

Uma Estratégia de Otimização Baseada no MOPSO Aplicada em Subsistemas Elétricos de *Data Centers*

F. M. Sousa Sobrinho¹, G. R. A. Callou¹, W. S. Leonardo¹, B. C. S. Nogueira²

¹Departamento de Computação – Universidade Federal Rural de Pernambuco (UFRPE)
Rua Dom Manoel de Medeiros, s/n. Recife/PE - Brasil.

²Instituto de Computação – Universidade Federal de Alagoas (UFAL)
Av. Lourival Melo Mota, s/n. Maceió, AL – Brasil.

{francisco.montesousa, gustavo.callou, wenderson.leonardo}@ufrpe.br

bruno@ic.ufal.br

Abstract. *Data center infrastructures must have high availability, low cost, and high energy efficiency. However, these objectives are often conflicting. This paper presents a strategy based on the MOPSO, a multi-objective particle swarm optimization, to improve the design of electrical data center architectures. To show the applicability of the proposed strategy, we present a comparative study between a brute force algorithm and the proposed strategy. Six models of electrical architectures were defined, the results showed that the application of the proposed strategy reduces the execution time by up to 870 times and showed that the algorithm can generate the approximate Pareto frontier with a difference of 3%.*

Resumo. *As infraestruturas de data centers devem ter alta disponibilidade, baixo custo e alta eficiência energética. No entanto, esses objetivos são muitas vezes conflitantes. Este artigo apresenta uma estratégia baseada no MOPSO, otimização multiobjetivo de enxame de partículas, para melhorar o projeto das arquiteturas elétricas de data centers. Para mostrar a aplicabilidade, apresentamos um estudo comparativo entre o algoritmo de força bruta e a estratégia proposta. Foram definidos seis modelos de arquiteturas elétricas, os resultados mostraram que a aplicação da estratégia proposta reduz o tempo de execução em até 870 vezes e mostram que o algoritmo consegue gerar a fronteira aproximada de Pareto com uma diferença de 3%.*

1. Introdução

Ultimamente, a alta demanda energética por parte dos *data centers* tem se mostrado um desafio para projetistas. Além disso, a necessidade de alta disponibilidade destes ambientes é imprescindível tanto para expansão de serviços tecnológicos como para prover suporte às mídias sociais e serviços de streaming [Koot and Wijnhoven 2021]. Um *data center* é composto por três pilares: (i) subsistema elétrico, (ii) subsistema de resfriamento e (iii) subsistema de tecnologia da informação. Em particular, o subsistema elétrico possui critérios que precisam ser atendidos por uma equipe capacitada tanto durante o projeto, quanto na manutenção [Isaak 2021]. Dentre estes critérios, uma infraestrutura elétrica dos *data centers* deve oferecer alta disponibilidade, com custo reduzido e eficiência energética

maximizada. Muitas vezes estes critérios são conflitantes, por exemplo, considerando duas arquiteturas similares, mas com equipamentos diferentes, em que a primeira possui maior disponibilidade que a segunda. Em contrapartida, a segunda, mesmo com menor disponibilidade, pode ter um custo inferior em relação à primeira. Esses cenários com objetivos conflitantes são problemas combinatórios complexos que os projetistas de *data center* precisam solucionar em tempo hábil.

Este trabalho propõe uma estratégia de otimização do projeto do subsistema elétrico de *data centers* considerando as seguintes métricas: disponibilidade, custo e eficiência energética. Como o problema envolve objetivos conflitantes, foi adotada uma meta-heurística multiobjetivo que busca otimizar de forma integrada a disponibilidade, o custo e o consumo energético. A fim de testar a aplicabilidade da estratégia proposta, será feita a comparação de seu conjunto solução com o conjunto solução ótimo de Pareto de uma técnica de otimização combinatória (algoritmo de força bruta) que avalie todos os cenários. Os resultados mostram que o conjunto das soluções da estratégia proposta se aproxima da curva de Pareto ótimo do algoritmo força bruta em uma diferença máxima de 3%. Vale ressaltar que esta estratégia foi codificada e integrada ao módulo de otimização da ferramenta *Stars* proposta por [Souza Leonardo and Callou 2021], visando auxiliar os projetistas durante a modelagem do subsistema elétrico de *data center*.

O artigo está organizado de acordo com as seções a seguir. A Seção II apresenta os trabalhos relacionados. A Seção III apresenta a fundamentação teórica para auxiliar no entendimento do trabalho. A Seção IV explica a metodologia adotada. A Seção V apresenta os modelos propostos para representar as arquiteturas, bem como poder avaliar as métricas (disponibilidade, custo e exergia operacional). A Seção VI explica a codificação da arquitetura e descreve o algoritmo proposto baseado no MOPSO. Por fim, a Seção VII conclui o trabalho e apresenta possíveis direcionamentos da pesquisa.

