

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

USO DE ALGORITMO GENÉTICO COM OPERADORES MODIFICADOS
PARA OTIMIZAÇÃO DE FUNÇÕES DE VARIÁVEIS REAIS

EDSON KOITI KUDO YASOJIMA

TD: 07/2019

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2019

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

EDSON KOITI KUDO YASOJIMA

USO DE ALGORITMO GENÉTICO COM OPERADORES MODIFICADOS
PARA OTIMIZAÇÃO DE FUNÇÕES DE VARIÁVEIS REAIS

TD: 07/2019

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2019

UNIVERSIDADE FEDERAL DO PARÁ
INSTITUTO DE TECNOLOGIA
PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA

EDSON KOITI KUDO YASOJIMA

USO DE ALGORITMO GENÉTICO COM OPERADORES MODIFICADOS
PARA OTIMIZAÇÃO DE FUNÇÕES DE VARIÁVEIS REAIS

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, PPGEE, da Universidade Federal do Pará, para a obtenção do grau de Doutor em Engenharia Elétrica na área de Computação Aplicada.

UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2019

EDSON KOITI KUDO YASOJIMA

USO DE ALGORITMO GENÉTICO MODIFICADO PARA OTIMIZAÇÃO DE
FUNÇÕES DE VARIÁVEIS REAIS

Tese de Doutorado apresentada ao Programa de Pós-Graduação em Engenharia Elétrica, PPGEE, da Universidade Federal do Pará, como requisito necessário à obtenção do título de Doutor em Engenharia Elétrica.

BANCA EXAMINADORA

Prof. Dr. Roberto Célio Limão de Oliveira (PPGEE/UFPA)

Membro da Banca Examinadora (Orientador)

Prof. Dr. Otávio Noura Teixeira (Campus Tucuruí/UFPA)

Membro da Banca Examinadora (Co-Orientador Informal / Externo ao Programa)

Prof. Dr. Jasmine Priscyla Leite De Araujo (PPGEE/UFPA)

Membro da Banca Examinadora (Examinadora Interna)

Prof. Dr. Filipe de Oliveira Saraiva (ICEN/UFPA)

Membro da Banca Examinadora (Examinador Externo ao Programa)

Prof. Dr. Alexandre Cesar Muniz De Oliveira (UFMA)

Membro da Banca Examinadora (Examinador Externo à Instituição)

Prof. Dr. Deam James Azevedo Silva (UFOPA)

Membro da Banca Examinadora (Examinador Externo à Instituição)

Lista de Figuras

Figura 1 Fluxograma do algoritmo genético. (Fonte: o autor)	20
Figura 2 Exemplo de regiões ótimas em um problema de minimização (Fonte: o autor).....	23
Figura 3 Tipos de Crossover (Fonte: o autor)	23
Figura 4 Cruzamento Binário Clássico (Fonte: o autor).....	24
Figura 5 Cruzamento Aritmético Clássico (Fonte: o autor)	24
Figura 6 Demonstração do ADX utilizando dois progenitores, pode-se perceber que o crossover sempre gera filhos com os genes paternos levemente modificados de acordo com o peso (Fonte: o autor).....	29
Figura 7 Exemplo da aplicação do CAM em um problema de minimização utilizando uma determinada população do AG (Fonte: o autor).....	32
Figura 8 Comportamento dos indivíduos afetados pelo método de ajuste proposto (Fonte: o autor)	32
Figura 9 Distribuição da população pelo espaço de busca do F1-30D conforme o avanço das gerações. Gráficos à esquerda são do ADX e da direita, do ARX (Fonte: o autor)	39
Figura 10 Distribuição da população pelo espaço de busca do F14-30D conforme o avanço das gerações. Gráficos à esquerda são do ADX e da direita, do ARX (Fonte: o autor)	39
Figura 11 Pontos do Caminho Ótimo desejado Pdc (Fonte: o autor)	41
Figura 12 Caminho gerado pelo ADX (Pgc) comparado com o Pdc. O caminho gerado é bastante idêntico ao caminho desejado (Pdc) (Fonte: o autor).....	43
Figura 13 Caminho gerado pelo SBX (Pgc) comparado com o Pdc. O SBX não obteve muito sucesso em gerar um caminho parecido com o desejado (Pdc), no entanto é possível perceber que com o uso do CAM, o caminho gerado (Pgc) é mais preciso (Fonte: o autor)	43
Figura 14 O ARX mostrou melhores resultados que o SBX, no entanto, pior que o ADX. Apresentou coordenadas falhas em alguns conectores, sendo que com o CAM, o caminho gerado (Pgc) foi mais preciso que a versão sem o ajuste. (Fonte: o autor)	44
Figura 15 Curvas de convergência de cada configuração. As configurações da esquerda utilizam o CAM e obtiveram uma curva mais próxima do eixo X em relação as configurações com mutação convencional (Gráficos à direita) (Fonte: o autor)	45
Figura 16 Gráficos de Diversidade de cada crossover para o problema DPV1. Os gráficos à esquerda utilizam o CAM, enquanto que os gráficos da direita utilizam a mutação convencional (Fonte: o autor)	50
Figura 17 Curvas de convergência das configurações testadas para o problema DPV1. O gráfico à esquerda utiliza o CAM, enquanto que o da direita utiliza a mutação convencional (Fonte: o autor).....	51
Figura 18 Gráficos de Diversidade de cada crossover para o problema DPV2. Os gráficos à esquerda utilizam o CAM, enquanto que os gráficos da direita utilizam a mutação convencional (Fonte: o autor)	52
Figura 19 Curvas de convergência das configurações testadas para o problema DPV1. O gráfico à esquerda utiliza o CAM, enquanto o da direita utiliza a mutação convencional (Fonte: o autor)	53
Figura 20 Gráficos de Diversidade de cada crossover para o problema WBD1. Os gráficos à esquerda utilizam o CAM, enquanto os gráficos da direita utilizam a mutação convencional (Fonte: o autor)	54
Figura 21 Curvas de convergência das configurações testadas para o problema WBD1. O gráfico à esquerda utiliza o CAM, enquanto o da direita utiliza a mutação convencional (Fonte: o autor)	55
Figura 22 Gráficos de Diversidade de cada crossover para o problema WBD2. Os gráficos à esquerda utilizam o CAM, enquanto os gráficos da direita utilizam a mutação convencional (Fonte: o autor)	56
Figura 23 Curvas de convergência das configurações testadas para o problema WBD2. O gráfico à esquerda utiliza o CAM, enquanto o da direita utiliza a mutação convencional (Fonte: o autor)	57

Figura 24 Gráficos de Diversidade de cada crossover para o problema MWTCS. Os gráficos à esquerda utilizam o CAM, enquanto os gráficos da direita utilizam a mutação convencional (Fonte: o autor)	58
Figura 25 Curvas de convergência das configurações testadas para o problema MWTCS. O gráfico à esquerda utiliza o CAM, enquanto o da direita utiliza a mutação convencional (Fonte: o autor)	59
Figura 26 Gráficos de Diversidade de cada crossover para o problema SRD. Os gráficos à esquerda utilizam o CAM, enquanto os gráficos da direita utilizam a mutação convencional (Fonte: o autor)	60
Figura 27 Curvas de convergência das configurações testadas para o problema SRD. O gráfico à esquerda utiliza o CAM, enquanto o da direita utiliza a mutação convencional (Fonte: o autor)	61
Figura 28 Gráficos de Diversidade de cada crossover para o problema TBT. Os gráficos à esquerda utilizam o CAM, enquanto os gráficos da direita utilizam a mutação convencional (Fonte: o autor)	62
Figura 29 Curvas de convergência das configurações testadas para o problema TBT. O gráfico à esquerda utiliza o CAM, enquanto o da direita utiliza a mutação convencional (Fonte: o autor)	63
Figura 30 Gráficos de Diversidade de cada crossover para o problema DGT. Os gráficos à esquerda utilizam o CAM, enquanto os gráficos da direita utilizam a mutação convencional (Fonte: o autor)	64
Figura 31 Curvas de convergência das configurações testadas para o problema DGT. O gráfico à esquerda utiliza o CAM, enquanto o da direita utiliza a mutação convencional (Fonte: o autor)	65
Figura 32 Curvas de Convergência para os problemas DPV1, DPV2, MWTCS e TBT (Fonte: o autor)	67
Figura 33 Curvas de Convergência para os problemas WBD1, WBD2, SRD e DGT (Fonte: o autor)	68

Lista de Tabelas

Tabela 1 Matriz que o algoritmo irá construir com os indivíduos da população no final de cada geração.....	30
Tabela 2 Resultados do ADX+CAM com 10 dimensões	36
Tabela 3 Resultados do ADX+CAM com 30 dimensões	36
Tabela 4 Resultado comparativo das funções unimodais (F1 e F2)	37
Tabela 5 Resultado comparativo das funções multimodais simples (F3 a F9)	37
Tabela 6 Resultado comparativo das funções híbridas (F10, F11 e F12)	38
Tabela 7 Resultado comparativo das funções compostas (F13, F14 e F15)	38
Tabela 8 Melhor resultado obtivo pelo ADX+CAM.....	45
Tabela 9 Níveis médios de SPD de todas as configurações para o problema DPV1	50
Tabela 10 Melhores resultados obtidos em cada configuração para o problema do DPV1	51
Tabela 11 Níveis médios de SPD de todas as configurações para o problema DPV2.....	52
Tabela 12 Melhores resultados obtidos em cada configuração para o problema do DPV2	53
Tabela 13 Teste Friedman entre as configurações para o problema DPV1	53
Tabela 14 Teste Friedman entre as configurações para o problema DPV2	53
Tabela 15 Níveis médios de SPD de todas as configurações para o problema WBD1	54
Tabela 16 Melhores resultados obtidos em cada configuração para o problema do WBD1	55
Tabela 17 Teste Friedman entre as configurações para o problema WBD1	55
Tabela 18 Níveis médios de SPD de todas as configurações para o problema WBD2.....	56
Tabela 19 Melhores resultados obtidos em cada configuração para o problema do WBD2.....	57
Tabela 20 Teste Friedman entre as configurações para o problema WBD2.....	57
Tabela 21 Níveis médios de SPD de todas as configurações para o problema MWTCS	58
Tabela 22 Melhores resultados obtidos em cada configuração para o problema do MWTCS	59
Tabela 23 Teste Friedman entre as configurações para o problema MWTCS.....	59
Tabela 24 Níveis médios de SPD de todas as configurações para o problema SRD	60

Tabela 25 Melhores resultados obtidos em cada configuração para o problema do SRD	61
Tabela 26 Teste Friedman entre as configurações para o problema SRD	61
Tabela 27 Níveis médios de SPD de todas as configurações para o problema TBT	62
Tabela 28 Melhores resultados obtidos em cada configuração para o problema do TBT	63
Tabela 29 Teste Friedman entre as configurações para o problema TBT.....	63
Tabela 30 Melhores resultados obtidos em cada configuração para o problema do DGT.....	65
Tabela 31 Teste Friedman entre as configurações para o problema DGT	65
Tabela 32 Resultados comparativos para os algoritmos evolutivos, os valores são a média de 20 execuções para cada algoritmo.....	66
Tabela 33 Teste T-pareado assumindo variâncias iguais entre o ADX+CAM e os outros dois algoritmos testados (PSO e SA).....	66
Tabela 34 Resultados Comparativos para o problema DPV1.....	69
Tabela 35 Resultados Comparativos para o problema DPV2.....	69
Tabela 36 Resultados Comparativos para o problema WBD1 (* - houve quebra de restrições)	69
Tabela 37 Resultados Comparativos para o problema WBD2	69
Tabela 38 Resultados Comparativos para o problema MWTCS	70
Tabela 39 Resultados Comparativos para o problema SRD.....	70
Tabela 40 Resultados Comparativos para o problema TBT	70
Tabela 41 Resultados Comparativos para o problema DGT	71

Lista de Publicações Relacionadas

1. YASOJIMA, E. K. K.; De Oliveira, R. ; Teixeira, O. ; Pereira, R. . CAM-ADX: A New Genetic Algorithm with Increased Intensification and Diversification for Design Optimization Problems with Real Variables. ROBOTICA (Cambridge), p. 1-46, 2019.
2. YASOJIMA, E. K. K.; DE OLIVEIRA, ROBERTO C. LIMA O ; TEIXEIRA, OTAVIO NOURA ; Pereira, Rodrigo Lisbôa . Um Algoritmo Genético com Cruzamento Ajustado e Ajuste Estatístico Populacional para Otimização de Problemas de Engenharia com Restrições em Variáveis. In: Simpósio Brasileiro de Automação Inteligente, 2017, Porto Alegre. Simpósio Brasileiro de Automação Inteligente - 2017, 2017.
3. YASOJIMA, E. K. K.; DE OLIVEIRA, ROBERTO C. LIMA O ; TEIXEIRA, OTAVIO NOURA ; Pereira, Rodrigo Lisbôa . A Genetic Algorithm with Adjusted Crossover and Global Adjustment Mechanism for Engineering Problems Optimization. In: Latin America Conference of Computational Intelligence, 2017, Arequipa. Latin America Conference of Computational Intelligence - LACCI, 2017.
4. YASOJIMA, EDSON KOITI KUDO; de Oliveira, Roberto Célio Limão ; Teixeira, Otávio Noura ; Lisbôa, Rodrigo ; Mollinetti, Marco . Analyzing Genetic Algorithm with Game Theory and Adjusted Crossover Approach on Engineering Problems. Advances in Intelligent Systems and Computing. 15ed.: Springer International Publishing, 2016, v. , p. 177-188.

Sumário

Resumo	11
<i>Abstract</i>	12
Capítulo 1. Introdução	13
1.1 Computação Evolutiva.....	13
1.2 Capacidade Exploratória e Intensificação.....	14
1.3 Estado da Arte e Motivação da Tese	15
Capítulo 2. Algoritmos Genéticos	20
2.1 Funcionamento de um algoritmo genético.....	20
2.2 Cruzamentos Utilizados	23
2.3 Exploração e Intensificação e Níveis de Diversidade.....	25
Capítulo 3. O Algoritmo Proposto.....	27
3.1 Cruzamento Ajustado (ADX)	27
3.2 Mutaç�o de Ajuste Correlacional (CAM).....	29
3.3 Custo Computacional dos M�todos Propostos	32
Capítulo 4. Problemas Utilizados e Resultados Obtidos	35
4.1 Problemas computacionais do CEC'2015 (IEEE <i>Congress on Evolutionary Computation</i>)	35
4.2 Cinem�tica Inversa em Rob�tica	40
4.2.1 Formula�o do Problema	40
4.2.2 Resultados	42
4.3 Problemas de Engenharia.....	46
4.3.1 Design de Vaso de Press�o / Design of Pressure Vessel (DPV)	46
4.3.2 Design de uma Viga de A�o / Welded Beam Design (WBD).....	46
4.3.3 Min. Tens�o-Peso sobre Mola / Minimize Tension-Compression on a Spring (MWTCS).....	47
4.3.4 Design de um Redutor de Velocidade / Speed Reducer Design (SRD) ...	48

4.3.5	Treliça de Três Barras / Tree-bar Truss (TBT).....	48
4.3.6	Design de Engrenagem de Trem / Design of Gear Train (DGT)	49
4.4	Análise de Diversidade e Resultados Obtidos para os Problemas de Engenharia	49
4.4.1	DPV 1 e 2.....	50
4.4.2	WBD 1 e 2.....	54
4.4.3	MWTCS	58
4.4.3	SRD	59
4.4.4	TBT	61
4.4.5	DGT.....	63
4.5	Comparação do ADX+CAM com outros Algoritmos Evolutivos Clássicos...	65
4.6	Comparação dos Resultados com outros Trabalhos da Literatura.....	68
	Capítulo 5. Considerações Finais	72
	Referências	74

Resumo

Este trabalho apresenta um Algoritmo Genético que utiliza uma nova operação de crossover (ADX – *Adjusted Crossover*) e um novo método de mutação através do ajuste baseado em correlação (CAM – *Correlation Adjustment Mutation*). Ambos os operadores propostos são executados na população do AG, otimizando soluções já existentes (ADX) e aumentando o poder exploratório global do algoritmo através do ajuste correlacional (CAM). Heurísticas puramente baseadas em solução oferecem um bom nível de otimização local (intensificação), no entanto, possuem baixo nível de exploração global do espaço de busca. Por outro lado, algoritmos evolutivos oferecem capacidades exploratórias globais melhores. A partir deste pressuposto, os métodos que buscam o aumento de tais capacidades exploratórias são vastamente utilizados para otimização de problemas com restrições de variáveis, devido à utilização vantajosa do poder de busca das duas abordagens (baseado em solução e evolutivos). Os métodos propostos neste trabalho aumentaram o poder de exploração e intensificação do algoritmo genético utilizado. De acordo com os testes executados, o algoritmo genético com os operadores propostos apresentou resultados melhores para a maioria dos problemas clássicos de engenharia quando comparados a outros métodos presentes na literatura. Além disso, obteve bons resultados em problemas de robótica e performance mediana quando utilizados com *benchmarks* propostos no evento CEC'2015.

