *Clause-Column table* não garante que o circuito seja livre de *hazards* de função de lógica, ou seja, pode ocorrer *glitches* na MEFA final, enquanto o HFMIN utilizado no SAGAAS garante ser livre de *hazards* e é exato no número de literais.

## VI. Considerações finais

O desenvolvimento deste projeto possibilitou a análise dos tipos de MEFs e suas aplicações. Entre elas a MEFA apresentou algumas vantagens econômicas e energéticas e em relação à MEF. Além disso, como a MEFA é composta de apenas de circuitos combinacionais possibilita utilização de algoritmos de otimização lógicas para gerar um circuito equivalente mínimo.

Durante o desenvolvimento do projeto as MEFAs foram modeladas em modo fundamental MIC, mais especificamente o circuito BM, pois está mais próximo dos circuitos reais, utilizados na indústria. Além de que possui diversos estudos dessa modelagem, assim como ferramentas que ajudam no processo de síntese. Logo, trouxe o projeto para problemas mais próximos dos problemas do mundo real.

Neste projeto foi mostrado um método de minimização lógica de MEFA utilizando os algoritmos *Clause-Column Table* e o Quine-McCluskey, na tentativa de obter um minimizador lógico eficiente. No entanto, os resultados demostraram que o método não é tão eficiente já que houve um aumento no número de literais e produtos do circuito final. Contudo, a diferença de soluções não é tão distante se comparados empiricamente. Embora, seja necessária uma análise estatística para comprovar essa hipótese.

Apesar do método garantir que o circuito seja livre de corrida crítica, uma deficiência é o fato que o *Clause-Column table* não garante que o circuito final seja livre de *hazards*. Diferentes de outros métodos já existentes na literatura, podendo a MEFA final possuir alguns *glitches* na saída, até que a MEFA entre em estado estável. Essa limitação é devido ao fato de ele só suportar funções completamente especificadas, ou seja, em que todas as entradas possuem um comportamento esperado e previsível.

Em relação ao tempo de execução, o desempenho foi satisfatório nos *benchmarks*. Porém, necessita de mais estudos e experimentos em MEFAs mais robustas, onde possuem um número maior de entradas e saídas para que tenha um resultado mais preciso. Pois, um aumento de 2 variáveis de entrada fez o tempo de execução que era de menos de um segundo saltar para mais de 6 segundos. Devido ao fato de o problema de gerar um conjunto de implicantes primos mínimo que represente todos os mintermos da função ser um problema NP-difícil e o Quine-Mccluskey ter complexidade exponencial, quanto mais variáveis a função tiver mais rapidamente o tempo execução aumentará.

Para futuros trabalhos, visando melhorar o método, é conveniente avaliar as meta-heurísticas para gerar o conjunto mínimo de implicantes primos da função para MEFAs maiores. Outro é adaptar o *Clause-Colum Table* para que possa garantir que o circuito final seja livre de *hazard* de função e lógico, utilizando expressões booleanas na forma de notação cubica.

## Referências

[1] C. D. P. do Nascimento Barbieri, J. V. De Paulo, and A. C. Rodrigues, "Aperfeiçoamento do método clause-column table para a geração eficiente de implicantes primos," *A*, vol. 1, p. M2, 2015.

**TABELA 3:** RESULTADOS DA METODOLOGIA PROPOSTA.

| Benchmark | Especificações | | | CCT/QM | | | SAAGAs/HFMIN | | |
|---|---|---|---|---|---|---|---|---|---|
| | E | S | EST | Pr | Li | T(s) | Pr | Li | T(s) |
| dme-e | 3 | 3 | 8 | 8 | 30 | 0,11 | 8 | 24 | 0,19 |
| it-control | 5 | 7 | 10 | 19 | 68 | 0,22 | 19 | 60 | 0,66 |
| hp-if-rf-control | 6 | 5 | 12 | 13 | 46 | 0,26 | 7 | 63 | 0,82 |
| stetson-p2 | 8 | 12 | 25 | 42 | 160 | 6,3 | 32 | 154 | 88,60 |
| stetson-p3 | 4 | 2 | 8 | 5 | 18 | 0,13 | 5 | 13 | 0,21 |
| isend | 4 | 3 | 10 | 21 | 81 | 0,22 | 19 | 59 | 0,27 |
| scsi-tsend-bm | 5 | 4 | 11 | 28 | 100 | 0,20 | 24 | 72 | 0,55 |
| opt-token | 4 | 4 | 12 | 15 | 48 | 0,25 | 14 | 50 | 0,68 |
| hp-ir | 3 | 2 | 6 | 3 | 11 | 0,07 | 4 | 12 | 0,05 |
| isend-bm | 5 | 4 | 10 | 24 | 88 | 0,2 | 20 | 61 | 0,23 |
| Total | | | | 178 | 650 | 7,5 | 152 | 568 | 92,26 |


[2] S. R. Das, A.-A. Amin, S. N. Biswas, M. H. Assaf, E. M. Petriu, and V. Groza, "An algorithm for generating prime implicants," in *Instrumentation and Measurement Technology Conference Proceedings (I2MTC), 2016 IEEE International*. IEEE, 2016, pp. 1–6.

[3] Z. Kohavi and N. K. Jha, *Switching and finite automata theory*. Cambridge University Press, 2009.

[4] T. Curtinhas, T. C. Cavalcante, D. L. Oliveira, L. A. Faria, and O. Saotome, "Minimization and encoding of high performance asynchronous state machines based on genetic algorithm," in *2015 28th Symposium on Integrated Circuits and Systems Design (SBCCI)*. IEEE, 2015, pp. 1–6.

[5] H. Taub, *Circuitos Digitais e Microprocessadores*. Editora McGraw-Hill, 1984.

[6] R. M. Fuhrer and S. M. Nowick, *Sequential optimization of asynchronous and synchronous finite-state machines: Algorithms and tools*. Springer Science & Business Media, 2012.

[7] S. M. Nowick, "Automatic synthesis of burst-mode asynchronous controllers," Ph.D. dissertation, Stanford University PhD thesis, 1993.

[8] R. F. Tinder, "Asynchronous sequential machine design and analysis: A comprehensive development of the design and analysis of clock-independent state machines and systems," *Synthesis Lectures on Digital Circuits and Systems*, vol. 4, no. 1, pp. 1–236, 2009.

[9] V. P. Nelson, H. T. Nagle, J. D. Irwin, and B. D. Carroll, *Digital logic circuit analysis and design*. Prentice Hall, 1995.

[10] S. Stergiou, K. Daskalakis, and G. Papakonstantinou, "A fast and efficient heuristic esop minimization algorithm," in *Proceedings of the 14th ACM Great Lakes symposium on VLSI*. ACM, 2004, pp. 78–81.

[11] M. Kalathas, D. Voudouris, and G. Papakonstantinou, "A heuristic algorithm to minimize esops for multiple-output incompletely specified functions," in *Proceedings of the 16th ACM Great Lakes symposium on VLSI*. ACM, 2006, pp. 357–361.

[12] Y. Lin and R. Geng, "Mlbm: Machine-learning-based minimization algorithm for boolean functions," in *Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on*. IEEE, 2009, pp. 1160–1165.

[13] F. Kudlačák and E. Gramatová, "Asynchronous sequential circuits synthesis by high-performance computing," in *2014 14th Biennial Baltic Electronic Conference (BEC)*. IEEE, 2014, pp. 65–68.

[14] E. J. McCluskey Jr, "Minimization of boolean functions," *Bell system technical Journal*, vol. 35, no. 6, pp. 1417–1444, 1956.

[15] H. M. Jacobson and C. J. Myers, "Efficient algorithms for exact two-level hazard-free logic minimization," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 21, no. 11, pp. 1269–1283, 2002.

# Computational experiment of error diffusion dithering for depth reduction in images

Leonardo Rezende Costa[1]

[1] *Universidade Estadual de Campinas, Institute of Computing, Campinas/SP, Brasil*

Reception date of the manuscript: 02/06/2020
Acceptance date of the manuscript: 08/06/2020
Publication date: 12/06/2020

**Abstract**—The halftone technique is a process that employs patterns formed by black and white dots to reduce the number of gray levels in an image. Due to the tendency of the human visual system to soften the distinction between points with different shades, the patterns of black and white dots produce a visual effect as if the image were composed of shades of gray and dark. This technique is quite old and is widely used in printing images in newspapers and magazines, in which only black (ink) and white (paper) levels are needed. There are several methods for generating halftone images. In this article we explore dithering with error diffusion and an analysis of different halftone techniques is presented using error diffusion to change the depth of the image. The results showed that the depth of the image changes 1/8 per channel, this halftone technique can be used to reduce an image weight, losing information but achieving good results, depending on the context.

**Keywords**—Error Diffusion Dithering, Processing Image, Depth Reduction

## I. Introduction

According to [1], error diffusion is a method for generating high quality halftones that are particularly suitable for low to medium resolution devices. The image resolution for halftones is determined by the screen ruling. The term screen ruling is used to describe the frequency of dots in a given area of image. For conventional screening (amplitude modulated screening or AM), the halftone dots are arranged in a grid structure termed the screen [2].

Also according to [1], to apply the error diffusion algorithm in producing a halftone $b(n_1,n_2)$, we need to scan the input image $f(n_1,n_2)$ in some manner. One of the most popular strategies is raster scanning, where we scan the image row by row from the top to the bottom, and within each row from the left to the right. At each pixel location, we perform the operations

$$u(n_1,n_2) = f(n_1,n_2) - \sum_{m_1,m_2} h(m_1,m_2)e(n_1-m_1,n_2-m_2) \tag{1}$$

$$b(n_1,n_2) = Q(u(n_1,n_2)) = \begin{cases} 1 & \text{if } u(n_1,n_2) \geq 0.5 \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

$$e(n_1,n_2) = b(n_1,n_2) - u(n_1,n_2) = Q(u(n_1,n_2)) - u(n_1,n_2) \tag{3}$$

The filter $h(m_1,m_2)$ generally has a low pass characteristic, and the coefficients often satisfy

$$\sum_{m_1,m_2} h(m_1,m_2) = 1 \tag{4}$$

Many research use error diffusion in wade of applications. Just to cite some of them: [3] employ a high speed ordered dithering and high quality error diffusion to simultaneously solve the problems we described above. As the experimental results demonstrate, this technique is able to enlarge the dynamic range of the histogram and improve the execution efficiency to around 15%-47% with the tested images.

Akarun, Ozdemir & Alpaydin presented in [4] an improvement on error diffusion dithering through a fuzzy error diffusion algorithm. In this method, the amount of error to be diffused is determined by considering the relative location of the pixel not only to the closest codebook vector, but also

Contact data: Leonardo Rezende Costa, leonardo.costa@students.ic.unicamp.br

to the color cluster it belongs to. The goal is to hide the quantization errors by error diffusion, while preventing the excess accumulation of errors. This is achieved through an attraction-repulsion schema according to a fuzzy membership function.

In [5], Lee, Horiuchi & Saito proposed an algorithm named “confined error diffusion (CED)" which has a well organized architecture between random error diffusion and ordered dither method to improve the image quality and gray level expression for the flat panel display.

A halftone watermarking technique of high watermark rate, robustness, and watermark-rate flexibility is presented in [6]. This technique, namely the parity-matched error diffusion (PMEDF) method, is capable of achieving an embedded watermark rate as high as 6.25–25% with good image quality and without the need for the original image as the reference for decoding.

In [7] is reported a new strategy to increase the speed of Fourier single-pixel imaging by two orders of magnitude. In this strategy, we binarize the Fourier basis patterns based on up sampling and error diffusion dithering. The results demonstrate a 20,000 Hz projection rate using a DMD and capture 256-by-256-pixel dynamic scenes at a speed of 10 frames per second.

In [8], Ozaki, Sako, Harada & Takasaki developed an error-diffusion dithering method for a reflective Memory-In-Pixel LCD to provide a high-quality moving image. In [9], error diffusion halftoning has been used to achieve image dithering based on complex wavelets. Similar to the wavelet-based dithering, Floyd-Steinberg error diffusion has been incorporated, but in addition a new set of sub band amplification factors is coined in the proposed method to further enhance the contrast in a dithered image. Experimental results show that the proposed method is superior to state-of-the-art methods in terms of subjective and objective assessments.

Thus, this paper describes a practical experiment of an error diffusion dithering algorithm, based on the following classical dithering methods, applied to reduce image depth. The methods are: Floyd–Steinberg[10], Stevenson-Arce[11], Burkes[12], Stucki[13] and Jarvis-Judice-Ninke [14].

## II. Implementation

This section describes the computational implementation of the descibre algorithm.

### *a. Diffusion error approach*

This approach aims to change the intensity to 0 (minimum) or 1 (maximum) and then propagate the difference from the original value to the neighbors of the pixel. This way, a region with more 0 pixels will be perceived by human eye as a dark region; a region with more 1 pixels will be perceived as a brighter region; a region with the same amount of black and white pixels distributed equally will be perceived as gray, and this logic continues for the proportion between black and white in the image’s regions.

