

UNIVERSIDADE FEDERAL DO PARÁ
GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

AUGUSTO FARIAS CRAVO
THAYS DHANDARAH RODRIGUES SÁ

**METAHEURÍSTICAS: O desenvolvimento de uma ferramenta *web* para acesso
dos algoritmos metaheurísticos *Particle Swarm Optimization* e
*Semi-Autonomous-PSO***

Belém - PA
2021

Augusto Farias Cravo
Thays Dhandarah Rodrigues Sá

METAHEURÍSTICAS:

O desenvolvimento de uma ferramenta *web* para acesso dos algoritmos metaheurísticos *Particle Swarm Optimization* e *Semi-Autonomous-PSO*.

Trabalho de conclusão de curso
submetido à banca da faculdade de
computação da Universidade Federal do
Pará como requisito para obtenção do
grau de Bacharel em Ciência da
Computação.

Orientador: Dr. Claudomiro de Souza Sales Júnior

METAHEURÍSTICAS: O desenvolvimento de uma ferramenta *web* para acesso dos algoritmos metaheurísticos *Particle Swarm Optimization* e *Semi-Autonomous-PSO*

Augusto Farias Cravo¹

Thays Dhandarah Rodrigues Sá²

Resumo: A aplicação de metaheurísticas em problemas complexos costumam gerar soluções inovadoras e variadas. Quando criadas, podem ser utilizadas para auxiliar tanto os veteranos, que já trabalham na área, quanto os pesquisadores iniciantes da área. Algoritmos como *Particle Swarm Optimization (PSO)*, *Genetic Algorithm (AG)*, *Ant Colony Algorithm (ACA)* e *Artificial Bee Colony Algorithm (ABC)* são exemplos de algoritmos de otimização existentes amplamente utilizados e com novas variações sendo desenvolvidas largamente. Seja por meio de adaptações das soluções já existentes ou pela criação de novos métodos, a literatura tem ganhado muitas contribuições sobre o assunto. Este trabalho aborda a construção de uma ferramenta *web* voltada para os algoritmos *PSO* e *SAPSO (Semi-Autonomous Particle Swarm Optimization)*, sendo o último uma otimização recentemente criada a partir do *PSO*. Devido a complexidade de seleção, implementação e aplicação dos métodos, além das diferentes técnicas de otimização que cada um aborda, este trabalho se propõe a desenvolver uma plataforma *web* simplificada que permita o uso dos algoritmos *PSO* e *SAPSO*, inicialmente. No *Metaheuristics Framework*, desenvolvido neste trabalho, estes algoritmos serão dispostos em uma interface, com permissões para que o usuário selecione o algoritmo, ajuste os parâmetros e escolha uma das funções de teste disponíveis na aplicação, gerando resultados através de gráficos e tabelas para visualização. Reduzindo assim, o esforço aplicado para utilização desses algoritmos para diferentes funções de teste ou problemas, visando a objetividade e praticidade de não implementar cada um deles e focando nas etapas de análise e aplicação dos resultados obtidos.

Palavras-chave: metaheurísticas, *pso*, *sapso*, internet, ferramenta.

¹ Graduando em Ciência da Computação pela UFPA. E-mail: tdrodrigues@gmail.com

² Graduando em Ciência da Computação pela UFPA. E-mail: augusto.f.cravo@gmail.com

Abstract

The application of metaheuristics to complex problems usually generates innovative and varied solutions. When added to a literature, they can be used to assist both veterans, who already work in the field, and beginning researchers in the field. Algorithms like: Particle Swarm Optimization (PSO), Genetic Algorithm (AG), Ant Colony Algorithm (ACA) and Artificial Bee Colony Algorithm (ABC) are examples of existing optimization algorithms widely used and with new variations being developed widely. Whether through adaptations of existing solutions or the creation of new methods, the literature has gained many contributions on the subject. This work addresses the construction of a web tool focused on the PSO and SAPSO (Semi-Autonomous Particle Swarm Optimization) algorithms, the latter being an optimization recently created from the PSO. Due to the complexity of selection, implementation and application of the methods, in addition to the different optimization techniques that each one addresses, this work proposes to develop a simplified web platform that allows access to the PSO and SAPSO algorithms, initially. These algorithms will be arranged in a web interface, with permissions for the user to select the algorithm, adjust the parameters and choose one of the test functions available in the application, generating results through graphs and tables for viewing. Thus reducing the effort applied in the stages of development and understanding about their execution, aiming at the objectivity and practicality of not implementing each one of them and focusing on the stages of analysis and application of the results obtained.

Keywords: metaheuristics.pso.sapso.web.tool

1 INTRODUÇÃO

Algoritmos metaheurísticos, comumente aplicados a problemas complexos, podem ser aplicados a uma diversidade de problemas existentes em diferentes áreas de pesquisa. A busca por soluções de otimização específicas para cada problema e que reduzam os custos computacionais exigidos, além da capacidade de adaptação de algoritmos metaheurísticos para problemas distintos, os tornam ferramentas úteis para pesquisadores, intensificando o estudo e as aplicações de metaheurísticas nos mais variados contextos. Segundo José Antonio Parejo (2014),

a aplicação de metaheurísticas podem ocorrer obedecendo o processo de execução estruturado em cinco etapas principais:

- Seleção, onde são escolhidas as metaheurísticas específicas para o problema;
- Adaptação, os algoritmos escolhidos são ajustados para o problema selecionado;
- Implementação: os algoritmos de otimização metaheurísticas são implementados;
- Ajuste: etapa em que os valores dos parâmetros são especificamente definidos;
- A Execução do algoritmo é a etapa final aplicada.

Os algoritmos metaheurísticos mais conhecidos na literatura são bioinspirados, utilizados amplamente para resolver problemas de otimização. Algoritmos como: *Particle Swarm Optimization (PSO)* (WANG et al., 2017), *Algoritmo Genético (AG)* (REEVES, 2010), *Ant Colony Algorithm (ACA)* (DORIGO et al., 2006) e *Artificial Bee Colony Algorithm (ABC)* (YANG; JIANG, 2015) são exemplos de algoritmos de otimização existentes na literatura e com novas variações sendo apresentadas com frequência.

Existem, ainda, algumas dificuldades para se trabalhar com algoritmos metaheurísticos no meio acadêmico. Os processos de seleção, adaptação, implementação e ajustes dos algoritmos se tornam tarefas exaustivas para pesquisadores e muito complexas para iniciantes na área de estudo sobre otimizações. Para reduzir o tempo e o esforço exigidos nesses processos, algumas ferramentas *web* disponibilizam aos usuários meios de acesso a esses algoritmos, permitindo o ajuste de parâmetros que o problema necessita e apresentando os resultados em formas de gráficos, tabelas e alguns casos apenas variáveis diretas.

Ainda que muito utilizados, os ajustes e adaptações feitas nos algoritmos para lidarem com problemas específicos acabam gerando novas variantes de algoritmos existentes na literatura e divulgar, aprender e implementar esses novos algoritmos também criam barreiras para os usuários. A dificuldade em executar e modificar um algoritmo de otimização está intrinsecamente relacionado ao tempo, o conhecimento e o esforço exigido do pesquisador para administrar tais tarefas. Uma ferramenta que trabalha com algoritmos de otimização e possui suporte *web*, permitindo que

possa ser acessada de qualquer local com conexão pode ser muito útil aos pesquisadores e aprendizes de tais conteúdos.

O algoritmo a ser tratado como foco deste trabalho é o algoritmo *Semi-Autonomous Particle Swarm Optimization (SAPSO)*, desenvolvido na tese de doutorado (SANTOS FILHO, 2019). *SAPSO* é um algoritmo baseado no *PSO* e apresenta diferenças ao usar em sua execução informações de gradiente, incorporando características determinísticas e matrizes de rotação, além do controle de diversidade (SANTOS FILHO, 2019).

Este trabalho apresenta uma proposta de ferramenta nomeada de *Metaheuristics Framework* para auxiliar os estudantes, pesquisadores e entusiastas da área de otimização a reduzir a complexidade de acesso e compreensão dos algoritmos *PSO* e *SAPSO*, minimizando o esforço de desenvolvimento e aprendizado sobre a execução dos mesmos, visando a possibilidade de se reaproveitar o mesmo algoritmo para várias situações diferentes. A ferramenta desenvolvida pelos autores disponibiliza inicialmente os algoritmos *PSO* e *SAPSO* para acesso, podendo incorporar futuramente outros algoritmos metaheurísticos. Considerando os diferentes métodos aplicados por cada algoritmo a interface *web* permite que o usuário selecione o algoritmo, ajuste os parâmetros e selecione funções de teste disponíveis na aplicação, gerando resultados em gráficos e tabelas para visualização.