Palavras-chave: Algoritmo Genético, Otimização, Variáveis Reais, Problemas de Engenharia, Exploração, Intensificação.

Abstract

This work presents a modified Genetic Algorithm using a new crossover operator (ADX) and a novel statistic correlation mutation algorithm (CAM). Both ADX and CAM work with population information to improve existing individuals of the GA and increase the exploration potential via the correlation mutation. Solution-based methods offers good local improvement of already known solutions while lacking at exploring the whole search space, evolutionary algorithms provide better global search in exchange of exploitation power. Methods that increase the search potential are widely used for constrained optimization problems due to increased global and local search capabilities. The GA with the proposed operators improves results of constrained problems by balancing the exploitation and exploration potential of the algorithm. The conducted tests present average performance for various CEC'2015 benchmark problems, while offering good reliability and superior results on path planning problem for redundant manipulator and most of the constrained engineering design problems tested when compared with current works in the literature and classic optimization algorithms.

Keywords: *Genetic Algorithm, Design Optimization, Real Variables, Engineering Design, Exploration, Exploitation.*

Capítulo 1. Introdução

Este capítulo tem como objetivo ambientar o leitor na área de algoritmos bioinspirados, bem como demonstrar o atual estado da arte em métodos de otimização de variáveis reais e mostrar a motivação que levou ao desenvolvimento dessa pesquisa.

1.1 Computação Evolutiva

Computação Evolutiva (CE) utiliza o poder computacional para simular a evolução de espécies na otimização de problemas de alta complexidade numérica e/ou combinatória. Existem inúmeras técnicas já implementadas e bem adaptadas dentro da comunidade científica da área de CE, podemos citar os Algoritmos Genéticos (AG ou GA - *Genetic Algorithm* [1]), Enxame de Partículas (PSO - *Particle Swarm Optimization* [2]), Colônia de Abelhas (ABC - *Artificial Bee Colony* [3]), Evolução Diferencial (DE - *Differential Evolution* [4]), Busca Harmônica (HS - *Harmony Search* [5]), entre outros. Além disso, existem modelos heurísticos híbridos que utilizam diferentes técnicas evolutivas combinadas com objetivos de aumentar o poder de otimização. Grande parte destas técnicas utilizam o conceito de população, composta por diversos indivíduos que representam uma possível solução para o problema. Esta população então evolui seguindo as fases e regras da técnica utilizada, e ao final da execução do algoritmo, o indivíduo mais evoluído da população é selecionado como a melhor possível resposta para o problema analisado. [1]

Cada técnica evolutiva depende de parâmetros que devem ser ajustados para realizar uma otimização satisfatória do algoritmo. Tais ajustes podem variar conforme o problema a ser resolvido, bem como a codificação das variáveis representativas de cada indivíduo representante da população. Algumas funções possuem determinados números de variáveis que podem ser parametrizados para maximizar ou minimizar os resultados de acordo com o objetivo do problema. Tal busca por parâmetros ideais em grande parte dos problemas reais são complexas quando utilizado um algoritmo de força bruta ou técnicas matemáticas tradicionais, sendo necessário algum tipo de heurística ou um processo estocástico refinado para conseguir resultados satisfatórios. [6][7]

1.2 Capacidade Exploratória e Intensificação

Para um algoritmo evolutivo baseado em população possuir uma capacidade de otimização aceitável, é preciso ter um nível balanceado de poder exploratório e de intensificação de soluções prévias. A capacidade de exploração (também conhecido como *exploration*) é o poder do algoritmo explorar novas áreas o espaço de busca do problema, possibilitando a descoberta de novas soluções e aumentando a diversidade da população do algoritmo. Já a intensificação (conhecido como *exploitation*) é a capacidade de melhorar as soluções já encontradas, permitindo que indivíduos melhorem seus genes e otimizem ainda mais a solução do problema [8].

Atingir um nível balanceado entre estas duas características não são simples ao algoritmo evolutivo, existem vários fatores diretamente ligados ao quanto um algoritmo pode ter níveis bons de exploração e intensificação, sendo eles: complexidade do problema a ser otimizado, tipos de codificação das variáveis, operadores evolutivos utilizados etc. A deficiência da capacidade exploratória do algoritmo pode gerar convergência prematura durante a sua execução, isto é, a população se “prende” em um ótimo local no espaço de busca do problema e a evolução do mesmo fica congelado por diversas iterações/gerações do programa. Já a deficiência de intensificação causa dificuldade no processo evolutivo individual do algoritmo, ou seja, os indivíduos da população não conseguem gerar herdeiros com genes “mais evoluídos”, impossibilitando o programa de achar uma solução ótima satisfatória.

É necessário a aplicação de métodos para haver o balanceamento adequado de exploração local (intensificação) e global (exploração). Bons níveis de diversidade de um algoritmo evolutivo em consequência de um bom poder exploratório, garante uma varredura do espaço de busca mais efetiva, juntamente com um bom nível de aprimoramento de soluções já conhecidas garantem uma otimização com resultados melhores.

Existem diversas metodologias de manutenção de diversidade da população em algoritmos evolucionários, alguns se baseiam na: alteração nos métodos de seleção, agrupamento de indivíduos, controle de elitismo, compartilhamento de genes, dentre outros. O estudo do comportamento dos indivíduos durante a execução do algoritmo permite verificar se a população tem uma boa capacidade de explorar o espaço de busca

do problema, possibilitando a utilização de mecanismos para evitar o agrupamento de muitos indivíduos em áreas de ótimo local [9][10][11][12][13][14].

1.3 Estado da Arte e Motivação da Tese

Os algoritmos evolucionários, mais especificamente os algoritmos genéticos, são técnicas bem difundidas no meio científico, sendo inicialmente criado por [15]. Atualmente modificações destes algoritmos ainda são aplicadas em diversas pesquisas científicas com objetivo de otimizar problemas específicos como por exemplo em [16][17][18][19]. Muitas destas propostas são caracterizadas como algoritmos evolucionários híbridos onde mistura-se recursos de diferentes algoritmos evolutivos, no geral possuindo capacidades de otimização melhores que um algoritmo evolucionário puro em virtude do agrupamento de vantagens de diferentes técnicas, tendo em troca um aumento na complexidade do programa e tempo de processamento do computador.

Crepinsek et. al. [20] faz uma vasta pesquisa sobre a questão do poder de busca global e local em algoritmos evolutivos. Em algoritmos genéticos, variações em operadores de seleção, cruzamento, mutação e outros operadores adicionais, podem aumentar determinada característica de busca (global ou local) do algoritmo. Os métodos de seleção do AG podem aumentar o poder de exploração global ou local do algoritmo, com um método de seleção rígido (melhores indivíduos têm muito mais chance), é possível convergir grande parte da população para uma região da busca aumentando o potencial local, enquanto que um método que favorece mais indivíduos menos aptos espalha a população pelo espaço de busca, dando o algoritmo um melhor poder de exploração global em troca da otimização local. Métodos de mutação e cruzamento, também têm papel importante nos AGs, podendo aumentar o poder de exploração global e local da otimização.

Para determinar a eficiência de um algoritmo genético, é necessário a dependência de alguns fatores e parâmetros. Um deles é a medida de diversidade da população. Uma população com uma taxa de diversidade baixa pode causar convergência prematura no AG [9]. Para haver uma eficácia no comportamento do AG, é necessário balancear as medidas de exploração (diversidade) e a intensificação (refinamento de indivíduos). Trabalhos como [20][21][22] elucidam diferentes metodologias para fazer a manutenção de exploração/intensificação de algoritmos evolutivos.

Além disso, o nível de diversidade da população é um forte indicativo da performance do algoritmo quanto a varredura do espaço do problema. Um baixo nível de diversidade durante as gerações iniciais, indica que o GA está tendo um comportamento prematuro, ou seja, provavelmente a população está presa em um ótimo local do problema, enquanto que um alto nível de diversidade indica que os indivíduos estão mais espalhados pelo espaço de busca, isso caracteriza uma boa capacidade exploratória global do método utilizado. Existem diversas pesquisas para fazer a manutenção da diversidade de uma população, onde mostra-se que é preciso analisar os parâmetros do algoritmo, junto ao problema a ser otimizado [9][11][12][13][14].

O estudo da codificação binária ou real para os algoritmos genéticos não é recente, desde 1998, F.Herrera et al. [23] fizeram um estudo considerando diversos métodos de cruzamento e mutação em AGs, onde mostraram diversas comparações entre as codificações binária e real utilizando diferentes combinações entre métodos de cruzamento e mutações, testando em problemas de otimização com números reais e variando os parâmetros do algoritmo como: população e número de gerações, constatando diversos pontos, onde a codificação real se mostrou bastante promissora quando verificados o poder de exploração local dos algoritmos em comparação à codificação binária. A codificação binária mostrou ter um bom desempenho na exploração global do problema, no entanto, não conseguiram otimizar os resultados previamente encontrados, resultando em uma otimização insatisfatória em relação aos outros métodos de cruzamento real testados.

A operação de crossover ainda é um tópico bastante explorado em algoritmos evolucionários para otimização de problemas complexos com codificação real. Métodos de otimização local como o *Hill Climbing* e Busca Tabu já são bastante difundidos, pois tem uma velocidade superior quando comparados a algoritmos baseados em população como o Enxame de Partículas e Algoritmos Genéticos na otimização de soluções existentes, isso acontece pois são especializados em otimizar indivíduos já criados, não focando na geração de novas soluções para o problema. Porém, estes métodos têm deficiências em relação à otimização de problemas que tenham muitos ótimos locais, tendo dificuldade em ‘varrer’ todo espaço de busca por uma solução global ótima quando comparados aos métodos populacionais, que possuem uma convergência mais lenta, no entanto são melhores quando utilizados nestes tipos de problemas, pois tem um poder de exploração maior. Baseado no comportamento dos algoritmos citados, fica

evidente que um método que tenha um poder de exploração global e local balanceado é o ideal para a otimização de problemas [20][24][25].

O *crossover* dos algoritmos genéticos é uma das fases mais importantes do algoritmo, visto que com o cruzamento é possível gerar novas soluções e explorar o espaço de busca de maneira eficaz baseado no tipo de dados e no cruzamento utilizado. Inúmeras pesquisas nos últimos anos vêm sendo feitas em operações de cruzamento em AGs, dentre os tipos de cruzamento mais difundidos com utilização de números com codificação real, pode-se citar o crossover aritmético (*Blend Crossover – BLX/ARX* [26]) e o crossover binário para números reais (*Simulated Binary Crossover – SBX*) [27][28]. Muitos destes cruzamentos encontram dificuldades em problemas com codificação com valores reais quando utilizados em problemas com espaços de busca com restrições, pois podem ficar presos em ótimos locais ou acabar gerando indivíduos inviáveis à solução. Além disso, os genes dos novos indivíduos criados ficam geralmente limitados a valores entre os genes dos pais, dificultando a verificação de outras áreas no espaço de busca quando a população se encontra muito concentrada em determinada área [29][30][31][32]. Atualmente, além dos passos básicos do algoritmo genético, existem algumas técnicas que podem ser implementadas para aumentar a capacidade de diversificação da população, provendo ao AG um maior poder exploratório. Dentre algumas técnicas pode-se citar o modelo de ilhas, algoritmo genético híbrido, *crowding*, aprendizado cultural, etc [20].

A introdução de novas técnicas e operadores também são pesquisas bastante emergentes em algoritmos genéticos. Com o surgimento de problemas cada vez mais complexos, é preciso o constante aprimoramento dos algoritmos utilizados para nivelar o poder computacional com a complexidade do problema. Seguindo as pesquisas na área dos algoritmos genéticos, é notável o trabalho nos operadores de cruzamento (*crossover*). A escolha do operador de *crossover* deve-se ao fato de ser um operador de importância crucial para o desenvolvimento da solução do algoritmo genético, e tem uma probabilidade muito maior de ocorrer em relação a mutação.

Para a otimização de problemas com restrições de variáveis, é possível notar um grande número de trabalhos que implementam algoritmos evolucionários híbridos. Estes algoritmos são capazes de balancear o potencial exploratório local e global, resultando em performance e otimização melhores em relação aos algoritmos evolucionários

clássicos puros. Patwal et al [33], propôs um novo PSO utilizando diferentes estratégias de mutação de indivíduos. Rani et al. [34] criou um ambiente de lógica fuzzy com modelos pessimistas e otimistas para otimização de problemas numéricos.

O pesquisador Harish Garg em seus trabalhos, implementou diversos algoritmos híbridos para otimização de problemas matemáticos. Em [18] implementou um algoritmo híbrido entre enxame de partículas e algoritmo genético, conseguindo bons resultados para problemas clássicos de engenharia com restrições de variáveis. Em [16] e [17] foi proposto um algoritmo baseado na biogeografia e outro PSO híbrido com lógica Fuzzy para solução de modelos matemáticos de alocação redundantes. Em [19] é proposto um PSO híbrido para análise de comportamento e sensibilidade em variáveis de sistemas industriais.

Quanto aos operadores de cruzamento em algoritmos genéticos, Li et al. [35] propõe um método de cruzamento de 1-ponto de corte, gerando mais de 2 filhos por cruzamento, no entanto preservando a quantidade original da população, eliminando os indivíduos menos aptos para passar à próxima geração. Em testes com funções numéricas, o algoritmo proposto se mostrou melhor em relação ao algoritmo genético tradicional. Qiongbing et al. [36] desenvolveram um método de cruzamento para indivíduos com genótipos de tamanho variado. Utilizado em problemas de ordenamento, o algoritmo proposto mostrou uma curva de convergência mais lenta, no entanto alcançou melhores resultados, e teve um menor custo em relação ao cruzamento tradicional para problemas de ordenamento. Shrestha e Mahmood [37] propuseram um conceito de DNA mitocondrial (mtDNA) para o genótipo de indivíduos do algoritmo genético. No trabalho proposto, o mtDNA funciona como um parâmetro a mais no crossover, que possibilita o “espalhamento” de genes bons para mais indivíduos além dos gerados apenas com o cruzamento da elite da população. Além disso o algoritmo utiliza o modelo de ilhas para processamento em paralelo. Aplicado a problemas de caixeiro viajante (TSP) o algoritmo teve melhor desempenho para diversos resultados em relação a outros trabalhos.

Todos os trabalhos citados implementam diferentes técnicas, criando métodos híbridos para otimização de problemas melhores que a implementação singular de algoritmos clássicos. Isso acontece devido ao crescimento do poder exploratório global e local de melhoramento dos algoritmos híbridos.

Baseado na pesquisa bibliográfica, esta pesquisa propõe um novo método de cruzamento focado no aumento da capacidade exploratória local do algoritmo genético e um método de mutação de ajuste correlacional, responsável por aumentar a capacidade exploratória global do AG, para problemas de variáveis reais. O cruzamento proposto cria filhos com genes ajustados dos pais baseado em informações prévias sobre os genes de parte da população, possibilitando uma varredura local do espaço de busca e aumentando o poder de intensificação, enquanto o ajuste estatístico proposto gera uma perturbação em toda população do AG, possibilitando uma varredura global do problema e conseqüentemente o poder de exploração. O efeito do método de cruzamento proposto e o ajuste de genes, será analisado por meio da medida de diversidade gerada pelas duas técnicas implementadas. Além disso, o desempenho do AG com o uso dessas duas novas técnicas será comparado com outros trabalhos que utilizam os mesmos problemas de engenharia, robótica e *benchmark* (CEC'2015) como teste.

Capítulo 2. Algoritmos Genéticos

Este capítulo irá abordar o funcionamento básico dos algoritmos genéticos, mostrar os principais cruzamentos utilizados nos testes e relatar a importância dos níveis de exploração, intensificação e diversidade dos algoritmos evolutivos.

2.1 Funcionamento de um algoritmo genético

O Algoritmo genético é uma das áreas de pesquisa de algoritmos evolucionários [15]. Desde a sua popularização, mostrou-se bastante efetivo para otimização de funções [6]. Ao invés de utilizar apenas uma solução, o GA trabalha com um conjunto de possíveis soluções dentro de um espaço de busca, onde evolui estas soluções para atingir um resultado próximo ao ótimo [7]. A estratégia de evolução dos algoritmos genéticos, foi inspirado na teoria da evolução de Darwin, ao qual uma determinada população de indivíduos gera descendentes mais aptos para a próxima geração, evoluindo cada vez mais em função do número de gerações passadas [15].