2. Trabalhos Relacionados

Vários trabalhos vêm aplicando modelagens das infraestruturas de *data centers* a partir de técnicas de otimização, buscando encontrar soluções sustentáveis e com alta disponibilidade. Em [Masanet et al. 2020], os autores fizeram uma análise das estimativas globais de uso de energia do *data center*. Este trabalho fornece aos formuladores de políticas e analistas de energia uma compreensão recalibrada do uso global de energia do *data center*, seus fatores e potencial de eficiência de curto prazo. Este trabalho apesar de mostrar que o estudo do impacto do consumo de energia e formas de otimização devam ser adotadas em seus projetos não detalha ou propõe uma estratégia de otimização multiobjetivo que minimize o impacto de consumo de energia, mantendo a disponibilidade dos *data centers*.

Os autores em [Abualigah and Diabat 2021] apresentaram um algoritmo de otimização híbrido para problemas de agendamento de tarefas multiobjetivo em ambientes de computação em nuvem. O algoritmo de otimização foi aprimorado pela utilização da evolução diferencial baseada na elite como uma técnica de busca local para melhorar sua capacidade de exploração e evitar ficar preso em ótimos locais. No entanto, este trabalho focou somente problemas de agendamento de tarefas não ampliando o estudo para os equipamentos que compõem a infraestrutura do ambiente de computação em nuvem.

Os autores em [Li et al. 2020] propuseram uma técnica para otimizar o controle

de resfriamento do *data center* por meio da estrutura emergente de aprendizado por reforço profundo (DRL). Foi introduzido um mecanismo de validação de subestimação (DUE) para a rede crítica para reduzir a potencial subestimação do risco causado pela aproximação neural. O foco dos autores foi somente na infraestrutura de resfriamento e, ainda assim, não focaram na avaliação da disponibilidade e do custo.

Nenhum dos artigos adotam uma estratégia de otimização multiobjetivo, que seja capaz de maximizar a disponibilidade e minimizar o custo e exergia operacional do sub-sistema elétrico de *data center*.

3. Fundamentação Teórica

3.1. Disponibilidade

A disponibilidade é a probabilidade de um sistema funcionar corretamente em um momento específico ou durante um período de tempo predefinido [Nguyen et al. 2019]. A disponibilidade de um sistema (D) pode ser computada a partir dos tempos de falhas (MTTF - Mean Time To Failure) e reparo (MTTR- Mean Time To Repair) do sistema de acordo com a equação: $D = \frac{MTTF}{MTTF+MTTR}$.

3.2. Custo

O custo é um dos objetivos a serem otimizados neste trabalho. O custo (C) de um modelo de infraestrutura elétrica de *data center* leva em conta dois outros custos. O primeiro é o custo de aquisição (CA), obtido por pesquisa de mercado, e o outro é o custo operacional (CO), computado por $CO = E_{consumida} \times T \times E_{custo} \times D$, onde o $E_{consumida}$ é a energia consumida, T o tempo de operação do equipamento e D a disponibilidade. Sendo assim, o custo é dado por: $C = CA + CO$.

3.3. Exergia

A exergia é uma métrica utilizada para quantificar a energia desperdiçada de um equipamento, ou seja, energia não convertida em trabalho útil. Seu conceito vem da segunda lei da termodinâmica. A quantificação da exergia operacional (Ex_{op_i}) [Callou et al. 2014] é dada por: $Ex_{op} = \sum_{i=1}^n Ex_{op_i} \times T \times D$, onde i representa cada equipamento, n a quantidade destes, T o tempo de operação do equipamento e D a disponibilidade.

3.4. Diagrama de Bloco de Confiabilidade (RBD)

O RBD é usado para mostrar as interdependências entre todos os elementos de um sistema. Esta modelagem permite representar uma infraestrutura elétrica do *data center*, onde um equipamento tem suas interações e dependências com outros equipamentos [Wang et al. 2004]. Esta modelagem formal pode representar componentes conectados em série ou em paralelo, conforme mostrado na Figura 1. Na composição em série Figura 1(a)), se houver a falha de um equipamento, o sistema todo para de funcionar. Já na disposição em paralelo (Figura 1(b)), o sistema para de funcionar se todos os equipamentos falharem.

O cálculo da disponibilidade do RBD em série com n blocos é dado pela Equação 1 e para uma disposição paralela pela Equação 2:

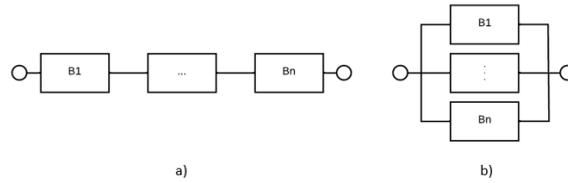


Figura 1. Estrutura básica do RBD. a) Série. b) Paralelo

$$D_s = \prod_{i=1}^n D_i \quad (1)$$

$$D_p = 1 - \prod_{i=1}^n (1 - D_i) \quad (2)$$

onde D_i é a disponibilidade do bloco i de um total de n blocos.