2 TRABALHOS RELACIONADOS

Desde que observado a falta de uma ferramenta com suporte *web* que disponibilize os algoritmos *SAPSO* e *PSO* para pesquisa e aprendizado, houve a busca por ferramentas que atendessem a essa demanda mesmo que utilizando outros algoritmos meta-heurísticos. Durante o processo de pesquisa foram encontrados os exemplos abaixo. O crescente número de pesquisas voltadas para aplicação de algoritmos de otimização na resolução de problemas de fontes diversificadas, como na bioinformática onde o *PSO* pode ser usado no problema de predição de proteínas (CAVALCANTI, 2018) ou na engenharia florestal onde pode ser usado no treinamento de redes neurais para estimativa de altura em povoamento

de eucalipto (MATIAS et al., 2018), fizeram surgir também algumas versões geradas a partir de otimizações de um algoritmo para trabalhar de acordo com o problema a ser resolvido. No entanto, essas variações existentes não são encontradas facilmente na literatura.

No decorrer deste trabalho, foram encontradas quatro ferramentas disponíveis que oferecem suporte para utilização de algoritmo de otimização dentro de projetos independentes e sem a complexidade de aplicação.

Epanel é uma ferramenta de aplicação que demonstra o processo de execução dos algoritmos metaheurísticos de Otimização de Enxame de Partículas e Colônia de abelhas Artificial, utilizando uma interface gráfica que exibe o valor de uma execução, a partir de uma equação inserida pelo usuário calculada em uma ou duas dimensões, usando espectro de cores. O Epanel também auxilia no entendimento do usuário sobre a distribuição do espaço da solução e mostra a posição de cada partícula em cada iteração, exibindo a movimentação de cada partícula até o melhor resultado final. Além disso, é possível visualizar a execução dos dois algoritmos simultaneamente, dando a possibilidade de comparar e avaliar a execução dos dois (YANG et al., 2018). Epanel foi produzida na linguagem JAVA e que pode ser embutida dentro de outras aplicações, desde que o *site* da ferramenta EPANEL seja acessado para instalação. Porém, o *site* não foi encontrado para acesso durante o tempo de execução desta pesquisa. Outro fator que pode se tornar uma desvantagem para os usuários desta aplicação é que os dados a serem inseridos para a execução dos algoritmos são feitos via documento formulado pelo próprio usuário com a extensão *.Epin*, de acordo com o modelo apresentado no *site*.

Web based educational Tool é um projeto educacional dirigido por Güncel SARIMAN e Ecir Uğur KÜÇÜKSİLLE em que foi desenvolvido um software *web* apresentando os algoritmos *Artificial Immune System* e *Artificial Bee Colony Algorithm* para serem observados por meio de uma interface *web* com o objetivo de facilitar o ensino e aprendizado de algoritmos metaheurísticos dando ao usuário a permissão de modificar os parâmetros do algoritmo escolhido e com a possibilidade de escolher entre as funções de teste disponíveis na plataforma ou inserir suas próprias funções. O projeto não especifica como o usuário pode trabalhar com os dados de forma *off-line*, se são disponibilizados via arquivo ou como uma nova aba do aplicativo (SARIMAN et al., 2018).

Solver é um suplemento do *Microsoft Excel*, disponibilizado com a nova versão desde 2010. Trata-se de um módulo para programação matemática integrado à planilha eletrônica que dispõe de três métodos para resolver problemas de otimização descritos por meio de equações escritas nas células da planilha. *Solver* é um *software* que tem o objetivo de aplicar fórmulas para calcular o resultado máximo ou mínimo de uma célula utilizando os métodos *Simplex LP*, usado nos problemas de programação linear; *GRG nonlinear*, usado nos problemas não lineares e o método *Evolutionary Solver*, usado se a célula objetivo e as células de restrições são determinados a partir de funções cujos os resultados possuem alta taxa de variação. O método *Evolutionary* utiliza o método evolucionário baseado na ideia desenvolvida por Darwin a respeito do processo de evolução e seleção natural das espécies (FrontlineSolvers, 2020)(RODRIGUES, 2020). Para utilização do *Solver* é necessário algum conhecimento sobre *Excel*, visto que ele precisa ser ativado dentro do programa para só então ser aplicado.

O *framework MOEA (Multiobjective Evolutionary Algorithms)* é uma biblioteca JAVA, gratuita e *open source*, para desenvolver e experimentar algoritmos evolutivos multiobjetivos, incluindo algoritmos genéticos, evolução gramatical, evolução diferencial, otimização de enxame de partículas e outros algoritmos de otimização multiobjetivo de propósito geral. *MOEA* fornece as ferramentas necessárias para projetar, desenvolver, executar e testar estatisticamente algoritmos de otimização (MOEA, 2019).

2.2 JUSTIFICATIVA

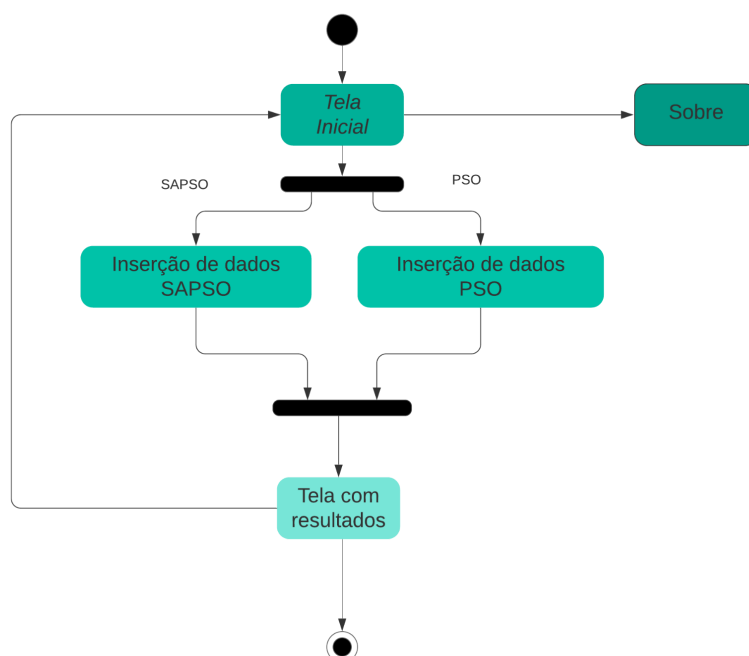
As limitações das ferramentas encontradas durante o processo de pesquisa deste projeto e a ausência de uma ferramenta que forneça vantagens para pesquisadores do algoritmo *PSO* e mais precisamente do algoritmo *SAPSO*, foram fatores importantes para a criação dessa ferramenta. Enquanto em *EPANEL* (YANG et al., 2018) se trabalha com inserção de dados via arquivo *.epin*, o *Metaheuristics Framework* trabalha com inserção de dados via interface e arquivo *.csv*, que formas mais fáceis e conhecidas para configuração de algoritmos e coleta de resultados. Outra função disponível neste projeto é a possibilidade de *download* dos resultados

gerados pela execução, através de um arquivo .csv que apresentam os dados em arquivos separados para melhor compreensão do usuário.

3 MATERIAIS E MÉTODOS

O método selecionado para aplicação deste projeto foi a construção de uma página *web* que permite o uso das ferramentas *PSO* e *SAPSO* com permissões de definição de valores das variáveis adaptáveis ao problema a ser resolvido. As etapas de desenvolvimento incluem a manipulação de um servidor para hospedar, processar e retornar os dados inseridos pelo usuário nas formas de arquivo para *download* e gráficos mostrados diretamente na interface. O ambiente foi desenvolvido utilizando um conjunto de *frameworks* para maximizar a construção do sistema e a qualidade de interação do usuário. Na figura 1 pode-se observar o Diagrama de Atividades da ferramenta.

Figura 1. Diagrama de atividades da ferramenta Metaheuristics.