Os algoritmos genéticos são caracterizados por possuir as seguintes etapas [Figura 1]:

1. Inicialização da população
2. Seleção
3. Cruzamento
4. Mutação

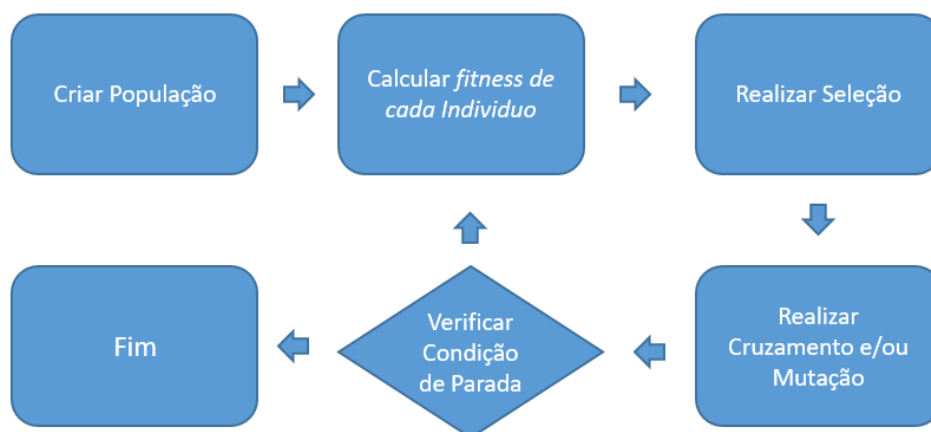


Figura 1 Fluxograma do algoritmo genético. (Fonte: o autor)

Após a inicialização da população, o GA executa iterativamente as operações de seleção, cruzamento e mutação até atingir o critério de parada (número máximo de iterações, atingir algum determinado resultado, etc.).

Na fase de inicialização, a população inicial geralmente é gerada aleatoriamente. Cada indivíduo gerado, representa uma solução para o problema. A codificação de cada indivíduo depende da implementação do algoritmo, podendo ter representação em binário e real. Por exemplo, para uma função $f(x, y)$, um indivíduo c poderia ser codificado com dois genes x e y .

Após a geração da população, cada indivíduo vai ser testado empiricamente em uma função e vai receber um valor de aptidão chamado *fitness*. Este valor geralmente é dado por uma função objetivo $f(x)$ que se deseja otimizar com o AG. Dependendo do problema a ser otimizado, o melhor valor pode ser o maior para casos de maximização, ou o menor para casos de minimização.

Depois que todos os receberem os valores de *fitness*, alguns indivíduos da geração atual serão selecionados para avançar para etapa de cruzamento. A operação de seleção no AG, segue os mesmos princípios da seleção natural, os indivíduos com melhor *fitness* têm mais chance de gerar filhos para próxima geração em comparação a indivíduos menos aptos.

Existem diversos tipos de operadores de seleção [15] dependendo do tipo de codificação do indivíduo. Alguns operadores de seleção clássicos são:

- *Roulette Wheel Selection*: Este método cria uma roleta em formato de ‘pizza’ onde cada fatia representa a porcentagem de escolha de determinado indivíduo, quanto melhor o *fitness*, maior a fatia correspondente na roleta. Este método não é muito efetivo quando a diferença de *fitness* dos indivíduos é muito alta, visto que *fitness* muito altos, tendem a pegar uma fatia muito grande, praticamente anulando as chances de seleção das fatias menores.
- *Tournament*: o método de torneio cria pequenos subgrupos de indivíduos selecionados aleatoriamente, dentro destes subgrupos é realizado uma disputa onde indivíduos com melhor *fitness* tem mais chance de ganhar. A chance e o número de indivíduos no subgrupo são definidos na implementação do AG. Este

método é mais efetivo em comparação com a roleta, pois gera uma seleção mais balanceada.

- *Stochastic Universal Sampling*: Este operador é muito semelhante a roleta, além disso este método faz uma ponderação do *fitness* dos participantes da roleta, criando uma fatia normalizada para cada um.

Após a seleção, os operadores de cruzamento são os responsáveis por misturar a carga genética de dois indivíduos e gerar filhos para a próxima geração. As operações de crossover são os responsáveis por evoluir a população, através da combinação dos genótipos dos pais, com a chance de melhorar a solução para o problema.

Junto, ou após a operação de cruzamento, os indivíduos podem sofrer mutações, que são pequenas perturbações geralmente aleatórias na carga genética do indivíduo, podendo assim aumentar as chances de evitar que o algoritmo fique preso em um ótimo local e melhore a busca por melhores resultados.

As condições de término do algoritmo geralmente são atingidas quando o GA:

- Atinge o resultado ótimo do problema
- Executou o número máximo de gerações definidas
- Ficou preso em algum ótimo local por número determinado de gerações

Para que um algoritmo evolucionário seja efetivo na otimização de problemas, é preciso verificar o poder de exploração local e global do algoritmo. A exploração global (diversificação) é a capacidade do algoritmo de varrer efetivamente o espaço de busca do problema, evitando que as soluções ótimas encontradas fiquem presas em seções chamados de “ótimos locais”. As áreas denominadas ótimos locais, são regiões dentro do espaço de busca onde o algoritmo sugere que encontrou a melhor solução para o problema e não consegue mais usar o seu poder exploratório num contexto global. É de grande importância o algoritmo percorrer todo espaço de busca do problema para verificar todas as ocorrências possíveis de ótimos locais, para assim depois iniciar o processo de otimização das soluções já encontradas (intensificação) [Figura 2].

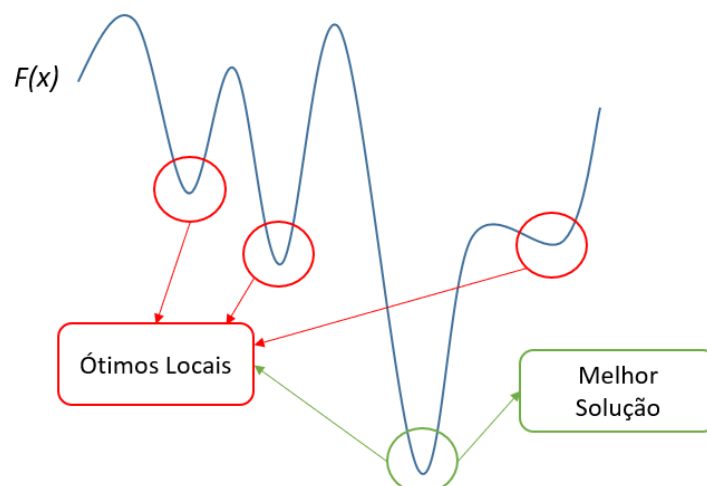


Figura 2 Exemplo de regiões ótimas em um problema de minimização (Fonte: o autor)

2.2 Cruzamentos Utilizados

O cruzamento é um operador/fase responsável por criar uma variabilidade de genes e a consequente otimização das soluções existentes no algoritmo genético. Matematicamente, o crossover, para genes com variáveis reais, cria genes com valores próximos ou iguais aos pais, seguindo a seguinte (Equação 1):

$$\beta = \left| \frac{c_1 + c_2}{p_1 + p_2} \right| \quad (\text{Equação 1})$$

Onde β é o fator de dispersão com relação aos genes dos filhos c_1 e c_2 a partir dos genes dos pais p_1 e p_2 . Dependendo do resultado o fator pode ser de três tipos: Contração, Dispersão ou Estacionário (Figura 3).

- Contração $\beta < 1$

- Valores dos genes gerados estão entre o limite superior e inferior dos genes paternos



- Dispersão $\beta > 1$

- Valores dos genes gerados estão além do limite superior e inferior dos genes paternos



- Estacionário $\beta = 1$

- Valores dos genes gerados são iguais aos genes paternos



Figura 3 Tipos de Crossover (Fonte: o autor)

O cruzamento binário (SBX) [27][28] realiza uma simples troca de genes entre os indivíduos selecionados, este tipo de cruzamento não tem um nível exploratório global muito efetivo, pois utiliza apenas genes existentes. No entanto, suas variantes são bastante utilizadas em problemas de análise combinatória, onde é preciso gerar indivíduos com diferentes combinações de valores existentes. O cruzamento binário é um crossover de comportamento estacionário.

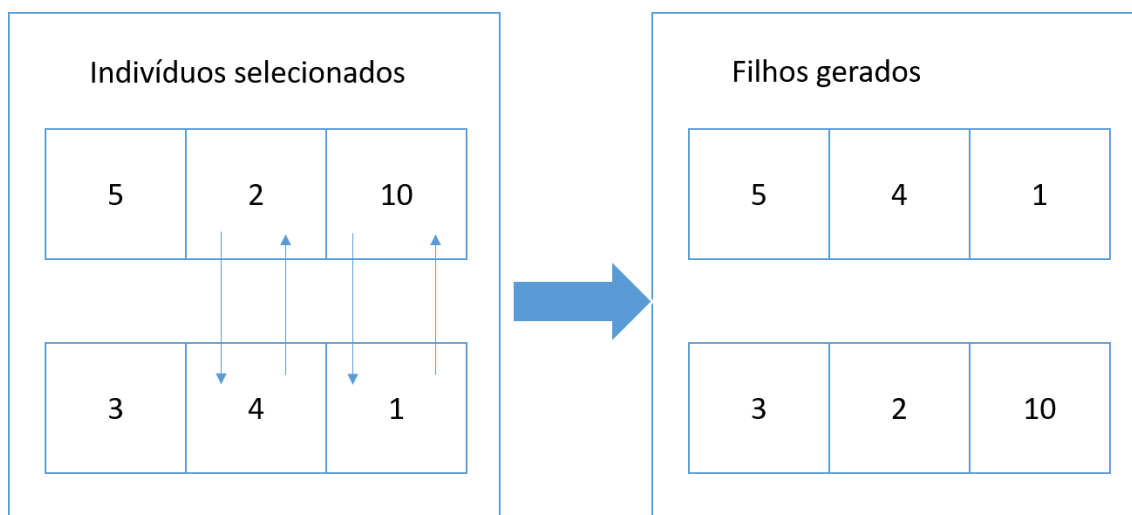


Figura 4 Cruzamento Binário Clássico (Fonte: o autor)

Outro cruzamento existente é o aritmético (ARX) [26], este cruzamento sempre gera um novo valor baseado nos genes dos indivíduos selecionados. Cada novo valor gerado é um sorteio aleatório de um número real, onde os limites inferior e superior são os genes dos pais. O cruzamento aritmético tem um comportamento de Contração.

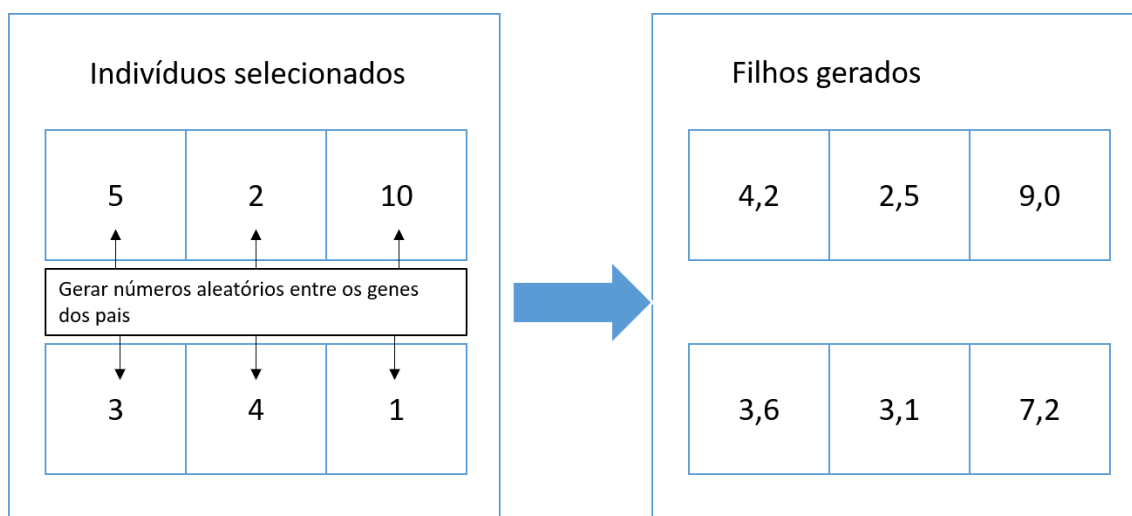


Figura 5 Cruzamento Aritmético Clássico (Fonte: o autor)

2.3 Exploração e Intensificação e Níveis de Diversidade

Balancar o poder exploratório global e local de um algoritmo evolutivo, não é trivial, diversos fatores influenciam no comportamento do algoritmo para determinado tipo de problema, parâmetros a respeito do algoritmo (e. g. tamanho da população, número de iterações, etc.) têm influência direta sobre o resultado da otimização. Atualmente, são feitos estudos em algoritmos evolucionários com diferentes abordagens, onde é verificado o nível de exploração global e local aliado a parametrização dos algoritmos propostos [20].

Em AGs, parâmetros como o número de indivíduos em uma população, quantidade de gerações, operações de seleção, cruzamento e mutação afetam o comportamento do algoritmo. A diversidade da população é um fator fundamental para que o AG consiga varrer o espaço de busca de maneira efetiva, tanto no âmbito global quanto local. Uma taxa de diversidade alta, é uma indicação positiva que o algoritmo consiga espalhar a população por todo espaço de busca do problema, evitando a convergência prematura da população e consequentemente caindo em um ótimo local [23].

A diversidade (SPD – *Standart Population Diversity*) descreve o nível de variação que uma determinada população possui. Através da análise da distância euclidiana de cada indivíduo do AG em relação a um resultado médio da população no espaço de busca, é possível inferir um determinado nível de diversidade. Quanto maior o nível, melhor a variabilidade da população [10][11].

No cálculo da SPD, primeiramente, calcula-se a média geral do valor de cada gene que compõe um indivíduo. Depois, é calculada a soma da distância euclidiana de cada indivíduo para a média geral.

Considerando uma população com P indivíduos (G_1, G_2, \dots, G_p), onde cada indivíduo G tem N genes, sendo então $G_i = (G_{i,1}, G_{i,2}, \dots, G_{i,N})$. A média geral de cada gene N é dado por (Equação 2):

$$G_N^{ave} = \frac{1}{P} \sum_{i=1}^P G_{i,N} \quad (\text{Equação 2})$$

Após isso, é calculada a somatória da distância euclidiana de cada gene N do indivíduo i em relação à média geral G_N^{ave} (Equação 3):

$$SPD_i = \sqrt{\sum_{n=1}^N (G_{i,N} - G_N^{ave})^2} \quad (\text{Equação 3})$$

Como o SPD_i pode ter seus valores muito variados dependendo do tipo e dimensão do problema a ser utilizado pelo GA, é aplicada uma normalização calculando-se o desvio padrão de cada gene N em relação à população P (Equação 4):

$$\sigma(G_N^{ave}) = \sqrt{\frac{1}{P} \sum_{i=1}^P (G_{i,N} - G_N^{ave})^2} \quad (\text{Equação 4})$$

Por fim, calcula-se o coeficiente de variação C_v da população em relação ao indivíduo médio. G^{ave} . Este coeficiente indica a variabilidade da população em cada geração do AG (Equação 5):

$$SPD = C_v(G^{ave}) = \frac{1}{N} \sum_{j=1}^N \left(\frac{\sigma(G_j^{ave})}{G_j^{ave}} \right) \quad (\text{Equação 5})$$

É importante lembrar que a capacidade de otimização não está diretamente ligada a um alto valor de SPD. O nível de diversidade alta no SPD é apenas um indicativo de que a população não está aglomerada em uma determinada área do espaço de busca e/ou presa em um ótimo local. É normal os níveis de SPD irem diminuindo conforme o avanço das gerações do AG, visto que a população converge para uma determinada área ótima, e conseqüentemente seus indivíduos irão possuir genes semelhantes.

Para esta pesquisa, o SPD é utilizado para medir a capacidade exploratória global do algoritmo no espaço de busca. O poder de intensificação (otimização local) é medido geralmente pelo melhor resultado médio (melhor fitness) dentro de um determinado número de execuções do AG.

Capítulo 3. Os Algoritmos Propostos

O capítulo a seguir irá mostrar as ideias principais do algoritmo proposto, com o novo método de cruzamento, chamado de ADX, e o novo método de mutação, chamado de CAM.

3.1 Cruzamento Ajustado (ADX)

Nos últimos anos, pesquisas envolvendo métodos de cruzamentos em algoritmos genéticos com números reais têm aumentado. Operadores de cruzamento (crossover) são essenciais para o AG realizar a varredura no espaço de busca. Métodos que trabalham com geração de números reais (não estacionários) aparentam ser mais promissores em relação aos métodos geralmente binários (estacionários). No entanto, ainda é preciso um estudo aprofundado nos níveis de exploração e intensificação nestes trabalhos [27][38][39][40].

O cruzamento ajustado (ADX) é proposto para algoritmos que utilizam codificação com números reais. Este operador realiza um ajuste nos genes dos pais, subtraindo ou somando valores normalizados através de um peso previamente determinado.

Diferente de outros cruzamentos que possuem uma certa aleatoriedade, como o binário que faz a troca de genes selecionando posições aleatórias ou o aritmético que gera um valor aleatório entre os limites dos genes dos pais, o cruzamento proposto sempre gera os mesmos filhos caso seja usado os mesmos progenitores. A geração de um descendente diferente de dois pais previamente utilizados, é feita através da mudança no valor do peso e/ou do vetor gradiente gerado pelo algoritmo, caracterizando uma busca semi guiada utilizando informações de todos os indivíduos selecionados.