### *b. Algorithm*

The methods can be described as kernels and the operation may seem like a convolution. The difference between

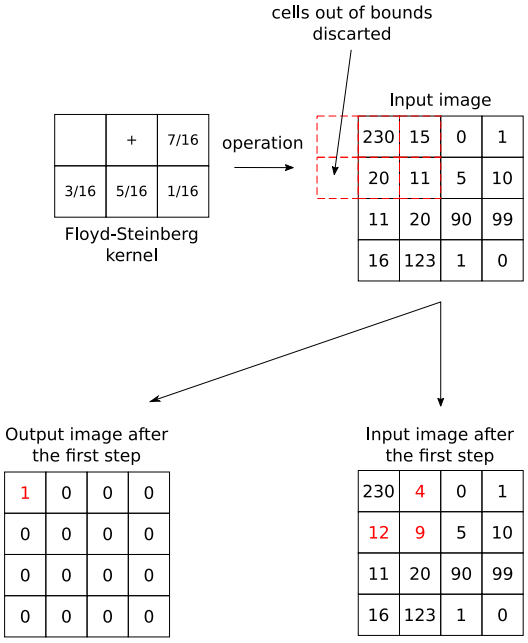



**Fig. 1:** The first step of a dithering operation with a Floyd-Steinberg kernel. Three pixels got updated in the first step. The red numbers indicates the changes made.

a dithering operation and a convolution is that the dithering updates the input image in order to propagate the error to the neighbours of the current pixel. With this principle, a generic function was developed to receive a dithering kernel, calculate the pixel value on the output image as

$$O[i,j] = round(\frac{I[i,j]}{255}) \tag{5}$$

where $O$ is the output image and $I$ is the input image. Then, the error is calculated as

$$error = I[i,j] - O[i,j] * 255 \tag{6}$$

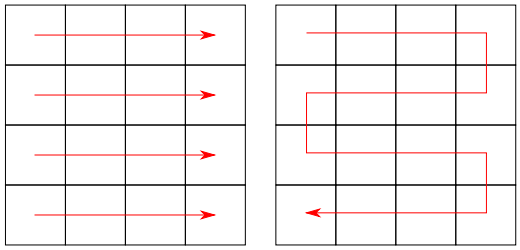
and the pixels around the original pixel are updated as

$$I[i+y, j+x] = I[i+y, j+x] + error * K[y,x] \tag{7}$$

where the $K$ is the dithering kernel, and the $y$ and $x$ are the position of the proportion of error diffused to the neighbour on that position. If a cell on the kernel is positioned out of the image, it’s ignored in the current step. Also, the updated value always respect the pixel range of $[0,255]$. The design of the kernels varies between the techniques listed on Section I. Fig.1 shows an example of the first step of a dithering operation.

### *c. Image scanning approaches*

The two approaches chosen to scan the image was **left to right** and **zigzag**, illustrated in the Fig.2. For this, a parameter called *alternate* is passed to the method, and when it is true, the horizontal range changes to right-left in the odd rows. Additionally, the kernel is flipped horizontally to propagate the errors in the scanning direction, never changing a pixel that already exists in the output image.



**(a)** Left to right **(b)** Zigzag

**Fig. 2:** Two different scan approaches used on the tests.

### *d. Colored images*

To process RGB (Red, Green, Blue) images, each channel was dithered separately, and then mounted as a 3D-array again. Fig.3 illustrates the steps followed by the algorithm for the Stucki's dithering method using the left-to-right approach.

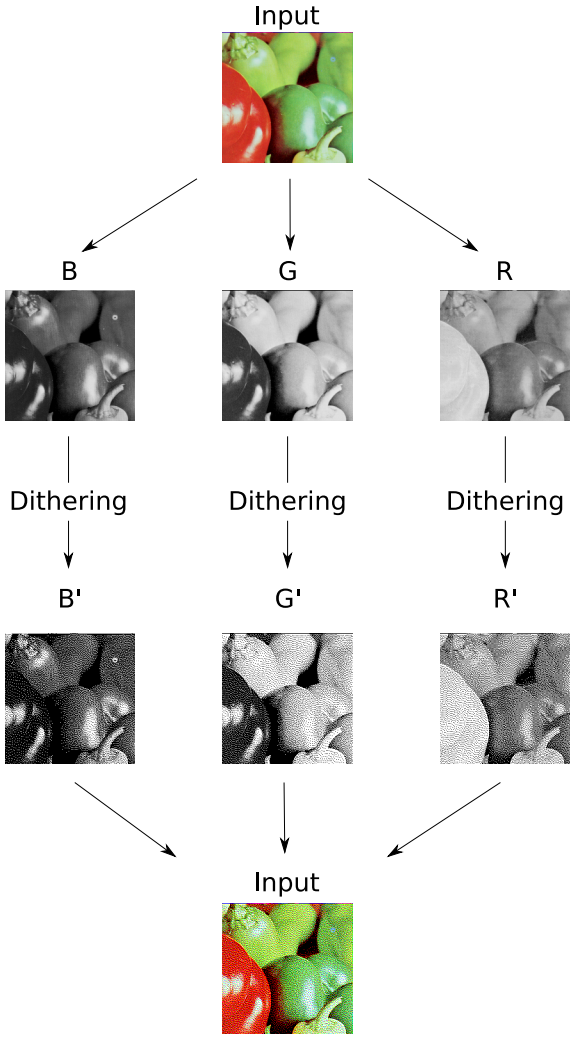



**Fig. 3:** Sequence of steps in the algorithm when performing dithering in a RGB image using Stucki's method and lef-to-right approach.

## III. Experiment and Results

For each of the approaches, the images *baboon*, *monalisa*, *peppers* and *watch*, represented on Fig.4 was used as input. As the approaches have different shapes, the pattern in the result changes along the tests. In Fig.5 it can be seen the difference between **Stevenson-Arce** [11] and **Floyd-Steinberg** results at the same location of the same input.

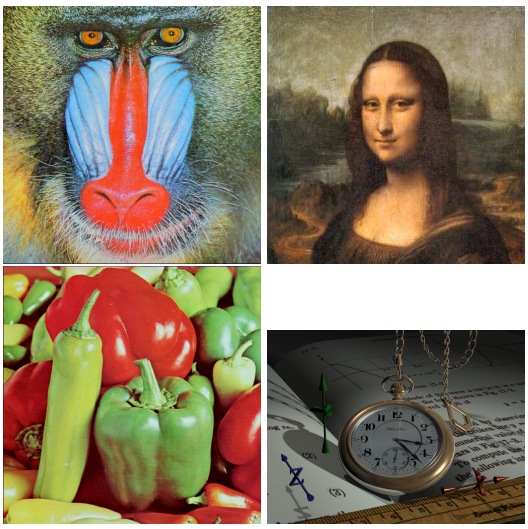


**Fig. 4:** Original images used as input.

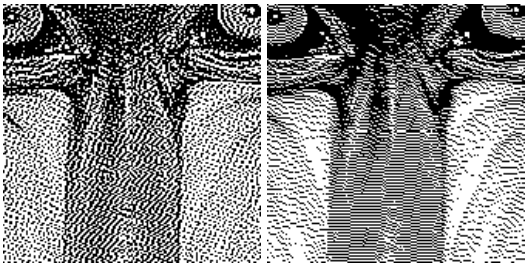


**Fig. 5:** Difference between patterns generated with different approaches. In the left, Stevenson-Arce. In the right, Floyd-Steinberg.

Fig. 6 shows the difference between the two scan approaches using Floyd-Steinberg kernel. As the mask reaches just some of 8-neighbors pixels and the weight is higher at the right side, when a pixel has an intensity near to the threshold, it propagates too much error to the right neighbor, creating a result with more horizontal lines. When the zigzag approach is used, this horizontal line is replaced by a pattern with more dots. This difference is less evident in the other approaches with a bigger mask, as the pixels are more influenced by more distant pixels that have different intensities and that distributes complementary errors.

Fig.7 shows the result of Burkes approach with zigzag scan in the *peppers* input. As each channel is converted to

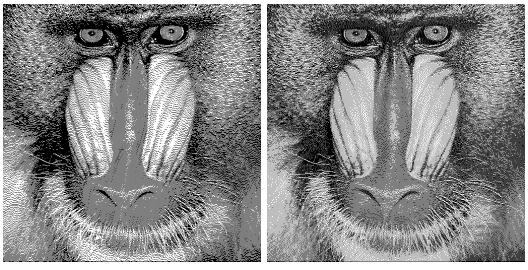


**Fig. 6:** Difference between left-to-right approach (left image) and zigzag approach (right image).

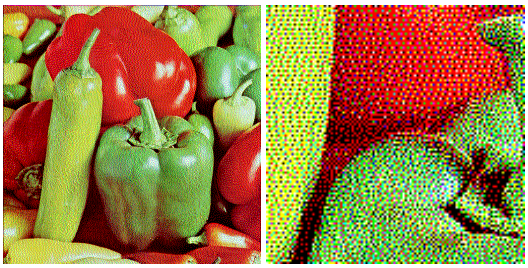
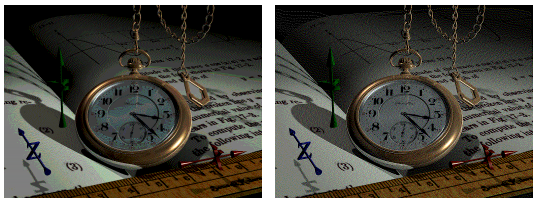


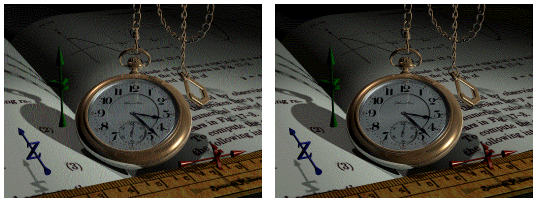
**Fig. 7:** Result from Burkes applied to *peppers* using the zigzag approach.

a binary image, the combination of the three channels result on colors: **red**, **green**, **blue**, **cyan**, **magenta**, **yellow**, **white** and **black**.

On Fig.8 and Fig.9 it can be seen the result of all approaches for *watch* and *monalisa* inputs using zigzag method. All images seams like the input when plotted with a small size. The difference is more evident on *monalisa*, especially because it's resolution is smaller than *watch*. But with a zoom in (as showed in Fig.7) the absolute pixel values and the difference between the approaches can be seen in all tests. Fig.10 highlights the dark region of **Jarvis, Judice and Ninke** and **Floyd-Steinberg** results. On the right one, almost all texture around the arrow was lost and became pure black. The left one shows that more pixels of the paper at the back of the clock appear after dithering, but when it's looked closer, it seams like noise.



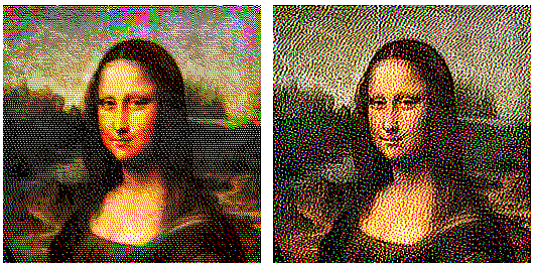
**(a)** Floyd-Steinberg **(b)** Steverson-Arce



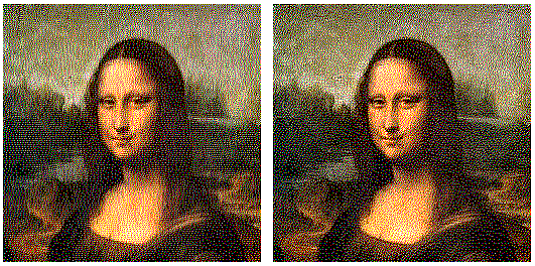
**(c)** Burkes **(d)** Stucki

**(e)** Jarvis, Judice and Ninke

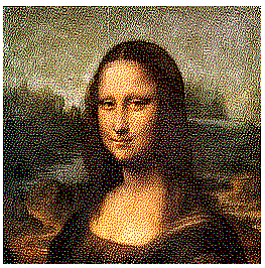
**Fig. 8:** Results for all methods using a zigzag approach on *Watch* input.



**(a)** Floyd-Steinberg **(b)** Steverson-Arce



**(c)** Burkes **(d)** Stucki



**(e)** Jarvis, Judice and Ninke

**Fig. 9:** Results for all methods using a zigzag approach on *Monalisa* input.

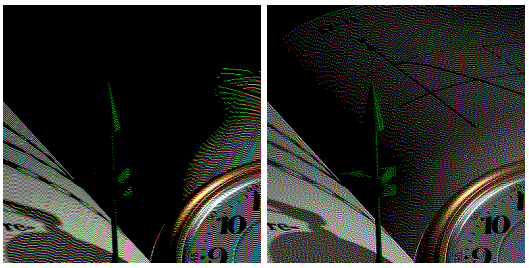


**Fig. 10:** Difference between **Jarvis, Judice and Ninke** and **Floyd-Steinberg** results from *watch* input using left-to-right approach.

## IV. Conclusion

The main purpose of dithering is to plot high resolution (high dimension in a small area) in context with low resource, like old newspaper printing (with monochrome images) or low cost magazine printing (with colorful images), so the result achieve the proposed goals. Another interest detail is that, as the depth of the image changes 1/8 per channel, this halftone technique can be used to reduce an image weight, losing information but achieving good results, depending on the context.

Another use of this technique is in led panels, that mix dithering with a blink technique, so more frames can be pro-

cessed at time and less micro-controller could be used to print something on the screen.