3.5. Redes de Petri Estocástica (SPN)

A SPN é um formalismo usado para especificar sistemas dinâmicos de eventos discretos, mas também permite análise estocástica quanto a mudança de estado deste sistema. Nas Redes de Petri estocásticas as transições podem disparar após um atraso probabilístico associado às transições temporizadas. A definição formal das SPNs pode ser encontrada em [Marsan 1990].

A Figura 2 mostra um exemplo em SPN que possui dois lugares para representar os estados de um equipamento (UPS). Um token no lugar UPS_ON representa o estado em pleno funcionamento, já um *token* no outro lugar (UPS_OFF) representa o estado quebrado. Esse modelo tem duas transições estocásticas, e que representam os períodos de falha e reparo do equipamento.

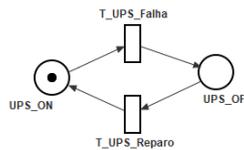


Figura 2. Estrutura SPN de um UPS ativo

4. Metodologia

A Figura 3 ilustra a metodologia proposta neste trabalho, cujas etapas para realizar a otimização dos modelos das arquiteturas elétricas de *data centers* são descritas ao longo desta seção. A primeira etapa da metodologia é o entendimento do sistema. Esse estudo modela o sistema elétrico de *data center*, segundo padrão adotado por [Association 2005] que é, na arquitetura base, composto por UPS (*Uninterruptible Power Supplies*), SDT (*Step Down Transformer*), *subpanel* e *powerstrip*.

A segunda etapa da metodologia proposta corresponde à criação dos modelos (ex., RBDs e SPNs). Esses modelos são criados através da ferramenta Stars

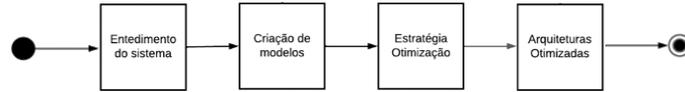


Figura 3. Metodologia

[Souza Leonardo and Callou 2021], que auxilia na integração da estratégia de otimização com o ambiente de avaliação do Mercury [Oliveira et al. 2017].

Na terceira ocorre a execução do algoritmo proposto e implementado na ferramenta *Stars*, sendo, baseado no MOPSO proposto por [Alvarez-Benitez et al. 2005], para realizar a otimização multiobjetivo. Basicamente, as arquiteturas modeladas serão codificadas como partículas que têm velocidade e uma posição aleatória no espaço de busca.

Na última etapa, para validação do conjunto de soluções gerados pelo algoritmo proposto é feita a análise estatística do conjunto solução gerado e do conjunto Pareto ótimo, este último é gerado através de um algoritmo combinatório multiobjetivo (força bruta). Caso as arquiteturas otimizadas não tenham desempenho similar àquela gerada pelo força bruta, é necessário voltar para etapa de Criação de modelos a fim de melhorar os resultados.

5. Modelos

Esta seção apresenta os modelos RBD e SPN adotados para avaliação das métricas de disponibilidade, custo e exergia operacional de subsistema elétricos de *data center*. Para criação dos modelos foram consideradas seis configurações de infraestrutura elétrica de *data center* baseados nas [Association 2005]. A Figura 4 apresenta os modelos das arquiteturas adotadas neste estudo.

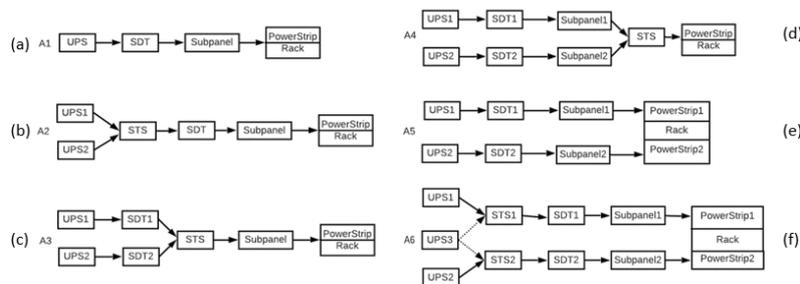


Figura 4. configuração das arquiteturas

A arquitetura A1 (Figura 4 (a)) não possui equipamentos em redundância. Nas Arquiteturas (A2-A5) a redundância de equipamentos estão em *hot standby* explicado na subseção 5.1. Já na Arquitetura A6 (Figura 4 (f)), além de ter equipamentos em redundância *hot standby*, o UPS_3 está em redundância *cold standby* (Seção 5.2).