O objetivo do cruzamento é aumentar a intensificação do algoritmo genético através do refinamento de resultados previamente conhecidos (progenitores), gerando filhos muito semelhantes aos pais, no entanto com pequenos ajustes nos genes. Este cruzamento pode ter um comportamento de contração ou de dispersão dependendo do vetor gradiente que será explicado a seguir.

Após a fase de seleção dos indivíduos que irão participar do cruzamento no algoritmo genético, o ADX cria um vetor gradiente $\nabla \vec{G}$ baseado nos indivíduos

ordenados ascendentemente conforme o valor de aptidão (*fitness*), ou seja, o algoritmo utiliza informação de parte da população para criar um vetor direcionado para cada gene. Tal vetor irá ter o mesmo tamanho da quantidade de genes presentes em um indivíduo, e cada espaço deste vetor irá ser preenchido com o valor 1 ou -1. Este valor é determinado após a seguinte análise: “Se determinado gene i dos indivíduos tende a aumentar de valor conforme o valor de aptidão melhora, então atribuir o valor 1 para o espaço $\nabla\vec{G}[i]$, ou -1 caso o gene i tenda a sofrer uma diminuição de valor conforme o valor de aptidão melhora”.

Após a geração do vetor gradiente, o ADX será realizado da seguinte maneira: dado um progenitor com os genes $P_1 = \{x_1, x_2, \dots, x_i\}$ e outro progenitor com $P_2 = \{y_1, y_2, \dots, y_i\}$, os descendentes criados $X_1 = \{g(1), g(2), \dots, g(i)\}$ e $X_2 = \{h(1), h(2), \dots, h(i)\}$ serão resultados da seguinte (Equação 6):

$$f(x_i, y_i) = (x_i - y_i) \times \alpha$$

$$g(i) = \begin{cases} x_i + f(x_i, y_i), & \nabla\vec{G}[i] = 1 \\ x_i - f(x_i, y_i), & \nabla\vec{G}[i] = -1 \end{cases} \quad (\text{Equação 6})$$

$$h(i) = \begin{cases} y_i + f(x_i, y_i), & \nabla\vec{G}[i] = 1 \\ y_i - f(x_i, y_i), & \nabla\vec{G}[i] = -1 \end{cases}$$

A variável α é um peso definido empiricamente e pode assumir valores reais entre 0 e 1. Os novos genes gerados g e h são resultado dos genes dos pais multiplicados pelo peso, e depois somado ou subtraído ao mesmo gene paterno baseado no vetor gradiente. De acordo com o nível de perturbação, os filhos gerados apresentam valores próximos aos genes dos pais, caracterizando uma otimização local. O principal papel deste cruzamento é realizar pequenas perturbações nos genes dos indivíduos do cruzamento, para “arrastá-los” pelo espaço de busca do problema evitando a quebra de restrições por mudanças bruscas nos valores dos genes.

Supondo um cenário onde do pior para o melhor indivíduo, os valores dos dois primeiros genes tendem a diminuir, e o valor de terceiro gene tende a crescer. Denota-se então que o vetor gradiente assume valores $[-1, -1, +1]$. A Figura 6, demonstra um exemplo do cruzamento ADX em ação no cenário descrito:

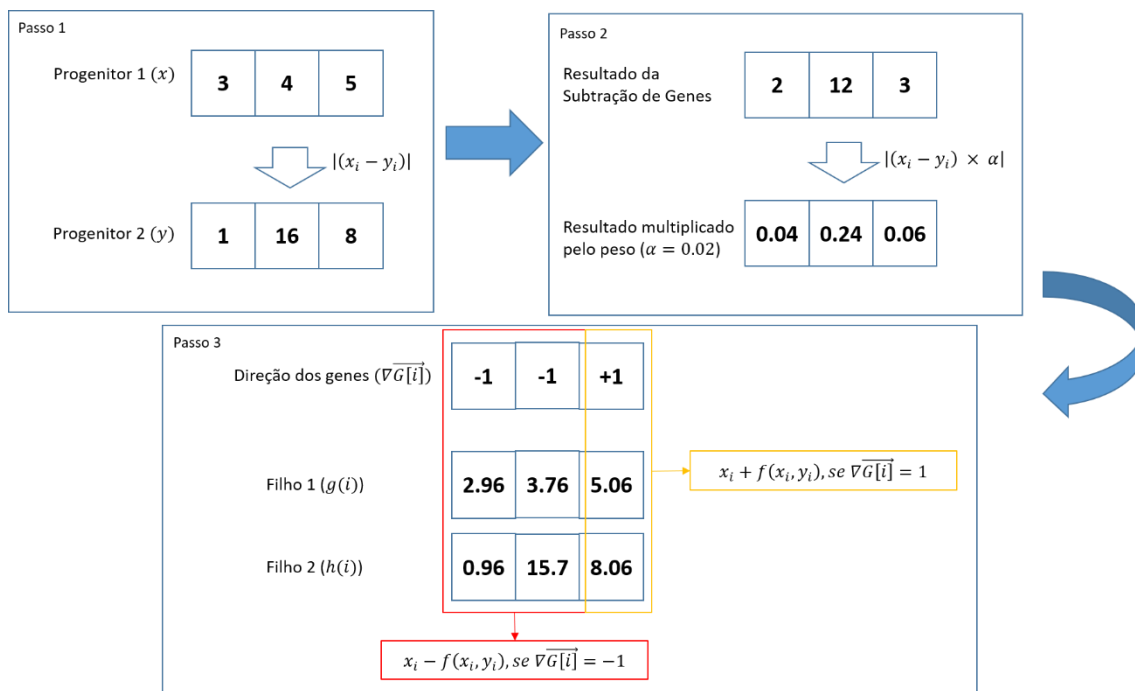


Figura 6 Demonstração do ADX utilizando dois progenitores, pode-se perceber que o crossover sempre gera filhos com os genes paternos levemente modificados de acordo com o peso (Fonte: o autor)

3.2 Mutação de Ajuste Correlacional (CAM)

Para manter alta a diversidade da população e aumento do potencial de exploração global, o método proposto a seguir realiza um ajuste em grande parte dos indivíduos baseado na análise de correlação dos genes de toda a população.

Após o crossover do algoritmo genético, tradicionalmente é realizado o processo de mutação e determinado o fim de uma geração do algoritmo, descartando-se os indivíduos menos aptos e preservando os mais aptos (elitismo) para a próxima geração. Buscando o aproveitamento de dados dos indivíduos menos aptos, descreve-se aqui o algoritmo CAM.

A Mutação de Ajuste Correlacional (CAM) é um algoritmo de ajuste estatístico utilizado na fase final de uma geração do algoritmo genético, substituindo a fase convencional de mutação. Ele utiliza dados de toda a população para verificar coeficientes de ajuste dos genes (variáveis) de uma maneira geral, e através de uma escolha de um subgrupo de indivíduos, tenta ajustar os genes de cada escolhido, tentando melhorar o fitness de cada indivíduo ajustado.

O CAM utiliza o teste de correlação do coeficiente tau de Kendall [41][42], que avalia o grau de similaridade entre dois conjuntos de dados. Os coeficientes de Kendall

podem variar com valores contínuos de $[-1, 1]$. Um valor positivo indica associação positiva. Um valor negativo indica associação negativa. Quanto maior a magnitude, mais forte será a associação.

Em outras palavras, o coeficiente de correlação verifica se determinado valor de uma variável irá aumentar se outra variável diminuir (relação discordante), ou se irá aumentar quando juntamente a outra variável aumentar (relação concordante). Por exemplo, dados um conjunto n de pares $(x_1, y_1), (x_2, y_2) \dots (x_n, y_n)$. A equação do coeficiente tau de Kendall é dado por (Equação 7):

$$\tau = \frac{(Z) - (Y)}{\frac{n(n-1)}{2}} \quad (\text{Equação 7})$$

Sendo:

- Z: Número de pares concordantes (quando $x_i > x_j$ e $y_i > y_j$ ou $x_i < x_j$ e $y_i < y_j$)
- Y: Número de pares discordantes (quando $x_i > x_j$ e $y_i < y_j$ ou $x_i < x_j$ e $y_i > y_j$)

Quanto mais próximo de 1 ou -1 o resultado, respectivamente significa que os valores têm forte associação positiva (se aumentar o valor de x, o valor de y também irá aumentar e vice-versa) ou negativa (se aumentar o valor de x, o valor de y irá diminuir e vice-versa).

Após a finalização do processo de crossover, o CAM irá montar uma matriz $N \times (M + 1)$ com os indivíduos da população utilizada durante a atual geração, considerando N como linhas e M como colunas [Tabela 1]:

- N é o número de indivíduos
- M é o número de genes (variáveis) que compõe um indivíduo
- A última coluna da matriz irá conter o valor de aptidão correspondente a cada indivíduo em N

Tabela 1 Matriz que o algoritmo irá construir com os indivíduos da população no final de cada geração

	Gene 1	Gene 2	...	Gene m	Fitness $F(x)$
Indivíduo 1	$G(1,1)$	$G(1,2)$...	$G(1,m)$	$F(g11,g12..g1m)$

Indivíduo 2	$G(2,1)$	$G(2,2)$...	$G(2,m)$	$F(g21,g22..g2m)$
...
Indivíduo n	$G(n,1)$	$G(n,2)$...	$G(n,m)$	$F(gn1,gn2..gnm)$

Após a criação da matriz, o CAM irá aplicar o teste de correlação de Kendall entre as colunas Gene $\{1, 2, \dots, m\}$ em relação aos dados da coluna *Fitness*. Com isso é possível mensurar se a variável irá afetar o *fitness* de maneira concordante (+) ou discordante (-).

Sendo c (coeficiente de precisão) e k (número de iterações) valores inteiros manualmente escolhidos na implementação. O CAM irá tentar ajustar os genes de um indivíduo, sendo que para cada indivíduo i escolhido tem-se (Equação 8):

$$\sum_{i=1}^m \sum_{0}^k G(i, m) = G(i, m) + G(i, m) \times \tau_m \times c \quad (\text{Equação 8})$$

Onde:

- $G(i,m)$: célula da matriz que contém o valor do gene m do indivíduo i
- c : coeficiente de precisão
- m : índice do Gene, segundo a [Tabela 1]
- k : número de iterações sobre um gene do indivíduo
- τ_m : resultado do teste de Kendall do gene m em relação ao *fitness*

Através deste ajuste em um âmbito geral da população, é possível “arrastá-los” pelo espaço de busca, evitando que os indivíduos fiquem presos em ótimos locais, aumentando a capacidade de busca global da otimização. Um exemplo pode ser visto na Figura 7, primeiramente é aplicado o teste de Kendall para todos os genes em relação ao *Fitness*, desta forma cada gene irá possuir um valor do teste (exceto o próprio *fitness*). Em seguida, a (Equação 8) é aplicada para cada gene dos indivíduos selecionados (no caso da figura, ilustra a operação para o gene 1 do indivíduo 1). Finalmente, os indivíduos ajustados são repassados para a próxima geração. A Figura 8 ilustra o comportamento destes indivíduos que são afetados pelo CAM.

Este método permite maior diversidade e exploração global do espaço de busca, visto que permite determinados indivíduos “explorar” novas áreas em um sentido mais

direcionado, ao invés de totalmente aleatório como uma mutação clássica. Particularmente, este método tem um bom resultado em problemas com restrições em variáveis, pois permite ajudar indivíduos de forma direcionada ao local ótimo e prevenindo a “quebra” das equações restritivas dos problemas de otimização.

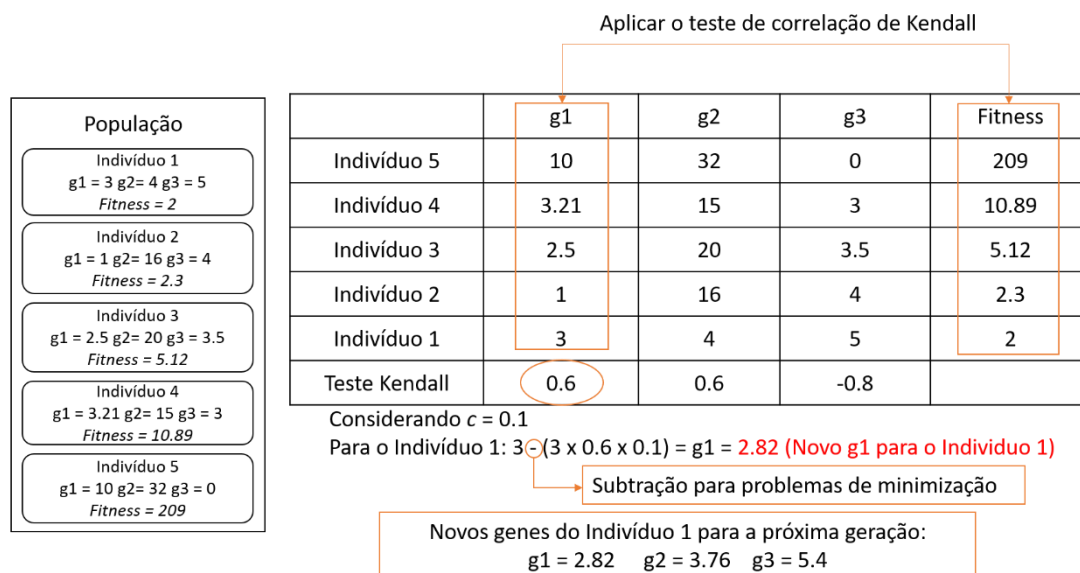


Figura 7 Exemplo da aplicação do CAM em um problema de minimização utilizando uma determinada população do AG (Fonte: o autor)

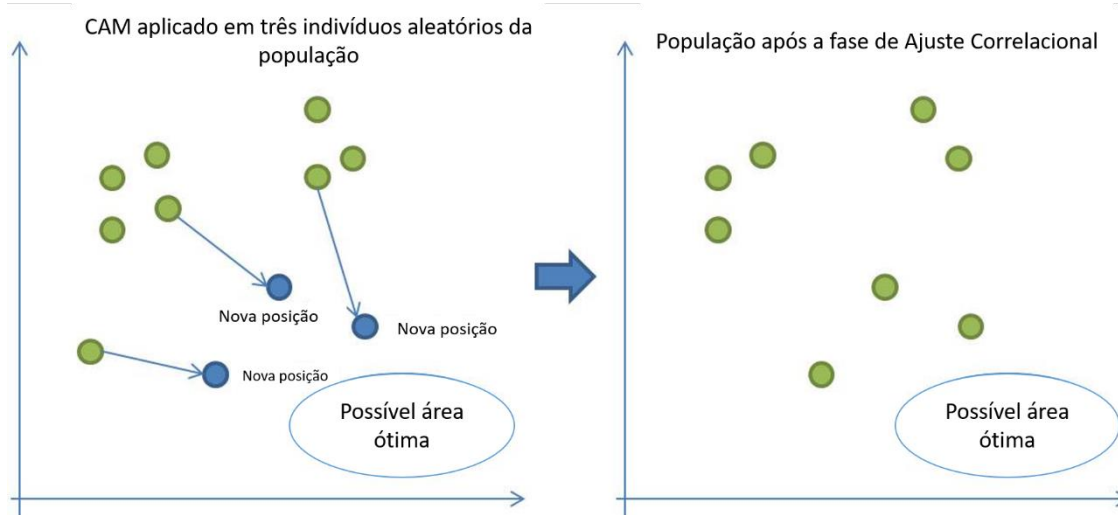


Figura 8 Comportamento dos indivíduos afetados pelo método de ajuste proposto considerando um problema de minimização (Fonte: o autor)

3.3 Custo Computacional dos Métodos Propostos

Para calcular o custo computacional de um algoritmo genético é preciso ter alguns parâmetros iniciais: o tempo de criação da população (T_p), tempo de cálculo de fitness de cada indivíduo (T_f), tempo de execução do operador de seleção (T_s), tempo de

execução do operador de cruzamento (T_c) e o tempo de execução do operador de mutação (T_m). Para realizar o cálculo do custo computacional do algoritmo genético proposto, será levando em consideração que T_p e T_s são os mesmos para todos os problemas testados, e T_f é variável de acordo com o problema otimizado.

O tempo T_c depende do operador de crossover utilizado. Considerando o SBX e o ARX citados anteriormente, os dois operadores utilizam sempre dois pais para criar dois filhos. Particularmente, o SBX realiza trocas de genes entre os pais e o ARX cria um valor aleatório real entre os limites maior e menor de genes dos pais. O tempo da realização de uma troca no SBX e da geração de números pelo ARX podemos denotar como T_{sc} . Cada T_{sc} ocorre n vezes durante g quantidade de gerações, onde P_n é o número de cada par selecionado para o cruzamento e G_n é o número de gerações do algoritmo genético. O custo computacional C_c de SBX e ARX pode ser dado pela (Equação 9):

$$C_c = G_n \times P_n \times T_{sc} \quad (\text{Equação 9})$$

O cruzamento ADX proposto também cria valores de acordo com um par de genes dos pais (T_{sc}), porém antes da criação destes valores, o operador inicializa e cria o vetor gradiente baseado em todos os indivíduos selecionados. Com isso, para o ADX devemos considerar que é preciso um tempo adicional T_i para a criação do vetor antes da fase efetiva de cruzamento. Portanto, o custo computacional do ADX pode ser denotado por $T_i + T_{sc}$.