## REFERENCES


[1] P. W. Wong, "8.1 - image quantization, halftoning, and printing," in *Handbook of Image and Video Processing (Second Edition)*, second edition ed., ser. Communications, Networking and Multimedia, A. BOVIK, Ed. Burlington: Academic Press, 2005, pp. 925 – 937. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780121197926501170

[2] R. Mumby, "19 - printing for packaging," in *Packaging Technology*, A. Emblem and H. Emblem, Eds. Woodhead Publishing, 2012, pp. 441 – 489. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9781845696658500196

[3] Jing-Ming Guo and Chih-Yu Lin, "High efficiency and contrast enhanced haiftoning with hybrid ordered dithering and error diffusion model," in *2005 5th International Conference on Information Communications Signal Processing*, 2005, pp. 31–35.

[4] L. Akarun, D. Ozdemir, and E. Alpaydin, "Fuzzy error diffusion of color images," in *Proceedings of International Conference on Image Processing*, vol. 3, 1997, pp. 46–49 vol.3.

[5] J. H. Lee, T. Horiuchi, and R. Saito, "Confined error diffusion algorithms for display device," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 1, 2007, pp. I–885–I–888.

[6] J.-M. Guo, S.-C. Pei, and H. Lee, "Watermarking in halftone images with parity-matched error diffusion," *Signal Processing*, vol. 91, no. 1, pp. 126 – 135, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0165168410002641

[7] Z. Zhang, X. Wang, G. Zheng, and J. Zhong, "Fast fourier single-pixel imaging via binary illumination," *Scientific Reports*, vol. 7, no. 1, 2017.

[8] T. Ozaki, K. Sako, T. Harada, and N. Takasaki, "74-3: Development of new error-diffusion dithering method for reflective memory-in-pixel (mip) lcd," *SID Symposium Digest of Technical Papers*, vol. 48, no. 1, pp. 1089–1092, 2017. [Online]. Available: https://onlinelibrary.wiley.com/doi/abs/10.1002/sdtp.11834

[9] S. Sharma, J. Zou, and G. Fang, "Contrast enhancement of dithered images using complex wavelets and novel amplification factors," in *2018 12th International Conference on Signal Processing and Communication Systems (ICSPCS)*, 2018, pp. 1–5.

[10] R. Floyd and L. Steinberg, "An adaptive algorithm for spatial grey scale," in *Proceedings of the Society of Information Display*, vol. 17, 1976, p. 75–77.

[11] R. L. Stevenson and G. R. Arce, "Binary display of hexagonally sampled continuous-tone images," *J. Opt. Soc. Am. A*, vol. 2, no. 7, pp. 1009–1013, Jul 1985. [Online]. Available: http://josaa.osa.org/abstract.cfm?URI=josaa-2-7-1009

[12] D. Burkes, "Presentation of the burkes error filter for use in preparing continuous-tone images for presentation on bi-level devices," *The file BURKES. ARC*, 1988.

[13] P. Stucki, "Mecca-a multiple-error correcting computation algorithm for bilevel image hardcopy reproduction," in *Research report RZ1060, IBM Research Lab.*, Zurich, Switzerland, 1981.

[14] J. F. Jarvis, C. N. Judice, and W. H. Ninke, "A survey of techniques for the display of continuous tone pictures on bilevel displays," *Computer Graphics and Image Processing*, vol. 5, pp. 13–40, 1976.

# 5G energy efficiency for Internet of Things: a survey

Felipe Sousa Barbosa[1], Rodrigo Fumihiro de Azevedo Kanehisa[1] and Alberico de Castro Barros Filho[2, 3]

[1] *Universidade Estadual de Campinas, Instituto de Computação, São Paulo, Brasil*
[2] *Faculdade de Informática e Administração Paulista, São Paulo, Brasil*
[3] *Universidade de São Paulo, Departamento de Engenharia de Sistemas Eletronicos, Escola Politécnica, São Paulo, Brasil*

Reception date of the manuscript: 03/06/2020
Acceptance date of the manuscript: 09/06/2020
Publication date: 12/06/2020

**Abstract**— The Internet of Things (IoT) consists of devices capable of measuring the environment and executing tasks without human intervention. Due to its size, these devices have restrictions in processing, memory, and battery. These devices can reach a trillion nodes and, therefore, requires network connections that are capable of both handle a large number of nodes connected and low energy transmission. The fifth generation of telecommunications technology (5G) is a key concept to address those requirements as new applications and business models require new criteria such as security trustworthy, ultra-low latency, ultra-reliability, and energy efficiency. Although the next generation of connections is at its early stage, progress has been made to achieve 5G enabled IoT technologies. This paper describes a review of the main technologies such as Cloud, Software Defined Network, device-to-device communication, Evolved Package Core and Network Virtual Function Orchestration that are planned to be applied for both fields of 5G and IoT.

**Keywords**—Internet of Things, 5G technology, Survey

## I. Introduction

IoT is one of the most researched subjects in the past five years due to its capability of connecting a vast number of affordable devices, and the numbers will overdue the number of humans by many folds until 2030 [1]. In this context, the connection technologies present a large number of limitations, such as packet size, interference mitigation, reliability, packet delay, and battery power consumption.

Repetitive transmission through radio network is one of the leading causes of energy drain [2], and many of the proposed applications are solely reliant on battery power such as; (i) utility metering, (ii) precision agriculture (iii) environmental sensors, and (iv) asset tracking (bikes, fleets, and goods). Therefore, new connection technology is needed to mitigate the power consumption problem and extend the life of IoT devices and improve their usability on the network [3]. Different approaches are taken in the search for a technology that can solve the connectivity problem of IoT devices. Some techniques were developed to allow restricted devices to be able to operate on legacy networks (3G and LTE) with emphasis on the Narrowband-IoT (NBIoT) proposal for 3GPP Release 14 [4, 5]. There are also other technologies developed focusing on this segment, such as Sigfox [6], IEEE 802.15.4g [7], and LoRaWAN [8], these use receivers with high sensitivity allowing transmitters to use the same energy in sending packets.

Technologists, however, address the device's power consumption during the *non-transmitting* phase of operation (idle or sleep mode) and do not propose methods to reduce the device's power consumption when it is transmitted. The 5th generation of connection technologies, under development by 3GPP releases 14, 15 and 16, bring a new implementation of the physical layer to: (1) support a large number of connected devices and (2) use relays and Device-to-Device (D2D) communication in order to increase network coverage and reduce energy costs in transmitting IoT devices [9, 10].

This paper is organized as follows. Section II presents a theoretical framework with the leading technologies related to 5G networks focusing on IoT operation. Section III works that address the problem of 5G networks as a solution for IoT scenarios. Section IV provides concluding remarks and future work. Finally, we present the acknowledgments and contributions to this work in Section VI.

## II. Background

This section put together theoretical foundations for the understanding of the problem of energy efficiency in 5G networks for Internet-of-Things. Figure 1, on its left side, demonstrates the two macro advances in computing that enabled IoT systems: Moore's Law, which allowed for area reduction, energy consumption, and higher performance, while

Contact data: Rodrigo Fumihiro de Azevedo Kanehisa, r230255@dac.unicamp.br

keeping costs lower, and Shannon's law, which allowed for faster and cheaper wireless network connectivity. Despite these advances, there are still problems to be addressed in order to enable large-scale IoT deployment.

The center of Figure 1 summarizes two of the primary needs of IoT systems: security and energy efficiency. The first aspect concerns overcoming security issues such as eavesdropping and denial of service [11]. The second addresses the need for the low energy consumption of these systems due to the lack of power in many cases.

### *a. Evolved Packet Core*

The Evolved Packet Core (EPC) is the latest evolution of the 3GPP core network architecture. EPC is the basic set of a telecommunication network, e.g., network outbound gateways, authentication, among others [13].

In GSM, the architecture relies on circuit-switching (CS). This means that circuits are established between the calling and called parties throughout the telecommunication network. In GSM, all services are transported over circuit-switches telephony principally, but short messages (SMS) and some data are also seen [13].

When designing the evolution of the 3G system, the 3GPP community decided to use IP (Internet Protocol) as the key protocol to transport all services. It was therefore agreed that the EPC would not have a circuit-switched domain anymore and that the EPC should be an evolution of the packet-switched architecture used in GPRS/UMTS.

This decision had consequences on the architecture itself but also on the way that the services were provided. IP-based solutions, in the long term, will replace the traditional use of circuits to carry voice and short messages.

### *b. Device-to-Device*

Device-to-Device Communication (D2D) is a proposed concept for 5G networks that has a direct impact on the deployment of IoT systems. Devices can communicate directly without going through a more extensive network infrastructure. This breakdown of communication decreases base station traffic, which increases network performance and reduces power consumption [14].

### *c. Cloud*

Cloud Computing plays a crucial role in 5G due to the move from dedicated hardware to software-based network elements [15]. Cloud Computing is, therefore, a vital technology that supports the 5G infrastructure and functionalities, and that can ultimately define the success of 5G deployments.

Cloud Computing solutions range from public offerings such as Amazon Web Services (AWS), Google Compute Engine, Microsoft Azure, to name a few, to private clouds instantiated in-house through software packages provided by leading software developers such as VMWare, Mirantis, Ericsson.

Supporting these offerings are many times Open Source software solutions that can be customized by companies such as OpenStack, OpenNebula, and Eucalyptus or built upon such as Docker, Kubernetes, XenServer.

The previous solutions provide either an ecosystem of resources (Processing, Storage, Network, IAM, and others.) or a single resource (Processing in the case of Docker, Kubernetes, and XenServer). To deploy a service on top of these resources, it is necessary to orchestrate those resources. In some cases, it is even necessary to connect resources belonging to different service providers or software providers. Virtual Infrastructure Management (VIM) is, therefore, a requirement for a successful and scalable Cloud resource usage.

#### *1. VIM Solutions*

VIM (Virtual Infrastructure Management) Solutions enable the network operator to make use of Cloud Resources (Processing, Storage, and Network) to build or instantiate their network elements. The VIM solution has a strong impact on the overall 5G architecture as it will enable or limit many of the foreseen 5G functionalities.

#### *2. OpenVIM*

OpenVIM is the VIM solution of Open Source MANO (OSM), an open-source project that implements ETSI NFV architecture. OpenVIM enables all-in-one installation and is fully integrated with the other OSM components such as VNF Configuration and Abstraction and the Resource Orchestrator. OpenVIM is, therefore, the reference VIM in OSM and provides Enhanced Platform Awareness (EPA).

OpenVIM is released with Apache 2.0 license, similarly to OpenStack. OSM (which OpenVIM is part of) governance is helped by ETSI and has the support of the European Telecommunication Industry.

The project is currently in its first release (Release ONE), although it had existed before being contributed to ETSI. This means that there are various previous versions, but that the project is mostly very recent.

OpenVIM is mostly a shoehorn VIM solution to demonstrate MANO concepts. Even in the scope of OSM, OpenVIM is just one of the possible VIM solutions, being OpenStack the other most prominent VIM. Nevertheless, OpenVIM integrates perfectly with the remaining OSM components and makes it the obvious solution when all-in-one MANO solutions are desired (such as for development purposes).

Feature-wise, OpenVIM provides just about the minimum requirements of a VIM in the scope of OSM. Choosing OpenVIM as a VIM solution should be tightly coupled with choosing OSM as the overall MANO solution, and even then, one should consider the limitations of the all-in-one approach of OpenVIM.

Since in 5GInFire, at least one partner is involved in OSM, OpenVIM can be fully supported and made an adequate candidate for the VIM solution.

We enumerate the main features of OpenVIM:

1. Fully compatible with Open Source MANO (OSM);
2. Open Source License;
3. Easy to use.

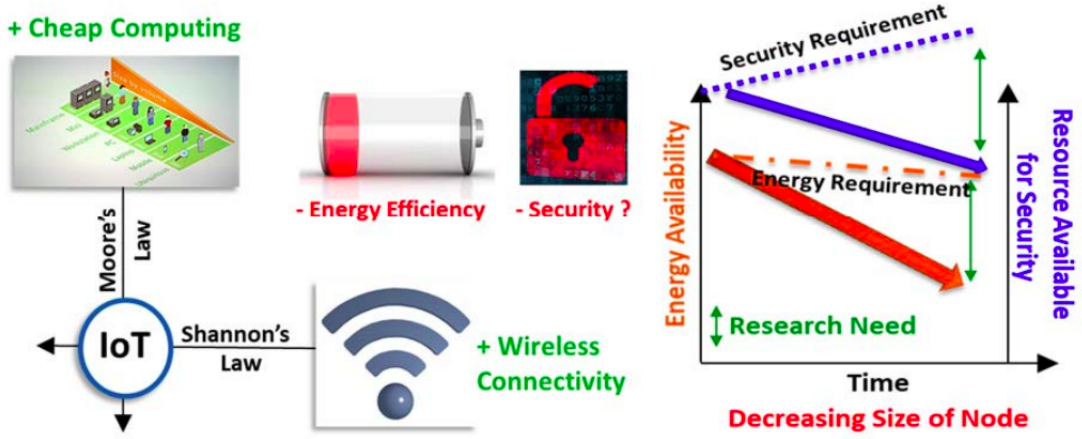



**Fig. 1:** Requirements for large-scale IoT system deployment. Sen et al. [12].

### *3. OpenStack*

OpenStack is the most prominent Open Source project on Cloud Computing. OpenStack consists of an ecosystem of several smaller projects that manages compute, storage, networking, and various support functions such as Identity, Orchestration, and Monitoring.