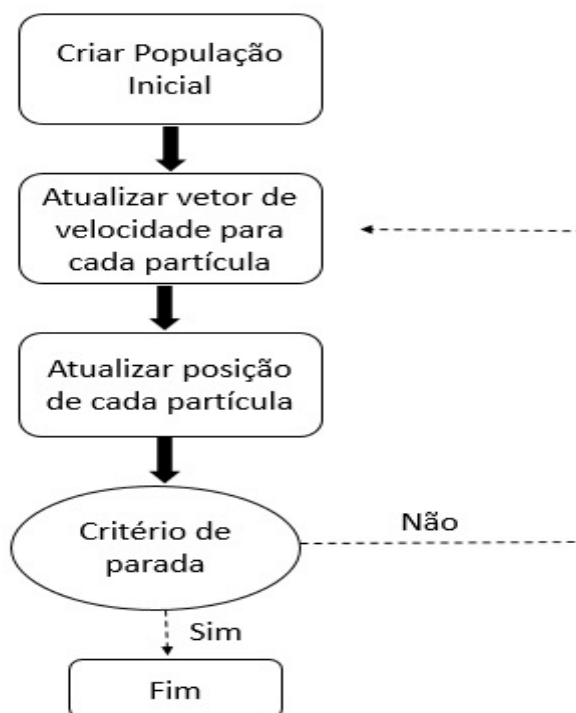
As ferramentas principais utilizadas foram os *frameworks* Meteor (versão 1.9) que fornece um conjunto de bibliotecas e outras funções que facilitam a construção dos processos *back-end*, utilizando a linguagem JavaScript e o *framework* Vue (2.6.11) que apresenta muitas funcionalidades que auxiliam a construção do *front-end*, utilizando as linguagens HTML, CSS e JavaScript. As linguagens de programação envolvidas neste projeto são: MatLab, JavaScript, HTML e CSS. Antes de ser inserido na ferramenta, o algoritmo *SAPSO* foi traduzido da sua linguagem original, o MatLab, para a linguagem JavaScript. Para disponibilizar a ferramenta e aplicar controle de versionamento do projeto, foi utilizada a plataforma GitHub.

Ainda para auxiliar na compreensão dos conceitos teóricos envolvidos no algoritmo *SAPSO* e também o *PSO*, foi anexada um *link* de informações sobre o conteúdo geral do projeto, assim como, uma breve definição de cada variável manipulada pelo usuário, com exemplos de valores obtidos do trabalho original (SANTOS et al., 2018).

3.1 Particle Swarm Optimization - PSO

De acordo com (EBERHART; KENNEDY, 1995), na otimização por enxame de partículas o sistema é inicializado com uma população de soluções aleatórias e a cada partícula é atribuída uma velocidade aleatória. Cada partícula armazena também suas coordenadas que estão associadas a melhor posição que a partícula esteve até agora (*pbest*). Outra informação conhecida são as coordenadas da partícula de melhor posição global dentro da população (*gbest*). Ainda segundo os autores, o conceito de otimização por enxame de partículas consiste em mudar a velocidade (aceleração) de cada partícula em direção à sua melhor posição local e à sua melhor posição global no espaço de busca, como demonstra o diagrama da Figura 2.

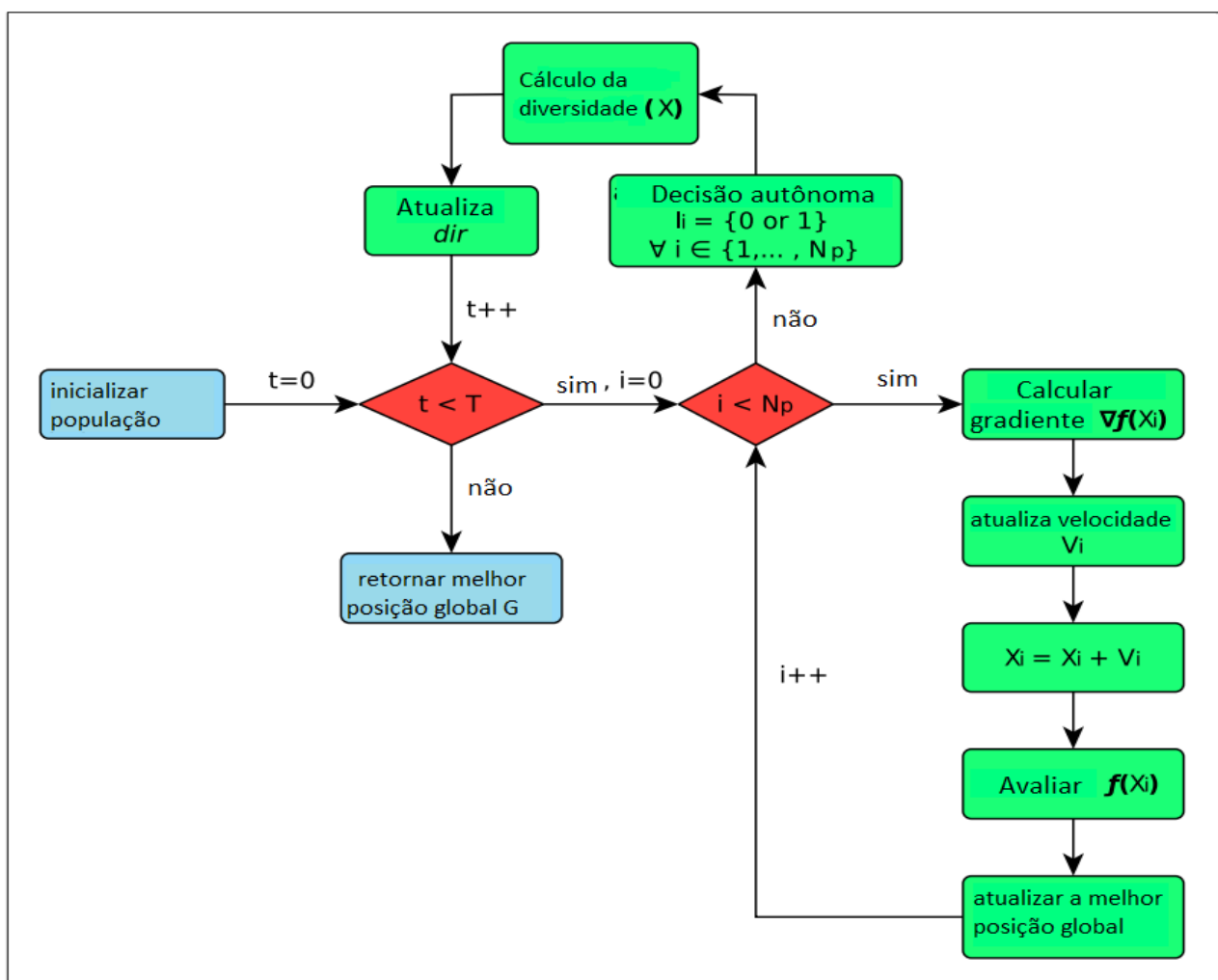
Figura 2. Diagrama do algoritmo PSO.



3.2 Sem Autonomous Particle Swarm Optimization - SAPSO

Em (SANTOS, 2019) o autor propõe o uso alternado de informação de gradiente e pesquisa estocástica baseado na melhor posição global atual encontrada até agora pela população para orientar a pesquisa em direção ao mínimo global. Ambas as direções são usadas por cada partícula do enxame. Enquanto algumas partículas estarão voando em direção às coordenadas de melhor posição global, outras estarão investigando sua vizinhança. Ainda que as partículas possam decidir individualmente em momentos diferentes durante o processo de busca, elas atuam sob uma regra geral de atração e repulsão, controlada pela diversidade do enxame (SANTOS, 2019), como demonstrado pela figura 3. estão

Figura 3. Diagrama do algoritmo SAPSO.



Fonte: SANTOS FILHO; R. C. (2019. p. 66)

4 RESULTADOS

Como resultado final, obteve-se uma plataforma interativa onde podem ser acessados os algoritmos *PSO* e *SAPSO*, por meio de uma página dedicada a cada um deles. A manipulação de algumas variáveis, selecionadas por serem altamente relevantes dentro de cada algoritmo, são feitas por meio de caixas de inserção de

dados ou *upload* de um arquivo no formato *csv* de acordo com o modelo disponibilizado pela interface, visando maior facilidade para testes que exigem repetidas inserções simultâneas. Dessa forma, permitem desde a alteração de valores de parâmetros que se adequem ao problema que o usuário deseja solucionar, quanto a seleção de funções de teste a serem calculadas, como mostrado nas figuras abaixo.