5.1. Modelo em Redundância *Hot Standby*

Na redundância *hot standby* o equipamento redundante sempre fica ativo, assim sendo, quando houver falha no equipamento principal, esse pode ser substituído pelo equipamento redundante sem um atraso de tempo de ativação. A Figura 5 apresenta a arquitetura

tra A5 que possui todos seus equipamentos principais com seus respectivos redundantes ativos.

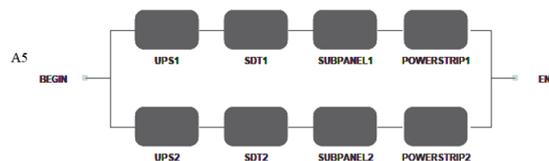


Figura 5. arquitetura A5 redundância hot standby

5.2. Modelo em Redundância Cold Standby

Em uma redundância cold standby o componente redundante não está ativo. Ele espera para ser ativado quando o módulo principal falhar. Quando o módulo principal falha, a ativação do módulo redundante ocorre em um certo período de tempo que é denominado de Mean Time To Activate (MTTA).

Existem cenários em que apenas o RBD seria insuficiente para se avaliar a métrica da disponibilidade. É o caso de sistemas que adotam redundâncias em cold standby, por exemplo. Nesses casos, a metodologia adotada indica a utilização da modelagem em SPN. Na arquitetura A6, o equipamento UPS3 só será ativado quando o UPS1 ou UPS2 estiverem em situação de falha. A Figura 6 apresenta o modelo SPN proposto para representar esse caso.

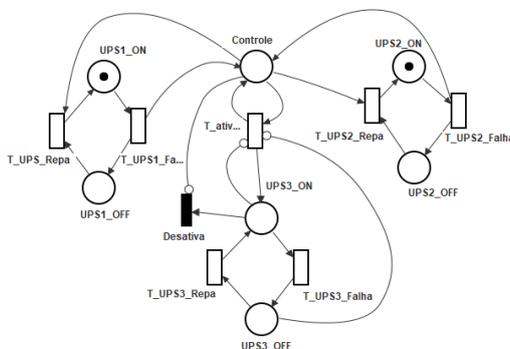


Figura 6. Modelo SPN representando UPS3 em redundância cold standby

Nesse modelo, a falha do UPS1 ou UPS2 faz com que um token seja gerado no lugar Controle, habilitando a transição T_ativ que é responsável pela ativação do UPS3. Após o tempo de ativação do UPS3, o token de Controle é consumido e o lugar UPS3_ON, inicialmente vazio, receberá um token. A disponibilidade desse modelo é computada por: $P\{(\#UPS1_ON = 1)OR(\#UPS2_ON = 1)OR(\#UPS3_ON = 1)\}$.

A estratégia proposta faz uso da modelagem em conjunto de SPN e RBD. A Figura 7 apresenta tal estratégia proposta, onde a redundância em cold standby é modelada em SPN. O resultado da avaliação desse modelo é utilizado como parâmetro de entrada para um bloco RBD (B1), conforme mostrado na Figura 7(c). Uma vez que se tem o RBD completo, já com os blocos B1 e B2 parametrizados, pode-se avaliar todo o sistema e, assim, obter a disponibilidade de toda arquitetura analisada.

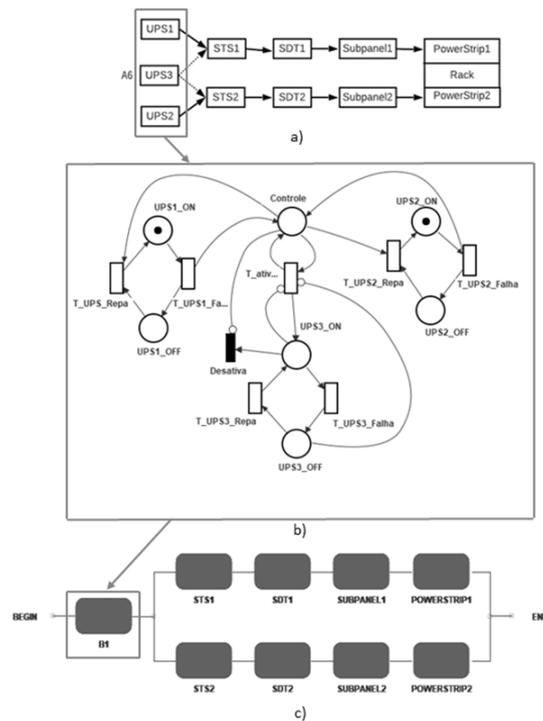


Figura 7. Integração entre modelos. Modelagem Cold standby (UPS1, UPS2 e UPS3) em SPN convertida em um bloco RBD - B1

6. Estratégia Proposta

A estratégia adotada é baseada no MOPSO (do inglês, *Multi-Objective Particle Swarm Optimization*), técnica utilizada para lidar com problemas de otimização multiobjetivo [Alvarez-Benitez et al. 2005]. O Algoritmo 1 apresenta uma adaptação do MOPSO tendo como destaque a codificação da posição da partícula (arquitetura) no espaço de busca codificada por uma tupla de três elementos sendo cada um deles o valor da função objetivo explorada neste trabalho (disponibilidade, custo e exergia operacional). Há também, um operador para descartar as piores soluções, foi escolhido o *crowding distance* (CD) [Sun et al. 2020].