OpenStack adopts Apache 2.0 license, which allows for the use of the software for any purpose, to distribute it, to modify it, and to distribute modified versions of the software, under the terms of the license, without concern for royalties. This licensing enables commercial companies to profit from OpenStack results and to include OpenStack on its products and services.

OpenStack is a very mature project with a governance model laid out on top of the OpenStack Foundation. This governance model is very important as there are more than 60k contributors, many of which financed by various IT companies worldwide. With many such contributors, the vitality of the project is undeniable.

OpenStack is currently on its 14th release, and seven years have gone by since the initial release codenamed "Austin" in 2010. In these years, OpenStack has kept regular biannual release cycles which receive support from the community for periods ranging one year. The project claims to be carrier-grade, with many carriers advertising the use of OpenStack in their portfolios, nonetheless several the burden of running OpenStack is very high and requires dedicated and highly trained technicians. Documentation is extensive but spread around various locations and versions, which in turn may difficult the adoption.

OpenStack has come to dominate the Private Cloud much the same way as AWS dominates the Public Cloud. In an area where defacto standards dominate the market, it is widely accepted that OpenStack API is the most supported API in the industry. OpenStack is incredibly featured rich, which means that the number of features supported by OpenStack far exceeds the needs of 5GinFire. 5GInFire requires a VIM solution that can be responsible for controlling and managing the NFV infrastructure (NFVI) compute, storage, and network resources, these easily map to OpenStack components Nova, Cinder, and Neutron. In 5GInFire, several partners are involved in OpenStack, which makes it an excellent candidate to be adopted by the project, as these partners have the right expertise to support OpenStack in the scope of the project.

We enumerate the main features of OpenStack:

1. Prominent open-source project with Apache 2.0 license;
2. Mature project with strong community;
3. Well documented;
4. Feature rich.

## *d. SDN*

SDN (Software-defined networking) is a novel architecture that could be briefly described by providing separation of the control plane and forwarding plane and allowing through open APIs to program network dynamically. It proposes a centrally managed architecture via SDN controllers, and the role of this chapter is to strive for providing the SDN controller for the 5GinFIRE project.

### *1. SDN Controllers*

We decided to focus on 4 of the most emerging SDN controllers that are OpenDaylight, ONOS, OpenContrail, and Ryu. We studied them by following the aspects described in section 2.2.

### *2. OpenDaylight*

OpenDaylight is an open-source SDN controller first released in February 2014. The OpenDaylight Foundation is part of the Linux Foundation, a large open-source community pushing several projects in different fields. The architecture of OpenDaylight (Figure 2) reflects the general concepts of SDN. On the north are located the applications that control the network. They use the controller to gather information about the network and push new rules. The central control platform implements a set of pluggable modules to perform all the required actions. On the south are located the different routing elements of the network, either real or virtual. OpenDaylight southbound API implements a set of protocols to communicate with those devices, such as OpenFlow or Netconf.

A vast community backs OpenDaylight. In 2016, 524 developers contributed to the project, for a total of 920 develop-

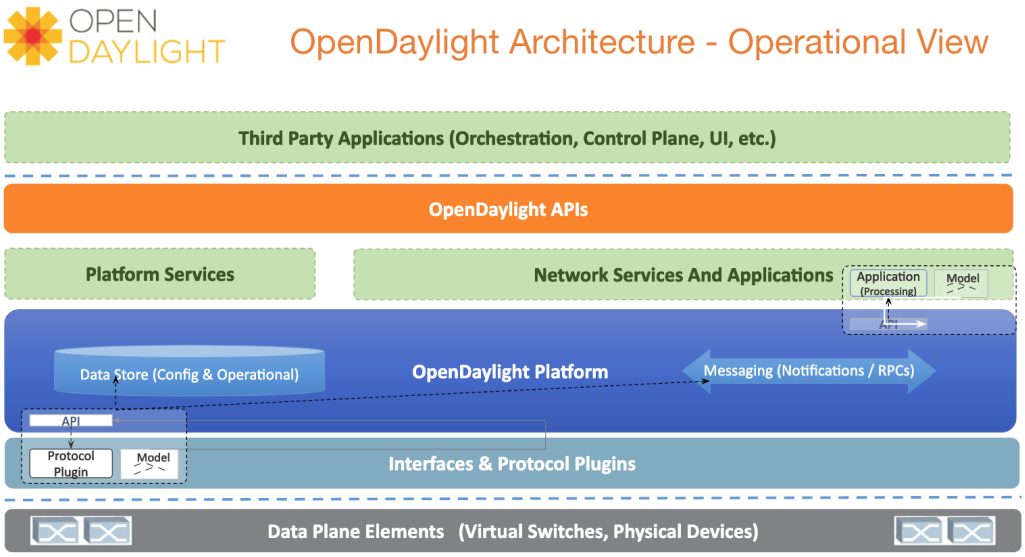



**Fig. 2:** OpenDaylight: An Operational view [16].

ers since the beginning of OpenDaylight, ranking OpenDaylight project team among the top teams of the open-source world. Moreover, the evolution of the project is keeping the same frequency compared to 2015. There are about 2000 commits per month [2]. The community can gather regularly during the OpenDaylight summits that take place once a year, or for more specific events such as the developer's forum. Besides the developers that participate in the project as volunteers, the companies can join the project as members. Today many companies are part of the project, with different levels of involvement. Among the most involved and most influent members (called platinum members), we can quote Cisco, Intel, or Red Hat [3]. This demonstrates the real interest of the communities of developers and enterprises for the OpenDaylight project. Thanks to this active community, the 5th version of the controller, Carbon, is already under development.

OpenDaylight is not only a demonstration product, as some SDN controllers have been in the past. It includes characteristics of carrier-grade solutions. OpenDaylight controllers can be clustered [4]. Clustering is a key feature since it allows the enforcement of High Availability and the fast scaling of resources, especially in the cloud context. OpenDaylight also uses microservices architecture to deal with complex problems with simple functions. This architecture is an asset for large, complex systems. Indeed, it makes each component lighter, simpler, more accessible to upgrade, and more efficient in its work.

As part of the Linux Foundation, OpenDaylight is a fully open-source, published under the Eclipse public license v1.0 [16]. Thanks to the broad community, which activities have been presented in the previous section, the project has quickly evolved to reach a high level of maturity, with about one new version every eight months. First released in February 2014 with the Hydrogen version, the project is now on its fourth stable version, Boron, released in December 2016. The next version, Carbon, is under development [5]. In addition to this swift-evolution, OpenDaylight has been proven to be stable and mature enough to be used in the industry. Per a 2016 study [6], 61% of the enterprises that have deployed an SDN solution have chosen OpenDaylight, and most companies that consider deploying SDN solutions in the future also want to start with this controller. Among the solutions using (or based on) OpenDaylight, we can find HPE Carrier SDN, Huawei Agile controller, Brocade SDN Controller, Extreme networks oneController, Inocybe Open Networking Platform or Virtuora Network Controller (Fujitsu). This adoption by industry tends to prove that OpenDaylight is a carrier-grade controller, and this is confirmed by the carrier-grade features that are implemented, such as clustering or microservices architecture.

Such popularity is probably due to the vast number of features displayed by OpenDaylight. On its southbound interface, the controller can handle various standardized protocols, including OpenFlow (all versions), Netconf [7], and OVSDB [8]. The northbound APIs are not standardized, as in any other SDN controllers, but they still offer many possibilities. Among all the available features, we can highlight NEMO [17], an intent NBI developed by Huawei that could represent a kind of standardized NBI.

If the available northbound applications or plugins do not fit the needs of the project, OpenDaylight offers rich documentation to help developers to code their features. This documentation is updated with each new release, which represents a high frequency.

Besides its efficiency as a standalone controller, OpenDaylight can also be integrated into a more general architecture such as the NFV architectural framework defined by ETSI [10]. In this framework, the controller is tightly linked with the MANagement Organization (MANO) component and the Virtual Infrastructure Manager (VIM). As part of the Linux Foundation, MANO Open-O is specifically designed to be able to work with a controller, and especially with the two controllers supported by the foundation: OpenDaylight and ONOS [18]. However, other MANO components, such as Open Source MANO (ex OpenMANO), can be used through some manipulations.

OpenDaylight profits from strong popularity lead the SDN controllers' community, and 5GinFIRE's partners naturally work with it on their internal projects and acquire then sig-

nificant competences. After having gone through all the partners, a good number of them claim to have high expertise with OpenDaylight: B-COM, ITav, UnivBris, UFU, and TID, who is a Contributor of the NetIDE project and a member of the ODL Advisory Board. Summary

As we saw in the previous section, OpenDaylight:

1. Is open-source, with a vast and active community;
2. Is mature;
3. Offers a rich standardized southbound API;
4. Offers a rich northbound API;
5. Possesses a lot of features and applications;
6. Can be integrated into an NFV architectural framework using Open-O or OSM;
7. Benefits of high partners' expertise.

### *3. ONOS*

With its first version released in December 2014, ONOS (Open Network Operating System) is the most recent mainstream, open-source controller. ONOS presents itself as an OS for networks [18], as opposed to the traditional SDN controller seen as experimental devices. As a network, OS ONOS has the same place in the architecture as a traditional SDN controller but is supposed to have much more responsibilities, such as:

1. Manage the limited resources and divide them between users;
2. Isolate different users from each other;
3. Provide abstraction to hide resource complexity;
4. Provide security;
5. Supply useful and basic features, so that developers do not have to code them again and again.

Those OS-oriented features tend to make ONOS more complete and useful than a traditional SDN controller, whose role is much more limited. ONOS supporters consider that traditional SDN controllers are rule-pushers, with not a lot of added values. Besides the attributes listed below, an essential aspect of ONOS, the most important perhaps, is the distributed aspect. ONOS is thought, from architecture to implementation, to be distributed. Distribution allows High Availability (HA) and easy scaling of resources by the addition of new servers. The Figure 3 describes the ONOS subsystem distribution.

As part of the Linux Foundation, ONOS is fully open-source. Although it is a recent project, ONOS is supported by a large community and has already many releases. The first one, Advocate, was delivered in December 2014. The 9th and last one, Ibis, was delivered in December 2016 [18]. ONOS is adopted by many professionals, which demonstrates its maturity. 23% of the enterprises that have deployed an SDN solution have chosen OpenDaylight, and 21% of the companies that consider deploying SDN solutions in the future also want to start with this controller [19]. Those figures show that ONOS is not as popular as OpenDaylight, by far. However, this might change in the future, for two reasons. Firstly, as explained in the previous section ONOS is more recent than OpenDaylight, which gives OpenDaylight an advantage, but at the same time, it is supposed to be more carefully designed, more mature, which could give it an essential advantage in the long run. Secondly, OpenDaylight and ONOS are not designed to play the same role: while OpenDaylight is more data center-oriented, ONOS is more adapted for WAN management. Since SDN is more deployed in the data centers today, it seems logical that OpenDaylight is more adopted, but this might change in the future. Among the solutions using ONOS, or a controller based on ONOS, we can find Huawei, ECI, Virtuora Network Controller (Fujitsu) or Atrium (ONF).

ONOS architecture pays much attention to its NBI and SBI. As an OS, ONOS is designed to accept any new southbound protocol to communicate with any network device and hide the complexity and diversity of the network to higher levels. Among the southbound protocols already supported, we can quote OpenFlow (all versions), Netconf, and OVSDB. The north API is not standardized, but ONOS provides a REST API for northbound applications. Besides this generic API, ONOS provides an intent framework to simplify application developers' work and a general view of the network [18].

If the available northbound applications or plugins do not fit the needs of the project, ONOS offers rich documentation to help developers to code their own features. This documentation is updated with each new release, which represents a high frequency. Regarding MANO architecture, ONOS is equivalent to OpenDaylight. As part of the Linux Foundation, Controllers Open-O is designed to integrate it [20]. It is also possible to use it in Open Source MANO.

Even if its notoriety cannot be compared to OpenDaylight, ONOS is a serious alternative and very promising. By polling partners of the project, few of them declared having strong expertise with ONOS, but for most of them, they already started considering it and improving their skills.

As we saw in the previous section ONOS:

1. Is open-source, with a larger and larger community;
2. Is mature, although it is not as much adopted as OpenDaylight;
3. Offers a rich standardized southbound API;
4. Offers a rich northbound API, including a homemade intent API;
5. Possesses features and application, maybe less than OpenDaylight;
6. Can be integrated into an NFV architectural framework using Open-O or OSM;
7. Benefits of medium expertise from the partners.