Para cada geração G_n do algoritmo, o ADX usa $G_n \times T_i$ mais tempo em comparação com os cruzamentos SBX e ARX, devido a inicialização do vetor gradiente em toda etapa de cruzamento. Finalmente, o custo computacional do ADX pode ser dado como (Equação 10):

$$C_c = G_n [T_i + (P_n \times T_{sc})] \quad (\text{Equação 10})$$

A mutação clássica faz uma verificação aleatória em qual gene de um indivíduo pode sofrer uma mudança abrupta de valor antes de passar para a próxima geração. Para cada indivíduo i da população, existe uma chance r do gene sofrer uma mutação (T_m), onde r é geralmente um valor entre 0,01 (1%) a 0,05 (5%). O custo computacional de uma mutação clássica pode ser dado por (Equação 11):

$$Cc = Gn(i \times r \times Tm) \quad (\text{Equação 11})$$

Diferente da mutação clássica, o ajuste correlacional (CAM) precisa montar uma matriz $i \times j$ de correlação antes da etapa de mutação, onde i é o número de indivíduos selecionados para a próxima geração e j é o número de genes de um indivíduo. O tempo de criação da matriz pode ser dado por Tcm e após a sua criação, o algoritmo irá alterar o valor de todos os genes j de alguns indivíduos da população. Se for considerado Tm o tempo para alterar o valor de um gene, na fase de aplicação do CAM pode-se denotar que $j \times Tm$ é o tempo para mudar todos os genes de apenas um indivíduo. O custo computacional do CAM, pode ser mostrado por (Equação 12):

$$Cc = Gn[Tcm + (i \times j \times Tm)] \quad (\text{Equação 12})$$

Analisando visualmente as equações de custo computacional de ADX em relação aos cruzamentos SBX e ARX, e a mutação de ajuste correlacional em relação à mutação clássica tradicional. É possível perceber que ambos os operadores propostos no trabalho adicionam mais variáveis e complexidade a execução do algoritmo genético.

Apesar do aumento significativo na complexidade do algoritmo, ambos os métodos apresentaram melhores resultados em diversos problemas quando comparado ao SBX, ARX e mutação tradicional como será mostrado nos capítulos seguintes.

Capítulo 4. Problemas Utilizados e Resultados Obtidos

Neste capítulo serão apresentados todos os resultados obtidos do AG com ADX+CAM empregado em diversos problemas presentes na literatura, além de comparar com outros algoritmos evolutivos propostos pela comunidade científica.

4.1 Problemas computacionais do CEC'2015 (IEEE *Congress on Evolutionary Computation*)

O CEC (Congress on Evolutionary Computation) da IEEE (Institute of Electrical and Electronics Engineers) está entre os eventos mais famosos dentro da área de otimização utilizando algoritmos evolucionários. Nesse evento científico é comum a realização de um campeonato para encontrar o melhor algoritmo na solução de problemas de otimização. Para a sua realização, os organizadores propõem problemas de benchmark clássicos com algumas alterações e restrições de parametrização dos algoritmos participantes. Esta pesquisa utilizou problemas da edição de 2015 (CEC'2015) do evento, consistindo em 15 problemas (F1 a F15), separados nas seguintes categorias de funções [44]:

- Unimodais: F1 e F2
- Multimodais simples: F3, F4 e F5
- Híbridas: F6, F7, F8
- Compostas: F9 a F15

Para realização dos testes, o algoritmo não pode ultrapassar 500 verificações de função para $10D$ e 1500 verificações para $30D$, onde D é a dimensão do vetor de cada possível solução do problema (indivíduo). Cada indivíduo inicia com valores reais aleatórios com limites $[-100,100]$. Nos testes com $10D$ foi utilizado uma população com 5 indivíduos e 100 gerações, enquanto para $30D$ foram utilizados 6 indivíduos com 250 gerações. Apenas 1 indivíduo da população (elite) de ambas as configurações perpetua para gerações futuras, o peso do ADX e o coeficiente do CAM foram testados com valor 0.01, empiricamente atribuído.

Os resultados para $10D$ e $30D$ utilizando o algoritmo proposto pode ser visto na Tabela 2 e Tabela 3 respectivamente.

Tabela 2 Resultados do ADX+CAM com 10 dimensões

10D	Função	Melhor	Pior	Mediana	Média	D. Padrão
Unimodal	<i>F1</i>	1,16E+09	7,80E+09	4,78E+09	4,63E+09	1,98E+09
	<i>F2</i>	1,73E+04	1,48E+05	6,03E+04	7,19E+04	3,43E+04
Multimodal Simples	<i>F3</i>	3,08E+02	3,13E+02	3,11E+02	3,11E+02	1,66E+00
	<i>F4</i>	1,37E+03	2,75E+03	1,95E+03	1,97E+03	3,70E+02
	<i>F5</i>	5,01E+02	5,04E+02	5,03E+02	5,03E+02	9,00E-01
	<i>F6</i>	6,01E+02	6,04E+02	6,03E+02	6,03E+02	7,57E-01
	<i>F7</i>	7,05E+02	7,38E+02	7,24E+02	7,23E+02	9,20E+00
	<i>F8</i>	8,81E+02	5,45E+04	4,34E+03	1,42E+04	1,74E+04
	<i>F9</i>	9,03E+02	9,05E+02	9,04E+02	9,04E+02	3,07E-01
Híbrido	<i>F10</i>	4,49E+04	1,71E+07	1,16E+06	2,21E+06	3,80E+06
	<i>F11</i>	1,11E+03	1,17E+03	1,12E+03	1,13E+03	1,90E+01
	<i>F12</i>	1,27E+03	2,76E+03	1,56E+03	1,60E+03	2,97E+02
Composto	<i>F13</i>	1,65E+03	2,25E+03	1,73E+03	1,83E+03	1,92E+02
	<i>F14</i>	1,60E+03	1,64E+03	1,61E+03	1,62E+03	1,16E+01
	<i>F15</i>	1,92E+03	2,17E+03	2,02E+03	2,02E+03	7,83E+01

Tabela 3 Resultados do ADX+CAM com 30 dimensões

30D	Função	Melhor	Pior	Mediana	Média	D. Padrão
Unimodal	<i>F1</i>	1,68E+10	3,28E+10	2,22E+10	2,31E+10	4,21E+09
	<i>F2</i>	7,02E+04	4,64E+05	1,90E+05	2,10E+05	9,65E+04
Multimodal Simples	<i>F3</i>	3,31E+02	3,48E+02	3,39E+02	3,39E+02	3,93E+00
	<i>F4</i>	4,23E+03	6,60E+03	5,29E+03	5,40E+03	6,31E+02
	<i>F5</i>	5,02E+02	5,06E+02	5,04E+02	5,04E+02	9,67E-01
	<i>F6</i>	6,03E+02	6,04E+02	6,04E+02	6,04E+02	3,64E-01
	<i>F7</i>	7,24E+02	7,73E+02	7,41E+02	7,42E+02	1,22E+01
	<i>F8</i>	1,02E+04	2,14E+06	2,62E+05	3,84E+05	4,74E+05
	<i>F9</i>	9,13E+02	9,14E+02	9,14E+02	9,14E+02	2,17E-01
Híbrido	<i>F10</i>	4,54E+06	1,66E+08	2,00E+07	3,36E+07	4,22E+07
	<i>F11</i>	1,14E+03	1,37E+03	1,21E+03	1,22E+03	6,76E+01
	<i>F12</i>	1,81E+03	3,42E+03	2,40E+03	2,44E+03	4,58E+02
Composto	<i>F13</i>	1,65E+03	2,25E+03	1,73E+03	1,83E+03	1,92E+02
	<i>F14</i>	1,60E+03	1,64E+03	1,61E+03	1,62E+03	1,16E+01
	<i>F15</i>	1,92E+03	2,17E+03	2,02E+03	2,02E+03	7,83E+01

As tabelas seguintes mostram o desempenho do algoritmo em relação aos cinco melhores algoritmos da competição, os melhores resultados estão em negrito.

A Tabela 4 mostra resultados para a F1 e F2, o algoritmo proposto alcançou melhores resultados que o *humanCog* [45] nas configurações F1-30D e F2-10D.

Tabela 4 Resultado comparativo das funções unimodais (F1 e F2)

	Funções Unimodais			
	F1		F2	
	10D	30D	10D	30D
MVMO [47]	1,93E+02	2,09E+03	1,68E-02	6,93E-03
TunedCMAES [48]	1,17E+06	1,52E+06	4,78E+04	1,44E+05
CMAS-ES_QR [44]	4,43E+06	8,50E+05	2,58E+04	9,17E+04
iSRPSO [49]	7,40E+06	7,19E+08	3,19E+04	7,67E+04
<i>humanCog</i> [45]	3,27E+09	4,74E+10	7,80E+04	1,13E+05
ADX+CAM	4,63E+09	2,31E+10	7,19E+04	2,10E+05

Na categoria de multimodal simples (Tabela 5), o ADX+CAM obteve melhor resultado que o *humanCog* na F4-10D e F8-30D. Mostrou ter bom desempenho na F4-30D ficando em segundo lugar. Para os problemas F3, F5, F6, F7, F8-10D e F9, o ADX+CAM mostrou piores resultados em relação aos outros participantes.

Tabela 5 Resultado comparativo das funções multimodais simples (F3 a F9)

	Funções Multimodais Simples							
	F3		F4		F5			
	10D	30D	10D	30D	10D	30D		
MVMO [47]	9,40E+00	3,79E+01	4,65E+02	1,43E+03	1,13E+00	1,68E+00		
TunedCMAES [48]	7,62E+00	2,43E+01	1,34E+03	6,11E+03	2,77E+00	3,13E+00		
CMAS-ES_QR [44]	2,79E+00	1,15E+01	1,73E+03	6,68E+03	3,20E+00	4,55E+00		
iSRPSO [49]	6,60E+00	2,57E+01	9,25E+02	5,41E+03	2,46E+00	4,24E+00		
<i>humanCog</i> [45]	1,12E+01	4,13E+01	2,09E+03	7,99E+03	2,82E+00	4,39E+00		
ADX+CAM	3,11E+02	3,39E+02	1,97E+03	5,40E+03	5,03E+02	5,04E+02		
	F6		F7		F8		F9	
	10D	30D	10D	30D	10D	30D	10D	30D
MVMO [47]	3,26E-01	5,20E-01	6,37E-01	4,39E-01	4,14E+01	4,03E+02	4,01E+00	1,34E+01
TunedCMAES [48]	6,00E-01	7,16E-01	6,31E-01	7,28E-01	3,68E+01	2,84E+01	4,17E+00	1,39E+01
CMAS-ES_QR [44]	4,17E-01	7,28E-01	5,52E-01	7,47E-01	4,68E+00	1,74E+01	3,96E+00	1,34E+01
iSRPSO [49]	5,29E-01	6,35E-01	5,71E-01	5,68E-01	5,03E+00	6,26E+02	3,95E+00	1,36E+01
<i>humanCog</i> [45]	3,63E+00	5,03E+00	2,74E+01	8,86E+01	7,77E+03	5,24E+06	4,16E+00	1,39E+01
ADX+CAM	6,03E+02	6,04E+02	7,23E+02	7,42E+02	1,42E+04	3,84E+05	9,04E+02	9,14E+02

Nas funções Híbridas (Tabela 6) e Compostas (Tabela 7), o ADX+CAM obteve melhoras apenas na função F10-30D em relação ao *humanCog*. Para todas as outras

configurações, o algoritmo proposto obteve resultados piores. Como são funções mais complexas comparado com as anteriores, o ADX+CAM provavelmente não teve gerações suficientes para realizar ajustes significativos nos genes da população, levando o algoritmo a ficar restrito a um ótimo local.

Tabela 6 Resultado comparativo das funções híbridas (F10, F11 e F12)

	Funções Híbridas					
	F10		F11		F12	
	10D	30D	10D	30D	10D	30D
MVMO [47]	4,97E+02	9,29E+04	1,17E+01	1,43E+02	2,00E+02	8,60E+02
TunedCMAES [48]	5,38E+05	4,89E+06	7,45E+00	2,11E+01	2,39E+02	7,66E+02
CMAS-ES_QR [44]	2,25E+05	3,25E+06	7,63E+00	2,46E+01	2,35E+02	6,27E+02
iSRPSO [49]	3,53E+05	6,83E+06	7,26E+00	5,09E+01	1,82E+02	7,36E+02
humanCog [45]	1,19E+06	5,60E+07	2,16E+01	2,76E+02	3,08E+02	1,60E+03
ADX+CAM	2,21E+06	3,36E+07	1,13E+03	1,22E+03	1,60E+03	2,44E+03

Tabela 7 Resultado comparativo das funções compostas (F13, F14 e F15)

	Funções Compostas					
	F13		F14		F15	
	10D	30D	10D	30D	10D	30D
MVMO [47]	3,16E+02	3,44E+02	2,06E+02	2,76E+02	4,76E+02	1,19E+03
TunedCMAES [48]	3,47E+02	4,15E+02	2,05E+02	2,47E+02	4,42E+02	8,01E+02
CMAS-ES_QR [44]	3,26E+02	3,80E+02	1,97E+02	2,35E+02	3,79E+02	4,90E+02
iSRPSO [49]	3,31E+02	4,00E+02	2,01E+02	2,65E+02	3,00E+02	9,51E+02
humanCog [45]	4,33E+02	8,35E+02	2,15E+02	3,94E+02	4,74E+02	1,49E+03
ADX+CAM	1,83E+03	1,83E+03	1,62E+03	1,62E+03	2,02E+03	2,02E+03

O baixo número de verificações de função permitidas, prejudicou a eficácia do CAM, visto que ele trabalha analisando a população geral. Com um número reduzido de indivíduos, a amostra populacional não mostrou ser relevante para a atuação do ajuste correlacional.

Para verificar o aumento de diversidade que o CAM provê ao algoritmo nos problemas do CEC'2015, foram mapeados os dois primeiros genes ($x1$ e $x2$) da população dos problemas F1-30D e F14-30D durante a execução do ADX e também do ARX. Para ambos, a Figura 9 e a Figura 10 mostram o “caminhamento” da população pelo espaço de busca do problema conforme o avanço das gerações, sendo a cor mais escura as gerações iniciais, e gradualmente clareando conforme o tempo de execução. Nos gráficos na parte superior usando CAM, é possível notar que com a utilização do

ajuste correlacional a população fica mais diversificada, varrendo o espaço de busca com mais efetividade em relação aos gráficos inferiores. Além disso, comparando o cruzamento ADX com o ARX, nos gráficos do lado esquerdo (ADX) é possível notar também que a população fica mais distribuída no espaço em relação aos gráficos da parte direita (ARX).