É importante destacar que em cada arquitetura analisada iremos otimizar as métricas de disponibilidade, custo e exergia operacional. Cada tipo de arquitetura possui seu nível de tolerância a falha devido a redundância dos equipamentos que a compõem. A codificação de cada equipamento é feita a partir de uma base de dados que contém uma coleção para cada equipamento com os seguintes parâmetros distintos: MTTF, MTTR, custo de aquisição e eficiência energética.

Os indivíduos são constituídos tomando por base uma arquitetura pré-definida (A1-A6) no momento do uso algoritmo. Quando criado um indivíduo, este recebe um conjunto de equipamentos. Cada um desses equipamentos possui apenas o seu tipo e conexão com os demais equipamentos na arquitetura. A Figura 8 ilustra como é feita a construção de um indivíduo.

A codificação do indivíduo utiliza uma base de dados que é mapeada num dicionário usando como chaves Strings que se referem aos tipos de componentes exis-

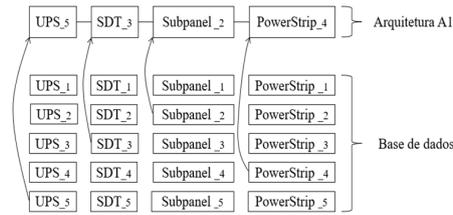


Figura 8. Codificação do Indivíduo

tentes. Então, os equipamentos da arquitetura são percorridos um a um, e a partir desse mapeamento uma função de escolha aleatória e sem repetição seleciona um equipamento do mesmo tipo e insere seus atributos (MTTF, MTTR, custo de aquisição e eficiência energética) no equipamento do indivíduo que está sendo percorrido no momento. Esse processo se repete até o último equipamento do indivíduo.

Algoritmo 1 – Estratégia baseada no MOPSO

Entrada: $N, g, f_k(X), tipo_Arquitetura \triangleright N$ tamanho da população, g gerações, $f_k(X)$ são as k funções objetivas, $tipo_arquitetura$ (A1-A6).

```

01: listArq, REP  $\leftarrow \emptyset$ ;
02: para  $i \leftarrow 0$  até  $N-1$  faça
03: listArqi.( $x$ )  $\leftarrow gerarPosiçãoAleatória$  ( $map < tipo\_Arquitetura, base\_dados >$ );  $\triangleright$  População inicial.
04:  $P_{best}^i \leftarrow listArq^i$ .( $x$ );
05: listArqi.( $vel$ )  $\leftarrow 0$ ;
06: para  $k \leftarrow 0$  até  $k-1$  faça
07: avaliarMetrica  $\leftarrow (listArq^i).metric^k$ ;  $\triangleright$  Avaliar métricas através do método metric ferramenta Stars.
08: fim para
09: fim para
10: REP  $\leftarrow ((listArq).ranquear) \triangleright non-dominated\ sorting$ 
11: para  $i \leftarrow 0$  até  $g-1$  faça
12: para  $j \leftarrow 0$  até  $N-1$  faça
13: listArqj.( $vel$ )  $\leftarrow W \cdot listArq^j$ .( $vel$ ) +  $R1 \times (P_{best}^j - listArq^j$ .( $x$ )) +  $R2 \times (REP^j$ .( $x$ ) - listArqj.( $x$ ));
14: listArqj.( $x$ )  $\leftarrow listArq^j$ .( $x$ ) + listArqj.( $vel$ );
15: listArqj.( $x$ )  $\leftarrow (listArq^j$ .( $x$ )).controleFronteira;
16: para  $k \leftarrow 0$  até  $k-1$  faça
17: calcularObj  $\leftarrow (listArq^j).metric^k$ ;  $\triangleright$  Avaliação das arquiteturas através do método metric ferramenta Stars.
18: fim para
19: se  $randNum(100) \leq 10$  então
20: (listArqj.( $x$ )).aplicarTurbulência;
21: fim se
22: fim para
23: REP  $\leftarrow ((listArq).ranquear)$ 
24: REP  $\leftarrow (REP).descartarPiores \triangleright crowding\ distance$ 
25:  $P_{best}^i \leftarrow listArq^i$ .( $x$ );
26: fim para
27: retorna REP

```

O Algoritmo 1 tem como entradas o tamanho da população, número de gerações do algoritmo, as funções alvo da otimização (disponibilidade, custo total e exergia operacional) e o tipo de arquitetura (A1-A6). O seu retorno é um conjunto solução Pareto aproximado contendo arquiteturas otimizadas. A primeira linha cria duas listas vazias, uma para armazenar o conjunto de arquiteturas iniciais (população inicial), e a outra para armazenar o conjunto das arquiteturas não-dominadas (conceito em [Deb and Goel 2001]). Na linha 2, é inicializado uma estrutura de repetição que atribui de forma aleatória a posição para cada arquitetura no espaço de busca (linha 3); o P_{best} , que representa a melhor posição da partícula, começa com sua posição inicial da arquitetura (linha 4) e vel representa a velocidade de cada arquitetura, que é inicializada com zero (0) na linha 5.