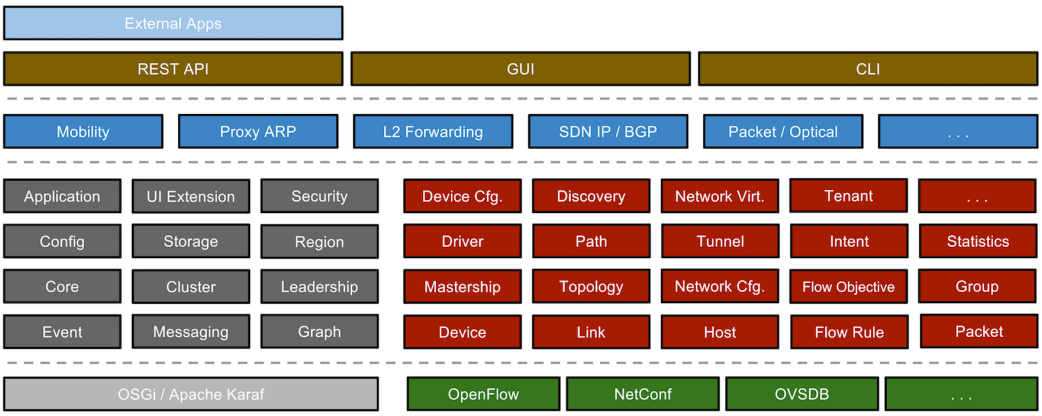



**Fig. 3:** ONOS subsystem. Bogineni et al. [18].

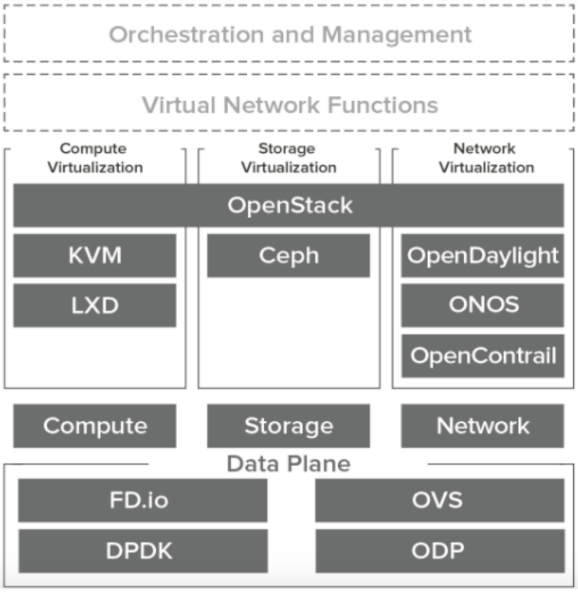



**Fig. 4:** NFV-MANO Architecture [21].

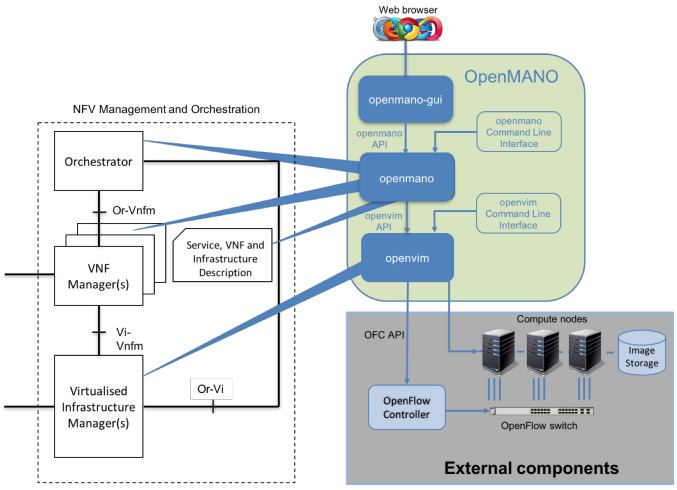



**Fig. 5:** OSM Mapping to ETSI NFV MANO [22].

### *e. NFV*

This section presents an overview of the main projects and technologies that have been identified by the 5GinFIRE consortium as relevant in the field of Network Functions Virtualization (NFV). In the following, we analyze the diverse functionalities provided by these solutions in the context of the NFV reference architecture defined by the ETSI Industry Specification Group for Network Functions Virtualization (ISG NFV). We conclude this section is presenting a summary of the main features and functionalities of the analyzed solutions, along with a set of considerations to motivate the selection of technologies to support NFV functionalities within 5GinFIRE.

The most relevant projects and open-source initiatives that focus on the development of specific solutions to support the management and orchestration of NFV services and resources. To serve as a reference, Figure 4 presents the structure of the NFV management and orchestration (MANO) system defined by ETSI and its interrelation with the other components of the NFV reference architectural framework. This subsection does not cover Virtualized Information Manager (VIM) solutions.

#### *1. Open Source MANO*

Open Source MANO (OSM) [22] is an ETSI-hosted open-source project involving leading network operators, NFV cloud providers and research and academic centers, including Telefónica, British Telecom, Telenor, Telecom Austria Group, Intel, Canonical, RIFT.io, Mirantis, VMware, 6WIND, IMDEA Networks and DELL, among others (the up-to-date list of members and participants is provided in the OSM website [23]). The project aims at providing a practical open-source implementation of an NFV Management and Orchestration (MANO) stack aligned with the NFV reference architectural framework defined by the ETSI NFV ISG [24].

The Network Service Orchestrator (NSO), based on RIFT.ware [23] from RIFT.io/Intel, takes care of the delivery of end-to-end network services, interacting with the Resource Orchestrator and the VNF Configuration & Abstraction components of the OSM architecture. It provides the point of contact in the OSM architecture to support the lifecycle management of network services, catalog management, and on-boarding/configuration of network services and VNFs, among others. The current OSM software stack implementation includes a Graphical User Interface (GUI), which provides an intuitive, easy-to-use mechanism to interact with the NSO, providing the necessary tools for VNF on-boarding and the creation and instantiation of network ser-

vices.

The resource orchestrator (RO), which is based on OpenMANO, coordinates the allocation and setup of the computing, storage, and network resources that are necessary for the instantiation and interconnection of VNFs. For this purpose, the RO may interact with multiple Virtualized Infrastructure Managers (VIMs), which may be of different types (see the Evaluation section). This component provides most of the functions associated with the NFV Orchestrator defined by the ETSI NFV framework and follows a plugin model to support the addition of other types of VIM and SDN Controllers.

Finally, the VNF Configuration and Abstraction layer (based on Juju Charms from Canonical) is aligned with the VNF Manager defined by the ETSI NFV reference architectural framework, overseeing VNF configuration per the corresponding VNF descriptors, following a model-driven approach.

This architecture of the OSM stack has been developed following four guiding principles:

1. Layering that allows the plugin replacement of the various layers (NSO, RO, VIM, etc.) with additional or alternative components;
2. abstraction, helping to clarify what is required by the different layers of the system;
3. Modularity, even within layers, that enables a plugin model to facilitate module replacements as the OSM develops and evolves;
4. Simplicity, trying to have the minimal complexity necessary to implement the governing information models in the solution.

OSM source code is available under the Apache License, Version 2.0 [22]. The implementation includes contributions at the different architectural levels from outstanding organizations in the field of NFV, particularly:

1. VIM: Telefonica's OpenMANO's OpenVIM;
2. VNFM: Juju from Canonical;
3. NFVO: Rift.ware, OpenMANO, Murano (by Mirantis).

Besides, the current release of the OSM software stack (OSM ONE), available from October 2016, includes a one-step installing process based on containers and Juju modeling, aiming at simplifying testing, customization, and deployment processes concerning the previous release (OSM ZERO). This version supports a plugin model framework that facilitates maintenance operations and future extensions of the platform while improving the interoperability with other components like VNFs, VIMs, or SDN controllers. OSM plugin model allows integrating multiple VIMs (the current release supports OpenStack, OpenVIM, and VMware vCloud Director). Release ONE allows the deployment of multi-site network services that span across multiple datacentres and includes OpenVIM as part of the OSM run-time environment to provide a reference VIM for all-in-one installations with the support of Enhanced Platform Awareness (EPA), aiming at having greater awareness about the capabilities of the platform under control.

Moreover, OSM provides a wiki with up-to-date documentation covering technical details concerning the project, along with an installation and a user guide. The initial expectation of the project is to provide new releases with a periodicity of six months.

Finally, we want to highlight that the 5GinFIRE consortium includes partners with direct involvement in the development of OSM, particularly Telefónica, as an OSM member, and UC3M, as an OSM participant.

In the following, we enumerate the main features of OSM, as previously presented:

1. ETSI-hosted open-source project;
2. Practical open-source implementation of an NFV MANO stack aligned with the ETSI NFV reference architectural framework;
3. Supported by leading network operators, NFV cloud providers, and research and academic centers (at the time of writing 26 members, 31 participants);
4. Licensed under Apache License 2.0;
5. Available up-to-date documentation;
6. Supports a plugin model framework to facilitate maintenance, future extensions, and interoperability;
7. Multi-VIM support;
8. One-step installing process;
9. Providing an intuitive, easy-to-use GUI.

## III. 5G Projects

This session discusses IoT-focused 5G standard projects. These projects were chosen because they are funded by the European Union's Horizon 2020 (H2020) project. It is noteworthy that projects linked to other areas such as mobile telephony will be treated briefly or ignored in this session.

H2020 is an EU Research and Innovation program with nearly €80 billion of funding available over seven years (2014 to 2020). Within this program is the 5G-Public-Private Partnership (5G-PPP), a stimulus program for the development of 5G (5G) telecommunication technologies.

This program aims to stimulate the study, validation, and implementation of technologies for connection in various (vertical) areas to ensure that all compatible devices can communicate with each other, even when they use different infrastructure. The requirements of the 5G standard include:

- Providing 1000 times higher wireless area capacity and more varied service capabilities compared to 2010
- Saving up to 90% of energy per service provided. The main focus will be in mobile communication networks where the dominating energy consumption comes from the radio access network
- Reducing the average service creation time cycle from 90 hours to 90 minutes
- Creating a secure, reliable and dependable Internet with a "zero perceived" downtime for services provision


**Fig. 6:** 5G-PPP Proposal [25].

- Facilitating very dense deployments of wireless communication links to connect over 7 trillion wireless devices serving over 7 billion people

- Ensuring for everyone and everywhere the access to a wider panel of services and applications at lower cost

### *a. 5G-Range*

5G-Range is a project focused on developing technologies to enable access to 5G infrastructure to provide affordable internet access with quality of service in remote areas [26].

The 5G-RANGE proposal has specified two use cases: Internet Access and High Mobility Application in Remote Areas. 5G-RANGE will be designed to achieve ten times more cell radius than the conventional 4G technology, supporting high data rates within cell sizes of 50 km radius. Based on these features, it is expected that this project will provide a cost-effective 5G system which can stimulate feasible business models for remote areas [26].

In the context of IoT, this project allows the connection of sensors and devices in remote environments, minimizing the cost of the necessary communication infrastructure. Among its applications can be mentioned use in agricultural environments.

### *b. 5G-HEART*

Healthcare, transport, and food verticals are hugely important in Europe, in terms of jobs, size, and export trade. Not only that, but they are also vital from a social point of view. 5G-HEART is one of 5G PPP Phase 3 projects focused on deploying a communications infrastructure for healthcare, transport and aquaculture industry partnerships.[27]

Providing better patient care or optimizing food production and distribution is one of the things 5G technology can be of great importance. 5G proves essential for these verticals in terms of providing a keyboard substrate for e-medicine and food supply focused devices.

Novel applications such as smart medical devices can use this technology as a communications infrastructure to provide more efficient patient care, optimizing today's healthcare system. Trials will run on sites of 5G-Vinni (Oslo), 5Genesis (Surrey), 5G-EVE (Athens), as well as Oulu and Groningen, which will be integrated to form a powerful and sustainable platform where slice concurrency will be validated at scale.

### *c. 5G!Drones*

5G!Drones is a 5G project for Drone-based Vertical Applications. The focus of this project is driving UAV verticals inside 5G networks infrastructure by providing low-latency and reliable communication, a massive number of connections, and high bandwidth requirements, simultaneously [28].

### *d. 5GSMART*

5G is foreseen as a key enabler for the future manufacturing ecosystem, termed Industry 4.0. 5G-SMART is focused on applying 5G in real manufacturing environments.

Its main idea is applying a 5G standard for technologies such as digital twins, industrial robotics, and machine vision-based remote operations. It also plans to explore new business models, including the roles of mobile network operators. The trial sites are an Ericsson factory in Kista (Sweden), a Fraunhofer IPT shopfloor in Aachen (Germany), and a Bosch semiconductor factory in Reutlingen (Germany) [29].

### *e. 5GInFire*

The 5GINFIRE aims to propose an architecture for open and Extensible 5G Reference ecosystem, along with integrating IoT projects such as FIWARE, FIRE to the 5G Networks. This environment will work as an experimental playground to validate devices, network function, and APIs before they are ported to industrially, Smart cities and other "mainstream" 5G network technologies [10]. Figure 7 demonstrates the infrastructure for IoT using the 5GInFire architecture model.

### *f. 5GTOURS*

This project aims to provide tourism and e-health services and mobility for tourists, citizens, and patients. Its test sites are Rennes, the safe city where e-health use cases will be demonstrated, Turin, the touristic city focused on media and broadcast use cases, and Athens, the mobility-efficient city that brings 5G to users in motion as well as to transport-related service providers [30].

### *g. 5G-SOLUTIONS*

5G-SOLUTIONS is the flagship ICT-19 RIA project supporting EC's 5G policy by implementing the last phase (Phase 3b) of the 5G PPP roadmap [31].