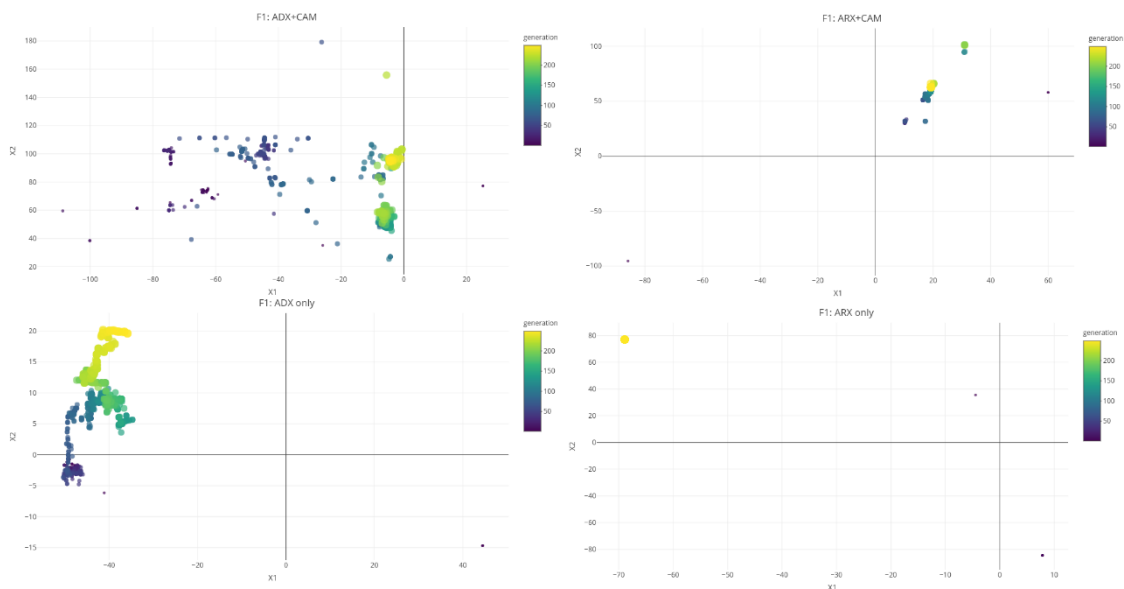


Figura 9 Distribuição da população pelo espaço de busca do F1-30D conforme o avanço das gerações. Gráficos à esquerda são do ADX e da direita, do ARX (Fonte: o autor)

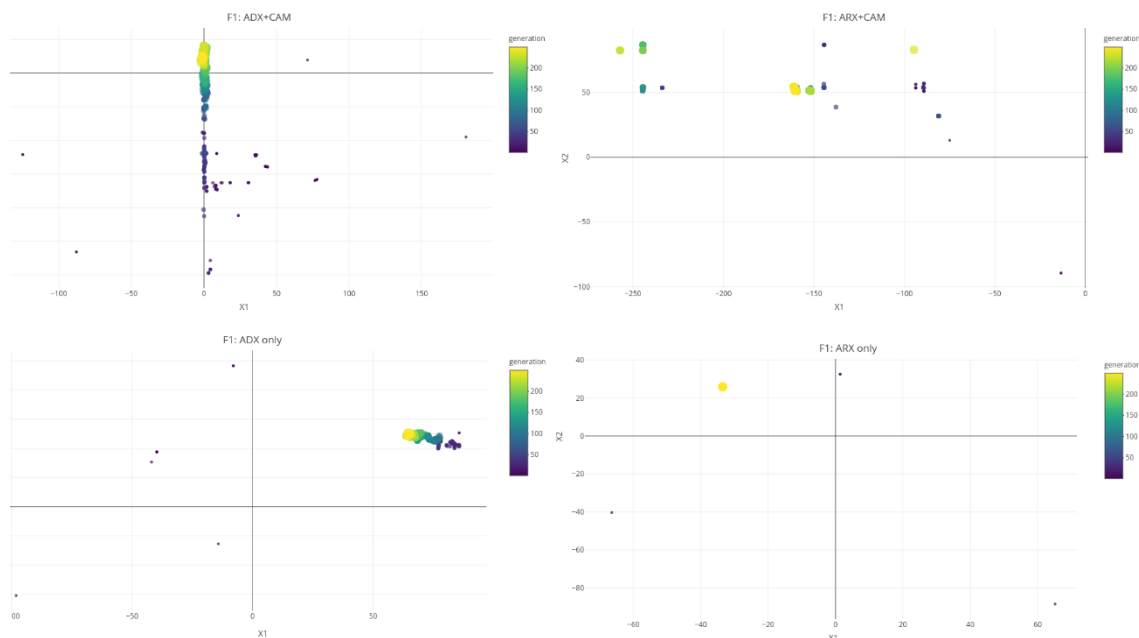


Figura 10 Distribuição da população pelo espaço de busca do F14-30D conforme o avanço das gerações. Gráficos à esquerda são do ADX e da direita, do ARX (Fonte: o autor)

4.2 Cinemática Inversa em Robótica

A Cinemática Inversa (CI) é uma área de bastante interesse em manipulação robótica. Basicamente, é o estudo no qual se busca otimizar o arranjo de conectores que compõe as peças de um manipulador robótico (ex: braço mecânico, alavancas, etc). Muitos dos problemas de CI possuem alta complexidade, são não-lineares, possuem muitas variáveis e são multi-soluções. Existem muitos trabalhos em análise e geração de trajetórias ponto-a-ponto de manipuladores e segundo [50][51], manipuladores não-redundantes com movimentação contínua receberam pouca atenção dentre os pesquisadores na área de CI.

Para analisar o comportamento do AG proposto, foi realizado testes com o trabalho de [50][51], onde o ADX+CAM é aplicado em um problema de otimização de um manipulador não-redundante com movimentação contínua. Basicamente, o objetivo do algoritmo evolutivo é gerar valores de uma matriz de coordenadas que se aproximem ao máximo de um caminho previamente definido.

4.2.1 Formulação do Problema

Considerando um robô com M graus de liberdade e N pontos nas coordenadas espaciais, dado um caminho desejado (Pdc). O objetivo é achar um caminho gerado (Pgc) onde o desvio padrão cumulativo entre Pdc e Pgc seja mínimo. Nk é o número de conectores i (pontos) que especificam a acurácia em relação ao caminho desejado Pdc . Ambos Pdc e Pgc são matrizes de $N \times Nk$ onde Pgc é calculado de uma matriz $P\theta$ formada por valores gerados pelo AG de tamanho $M \times Nk$. O desvio padrão $E(i)$ de qualquer ponto i é dado pela (Equação 13):

$$E(i) = \sum_{k=1}^N |Pdc(k, i) - Pgc(k, i)| \quad (\text{Equação 13})$$

Então, o desvio padrão cumulativo E (função fitness) entre o caminho desejado Pdc e o caminho gerado Pgc é dado pela (Equação 14):

$$E = \sum_{i=1}^{Nk} \sum_{k=1}^N |Pdc(k, i) - Pgc(k, i)| \quad (\text{Equação 14})$$

Para este determinado problema, o caminho desejado foi definido com 20 pontos/conectores ($Nk = 20$), mostrados na (Equação 15) e na Figura 11:

$$Pdc(1, i) = Xdc(i) = x_{inicial} + \frac{x_{final} + x_{inicial}}{Nk - 1} \times (i - 1)$$

$$Pdc(2, i) = Ydc(i) = 0.25 \quad (\text{Equação 15})$$

$$x_{inicial} = 0 ; x_{final} = 0.25$$

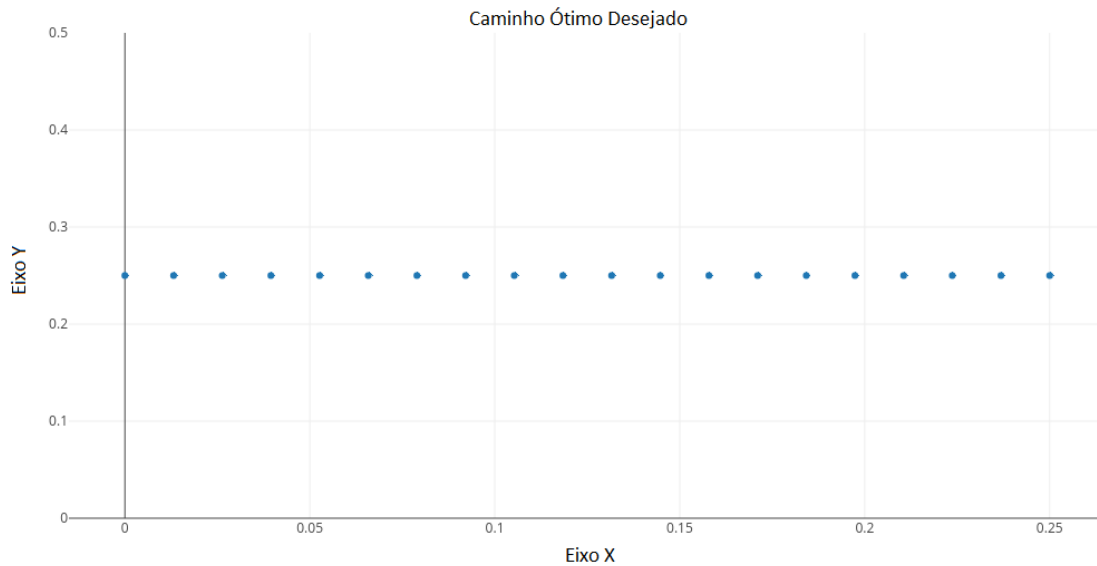


Figura 11 Pontos do Caminho Ótimo desejado Pdc (Fonte: o autor)

O modelo cinemático do manipulador que irá gerar o Pgc é dado pela (Equação 16):

$$Xgc(i) = \sum_{j=1}^M \left(L_j \times \cos \left[\sum_{k=1}^j \theta_{k,i} \right] \right)$$

(Equação 16)

$$Ygc(i) = \sum_{j=1}^M \left(L_j \times \sin \left[\sum_{k=1}^j \theta_{k,i} \right] \right)$$

Os parâmetros para o manipulador ficaram definidos como:

- $L1 = L2 = L3 = 0.5$
- $N = 2$
- $M = 3$
- $-180 \leq \theta(h) \leq 180$ para $h = 1, 2, 3$
- $Nk = 20$
- $1 \leq i \leq Nk$

4.2.2 Resultados

Para realização dos testes, foram realizadas 120 execuções do algoritmo genético, sendo divididos em 20 execuções para cada operador de crossover (ADX, SBX e ARX) com o CAM e 20 execuções para cada um sem o CAM. Cada possível solução do AG é uma matriz de 3 linhas e 20 colunas, onde cada linha é uma coordenada (x, y ou z) do manipulador e cada coluna representa um conector, como mostrado anteriormente na Figura 11. Considerar cada indivíduo do AG como uma matriz 3x20 irá acarretar uma elevada complexidade na execução ao algoritmo, levando bastante tempo computacional para finalizar poucas gerações em um computador comum de uso pessoal. Então, a estratégia utilizada para a realização dos testes foi dividir a execução em partes menores, onde cada indivíduo seria uma coluna i da matriz. Considerando $i = 20$, foram geradas populações de 20 indivíduos para cada i executando por 25 gerações cada, totalizando 10000 verificações de função para cada caso de teste, o mesmo utilizado em [50].

O ADX conseguiu alcançar os melhores resultados em comparação com os cruzamentos SBX e ARX, o desvio padrão cumulativo do ADX foi menor que 0.005, resultando em um caminho gerado ótimo chegando mais perto possível do desejado (Pdc), mostrado na Figura 12. Analisando o uso do CAM, o ADX não demonstrou ter muita variação de resultado final, no entanto, o SBX utilizando o CAM obteve uma melhoria visível no caminho gerado na Figura 13 quando comparado a versão sem o uso do ajuste correlacional. O ARX mostrou ter um resultado mediano entre os três operadores, apresentando erros em alguns conectores, a versão com o CAM mostrou ser mais precisa em relação a versão apenas com a mutação convencional (Figura 14).

Analisando as curvas de convergência para cada configuração na Figura 15, é possível ver que utilizando o ajuste correlacional (Gráficos à esquerda), os operadores de cruzamento alcançaram melhores resultados apresentando uma curva mais próxima do eixo X (próximo de zero) em relação as configurações sem a utilização do ajuste correlacional.

Para minimizar este problema de robótica, o ADX mostrou ser o operador de cruzamento mais eficiente, e o CAM mostrou melhorar os resultados dos outros cruzamentos testados. Os resultados obtidos utilizando o ADX+CAM equivalem ao

resultado mostrado por [50]. A Tabela 8 mostra os valores do melhor resultado obtido (ADX+CAM) para o problema do manipulador.

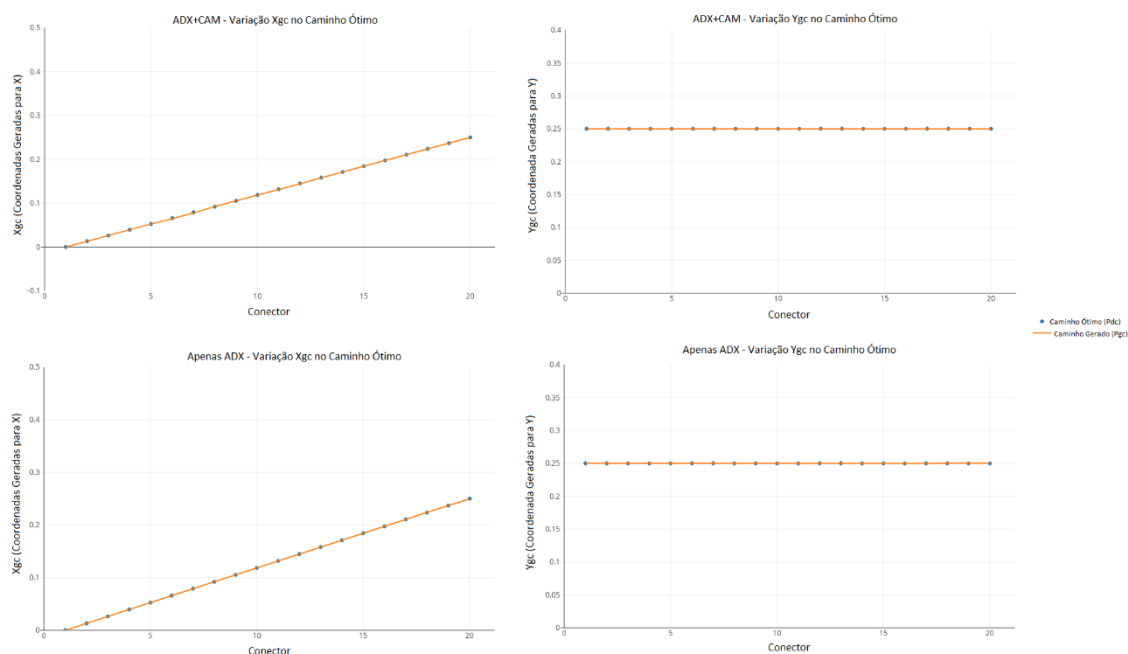


Figura 12 Caminho gerado pelo ADX (Pgc) comparado com o Pdc. O caminho gerado é bastante idêntico ao caminho desejado (Pdc) (Fonte: o autor)

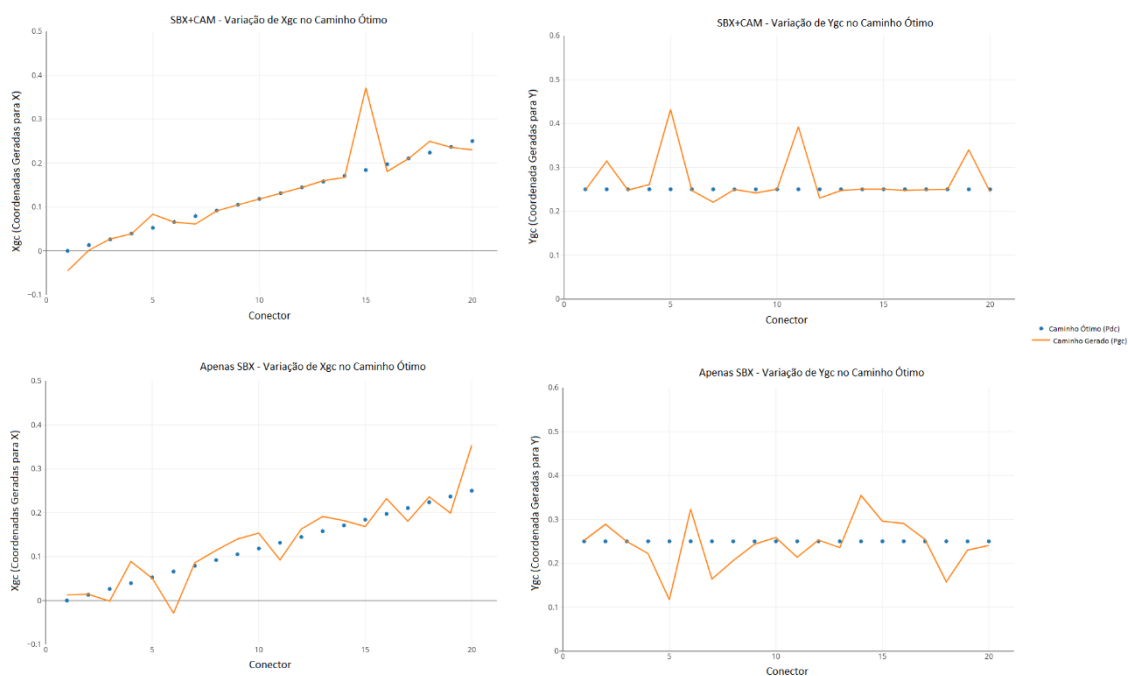


Figura 13 Caminho gerado pelo SBX (Pgc) comparado com o Pdc. O SBX não obteve muito sucesso em gerar um caminho parecido com o desejado (Pdc), no entanto é possível perceber que com o uso do CAM, o caminho gerado (Pgc) é mais preciso (Fonte: o autor)