As linhas 6 a 8 realizam chamadas sucessivas ao método *avaliarMetrica*, passo onde o algoritmo se comunica com o ferramenta *Stars* que por sua vez converte arquitetura passada como entrada em linguagem de *script* compatível com o *Mercury*. Neste último,

pelos métodos *metric* e *solve* se obtém o valor de cada função objetivo das arquiteturas. Sendo assim, para cada função objetivo das k funções objetivo é avaliado o seu valor e passado para cada arquitetura i (indivíduo) da *listArq* (lista de arquiteturas).

A linha 10 chama o método *ranquear* para realizar o ranqueamento das arquiteturas geradas e as salva na lista *REP*, isto é, a população é classificada baseada no método o ranqueamento das soluções não-dominadas (definido por [Deb and Goel 2001]) e são armazenadas em um repositório. O laço principal da estratégia inicia na linha 11, tendo como condição de parada o número de gerações. Para cada geração, cada arquitetura terá sua velocidade *vel* computada segundo a linha 13, onde W é o peso de inércia, valor entre (0...1]; $R1$ e $R2$ são constantes reais escolhidas de forma aleatória entre (0...1]. A nova posição x da arquitetura no espaço de busca é computada na linha 14. Como forma de não passar do domínio de busca o método *controleFronteira* é chamado na linha 15. Este método atribui valores extremos, mínimo ou máximo, para cada função objetivo da arquitetura que tenha seus valores limites ultrapassados.

As linhas 16 a 18 computam os valores das funções objetivo das arquiteturas geradas após o passo dado pela linha 14. Em seguida, as linhas 19 a 21 têm 10% de chance de invocar o método *turbulencia*. Esse método realiza a modificação na arquitetura gerada, trocando um equipamento aleatório desta arquitetura. Após todas as arquiteturas se movimentarem e ter suas funções objetivos avaliadas, a linha 23 *REP* recebe estas novas arquiteturas e há um novo ranqueamento. Na linha 24, há o descarte das piores arquiteturas utilizando o operador *crowding distance* e a melhor posição de cada arquitetura P_{best} é atualizada. Repete-se o laço principal até que seja atingida a condição de parada.

7. Estudo de Caso

Esta seção apresenta um estudo de caso para ilustrar a aplicabilidade da estratégia proposta. A Tabela 1 apresenta os parâmetros de entrada (MTTF, MTTR, eficiência energética (EE) e custo de aquisição) adotados com base em [Guimarães and Pereira 2020]. Intervalos (min e máx.) foram definidos a partir dos valores bases dessa tabela para criação da base de equipamentos utilizados para a otimização. Sendo assim, para a eficiência energética, foi definida uma variação de 15%; para o MTTF foi adotada a variação de 50%; e considerando o custo dos equipamentos se variou 15%.

Tabela 1. Parâmetros de Entrada

Equip	EE (%)	MTTF (h)	MTTR (h)	CUSTO (R\$)
UPS	95,3	50000	8,0	15000
STS	95,5	58284	6,0	1000
SDT	98,5	282000	156	680
Subpanel	95,5	304000	2,4	250
Power Strip	97	274097	3,8	256

Ao todo foram gerados 10 equipamentos de cada tipo (mapeando a base de dados (.txt) onde cada linha tem os atributos de um determinado componente da infraestrutura elétrica de *data center*), que se combinados na arquitetura mais simples (A1) geram 10000 combinações possíveis. A estratégia multiobjetivo proposta e o algoritmo de força bruta foram aplicados às 6 arquiteturas. O objetivo é maximizar a disponibilidade, diminuir

a exergia operacional e o custo. O computador utilizado para rodar os modelos teve a seguinte configuração: Win10, Intel i5 1.80 GHz, 8GB RAM, 1TB HD.

O algoritmo adotado teve como entrada uma população inicial de 100 indivíduos, 1000 gerações e foi executado 10 vezes, por se tratar de uma solução estocástica. Os resultados obtidos aplicando a estratégia de otimização na arquitetura A6 é apresentado no gráfico da Figura 9. Nele é possível observar que há similaridade dos resultados encontrados pelo Alg. Proposto quando comparado ao conjunto solução ótimo gerado pelo Força Bruta.