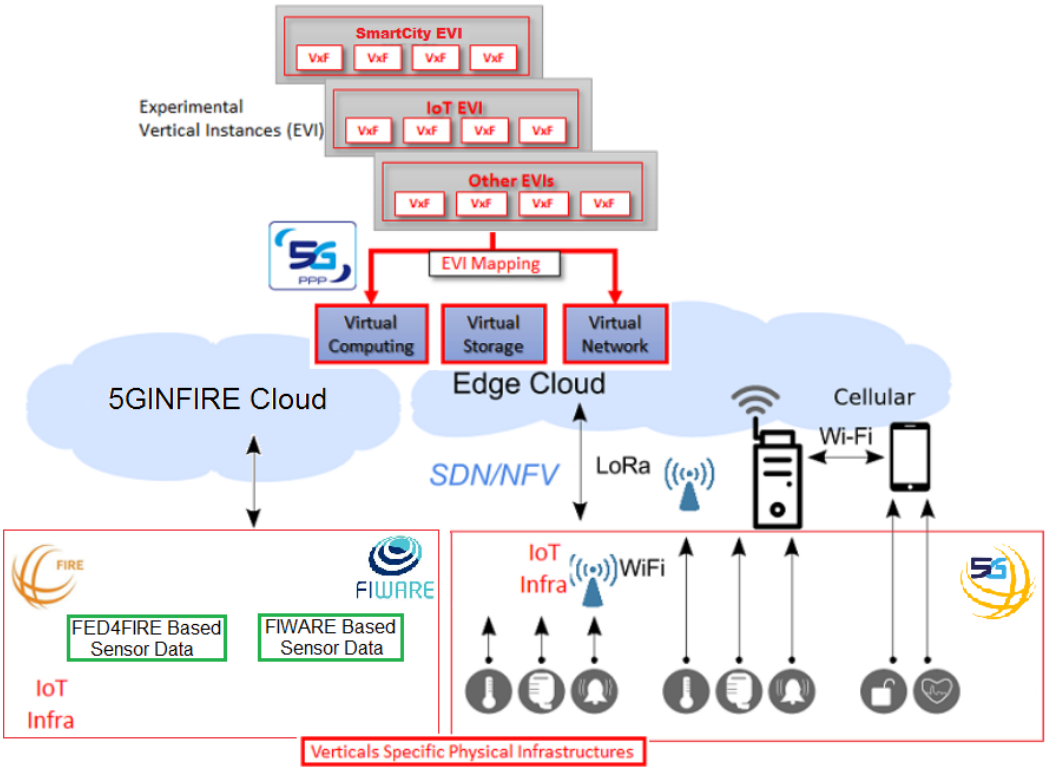



**Fig. 7:** 5GINFIRE IoT architecture. 5GinFire [10].

This project focuses on validating 5G technologies for their final implementation, and thus, bringing the 5G vision closer to realization. This will be achieved through conducting advanced field-trials of innovative use cases, directly involving end-users across five significant industry vertical domains in five countries:

- Factories of the Future, in Belgium, Ireland, and Norway
- Smart Energy, in Italy
- Smart Cities, in Ireland and Norway
- Smart Ports, in Norway
- Media & Entertainment in Greece and Norway

### *h. Comparison Table*

This session brings a comparison table between the 5G technologies mentioned. It also brings a glossary for a better understanding of the table.

On the range column, we mention the expected range of the technology. It is divided according to the list below into short, medium (mid), and long-range.

- Short Range: < 1 KM
- Mid Range: 2KM - 5KM
- Long Range: > 5KM

In the Throughput session, we mention the expected data throughput of the technology. It is divided according to the list below into short, medium (mid), and long-range.

- Small: < 200Kb/s
- Mid: 200Kb/s - 10Mb/s
- Large: > 10Mb/s

## IV. Conclusion

We can conclude that 5G is a set of technologies to provide telecommunication support for multiple different platforms. In order to meet the specific needs of each vertical, the 5G standard is subdivided into multiple designs.

Within these verticals, IoT has multiple specific needs, in scopes too narrow to be met by a single project. Therefore, several proposals were made to remedy these problems.

This survey described some of these technologies, their applications, and their scope. It is noteworthy that within the context of 5G, the standards are not yet fully defined, and are subject to change.

## References

[1] P. Cerwall, P. Jonsson, R. Möller, S. Bävertoft, S. Carson, and I. Godor, "Ericsson mobility report," *On the Pulse of the Networked Society. Hg. v. Ericsson*, 2015.

[2] N. Zaman, "Energy optimization through position responsiverouting protocol (prrp) in wireless sensor network," *International Journal of Information and Electronics Engineering*, 01 2012.

[3] P. Andres-Maldonado, P. Ameigeiras, J. Prados-Garzon, J. J. Ramos-Munoz, and J. M. Lopez-Soler, "Optimized lte data transmission procedures for iot: Device side energy consumption analysis," in *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE, 2017, pp. 540–545.

[4] Y.-P. E. Wang, X. Lin, A. Adhikary, A. Grovlen, Y. Sui, Y. Blankenship, J. Bergman, and H. S. Razaghi, "A primer on 3gpp narrowband internet of things," *IEEE communications magazine*, vol. 55, no. 3, pp. 117–123, 2017.

**TABLE 1:** COMPARISON BETWEEN TECHNOLOGIES AND PURPOSES IN 5G.

| Project | Range | Objective | Throughput |
|---|---|---|---|
| 5G-Range | Long Range | Long Range Communitations | Large |
| 5G-HEART | Short Range | E-Health | Large |
| 5G!Drones | Mid Range | UAV and Drones | Large |
| 5GSMART | Short Range | Manufacturing | Large |
| 5GInFire | Mid/Long Range | IoT | Small/Mid |
| 5GTOURS | Long Range | Tourism, E-Health, Mobility | Large |
| 5G-SOLUTIONS | Not Apply | Tests and Validation | Not Apply |


[5] A. Hoglund, X. Lin, O. Liberg, A. Behravan, E. A. Yavuz, M. Van Der Zee, Y. Sui, T. Tirronen, A. Ratilainen, and D. Eriksson, "Overview of 3gpp release 14 enhanced nb-iot," *IEEE Network*, vol. 31, no. 6, pp. 16–22, 2017.

[6] M. Lauridsen, B. Vejlgaard, I. Z. Kovacs, H. Nguyen, and P. Mogensen, "Interference measurements in the european 868 mhz ism band with focus on lora and sigfox," in *2017 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2017, pp. 1–6.

[7] U. Raza, P. Kulkarni, and M. Sooriyabandara, "Low power wide area networks: An overview," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 2, pp. 855–873, 2017.

[8] M. Rizzi, P. Ferrari, A. Flammini, and E. Sisinni, "Evaluation of the iot lorawan solution for distributed measurement applications," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 12, pp. 3340–3349, 2017.

[9] S. Parkvall, E. Dahlman, A. Furuskar, and M. Frenne, "Nr: The new 5g radio access technology," *IEEE Communications Standards Magazine*, vol. 1, no. 4, pp. 24–30, 2017.

[10] A. P. Silva, C. Tranoris, S. Denazis, S. Sargento, J. Pereira, M. Luís, R. Moreira, F. Silva, I. Vidal, B. Nogales *et al.*, "5ginfire: An end-to-end open5g vertical network function ecosystem," *Ad Hoc Networks*, vol. 93, p. 101895, 2019.

[11] J. Koo, R. K. Panta, S. Bagchi, and L. Montestruque, "A tale of two synchronizing clocks," in *Proceedings of the 7th ACM Conference on Embedded Networked Sensor Systems*, ser. SenSys '09. New York, NY, USA: ACM, 2009, pp. 239–252. [Online]. Available: http://doi.acm.org/10.1145/1644038.1644062

[12] S. Sen, J. Koo, and S. Bagchi, "Trifecta: Security, energy efficiency, and communication capacity comparison for wireless iot devices," *IEEE Internet Computing*, vol. 22, no. 1, pp. 74–81, Jan 2018.

[13] J. Roessler, "Lte-advanced (3gpp rel. 12) technology introduction white paper," *Rohde & Shwarz*, 2015.

[14] X. Lin, J. G. Andrews, A. Ghosh, and R. Ratasuk, "An overview of 3gpp device-to-device proximity services," *IEEE Communications Magazine*, vol. 52, no. 4, pp. 40–48, April 2014.

[15] D. Wübben, P. Rost, J. Bartelt, M. Lalam, V. Savin, M. Gorgoglione, A. Dekorsy, and G. Fettweis, "Benefits and impact of cloud computing on 5g signal processing," *IEEE Signal Processing Magazine*, vol. 31, p. 10, 11 2014.

[16] "Lithium." [Online]. Available: https://www.opendaylight.org/what-we-do/current-release/lithium

[17] "Openvim installation (release one)." [Online]. Available: https://osm.etsi.org/wikipub/index.php/OpenVIM_installation_(Release_One)

[18] K. Bogineni *et al.*, "Introducing onos—a sdn network operating system for service providers," *White Paper*, 2014.

[19] "Sdxcentral releases 2017 network virtualization and sdn controller report." [Online]. Available: https://www.epicos.com/article/154885/sdxcentral-releases-2017-network-virtualization-and-sdn-controller-report

[20] E. OSM, "Opensourcemano," 2016.

[21] D. Gomes, "5ginfire experimental infrastructure architecture and 5g automotive use case." [Online]. Available: https://bscw.5g-ppp.eu/pub/bscw.cgi/d302168/D2-2-5GINFIRE_Experimental_Infrastructure_Architecture_and_5G_Automotive_Use_Case-v1-0.pdf

[22] "Osm release six is now is open source is community is collaboration." [Online]. Available: https://osm.etsi.org/

[23] "Etsi sol interoperability with rift.ware." [Online]. Available: https://riftio.com/

[24] A. Israel, A. Hoban, A. Sepúlveda, F. Salguero, G. de Blas, K. Kashalkar, M. Ceppi, M. Shuttleworth, M. Harper, M. Marchetti *et al.*, "Osm release three," *Open Source MANO, Technical Overview*, 2017.

[25] G. PPP, "5g-ppp," https://5g-ppp.eu/, 2019, (Accessed on 12/01/2019).

[26] G. Range, "5g-range remote area access network for the 5th generation," http://5g-range.eu/wp-content/uploads/2019/02/5G-Range_D2.1_V3.pdf, 2018, (Accessed on 12/01/2019).

[27] "5g heart," Nov 2019, (Accessed on 12/01/2019). [Online]. Available: https://5gheart.org/

[28] "5g!drones." [Online]. Available: https://5gdrones.eu/

[29] "5g smart," Nov 2019, (Accessed on 12/01/2019). [Online]. Available: https://5g-ppp.eu/5g-smart/

[30] "5g tours," Nov 2019, (Accessed on 12/01/2019). [Online]. Available: https://5gtours.eu/

[31] "5g-solutions," Nov 2019, (Accessed on 12/01/2019). [Online]. Available: https://www.5gsolutionsproject.eu

# A Systematic Review About Use of TensorFlow for Image Classification and Word Embedding in the Brazilian Context

Thereza Patrícia Pereira Padilha[1], Lucas Estanislau Alves de Lucena[1]

[1]*Federal University of Paraiba - UFPB, Center of Applied Science and Education, Rio Tinto – PB, Brazil*

Reception date of the manuscript: 31/05/2020
Acceptance date of the manuscript: 09/06/2020
Publication date: 12/06/2020

**Abstract—** A great deal of data are available on the Internet, and it is possible to extract any type of implicit knowledge using Artificial Intelligence (AI) tools to support decision making. The open-source TensorFlow framework, developed by the Google Brain Team in 2015, is, presently, the most used tool for several AI applications, such as image classification, word embedding, and chatbot development. This paper presents results of a systematic review of the use of the TensorFlow framework for image classification and word embedding applications written in Portuguese language and in the Brazilian context. We used Google Scholar as Academic Search Engine and 90 were retrieved initially. However, just 12 were remained for reading and obtaining of the main information. Title, publication year, used domain, type of application and covered scope were collected from papers retrieved to accelerate studies in the AI area and to disseminate the potential of this framework for emerging challenges.

**Keywords—**Systematic Review, TensorFlow framework, Artificial Intelligence Applications.

## I. INTRODUCTION

The responsible area to process a high volume of data for implicit knowledge acquisition is Artificial Intelligence (AI) [1, 2]. AI is an area of Computer Science that presents several techniques and algorithms in the development of intelligent projects, allowing similar decision making to that of human beings [3]. For example, when you open the YouTube website or app, the AI implemented recommends some similar videos based on previously searched or watched videos. The most popular term in AI area today is Machine Learning, which is a method of data analysis that automates the analytical model's construction. Some Machine Learning applications include natural language processing, search engines, medical diagnostics, bioinformatics, speech recognition, handwriting recognition, image recognition, robot locomotion, and forecasting systems [4].

In the literature, we can find several frameworks for Machine Learning applications, such as TensorFlow (Google's framework) [5], Azure (Microsoft) [6], and AWS [7]. TensorFlow, for example, gained much popularity in the last years among developers who intend to apply machine learning to emerging challenges, such as preparing teachers to use intelligent tools in order to anticipate interventions, to provide personalized and real-time feedbacks, and so on. Thus, this paper aims to present results of a systematic review to identify and analyze Brazilian papers that used Google's AI framework, known as TensorFlow, for image classification and word embedding applications.

This paper is organized as follows: in section II, concepts and two kinds of applications (image recognition and word embedding) used with TensorFlow framework are presented. Section III presents the methodology employed to perform this systematic review. In section IV, the results are displayed and discussed, and, finally, in section V the conclusions are presented.

## II. TENSORFLOW FRAMEWORK AND APPLICATIONS

TensorFlow is an open-source library developed by the Google Brain Team (AI research group at Google) in 2015, written in C ++, for Artificial Intelligence applications. TensorFlow is multiplatform (Windows, MacOS, and Linux) and can be run on CPU (Central Processing Unit), GPU (Graphics Processing Unit) and TPU (Tensor Processing Units) [5]. Currently, it is considered the most used software for Machine Learning and Deep Learning applications [8]. Google, Intel, Uber, Airbnb, and DropBox, for example, are some companies that already use it in their solutions.