Figura 4. Tela inicial para escolha do algoritmo *SAPSO* ou *PSO*.

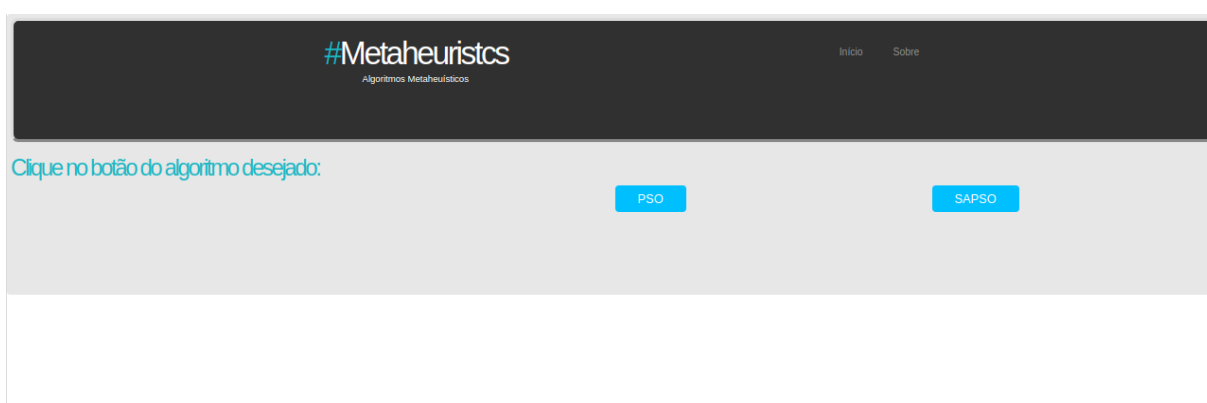


Figura 5. Tela de inserção de dados para a execução do algoritmo *SAPSO*.

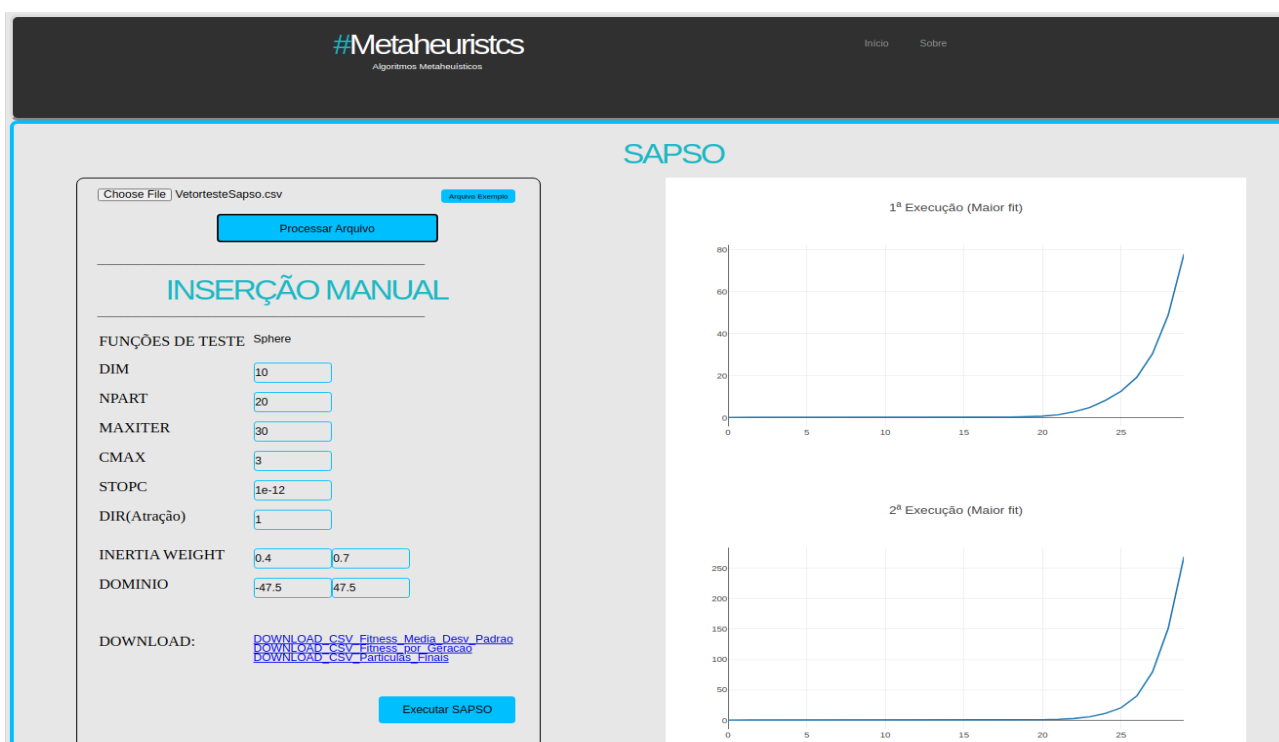


Figura 6. Tela de inserção de dados para a execução do algoritmo *PSO*.

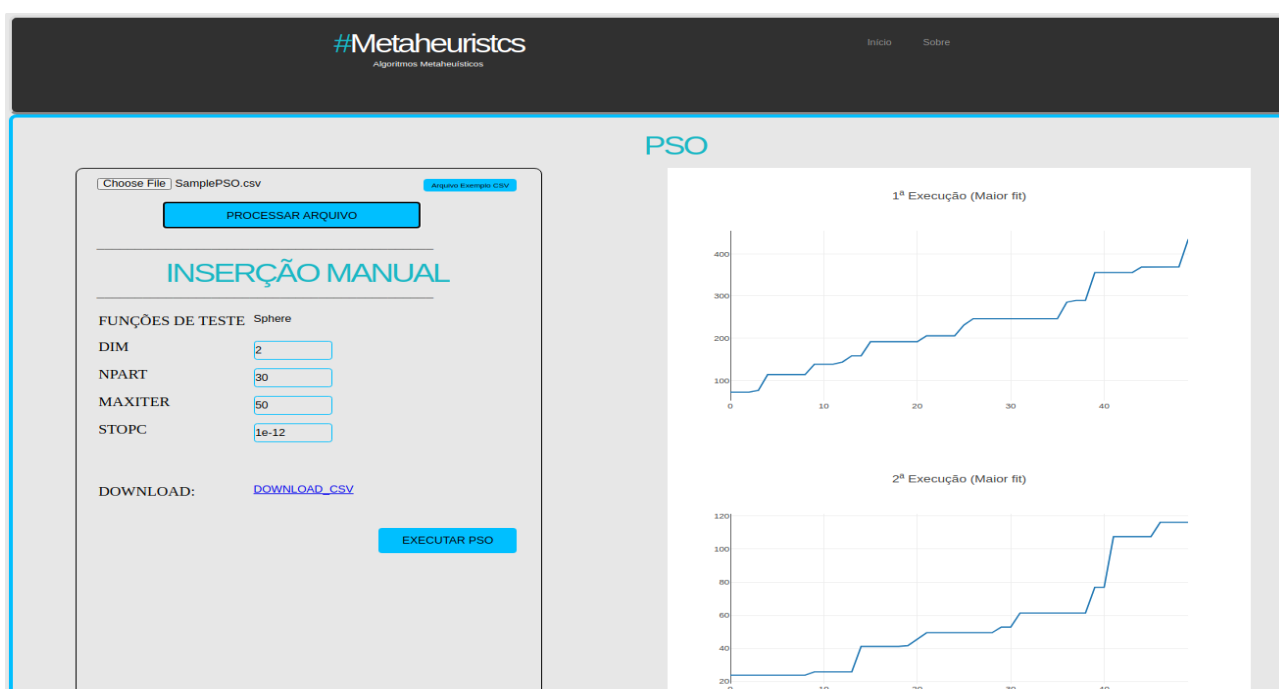


Figura 7. Tela de inserção de dados para a execução do algoritmo *SAPSO* com a lista de funções que podem ser escolhidas.