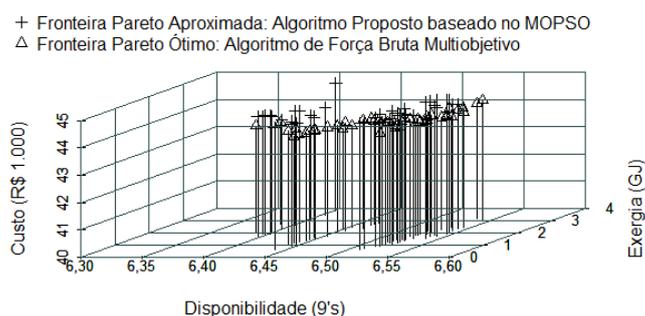


Figura 9. Gráfico Pareto Ótimo: Força Bruta x Algoritmo Proposto (arq. A6)

Nos gráficos da Figura 10 é possível comparar as funções otimizadas pelo força bruta e pela estratégia propostas sob três perspectivas. Na primeira (Figura 10 (a)) são apresentados os resultados da otimização na perspectiva Disponibilidade x Custo. Nele é possível observar a máxima diferença entre o conjunto solução e a curva de Pareto ótimo é de 1,68%. A segunda (Figura 10 (b)) traz a perspectiva da disponibilidade em função da exergia, onde a diferença máxima entre as curvas ficam em 3%. A Figura 10 (c) traz a terceira perspectiva (Exergia x Custo), onde a diferença máxima entre as curvas ficam em 1,64%.

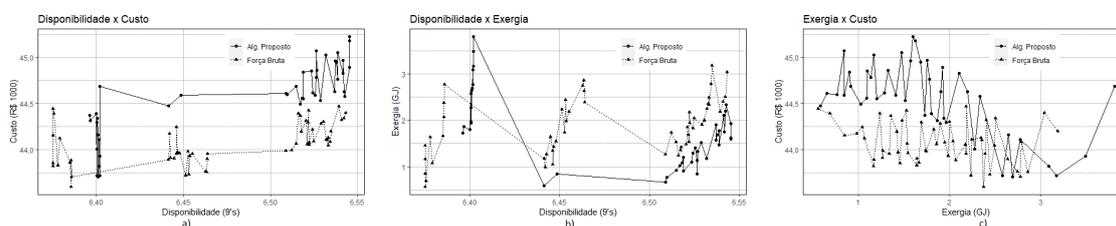


Figura 10. Gráficos Comparativos em Três Perspectivas (arq. A6)

É importante destacar que foram computados os tempos de execução demandados para cada abordagem. Por exemplo, assumindo a arquitetura A1, o tempo com o algoritmo de otimização proposto demandou em média aproximadamente 10s em comparação com os 5600s necessários utilizando a força bruta. Já para a arquitetura A6, o tempo demandado médio foi de 453s para o algoritmo de otimização proposto versus 394.110s no algoritmo de busca exaustiva. É considerável a diferença entre o tempo gasto pelos algoritmos, principalmente quando são arquiteturas mais complexas. Enquanto o algoritmo

baseado no MOPSO tem crescimento linear do tempo, o Força Bruta tem aumento de tempo de execução de forma exponencial. A Figura 11 apresenta o gráfico com a média e barra de erro por desvio padrão do conjunto das arquiteturas do tipo A6 otimizadas por função objetivo.

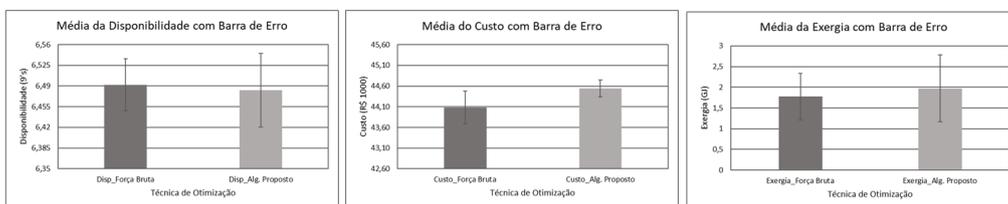


Figura 11. Gráficos Média com Barra de Erro por Funções Objetivo (arq. A6)

A Tabela 2 apresenta os resultados dos valores das médias de cada métrica analisada, os desvios padrões (Desv. Pad.) e o intervalo de confiança (IC) da última geração da arquitetura A6 do algoritmo proposto e também do força bruta (que traz o conjunto solução ótimo). Sendo assim, pode-se perceber que os intervalos de confiança baseado nas médias das funções objetivas do algoritmo proposto baseado no MOPSO sobreescrevem a média do algoritmo de Força Bruta, validando assim a estratégia proposta.