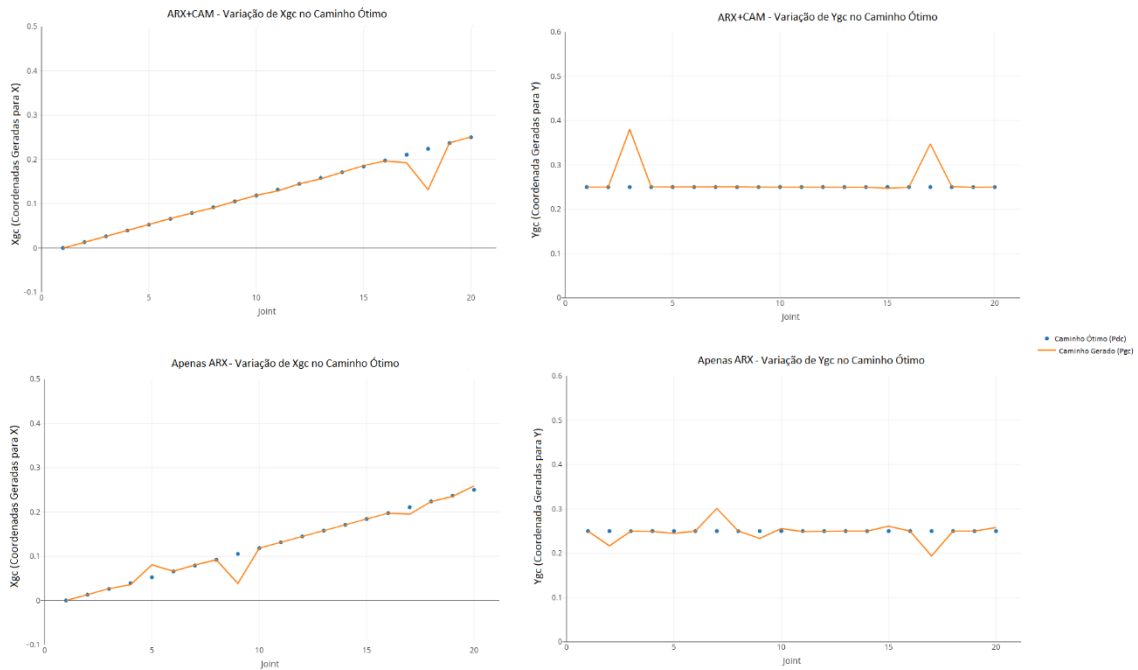


Figura 14 O ARX mostrou melhores resultados que o SBX, no entanto, pior que o ADX. Apresentou coordenadas falhas em alguns conectores, sendo que com o CAM, o caminho gerado (Pgc) foi mais preciso que a versão sem o ajuste. (Fonte: o autor)

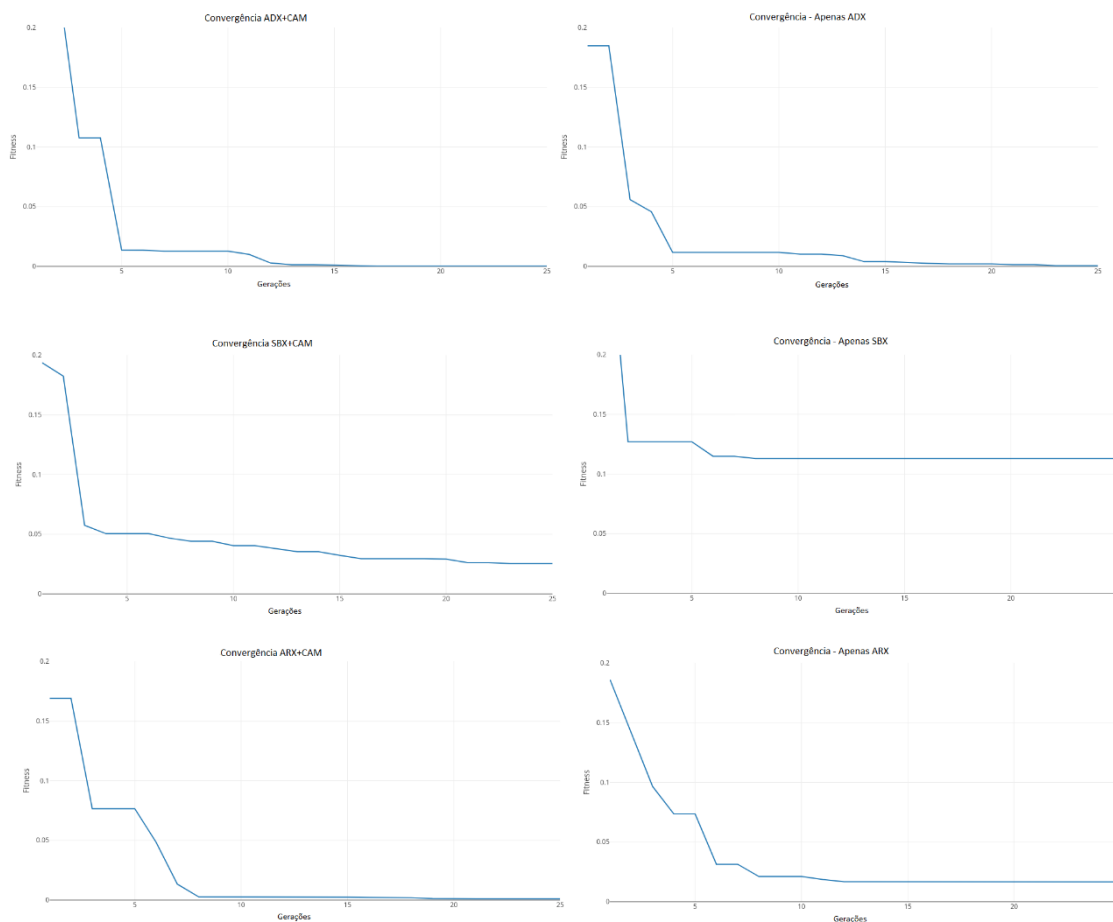


Figura 15 Curvas de convergência de cada configuração. As configurações da esquerda utilizam o CAM e obtiveram uma curva mais próxima do eixo X em relação as configurações com mutação convencional (Gráficos à direita) (Fonte: o autor)

Tabela 8 Melhor resultado obtivo pelo ADX+CAM

Nk (Conector)	$P\theta (M = 3)$			$Pgc (N = 2)$	
	θ_1	θ_2	θ_3	Xgc	Ygc
1	84,58663755	99,2398954	150,7892837	1,17E-05	0,250059139
2	85,61922796	103,086221	151,0746065	0,013138846	0,250049375
3	114,2616067	132,2130647	143,8781706	0,026435948	0,250059401
4	81,46650331	104,7391331	151,4081549	0,039442494	0,249992152
5	48,00699685	83,21534443	144,2344777	0,052576298	0,249958196
6	66,4017937	94,96775851	151,2184117	0,064721446	0,249799461
7	121,7339008	143,5690452	134,9242755	0,077813525	0,249839475
8	57,47721288	90,89468502	151,5295011	0,092337034	0,250348353
9	96,8534707	134,163299	145,5164409	0,105115727	0,250047758
10	8,398956245	80,9428497	130,9056913	0,118750851	0,249954193
11	64,44954803	105,4543423	154,8189469	0,131579235	0,249986724
12	58,51153224	100,1570313	155,5557187	0,144152013	0,249946299
13	177,5308716	155,7420472	91,77804148	0,157949669	0,250011596
14	48,37411046	90,83852409	156,5627239	0,170984385	0,250122902
15	101,0230329	149,9188693	136,6626152	0,184218357	0,24986417
16	95,30811879	149,5333462	139,3261017	0,197357989	0,249991647
17	89,47678319	148,8711804	142,2115573	0,210340186	0,249943925
18	73,14986278	139,8545406	152,0173438	0,223701118	0,24998287
19	69,90537288	139,8235808	153,524534	0,236789409	0,249991239
20	122,7452598	162,3170825	113,5803746	0,250010064	0,249951106

4.3 Problemas de Engenharia

Para realização de testes, foram utilizados os problemas descritos a seguir com a aplicação de três tipos diferentes de cruzamento citados anteriormente. O cruzamento binário (1-ponto de corte) (SBX), o cruzamento aritmético (ARX) e o cruzamento ajustado (ADX). Para cada cruzamento, o algoritmo genético foi executado 40 vezes, cada uma com população inicial diferente, sendo 20 com o ajuste correlacional (CAM) e 20 vezes sem o ajuste, para balancear a utilização do CAM em comparação as execuções sem ele (Mutaç o tradicional), as execuções foram parametrizadas com uma população de 500 indiv duos e 100 gerações, totalizando 50000 verificações de função para o AG, com taxa de elitismo em 70% da população.

4.3.1 Design de Vaso de Pressão / Design of Pressure Vessel (DPV)

Minimizar o custo total $f(x)$ (Equação 17), incluindo o custo do material, de corte e solda de um recipiente cil ndrico, que est  limitado em ambas as extremidades por cabeças hemisféricas. Ele tem restrições em vari veis: (T_s , x_1)   a espessura da casca; (T_h , x_2) espessura da cabeça; (R , x_3)   o raio interno; (L , x_4) comprimento da secc o cil ndrica do vaso. O problema possui 4 restrições $g(1-4)$ [52][53][18].

O DPV possui duas variantes, para o DPV1 [53] o x_4 vai de 10 a 240 e o DPV2 [18] tem o x_4 indo de 10 a 200.

$$\begin{aligned}
 \text{Minimize } f(x) &= 0.6224x_1x_3x_4 + 1.7781x_2x_3^2 + 3.1661x_1^2x_4 \\
 &\quad + 19.84x_1^2x_3 \\
 \text{s. t. } g_1(x) &= -x_1 + 0.0193x_3 \leq 0 \\
 g_2(x) &= -x_2 + 0.00954x_3 \leq 0 \\
 g_3(x) &= -\pi x_3^2x_4 - \frac{4}{3}\pi x_3^3 + 1296000 \leq 0 \\
 g_4(x) &= x_4 - 240 \leq 0 \\
 1 \times 0.0625 &\leq x_1, x_2 \leq 99 \times 0.0625 ; 10 \leq x_3 \leq 200 ; \\
 10 &\leq x_4 \leq 240 \text{ (DPV1) ou } 200 \text{ (DPV2)}
 \end{aligned}
 \tag{Equa o 17}$$

4.3.2 Design de uma Viga de Aço / Welded Beam Design (WBD)

O objetivo   minimizar $f(x)$ (Equa o 18) com algumas restrições sobre as vari veis : (H , x_1)   a espessura de solda; (t , x_2)   a largura do feixe; (b , x_3) a espessura do feixe; (l , x_4)   o comprimento da junta soldada. O problema deve obedecer a sete restrições $g(1-7)$ [53][54][18].

O problema do WBD possui duas variantes, onde o WBD1 [53] tem sete restrições e o WBD2 [18] tem seis.

$$\begin{aligned}
 & \text{Minimize } f(x) = 1.10471x_1^2x_2 + 0.04811x_3x_4(14 + x_2) \\
 & \text{s.t } g_1(x) = \tau(x) - \tau_{max} \leq 0 \\
 & \quad g_2(x) = \sigma(x) - \sigma_{max} \leq 0 \\
 & \quad g_3(x) = x_1 - x_4 \leq 0 \\
 & \quad g_4(x) = 0.125 - x_1 \leq 0 \\
 & \quad g_5(x) = \delta(x) - 0.25 \leq 0 \\
 & \quad g_6(x) = P - P_c(x) \leq 0 \\
 \\
 & \tau(x) = \sqrt{\tau_1^2 + 2\tau_1\tau_2\left(\frac{x_2}{2R}\right) + \tau_2^2} ; \tau_1 = \frac{P}{\sqrt{2}x_1x_2} ; \tau_2 = \frac{MR}{J} \\
 & M = P\left(L + \frac{x_2}{2}\right) ; J(x) = 2\left\{\frac{x_1x_2}{\sqrt{2}}\left[\frac{x_2^2}{12} + \left(\frac{x_1 + x_3}{2}\right)^2\right]\right\} \\
 & R = \sqrt{\frac{x_2^2}{4} + \left(\frac{x_1 + x_3}{2}\right)^2} ; \sigma(x) = \frac{6PL}{x_4x_3^2} \\
 & \delta(x) = \frac{4PL^3}{Ex_3^3x_4} ; P_c(x) = \frac{4.013\sqrt{\frac{EGx_3^2x_4^6}{36}}}{L^2}\left(1 - \frac{x_3}{2L}\sqrt{\frac{E}{4G}}\right) \\
 \\
 & G = 12 \times 10^6 ; E = 30 \times 10^6 ; P = 6000 ; L = 14 \\
 & 0.1 \leq x_1 \leq 2 ; 0.1 \leq x_2 \leq 10 ; 0.1 \leq x_3 \leq 10 ; 0.1 \leq x_4 \leq 2
 \end{aligned}
 \tag{Equação 18}$$

4.3.3 Min. Tensão-Peso sobre Mola / Minimize Tension-Compression on a Spring (MWTCS)

Minimizar o peso da uma tensão/compressão $f(x)$ (Equação 19) sobre uma mola. Obedecendo a quatro restrições $g(1-4)$. Onde: (d, x_1) é o diâmetro do fio; (D, x_2) diâmetro médio da bobina; (P, x_3) é o número de bobinas ativas [53][55].

$$\begin{aligned}
 & \text{Minimize } f(x) = (x_3 + 2)x_2x_1^2 \\
 & \text{s.t } g_1(x) = 1 - \frac{x_2^3x_3}{71785x_1^4} \leq 0 \\
 & \quad g_2(x) = \frac{4x_2^2 - x_1x_2}{12566(x_2x_1^3 - x_1^4)} + \frac{1}{5108x_1^2} - 1 \leq 0 \\
 & \quad g_3(x) = 1 - \frac{140.45x_1}{x_2^2x_3} \leq 0 \\
 & \quad g_4(x) = \frac{(x_1 + x_2)}{1.5} - 1 \leq 0 \\
 \\
 & 0.05 \leq x_1 \leq 2 ; 0.25 \leq x_2 \leq 1.3 ; 2 \leq x_3 \leq 15
 \end{aligned}
 \tag{Equação 19}$$

4.3.4 Design de um Redutor de Velocidade / Speed Reducer Design (SRD)

Minimizar $f(x)$ (Equação 20) para projetar um redutor de velocidade com 11 restrições $g(1-11)$, onde: (x_1) Largura do rosto; (x_2) Módulo de dentes; (x_3) Número de dentes do pinhão; (x_4) Comprimento do primeiro eixo entre os rolamentos; (x_5) Comprimento do segundo eixo entre os rolamentos; (x_6) Diâmetro do primeiro eixo; (x_7) Diâmetro do segundo eixo [56].

$$\begin{aligned} \text{Minimize } f(x) = & 0.7854x_1x_2^2(3.333x_3^2 + 14.9334x_3 - 43.0934) \\ & - 1.508x_1(x_6^2 + x_7^2) + 7.4777(x_6^2 + x_7^2) \\ & + 0.78054(x_4x_6^2 + x_5x_7^2) \end{aligned}$$

$$s. t \ g_1(x) = \frac{27}{x_1x_2^2x_3} - 1 \leq 0$$

$$g_2(x) = \frac{397.5}{x_1x_2^2x_3} - 1 \leq 0$$

$$g_3(x) = \frac{1.93x_4^3}{x_2x_3x_6^3} - 1 \leq 0$$

$$g_4(x) = \frac{1.93x_5^3}{x_2x_3x_7^3} - 1 \leq 0$$

$$g_5(x) = \frac{1}{110x_6^3} \sqrt{\left(\frac{750x_4}{x_2x_3}\right)^2 + 16.9 \times 10^6} - 1 \leq 0$$

$$g_6(x) = \frac{1}{85x_7^3} \sqrt{\left(\frac{750x_5}{x_2x_3}\right)^2 + 157.5 \times 10^6} - 1 \leq 0$$

(Equação 20)

$$g_7(x) = \frac{x_2x_3}{40} - 1 \leq 0$$

$$g_8(x) = \frac{5x_2}{x_1} - 1 \leq 0$$

$$g_9(x) = \frac{x_1}{12x_2} - 1 \leq 0$$

$$g_{10}(x) = \frac{1.5x_6 + 1.9}{x_4} - 1 \leq 0$$

$$g_{11}(x) = \frac{1.1x_7 + 1.9}{x_5} - 1 \leq 0$$

$$2.6 \leq x_1 \leq 3.6 ; 0.7 \leq x_2 \leq 0.8 ; 17 \leq x_3 \leq 28 ; 7.3 \leq x_4 \leq 8.3$$

$$7.3 \leq x_5 \leq 8.3 ; 2.9 \leq x_6 \leq 3.9 ; 5 \leq x_7 \leq 5.5$$

4.3.5 Treliça de Três Barras / Tree-bar Truss (TBT)

O objetivo é minimizar $f(x)$, com duas variáveis contínuas e três restrições como mostrado na (Equação 21) [57].

$$\text{Minimize } f(x) = (2\sqrt{2}x_1 + x_2) \times l$$

(Equação 21)

$$\begin{aligned}
 \text{s. t. } g_1(x) &= \frac{\sqrt{2}x_1 + x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0 \\
 g_2(x) &= \frac{x_2}{\sqrt{2}x_1^2 + 2x_1x_2} P - \sigma \leq 0 \\
 g_3(x) &= \frac{1}{x_1 + \sqrt{2}x_2} P - \sigma \leq 0
 \end{aligned}$$

$$0 \leq x_1 \leq 1 ; 0 \leq x_2 \leq 1 ; l = 100 ; P = 2KN/cm^2 ; \sigma = 2KN/cm$$

4.3.6 Design de Engrenagem de Trem / Design of Gear Train (DGT)

O objetivo do problema é minimizar $f(x)$ representado na (Equação 22). Este é um problema sem restrições, exceto no limite mínimo e máximo de cada variável que vai de 12 a 60 e são do tipo inteiro [18].