The screenshot shows the SAPSO web application interface. At the top, there is a dark header with the logo "#Metaheuristics" and the text "Algoritmos Metaheurísticos". To the right of the logo are links for "Inicio" and "Sobre". Below the header, the main content area is titled "SAPSO" in large blue letters. On the left side, there is a form titled "INSERÇÃO MANUAL". The form includes a file upload section with a "Choose File" button and a "Processar Arquivo" button. Below this, there is a section for "FUNÇÕES DE TESTE" with a list of functions: DIM, NPART, MAXITER, CMAX, STOPC, DIR(Atração), and INERTIA WEIGHT. A dropdown menu is open, showing a list of functions: Ackley, Alpine, Griewank, Rastrigin, Rosenbrock, Sphere, Debi, Himmelblau, Impossible, Levi, Levi13, Schaffer2, Schaffer4, Schwefel, and Testando. Below the list, there are input fields for "WMIN" and "WMAX". At the bottom of the form, there is a "DOMINIO" section with input fields for "Ex. [-47.5]" and "Ex. [47.5]". A "DOWNLOAD:" label is also present. At the bottom right of the form, there is a blue button labeled "Executar SAPSO".

Figura 8. Gráfico da 1ª execução do algoritmo SAPSO.

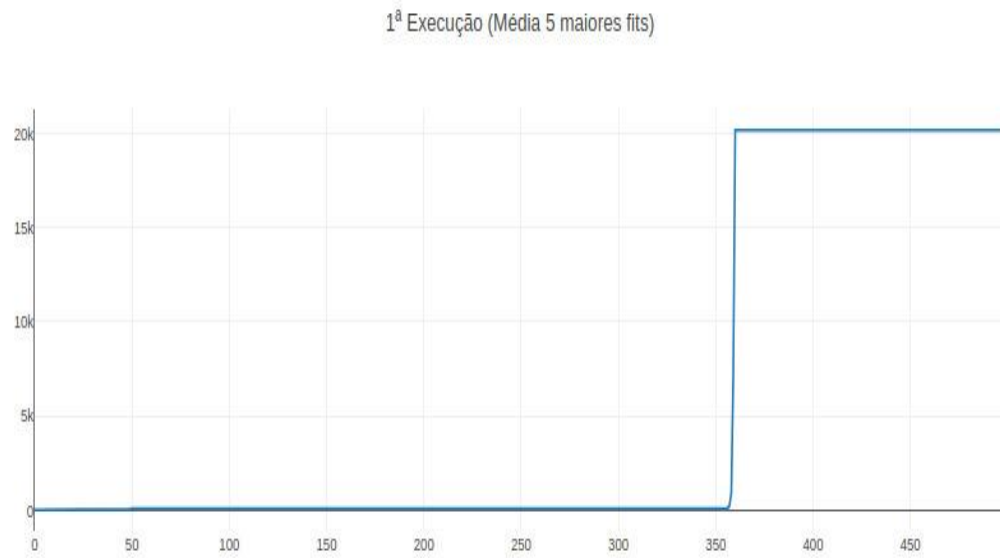


Figura 9. Gráfico da 2ª execução do algoritmo SAPSO.

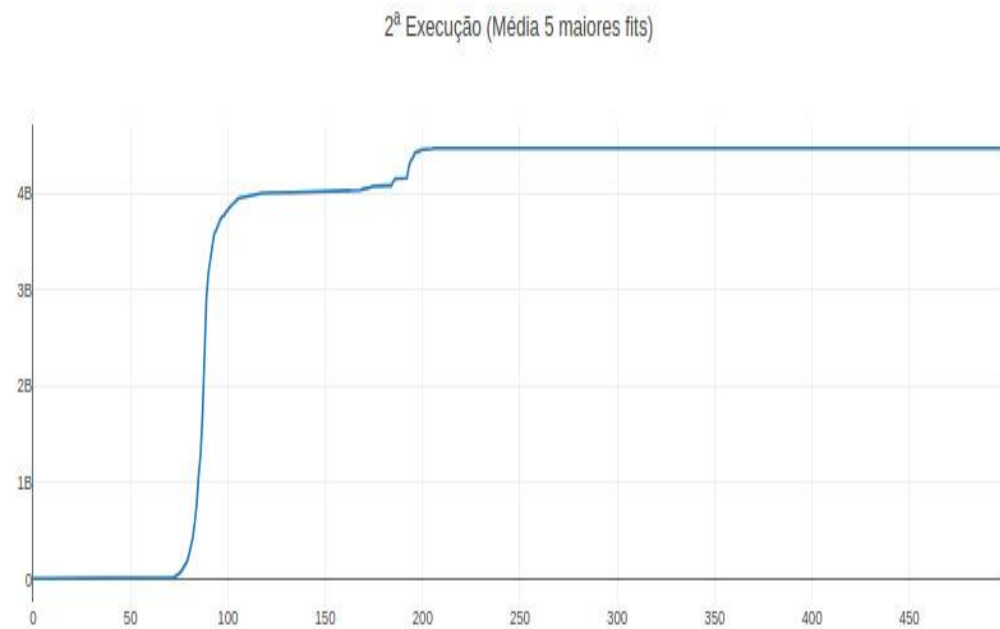


Figura 10. Gráfico da 3ª execução do algoritmo SAPSO.

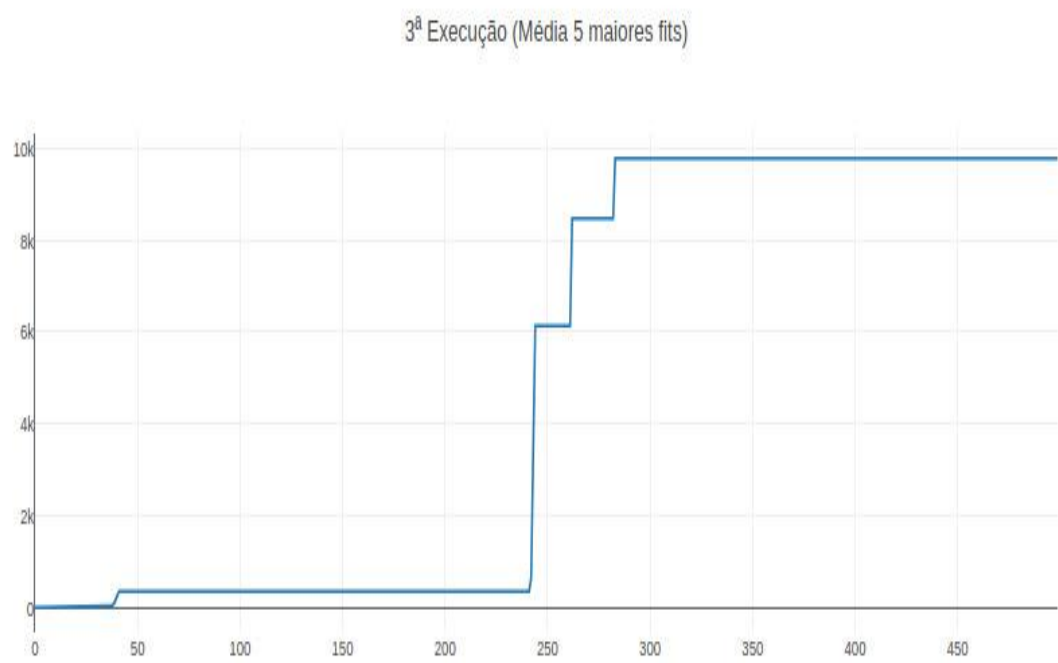


Figura 11. Gráfico da 4ª execução do algoritmo SAPSO.