The differential of this framework is the use of data flow graphs, with each node representing a mathematical operation and each edge representing a data matrix (multidimensional tensor). Its main feature is the ability to quickly generate a trained predictive model, eliminating the need to reimplement it and with high precision.

There are a variety of applications that TensorFlow can be used for, such as image classification, word embedding, recommendation systems, stock prediction, and chatbot building. However, in this work, the systematic review focused only on applications such as image recognition and word embedding because they are the most used by TensorFlow. In addition, these two types of applications have transformed basic everyday actions such as unlocking a mobile screen through facial recognition or writing a

Contact data: Thereza Patrícia Pereira Padilha, thereza@dcx.ufpb.br

faster message on WhatsApp or Email by proposing a set of words according to the context presented.

Image classification applications serve, for example, to unlock a cell phone screen, to find a suspect, or even to identify an unknown object. Basically, the image classification process (for people, animals, objects, places, etc.) uses algorithms that search, compare, and try to find relationships among a given image (shown on the camera or not) and other stored images in a database.

Word embedding applications are other tasks that can be implemented by TensorFlow. For example, when you are writing a message, this type of application suggests some words according to the meaning presented, reducing the user's effort and time.

## III. METHODOLOGY

A systematic review is a form of research that uses literature on an established topic as a data source. This type of investigation provides a summary of the evidence related to a specific intervention strategy through the application of explicit and systematic methods of searching, critical appraisal, and synthesis of the selected information. In this work, the systematic review model proposed by [9] was adapted, which consisted of 3 phases: **(1) Input** – definition of one or more research questions, search string, search engine, and inclusion and exclusion criteria; **(2) Processing** - search for papers according to criteria and the defined search location; and **(3) Output** – list of filtered papers with yours relevant information.

## IV. RESULTS

### *a. Input Phase*

This work attempts to answer the following research questions:

• Primary question:

- What are the most recent papers related to TensorFlow written in Portuguese?

• Secondary questions:

- What are the most recent papers related to TensorFlow for Image Classification application written in Portuguese?

- What are the most recent papers related to TensorFlow for Word Embedding application written in Portuguese?

A search string is a string of characters used to symbolize words or terms in order to do a search on an Academic Search Engine (MBA). This systematic review was limited to only Brazilian scientific works, with the search string being searched in Portuguese, with the purpose of verifying the progress of research on this topic in Brazil. From the previously defined questions, the search string used was:

((“TensorFlow” OR “Tensorflow”) AND (“classificação de imagens” OR “incorporação de palavras”))

The search source chosen was Google Scholar (https://scholar.google.com/), because it has many papers from different areas of knowledge in several languages, is free and well known, allows searches for dates, and creates alerts and filters.

The inclusion (IC) and exclusion (EC) criteria defined for this systematic review are shown in Table 1.

**TABLE 1:** INCLUSION AND EXCLUSION CRITERIA.

| Inclusion Criteria | Exclusion Criteria |
|---|---|
| IC1 - Papers published between 2010 and 2019 | EC1 - Duplicated |
| IC2 - Papers available for download | EC2 – Papers have at least a minimum of 4 pages to a maximum of 20 pages |
| | EC3 - Papers do not related to Image Classification or Word Embedding |

### *b. Processing Phase*

Using the academic search engine Google Scholar and the search string mentioned previously, 90 documents were found. Table 2 illustrates, in more details, the number of documents remaining after the application of IC1, IC2, EC1, EC2, and EC3 criteria.

**TABLE 2:** APPLICATION OF INCLUSION AND EXCLUSION CRITERIA.

| Inclusion/Exclusion Criteria | # of Documents |
|---|---|
| Initial List | 90 |
| IC1 | 78 |
| IC2 | 74 |
| EC1 | 0 |
| EC2 | 61 |
| EC3 | 1 |
| Final List | 12 |

The IC1 criterion (papers between 2010 and 2019) kept 78 of the 90 documents recovered. The EC2, which corresponds to the number of pages, was the one that had the greatest impact on filtering, since it excluded 61 documents that, basically, were dissertations or theses. After applying the criteria, only 12 papers remained to be analyzed, 10 of which focused on Image Classification and 2 on Word Embedding.

### *c. Output Phase*

Table 3 presents a set of basic information of filtered papers, authors' names, title of the paper, publication year, used domain, type of application (Word Embedding or Image Classification) and the scope of the content covered (Theory, Test and/or Implementation).

**TABLE 3:** FINAL LIST OF PAPERS.

| # | Authors | Tittle | Year | Domain | Application | Coverage |
|---|---|---|---|---|---|---|
| 1 | Pedro A. Lelis; Filipe I. Fazanaro; J. R. Sato | Análise da influência dos hiperparâmetros no treinamento de word embeddings para a língua portuguesa | 2018 | Portuguese Language | Word Embedding | Test |
| 2 | Rafael de Sá; Kaio Ribeiro; André Aranha | Detecção e reconhecimento de faces, objetos e ações em tempo real em vídeo com raspberry pi através de uma rede neural convolucional | 2018 | Home Security | Image Classification | Test |
| 3 | Andre G. C. Pacheco | Classificação de espécies de peixe utilizando redes neurais convolucional | 2016 | Fishing | Image Classification | Test |
| 4 | Amanda C. Spolti | Classificação de vias através de imagens aéreas usando deep learning | 2018 | Traffic | Image Classification | Implementation |
| 5 | Raí G. Carvalho; Leticia T. M. Zoby | Convolutional neural networks for leaf disease classification | 2018 | Agriculture | Image Classification | Test |
| 6 | Vinicius E. Martins | Deep learning para classificação de imagens | 2018 | Livestock | Image Classification | Theory |
| 7 | Marcelo S. de Melo | Dual Scaling: Uma implementação em gpu com o TensorFlow | 2018 | Prograamming | Image Classification | Implementaion |
| 8 | Gabriel de C. Vasconcelos | Identificação da praga bicho-mineiro em plantações de café usando imagens aéreas e deep learning | 2019 | Agriculture | Image Classification | Implementaion |
| 9 | Flávio H. D. Araújo; Allan C. Carneiro; Romuere R. V. Silva; Fátima N. S. Medeiros; Daniela M. Ushizima | Redes neurais convolucionais com TensorFlow: teoria e prática | 2017 | Human Health | Image Classification | Test and Implementation |
| 10 | Maxwell M. Ribeiro; Samuel S. Guimarães | Redes Neurais utilizando TensorFlow e Keras | 2018 | Programming | Image Classification | Test and Implementation |
| 11 | Humberto da S. Neto; Clebeson C. dos Santos; Mariana R. Fernandes | Transfer Learning for Facial Emotion Recognition | 2018 | Human Emotion | Image Classification | Test |
| 12 | Rafael A. S. Andrade; Franciny M. Barreto; Esdras L. Bispo Jr. | Uso de mineração de dados para o auxílio de produção de material didático em disciplinas de algoritmos | 2019 | Education | Word Embedding | Test and Implementation |

## V. CONCLUSIONS

This paper presented a systematic review of papers written in Portuguese (Brazilian) about image classification and word embedding applications with the TensorFlow framework. After a systematic search, 12 papers were retrieved in this area, 10 being for image classification and 2 for word embedding applications.

This revealed that there are many domains for the use of these applications such as pest control in plants and emotional recognition. We recommended the papers "Use of Data Mining to Aid the Production of Didactic Material in Algorithmic Disciplines" and "Classification of Pathways through Aerial Images using Deep Learning" for reading about word embedding and image classification, respectively, because they have presented, in more details, how to use TensorFlow framework to solve emerging challenges.

In order to obtain more relevant papers, we suggest extending the search string, allowing papers written in English, and using other research sources, such as ACM or IEEE.

## VI. ACKNOWLEDGMENT

We thank the Lecturer Carlos Valdes of the Georgia Tech Language Institute for his special English review.

## REFERENCES

[1] W. L. J. Teixeira, "Big Data, Jornalismo Computacional e Data Journalism: estrutura, pensamento e prática profissional na Web de dados," *Estudos em Comunicação*, no. 12, pp. 207-222, 2012.

[2] S. Loh, *BI na era do big data para cientistas de dados: indo além de cubos e dashboards na busca pelos porquês, explicações e padrões*, kindle edition, Porto Alegre, RS, 2014.

[3] A. Bottini and A. Laurentini, "Experimenting with non instructive motion capture in a virtual environment," *The Visual Computer*, no. 17, pp. 14–29, 2001.

[4] M. J. V. Amorim, D. Barone and A. U. Mansur "Técnicas de Aprendizado de Máquina Aplicadas na Previsão de Evasão Acadêmica," in *Proc.* XIX Simpósio Brasileiro de Informática na Educação, Fortaleza, CE, Brasil, 2008, pp. 1-9.

[5] TensorFlow. An end-to-end open source machine learning platform. (2019, Set. 23). Available: https://www.tensorflow.org/

[6] Azure Machine Learning. Enterprise-grade machine learning service to build and deploy models faster. (2020, May. 27). Available: https://azure.microsoft.com/en-us/services/machine-learning/

[7] Machine Learning on AWS. Putting machine learning in the hands of every developer. (2020, May. 27). Available: https://aws.amazon.com/machine-learning/

[8] 10 Most Popular Machine Learning Software Tools in 2020 (updated). (2020, Jun. 8). Available: https://towardsdatascience.com/10-most-popular-machine-learning-software-tools-in-2019-678b80643ceb

[9] E. C. Conforto, D. C. Amaral and S. L. D. Silva "Roteiro para revisão bibliográfica sistemática: aplicação no desenvolvimento de produtos e gerenciamento de projetos," in *Proc* Congresso Brasileiro de Gestão de Desenvolvimento de Produtos, Porto Alegre, RS, Brasil, 2011, pp. 1-12.

# Multilayer Perceptron Perceptron optimization through Simulated Annealing and Fast Simulated Annealing

Pedro H. Cardoso Camelo[1] e Rafael Lima de Carvalho[1]

[1] *Curso de Ciência da Computação, Universidade Federal do Tocantins, Tocantins, Brasil*

Data de recebimento do manuscrito: 30/05/2020
Data de aceitação do manuscrito: 10/06/2020
Data de publicação: 12/06/2020

**Abstract**— The Multilayer Perceptron (MLP) is a classic and widely used neural network model in machine learning applications. As the majority of classifiers, MLPs need well-defined parameters to produce optimized results. Generally, machine learning engineers use grid search to optimize the hyper-parameters of the models, which requires to re-train the models. In this work, we show a computational experiment using metaheuristics Simulated Annealing and Fast Simulated Annealing for optimization of MLPs in order to optimize the hyper-parameters. In the reported experiment, the model is used to optimize two parameters: the configuration of the neural network layers and its neuron weights. The experiment compares the best MLPs produced by the SA and FastSA using the accuracy and classifier complexity as comparison measures. The MLPs are optimized in order to produce a classifier for the MNIST database. The experiment showed that FastSA has produced a better MLP, using less computational time and less fitness evaluations.

**Keywords**— Neural Network, Multilayer Perceptron, Simulated Annealing, MNIST Database

## I. Introdução

As redes neurais artificiais (RNAs) se popularizaram grandemente pela capacidade de aproximação de funções não-lineares. Alguns problemas de classificação múltipla podem ser aproximados por tais funções, por exemplo. O modelo *Multilayer Perceptron* (MLP) teve seu auge inicial na década de 60 [1], ao ser incorporado o algoritmo *Backpropagation*, para realizar o treinamento de seus pesos. Uma MLP é composta de pelo menos três camadas: a camada de entrada, uma camada oculta e uma camada de saída. A quantidade de neurônios na camada de entrada é dependente diretamente da quantidade de características envolvidas na aplicação. O número de neurônios assim como o número de camadas ocultas compõem parâmetros ajustáveis da rede. A camada de saída, para problemas de classificação, em geral possuem o mesmo número de classes da previsão desejada [2].

Em geral, ao se utilizar uma MLP para classificação, os parâmetros devem ser ajustados de maneira a permitir a otimização do resultado do algoritmo. Existem diversos algoritmos que auxiliam nesta etapa, como o RPROP [3] e até algoritmos evolutivos como o Algoritmos Genéticos [4]. No caso das MLPs, a escolha de seus parâmetros como configuração das camadas, peso dos neurônios e taxa de aprendizado influenciam na sua eficiência. Dentre os algoritmos capazes de realizar a otimização de tais parâmetros, este experimento considerou o algoritmo *Simulated Annealing* (SA) [5].

*Simulated Annealing* é um algoritmo metaheurístico para otimização, consistindo em uma técnica de busca local probabilística que simula o processo de recozimento utilizado na metalurgia para obtenção de estados de baixa energia em sólidos [6]. O processo metalúrgico envolve duas principais etapas [7]:

1. Aquecer o material a uma temperatura próxima de 1100$^{o}$C
2. Realizar o resfriamento acompanhado até que o material se solidifique.

Em geral o processo de SA pode ser acelerado, ao ser incorporado outras funções de resfriamento. Uma das variações populares consiste no *Fast Simulated Annealing (FastSA)*. A exemplo de sua melhora da eficiência, em [8], os autores o aplicaram com o objetivo de reduzir o tempo de execução do algoritmo para a resolução do problema de agendamento de horários para exames avaliativos.