Tabela 2. Análise Estatística

	Força Bruta			Alg. Proposto		
	Média	Desv. Pad.	IC	Média	Desv. Pad.	IC
Disp. (9's)	6,5	0,058	[6,48; 6,51]	6,47	0,061	[6,45; 6,48]
Exergia (GJ)	1,77	0,611	[1,65; 1,81]	1,86	0,82	[1,69; 2,02]
Custo (R\$)	44059	223,69	[44015; 44102]	44546	395,2	[44058; 44623]

8. Conclusão

Este trabalho propôs uma estratégia de otimização multiobjetivo baseada no MOPSO aplicada a sistemas elétricos de *data centers*. Foi utilizada uma abordagem híbrida dos modelos RBD e SPN, a fim de proporcionar dados robustos para um melhor entendimento. O estudo de caso mostrou que o conjunto dos modelos das arquiteturas elétricas de *data center* otimizadas pelo algoritmo proposto trouxeram resultados que atingem a curvatura de Pareto ótimo com uma diferença de aproximadamente 3%.

Ressalta-se que a estratégia apresentada conseguiu otimizar as métricas em estudo (disponibilidade, exergia operacional e custo) em todos os modelos abordados, tendo na modelagem mais complexa (A6) um tempo de execução 870 vezes menor, quando comparada ao algoritmo de força bruta. Esse trabalho também apresentou uma análise estatística, que permite validar a estratégia proposta, onde após 1000 gerações a população de arquiteturas possuem indivíduos com melhores disponibilidade, custo e exergia operacional, quando comparados com sua população inicial. Como trabalho futuro sugere-se aplicar a estratégia proposta para outros subsistemas de *data center* (refrigeração e TI).

Agradecimentos

Os autores gostariam de agradecer ao CNPq, CAPES e à FACEPE pelo suporte financeiro para a realização desta pesquisa.

Referências

- Abualigah, L. and Diabat, A. (2021). A novel hybrid antlion optimization algorithm for multi-objective task scheduling problems in cloud computing environments. *Cluster Computing*, pages 205–223.
- Alvarez-Benitez, J., Everson, R., and Fieldsend, J. (2005). A mopso algorithm based exclusively on pareto dominance concepts. volume 3410, pages 459–473.
- Association, T. I. (2005). *Telecommunications Infrastructure Standard for Data Centers ANSI/TIA-942*. Telecommunication Industry Association.
- Callou, G., Ferreira, J., Maciel, P., Tutsch, D., and Souza, R. (2014). An integrated modeling approach to evaluate and optimize data center sustainability, dependability and cost. *Energies*, 7(1):238–277.
- Deb, K. and Goel, T. (2001). Controlled elitist non-dominated sorting genetic algorithms for better convergence. In *International conference on evolutionary multi-criterion optimization*, pages 67–81. Springer.
- Guimarães, A. P. and Pereira, A. (2020). Análise de aspectos de dependabilidade em sistemas de data centers integrando as infraestruturas de comunicação, de potência e de refrigeração. *Revista Brasileira de Administração Científica*.
- Isaak, P. (2021). Architecture: Data center rack floor plan and facility layout design. *Data Center Handbook: Plan, Design, Build, and Operations of a Smart Data Center*.
- Koot, M. and Wijnhoven, F. (2021). Usage impact on data center electricity needs: A system dynamic forecasting model. *Applied Energy*, 291:116798.
- Li, Y., Wen, Y., Tao, D., and Guan, K. (2020). Transforming cooling optimization for green data center via deep reinforcement learning. *IEEE Transactions on Cybernetics*.
- Marsan, M. A. (1990). *Stochastic Petri Nets: An Elementary Introduction*, page 1–29. Springer-Verlag, Berlin, Heidelberg.
- Masanet, E., Shehabi, A., Lei, N., Smith, S., and Koomey, J. (2020). Recalibrating global data center energy-use estimates. *Science*, 367(6481):984–986.
- Nguyen, T. A., Min, D., Choi, E., and Tran, T. D. (2019). Reliability and availability evaluation for cloud data center networks using hierarchical models. *IEEE Access*.
- Oliveira, D., Matos, R., Dantas, J., Ferreira, J. a., Silva, B., Callou, G., Maciel, P., and Brinkmann, A. (2017). Advanced stochastic petri net modeling with the mercury scripting language. New York, NY, USA. Association for Computing Machinery.
- Souza Leonardo, W. and Callou, G. (2021). Stars: um ambiente integrado para avaliação de disponibilidade, custo e consumo de energia de sistemas. *Revista Brasileira de Computação Aplicada*, 13(3):10–21.
- Sun, J., Deng, J., and Li, Y. (2020). Indicator & crowding distance-based evolutionary algorithm for combined heat and power economic emission dispatch. *Applied Soft Computing*, 90:106158.
- Wang, W., Loman, J. M., Arno, R. G., Vassiliou, P., Furlong, E. R., and Ogden, D. (2004). Reliability block diagram simulation techniques applied to the ieee std. 493 standard network. *IEEE Transactions on Industry Applications*, 40(3):887–895.