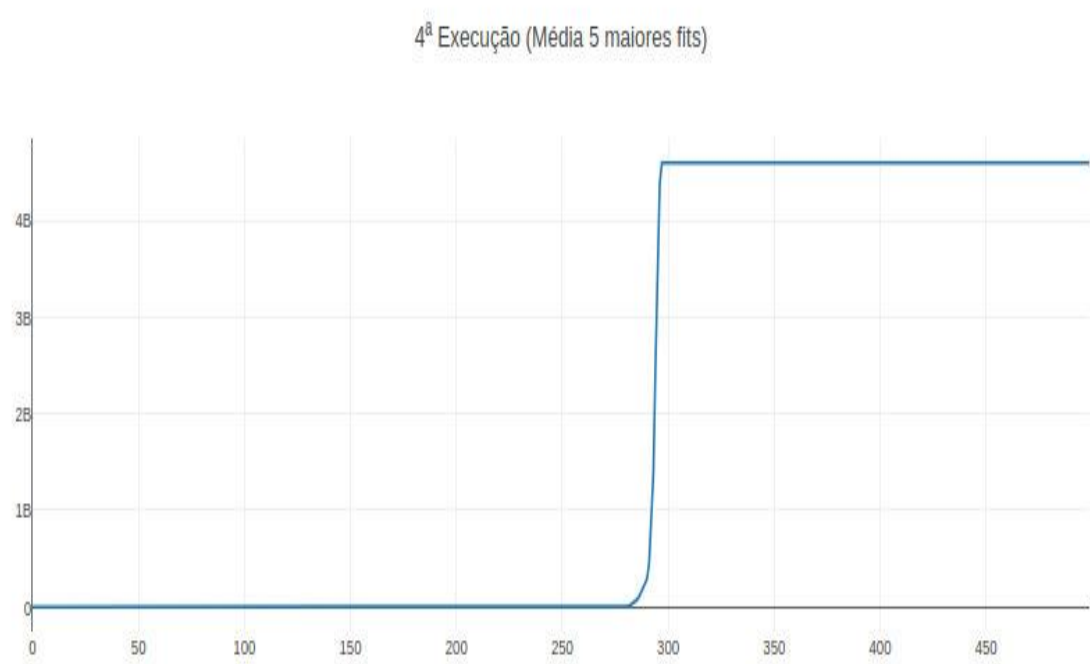


Figura 12. Gráfico da 5ª execução do algoritmo SAPSO.

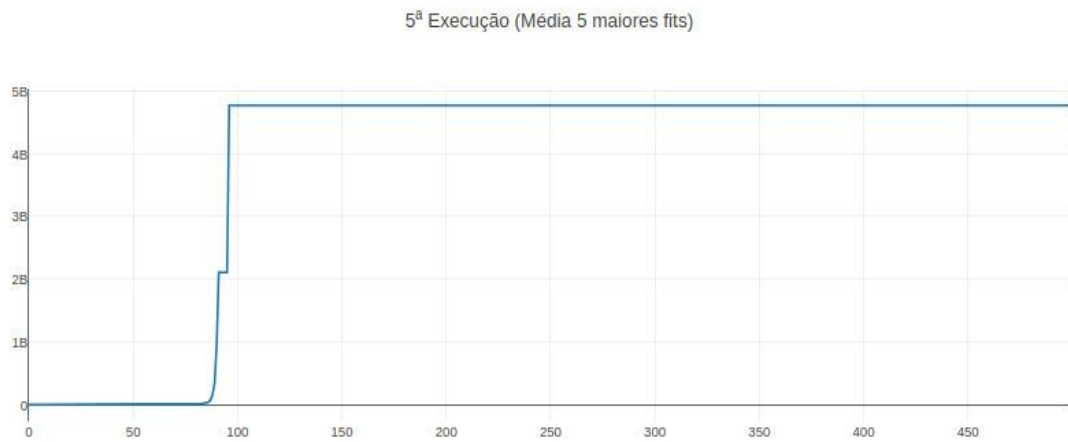
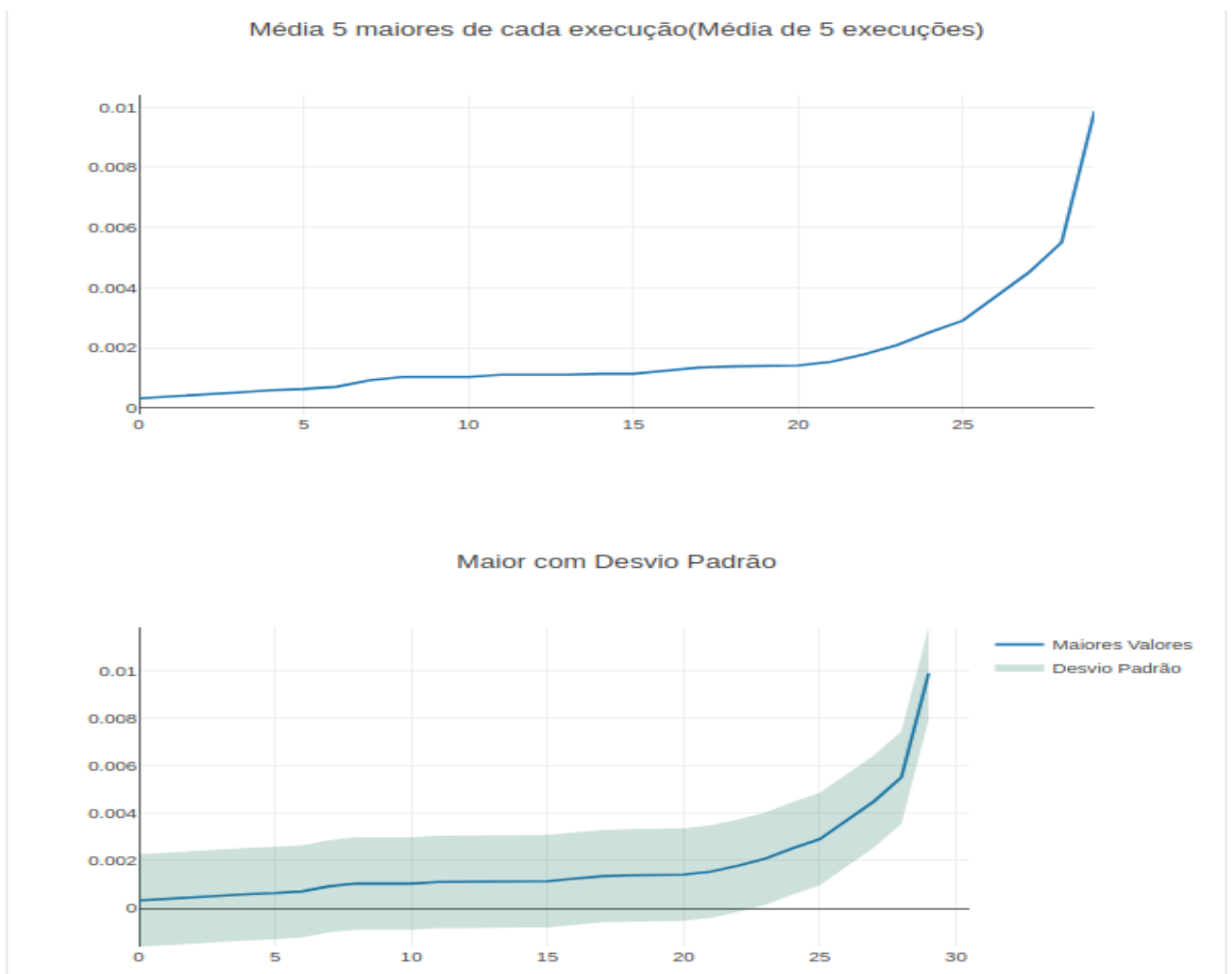


Figura 13. Média dos 5 gráficos anteriores e o mesmo gráfico com seu desvio padrão

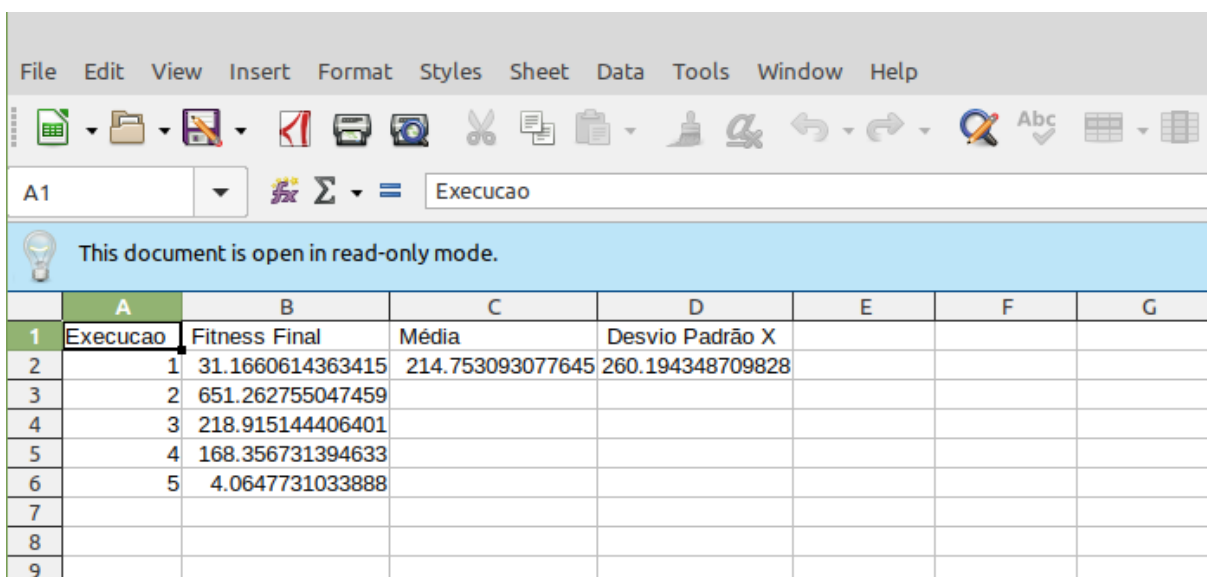


A apresentação dos resultados para o usuário consiste na representação dos gráficos das 5 execuções, contendo os valores dos *fitness* e o cálculo do desvio padrão encontrados durante as execuções. Assim como demonstrado nas Figuras 8 a 13.