Diante do exposto, este artigo apresenta um experimento computacional, mostrando a aplicação do Simulated Annealing e sua variação Fast Simulated Annealing no problema de otimização de parâmetros de uma rede neural MLP para classificação da base de dados de dígitos manuscritos MNIST [9]. O restante do artigo está organizado da seguinte maneira: A Seção II apresenta os algoritmos utilizados, a estruturação da base de dados e como os algoritmos foram configurados para resolução do problema em questão. Na Seção III são apresentados os resultados das simulações implementadas sob

Dados de contato: Pedro H. Cardoso Camelo, pedrohc8407@gmail.com

as medidas adotadas para o desempenho do algoritmo de classificação. Por fim, a Seção IV apresenta as considerações finais do experimento.

## II. METODOLOGIA

Nesta seção são descritos os algoritmos SA e FastSA, a base de dados MNIST e a aplicação destes para a resolução do problema.

### *a. Simulated Annealing*

O algoritmo se inicia definindo uma solução inicial aleatória dentro do espaço do problema ($curr \in P$), temperatura inicial(T) e mínima (Tmin), taxa de resfriamento ($Tx$), número de iterações(k). Em seguida ele inicia o processo de "resfriamento", em que a temperatura decresce a uma taxa definida ($C(T) = Tx * T$) até que atinja o limite mínimo ($T \geq Tmin$), geralmente 0, sendo cada iteração de resfriamento chamada de *época*. A cada época de resfriamento, é feito um laço com o máximo de $k$ iterações. Nele, a cada iteração $k$, é escolhida uma nova solução pertencente à vizinhança atual, chamada de vizinho($v \in N$), as duas então são comparadas de acordo com uma função de energia ($E$) e, caso o vizinho apresente uma melhor energia, ele se torna a solução atual, configurando um movimento. Mesmo se o vizinho se mostrar pior, ele ainda pode ser aceito de acordo com o fator de Boltzmann, que é dado por:

$$e^{-\frac{\Delta}{T}}$$

Onde $\Delta$ é a diferença das avaliações das duas soluções consideradas. Ao final a solução atual é retornada como a melhor solução. A Figura 1 mostra um pseudocódigo do SA.

**Algorithm 1:** Algoritmo Simulated Annealing

**Input:** Taxa de resfriamento
$s \leftarrow s_0$ ; ▷ *Solução inicial*
$T \leftarrow T_{max}$ ; ▷ *Temperatura inicial*
**repeat**
  **repeat** ▷ *A uma temperatura fixa*
    Gerar um vizinho aleatório $nb \in N(s)$
    Comparar as energias

$$\triangle E = \frac{E(nb) - E(s)}{E(s)}$$

    **if** $\triangle E \leq 0$ **then**
      Aceite *nb* como a nova melhor solução;
    **else if** $random(0,1) \leq B$ **then**
      Aceite *nb* como a nova melhor solução;
  **until** *Máximo de iterações (k)*;
  $T \leftarrow C(T)$ ; ▷ *Atualização da temperatura*
**until** *Temperatura mínima* ($T < T_{min}$);
**Output:** Melhor solução encontrada

### *b. Fast Simulated Annealing*

Desenvolvido por Nuno Leitea, Fernando Melícioa e Agostinho C. Rosac [8] para resolver o problema de alocação de horários para exames avaliativos, o FastSA é um algoritmo derivado do SA, possuindo alterações com o intuito de reduzir o número de avaliações (comparações entre as soluções) realizadas pelo algoritmo. Nele, são armazenados os movimentos feitos em cada época. Quando se seleciona um vizinho, é verificado se houve movimentos para ele na época anterior. Em caso negativo, ele não é avaliado e não é feito nenhum movimento. De acordo com os autores, este algoritmo consegue resultados mais rápidos, ao custo da degradação dos resultados. A Figura **??** apresenta o pseudocódigo do algoritmo utilizado para o problema dos autores.

### *c. Base de dados MNIST*

MNIST é uma base de dados de dígitos manuscritos composta por 70.000 imagens, sendo bastante popular para tarefas de aprendizado e reconhecimento de padrões. É dividida em 60.000 exemplos de treinamento e 10.000 exemplos de teste, como imagens devidamente normalizadas e centralizadas, gerando, assim, uma economia de tempo de pré-processamento. Na Figura 1 é exibida uma amostragem desta base de dados.

**Figura 1:** Amostra da base de dados MNIST [10].

### *d. Aplicação*

Nesta seção são descritos os algoritmos aplicados ao problema que este trabalho busca resolver.

#### *1. Pré-Processamento*

Antes de se iniciar a aplicação, os dados precisam ser pré-processados, sendo um procedimento padrão na maioria das atividades de aprendizado de máquina. Mesmo com o auxílio da própria base de dados, que já se encontra pré-processada, algumas características ainda precisam ser tratadas. Neste trabalho, as amostras foram transformadas em vetores unidimensionais de 784 elementos, sendo estes elementos os próprios pixels das imagens de tamanho 28x28, codificados em escala de cinza. O vetor de classes das amostras foi transformado em uma matriz bidimensional através da técnica de *One Hot Encoding*, que facilita a classificação pela rede.

#### *2. Soluções*

As soluções utilizadas no algoritmo são redes neurais MLP. A solução inicial é uma MLP com os parâmetros:

- Taxa de Aprendizado: 0.1;
- Camada Interna: 3 níveis de tamanhos iguais com valor aleatório entre 1 e 100 neurônios;

**Algorithm 2:** Algoritmo Fast Simulated Annealing

**Input:** Taxa de resfriamento

*prevBin* $\leftarrow$ *nil* ▷ *Lista contendo "n" movimentos aceitos na época anterior, inicialmente vazio*
*currBin* $\leftarrow$ *createArray*(*numExams*) ▷ *Lista contendo "n" movimentos aceitos na época atual, inicializada com zeros*
$s \leftarrow s_0$ ; ▷ *Solução inicial*
$T \leftarrow T_{max}$ ; ▷ *Temperatura inicial*
**repeat**
  **repeat** ▷ *A uma temperatura fixa*
    Gerar um vizinho aleatório $nb \in N(s)$; *examToMove* $\leftarrow$ *selectExamFromSolution*(*nb*) ▷ *Seleciona um exame que será movido*
    **if** *prevBin* = *nil or*(*prevBin* <> *nil and prevBin*[*examToMove*] > 0) **then**
      Calcular a energia ($\triangle E$)
      **if** $\triangle E \leq 0$ **then**
        Aceite *nb* como a nova solução;
      **else if** $random(0,1) \leq B$ **then**
        Aceite *nb* como a nova solução;
      **if** *nb foi aceito* **then**
        *currBin*[*examToMove*] $\leftarrow$ *currBin*[*examToMove*] + 1
  **until** *Condição de equilíbrio* ▷ *Exemplo: máximo de iterações alcançado*;
  $T \leftarrow C(T)$; ; ▷ *Atualização de temperatura*
  prevBin $\leftarrow$ updateBin(T,currBin) ; ▷ *Se T>=Tmin, prevBin*$\leftarrow$ *currBin e zere currBin*
**until** *Critério de parada*;

**Output:** Melhor solução encontrada

- Máximo de iterações: 200
- Função de ativação: ReLU, Unidade Linear Rectificada;
- Camada de entrada: 784 neurônios, sendo os pixels das imagens de tamanho 28$x$28.
- Camada de Saída: 10 neurônios, sendo as classificações possíveis, que vão de 0 a 9.

Os parâmetros que foram alterados para seleção da melhor rede são a Taxa de Aprendizado e a configuração da Camada Interna. Assim, a solução vizinha é escolhida a partir da solução atual, alterando os parâmetros da Camada Interna e Taxa de Aprendizado, seguindo as regras:

- Quantidade de níveis da camada interna: Soma a um valor inteira aleatório entre -2 e 2, gerando a possibilidade de aumentar, diminuir ou se manter inalterada;
- Quantidade neurônios por nível: Soma ao o valor -4, 0 ou 4, visto em testes que as mudanças mais significativas na energia se davam ao se utilizar estes valores;
- Taxa de aprendizado: Soma a um valor aleatório obtido de um distribuição uniforme entre -0,05 e 0,05.

### *3. Energia*

A função de energia é dada pela acurácia da solução, que é o resultado de seu aprendizado.

$$E(s) = score(S)$$

### *4. Resfriamento*

O resfriamento é realizado através da função:

$$C(T,k) = T/(1+n).$$

Sendo $n$ a época em que se encontra o algoritmo.

### *e. Treinamento*

### *1. Busca*

Para os dois algoritmos, SA e FastSA, foram definidos os parâmetros $k$, $T$ e $Tmin$ como 5,0, 2,0 e 0,001 respectivamente. Para aumentar ainda mais a confiabilidade dos resultados, os algoritmos foram executados, cada um, 10 vezes, gerando 10 redes cada. Estas então foram comparadas de acordo com sua acurácia e algoritmo utilizado, resultando em 2 redes, uma resultante do SA e outra do FastSA.

### *2. Tamanho da Base*

Como os algoritmos demandam um tempo de execução razoavelmente longo, o processo pode se tornar custoso pois os dados são utilizados extensivamente durante a execução, principalmente no processo do SA. Para sanar este problema, foi utilizada inicialmente uma parcela de dez por cento (10%) da base, respeitando as proporções entre a divisão de treinamento e teste. Após a finalização da busca, as redes resultantes são treinadas com a totalidade dos dados.

## III. RESULTADOS

Para a avaliação dos algoritmos, foram utilizados os resultados de tempo de execução, número de avaliações e acurácia da rede obtida. Para um maior aprofundamento, foram utilizadas outras métricas para avaliar as redes. As Tabelas 1 e 2 mostram as comparações entre os resultados de execução dos algoritmos e suas respectivas redes obtidas.

Como se pode observar, o algoritmo FastSA obteve resultados superiores em relação ao tempo computacional e a quantidade de chamadas à função de avaliação. Além do mais, ainda que a degradação dos resultados é um fator apontado como esperado, o FastSA permitiu gerar uma

**TABELA 1:** RESULTADO DE DESEMPENHO DOS ALGORITMOS AVALIADOS.

| Alg. | Tempo (seg.) | Acc. (Parcial) | Nº de Aval | Acc (Total) |
|---|---|---|---|---|
| SA | 4350,06 | 0,9070 | 300 | 0,9656 |
| FastSA | 482,58 | 0,9090 | 50 | 0,9663 |

**TABELA 2:** MELHORES CONFIGURAÇÕES DE MLPS OBTIDAS PELOS ALGORITMOS SELECIONADOS.

| Alg. | Camada Interna | Taxa de Aprendizado |
|---|---|---|
| SA | 4 níveis<br>85 Neur. / nível | 0,14 |
| FastSA | 3 níveis<br>79 Neur. / nível | 0,06 |

MLP menos complexa, enquanto que a diferença nas acurácias foi mínima, sobretudo ainda superior na configuração encontrada pelo FastSA.

## IV. CONCLUSÃO

Este artigo apresentou a utilização do *Simulated Annealing* e sua variante *Fast Simulated Annealing* para otimização dos parâmetros de uma rede neural MLP, com objetivo de maximizar a acurácia de classificação na base de dados de dígitos manuscritos MNIST. Pelo experimento, observou-se que o algoritmo *Fast Simulated Annealing* obteve resultados superiores em relação ao *Simulated Annealing* convencional nos seguintes índices: 83,33% menos chamadas à função de avaliação, tempo computacional 88,91% menor.

Em relação à Rede Neural gerada, percebeu-se que a acurácia dos melhores classificadores resultantes tanto no SA quanto no FastSA foi basicamente a mesma (ou seja, ambos permitiram ajustes para um classificador otimizado). Sobretudo, vale destacar que a rede neural gerada pelo FastSA é de menor complexidade.

## REFERÊNCIAS

[1] S. Haykin, *Neural networks: a comprehensive foundation*. Prentice Hall PTR, 1994.

[2] S. Raschka, *Python machine learning*. Packt Publishing Ltd, 2015.

[3] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: the rprop algorithm," in *IEEE International Conference on Neural Networks*, March 1993, pp. 586–591 vol.1.

[4] W. Kinnebrock, "Accelerating the standard backpropagation method using a genetic approach," *Neurocomputing*, vol. 6, no. 5-6, pp. 583–588, 1994.

[5] P. A. Castillo, J. J. Merelo, J. González, V. Rivas, and G. Romero, "Saprop: Optimization of multilayer perceptron parameters using simulated annealing," in *International Work-Conference on Artificial Neural Networks*. Springer, 1999, pp. 661–670.

[6] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[7] S. Kirkpatrick, "Optimization by simulated annealing: Quantitative studies," *Journal of statistical physics*, vol. 34, no. 5-6, pp. 975–986, 1984.

[8] N. Leite, F. Melício, and A. C. Rosa, "A fast simulated annealing algorithm for the examination timetabling problem," *Expert Systems with Applications*, vol. 122, pp. 137–151, 2019.

[9] L. Deng, "The mnist database of handwritten digit images for machine learning research [best of the web]," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 141–142, 2012.

[10] J. Steppan. (2020) Own work, cc by-sa 4.0. Tomado de https://commons.wikimedia.org/w/index.php?curid=64810040 (30/05/2020).