- A função objetivo utilizada para os gráficos das figuras 8 a 13 (para todos os gráficos) foi:
 - Coordenada X : Número da iteração do algoritmo.
 - Coordenada Y: Média dos 5 maiores fitness encontrados na iteração atual.
- A função objetivo utilizada para os gráficos da figura 13 foi:
 - Para primeiro gráfico:
 - Coordenada X : Número da iteração do algoritmo.
 - Coordenada Y : Média das coordenadas y dos gráficos das figuras 8 a 13.
 - Para o segundo gráfico:
 - Coordenada x : Número da iteração do algoritmo.
 - Coordenada y: Média das coordenadas y dos gráficos das figuras 8 a 13.
 - Área azul sombreada: Representa o desvio padrão para cada ponto considerando-se a variação para cima e para baixo.

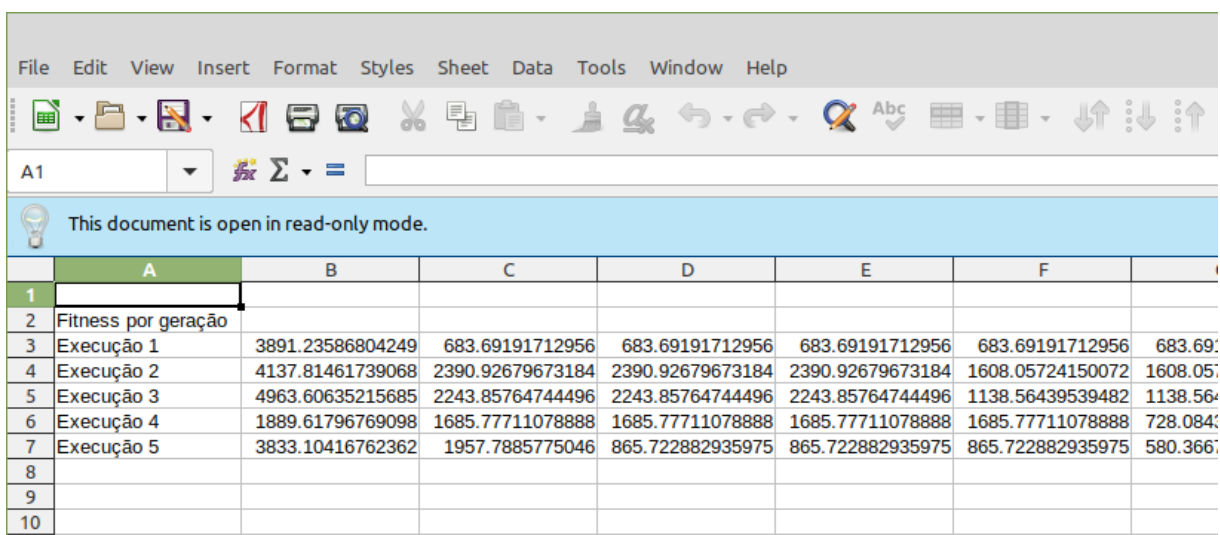
Os dados também são disponibilizados para *download* via arquivos, também no formato *csv*, para que o usuário possa reproduzir estes gráficos em outros ambientes, se assim desejar. As figuras 14, 15, e 16 são fragmentos dos modelos de arquivo *csv* gerados para *download*.

Figura 14. Arquivo de download com Fitness, Média e Desvio Padrão.



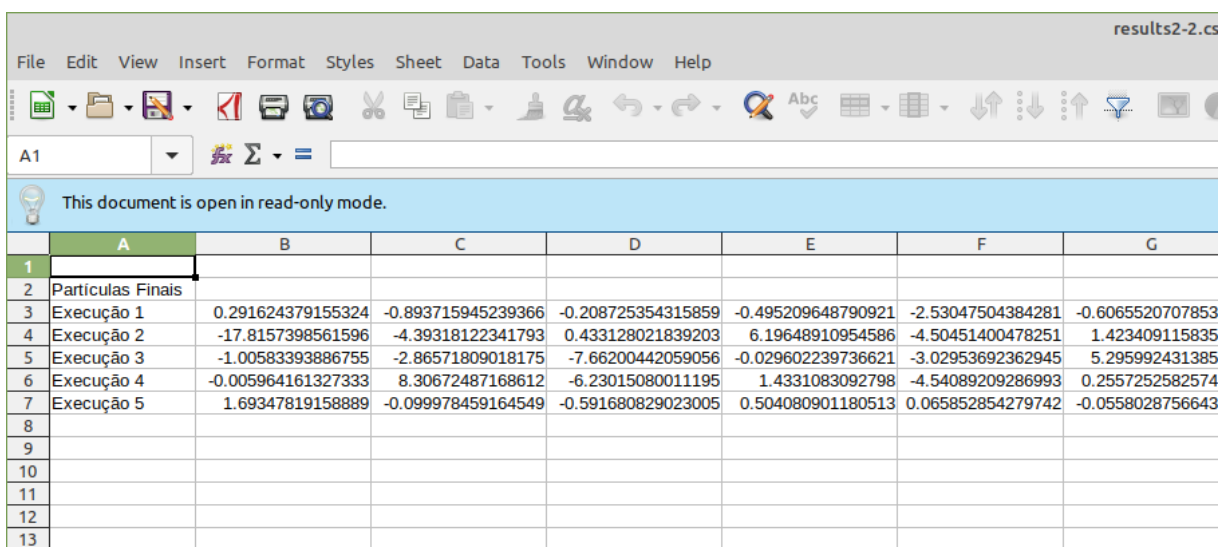
	A	B	C	D	E	F	G
1	Execucao	Fitness Final	Média	Desvio Padrão X			
2	1	31.1660614363415	214.753093077645	260.194348709828			
3	2	651.262755047459					
4	3	218.915144406401					
5	4	168.356731394633					
6	5	4.0647731033888					
7							
8							
9							

Figura 15. Arquivo de download com Fitness por Geração



	A	B	C	D	E	F	G
1							
2	Fitness por geração						
3	Execução 1	3891.23586804249	683.69191712956	683.69191712956	683.69191712956	683.69191712956	683.69191712956
4	Execução 2	4137.81461739068	2390.92679673184	2390.92679673184	2390.92679673184	1608.05724150072	1608.05724150072
5	Execução 3	4963.60635215685	2243.85764744496	2243.85764744496	2243.85764744496	1138.56439539482	1138.56439539482
6	Execução 4	1889.61796769098	1685.77711078888	1685.77711078888	1685.77711078888	1685.77711078888	728.08411078888
7	Execução 5	3833.10416762362	1957.7885775046	865.722882935975	865.722882935975	865.722882935975	580.36611078888
8							
9							
10							

Figura 16. Arquivo de download com as Partículas Finais.



	A	B	C	D	E	F	G
1							
2	Partículas Finais						
3	Execução 1	0.291624379155324	-0.893715945239366	-0.208725354315859	-0.495209648790921	-2.53047504384281	-0.6065520707853
4	Execução 2	-17.8157398561596	-4.39318122341793	0.433128021839203	6.19648910954586	-4.50451400478251	1.423409115835
5	Execução 3	-1.00583393886755	-2.86571809018175	-7.66200442059056	-0.029602239736621	-3.02953692362945	5.295992431385
6	Execução 4	-0.005964161327333	8.30672487168612	-6.23015080011195	1.4331083092798	-4.54089209286993	0.2557252582574
7	Execução 5	1.69347819158889	-0.099978459164549	-0.591680829023005	0.504080901180513	0.065852854279742	-0.0558028756643
8							
9							
10							
11							
12							
13							

Na Figura 14, a coluna “Execução” se refere a cada execução realizada pelo algoritmo. “*Fitness Final*” representa o último *fitness* encontrado pelo algoritmo. “Média” é a média de todos os *fitness* finais apresentados no campo “*Fitness Final*”. “Desvio Padrão X” é o desvio padrão de todos os *fitness* finais apresentados no campo “*Fitness Final*”.

O campo, na Figura 15, “*Fitness por geração*” apresenta o melhor *fitness* encontrado por execução em cada iteração realizada. Cada linha representa uma execução diferente. “Partículas Finais”, na Figura 16, apresenta as coordenadas das partículas finais em cada uma das cinco execuções realizadas. Cada linha corresponde a uma execução diferente.

A interface conta também com um *link* ‘*about*’ que contém explicações sobre cada algoritmo e variáveis e algumas *tips* na *link* dedicada do próprio algoritmo com uma explicação breve e objetiva sobre cada variável, bastando apenas que o cursor esteja sobre elas. Os campos de inserção de dados também mostram exemplos de possíveis valores a serem inseridos pelo usuário.

5 TRABALHOS FUTUROS

Nesta seção apresentamos algumas propostas de trabalhos que foram identificados como potenciais contribuições futuras para este projeto:

- Otimização do algoritmo *SAPSO* dentro do servidor;
- Implementação de mais recursos de Interação Humano Computador(IHC);
- Adição de novas metaheurísticas;
- Implementação do método de adição de uma função customizada do usuário na ferramenta web;
- Implementação de campo para a escolha do usuário de quantas vezes o algoritmo deve ser executado;
- Implementação de forma de se obter a semente da execução realizada nos algoritmos;
- Implementação de forma de se definir a semente da execução realizada nos algoritmos.

6 CONCLUSÃO

A proposta inicial deste projeto foi desenvolver uma aplicação online que desse ao usuário a possibilidade de utilizar algoritmos metaheurísticos de maneira que permita a adaptação dos elementos de acordo com a necessidade do usuário. Um dos objetivos foi manter a interface em um formato simples que abranja os diferentes tipos de conhecimentos que os usuários tenham sobre metaheurísticas ao utilizarem a ferramenta.

Considerando-se as cinco etapas consideradas por José Antonio Parejo (2014) (Seleção, Adaptação, Implementação, Ajuste e Execução), foi alcançado êxito em todas elas. No critério de Seleção, foi adicionado a duas formas de metaheurísticas que podem ser escolhidas pelo usuário, PSO e SAPSO. Quanto a

Adaptação, é disponibilizado a possibilidade de se escolher diferentes funções de área inicial dos algoritmos antes da execução, possibilitando parcial ajuste para o problema selecionado. Na questão de Ajuste, foram determinadas formas de se alterar diversos parâmetros dos algoritmos para execução. Por fim, a construção do algoritmo e a possibilidade de poder ser executado cobrem as etapas de Implementação e Execução.

O produto final foi obtido satisfatoriamente de acordo com os objetivos descritos na introdução, e a ferramenta desenvolvida pode ser utilizada por usuários iniciantes e/ou experientes dentro de contextos e ambientes acadêmicos diferentes.

REFERÊNCIAS

CAVALCANTI, D. M. Algoritmos de Otimização de Inteligência de Enxame Aplicados ao Problema de Predição de Proteína, 2018. In.: <<https://repositorio.ufu.br/bitstream/123456789/24120/3/AlgoritmosOtimizacaoInteligencia.pdf>>. Acesso em: dia 10 de dezembro de 2020.

DAO, S. D.; ABHARY, K.; MARIAN, R. *“An innovative framework for designing genetic algorithm structures”*, 2017. In.: <<https://www.sciencedirect.com/science/article/abs/pii/S0957417417305572>>. Acesso em: dia 14 de junho de 2020.

DORIGO, M.; BIRATTARI, M.; STUTZLE, T. Ant Colony Optimization. IEEE Computational Intelligence Magazine. 2006. In.: [https://www.researchgate.net/publication/308953674_Ant_Colony_Optimization]. Acesso em: 10 de junho de 2021.

EBERHART, R.; KENNEDY, J. A New Optimizer Using Particle Swarm Theory. Sixth International Symposium on Micro Machine and Human Science. 1995. In.: [http://www.ppgia.pucpr.br/~alceu/mestrado/aula3/PSO_2.pdf]. Acesso em: 15 de junho de 2021.

A Framework for Innovation. MOEA Framework: A Free and Open Source Java Framework for Multiobjective Optimization, 2019. In.: <<http://moeaframework.org>>. Acesso em: dia 10 de abril de 2020.

Genetic Algorithms and Evolutionary Algorithms- Introduction. FrontlineSolvers, 2020. In.:

[https://www.solver.com/genetic-evolutionary-introduction?utm_source=Google&utm_medium=PPC&utm_campaign=Tier1ConventionalOptimization&utm_term=genetic%20algorithms%20c%23&gclid=CjwKCAjwxqX4BRBhEiwAYtJX7e29gDMbimfCi0Qk1p-AZoT9Rm7YQDDBaCYr1PcNrwl6mllkoHVmbBoC_IIQAvD_BwE]. Acesso em: dia 07 de fevereiro de 2021.

MATIAS, H. de B.; BINOTI, D. H. B.; SILVA, G. F.; LEITE, H. G.. Algoritmo *particle swarm optimization* (PSO) no treinamento de redes neurais para estimativa de altura em povoamento de eucalipto, 2018. In.: <<https://periodicos.ufpel.edu.br/ojs2/index.php/cienciadamadeira/article/view/8852>>. Acesso em: dia 15 de novembro de 2020.

PAREJO, J. A. *MOSES: A Metaheuristic Optimization Software EcoSystem. AI Communications*. 2014. In.: <https://www.researchgate.net/publication/291246714_MOSES_A_Metaheuristic_Optimization_Software_EcoSystem> . Acesso em: dia 06 de janeiro de 2020.

REEVES, C. R. Genetic Algorithms. 2010. In.: [https://www.researchgate.net/publication/226462334_Genetic_Algorithms]. Acesso em: 11 de junho de 2021.

RODRIGUES, L. Solver: como usar essa função do Excel?. Voitto, 2020. In.: <<https://www.voitto.com.br/blog/artigo/solver>>. Acesso em: 08 de março de 2021.

SANTOS FILHO, R. C. Global Continuous Optimization with Particle Swarm Enhancements. Tese(Doutorado em Filosofia em Ciência da Computação) - Faculdade de Computação da Universidade Federal do Pará. Pará, p. 60. 2019.

SARIMAN, G. I. ; KUCUKSILLE, E. U. *Web Based Educational Tool for Metaheuristic Algorithms. Pamukkale University Journal of Engineering Sciences*. 2014. In.: [https://www.academia.edu/31041970/Web_Based_Educational_Tool_for_Metaheuristic_Algorithms]. Acesso em: dia 24 de março de 2020.

WANG, D., TAN, D. & LIU, L. Particle swarm optimization algorithm: an overview. *Soft Comput*. 2018. In.: [<https://doi.org/10.1007/s00500-016-2474-6>]. Acesso em: 12 de junho de 2021.

YANG, Chia-Yun; YANG, Li-Sheng; KUO, Shu-Yu; CHOU, Yao-Hsin. "EPanel 2.0: The Visualization Tool for Combinatorial Optimization and Deployment Problem," 2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC), Miyazaki, Japan, 2018, pp. In.: [<https://ieeexplore.ieee.org/document/8616266>]. Acesso em: dia 20 maio de 2020.

YANG, S.; JIANG, H. An Overview of Artificial Bee Colony Algorithm. International Conference on Advances in Mechanical Engineering and Industrial Informatics. 2015. In.: [<https://www.atlantispress.com/proceedings/ameii-15/21898>]. Acesso em: 13 de junho de 2021.