Para este problema em particular, o ADX e o CAM foram alterados para fazer ajustes unitários inteiros. Ou seja, o peso do cruzamento e do ajuste correlacional foram atribuídos como $p = 1$ para gerar apenas números inteiros.

$$\text{Minimize } f(x) = \left(\frac{1}{6.931} - \frac{x_1x_2}{x_3x_4} \right)^2 \quad \text{(Equação 22)}$$

$$12 \leq x_1, x_2, x_3, x_4 \leq 60 ; x_i' \in \mathbb{Z}^+$$

4.4 Análise de Diversidade e Resultados Obtidos para os Problemas de Engenharia

Utilizando o CAM juntamente com o ADX, é possível obter um nível balanceado entre exploração local e global (diversificação e intensificação). Permitindo ao algoritmo genético varrer efetivamente o espaço de busca e melhorar os resultados já obtidos.

Para cada problema, foi gerado um gráfico de diversidade e convergência com a finalidade de analisar o comportamento do algoritmo conforme o avanço das gerações. Os valores obtidos das otimizações estarão dispostos em tabelas com o melhor valor médio, desvio padrão, melhor valor obtido e os valores das variáveis da melhor solução. As melhores soluções mostradas para todos os problemas de engenharia não violam nenhuma restrição.

Visto que os algoritmos evolutivos têm comportamento estocástico, para a confirmação de que uma determinada configuração é melhor que outra, foi aplicado o

teste de Friedman [21][58][59]. O teste analisa os resultados obtidos depois de 20 execuções do algoritmo e gera um valor p para um par de configurações, onde $p > 0,05$ indica que as duas configurações testadas possuem a mesma performance e o melhor resultado pode alternar entre as duas ao acaso. Enquanto que um valor $p < 0,05$ indica que determinada configuração realmente irá obter o melhor resultado mais de 95% das execuções.

4.4.1 DPV 1 e 2

A Figura 16 mostra os gráficos de diversidade de todas as configurações testadas. Utilizando o ajuste correlacional, é possível ver que a população apresentou um nível de diversidade maior, variando entre 0 a 0,8 (Eixo Y) para o ADX e o SBX. Sem a utilização do CAM, a variação máxima dos cruzamentos testados ficou em 0,6. Na Tabela 9 mostra-se os níveis médios de diversidade da população, quanto maior, mais diversificada a população será.

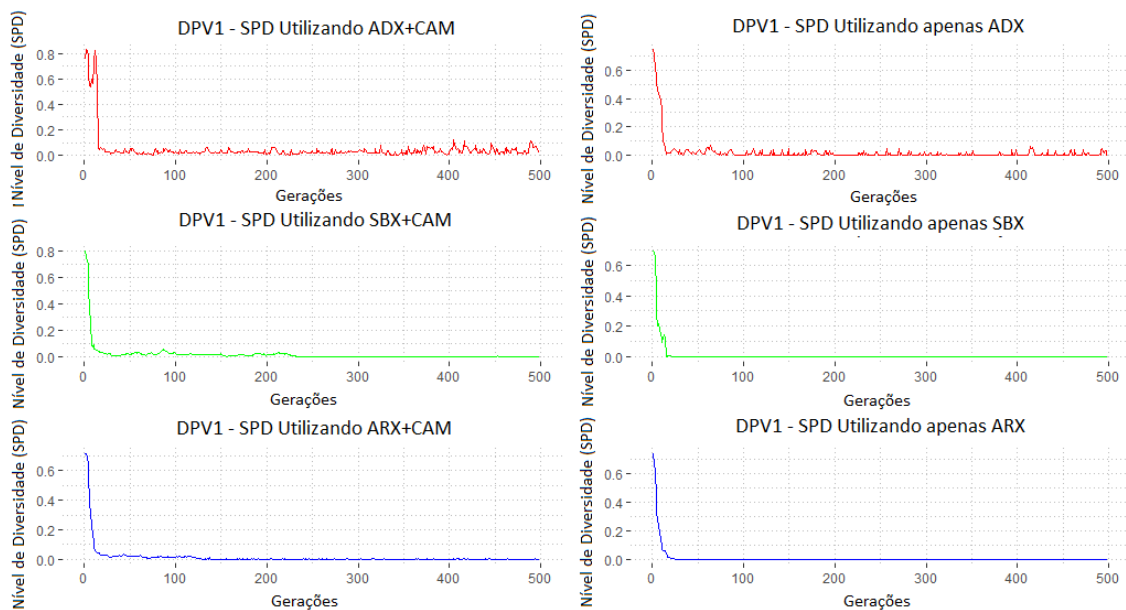


Figura 16 Gráficos de Diversidade de cada crossover para o problema DPV1. Os gráficos à esquerda utilizam o CAM, enquanto que os gráficos da direita utilizam a mutação convencional (Fonte: o autor)

Tabela 9 Níveis médios de SPD de todas as configurações para o problema DPV1

Crossover	CAM	SPD médio
ADX	Sim	0,046783466
SBX	Sim	0,018913272
ARX	Sim	0,013920248
ADX	Não	0,019719356
SBX	Não	0,008308606
ARX	Não	0,008320074

Analisando as otimizações obtidas pelas configurações, o ADX+CAM obteve o melhor e mais confiável resultado para este problema, visto que na Tabela 10 o Fitness Médio, Melhor Fitness e o Desvio Padrão são os menores dentre as demais configurações testadas. Na Figura 17 é notável que com a utilização do CAM, os operadores de cruzamento convergem para um ótimo muito mais rápido em comparação aos que utilizaram a mutação convencional.

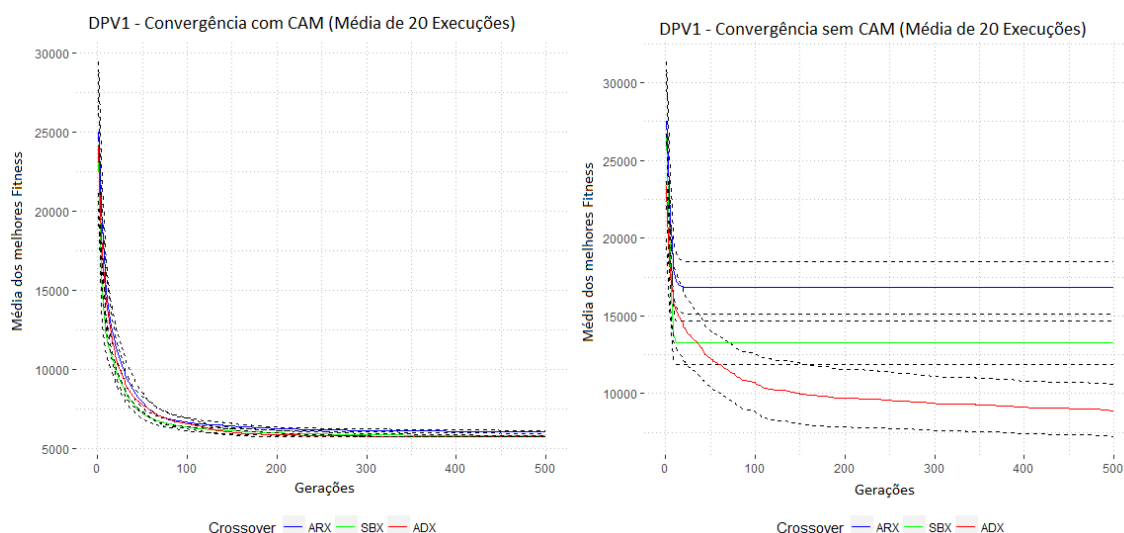


Figura 17 Curvas de convergência das configurações testadas para o problema DPV1. O gráfico à esquerda utiliza o CAM, enquanto que o da direita utiliza a mutação convencional (Fonte: o autor)

Tabela 10 Melhores resultados obtidos em cada configuração para o problema do DPV1

Crossover	CAM	Fitness Médio	Desv. Padrão	Melhor Fitness	X1	X2	X3	X4
ADX	Sim	5804,376773	0,001511001	5804,376257	0,727590953	0,359648586	37,69901308	239,9999799
SBX	Sim	5820,066814	49,92834938	5804,376291	0,727590943	0,359648587	37,69901234	239,9999925
ARX	Sim	6074,118786	216,3093363	5804,376473	0,727590989	0,359648607	37,69901408	239,9999645
ADX	Não	8895,457332	3795,00936	5831,165675	0,738372148	0,365271179	38,25116869	231,397417
SBX	Não	13248,75369	3178,245363	7498,185704	1,125	0,5	51,40280072	96,46087191
ARX	Não	16819,61528	3825,898879	9877,200681	1,031608158	0,988299592	46,41165417	154,1187151

Na variante DPV2, analisando os gráficos de diversidade na Figura 18 é possível notar que o ADX+CAM obteve a maior variabilidade de SPD da população, indicando um maior nível de diversidade em comparação as outras configurações. Os valores médios de SPD podem ser vistos na Tabela 11, onde o ADX+CAM possui a maior variabilidade.

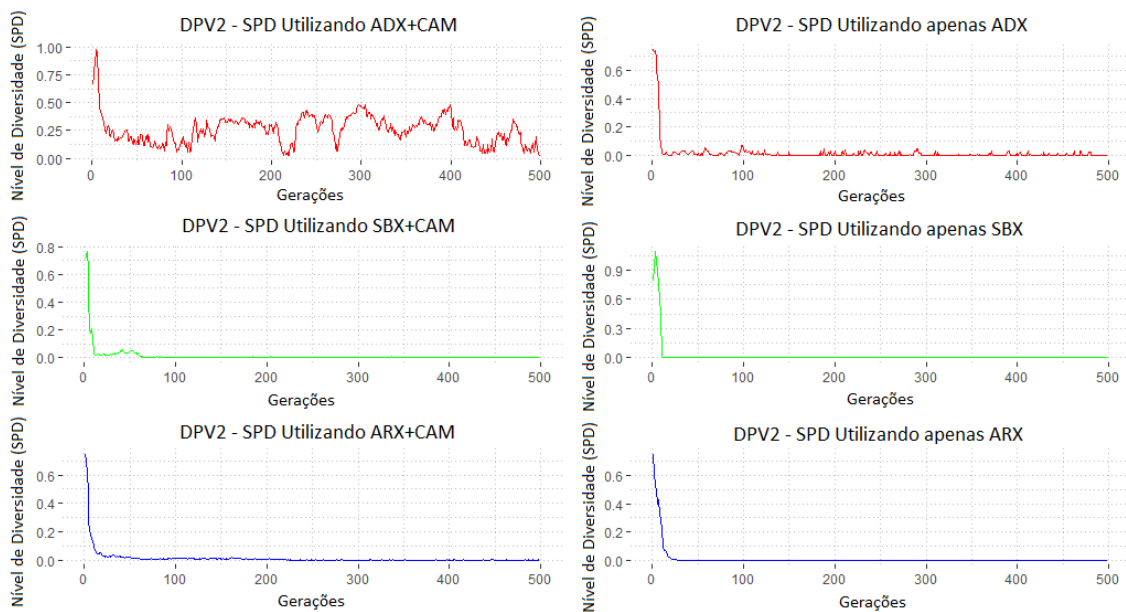


Figura 18 Gráficos de Diversidade de cada crossover para o problema DPV2. Os gráficos à esquerda utilizam o CAM, enquanto que os gráficos da direita utilizam a mutação convencional (Fonte: o autor)

Tabela 11 Níveis médios de SPD de todas as configurações para o problema DPV2

Crossover	CAM	SPD médio
ADX	Sim	0,253794
SBX	Sim	0,012102
ARX	Sim	0,012865
ADX	Não	0,016853
SBX	Não	0,015105
ARX	Não	0,010446

Em relação aos melhores resultados para o DPV2, o ADX+CAM também apresentou a melhor otimização dentre as outras configurações mostradas na Tabela 12. Na curva de convergência na Figura 19 Curvas de convergência das configurações testadas para o problema DPV1. O gráfico à esquerda utiliza o CAM, enquanto o da direita utiliza a mutação convencional, as configurações utilizando o CAM alcançaram melhores otimizações quando comparados às respectivas versões com a mutação convencional.

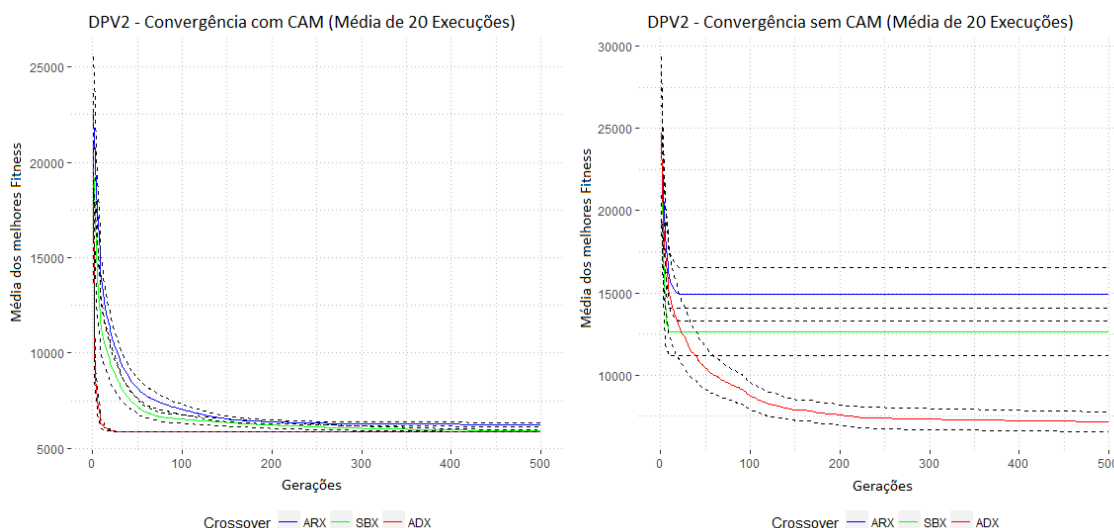


Figura 19 Curvas de convergência das configurações testadas para o problema DPV1. O gráfico à esquerda utiliza o CAM, enquanto o da direita utiliza a mutação convencional (Fonte: o autor)

Tabela 12 Melhores resultados obtidos em cada configuração para o problema do DPV2

Crossover	CAM	Fitness Médio	Desv. Padrão	Melhor Fitness	X1	X2	X3	X4
ADX	Sim	5885,333195	0,000136126	5885,332966	0,77816872	0,384649211	40,3196226	199,999946
SBX	Sim	5912,699038	67,6716489	5885,332866	0,778168661	0,384649169	40,31961926	199,9999927
ARX	Sim	6240,630536	251,6894027	5885,333039	0,778168702	0,384649184	40,31962048	199,9999756
ADX	Não	7148,721563	1345,430944	5899,012929	0,78587434	0,388584768	40,71887648	194,515903
SBX	Não	12632,48067	3354,446078	7674,682181	0,928420318	0,674013419	41,66909929	182,0824132
ARX	Não	14892,40824	3700,728735	7496,664921	1,05512907	0,418332934	43,5389206	159,5691149

Para a confirmação de que uma determinada configuração é melhor que outra, foi aplicado o teste Friedman mencionado anteriormente. Nas duas variações do problema DPV mostrados na Tabela 13 e Tabela 14, o ADX+CAM proposto neste trabalho obteve $p < 0,05$ em comparação a todas as outras configurações, confirmando a superioridade do algoritmo para este problema.

Tabela 13 Teste Friedman entre as configurações para o problema DPV1

	ADX + CAM	SBX + CAM	ARX + CAM	ADX sem CAM	SBX sem CAM
SBX + CAM	6,62E-06	NA	NA	NA	NA
ARX + CAM	1,99E-23	4,10E-13	NA	NA	NA
ADX sem CAM	3,46E-32	8,39E-23	1,72E-05	NA	NA
SBX sem CAM	1,35E-44	3,31E-37	1,72E-21	8,74E-12	NA
ARX sem CAM	3,34E-50	1,07E-43	7,97E-30	7,27E-21	3,18E-05

Tabela 14 Teste Friedman entre as configurações para o problema DPV2

	ADX + CAM	SBX + CAM	ARX + CAM	ADX sem CAM	SBX sem CAM
SBX + CAM	0,032039571	NA	NA	NA	NA

ARX + CAM	1,33E-17	4,85E-13	NA	NA	NA
ADX sem CAM	3,06E-26	5,79E-22	0,000101978	NA	NA
SBX sem CAM	5,70E-40	1,31E-36	9,74E-21	8,08E-12	NA
ARX sem CAM	8,17E-45	8,84E-42	2,18E-27	7,52E-19	0,001852786

4.4.2 WBD 1 e 2

No problema WBD, na variante WBD1 a população das três configurações utilizando o CAM tiveram mais variabilidade que suas respectivas versões com a mutação convencional mostrado na Tabela 15. No entanto, o cruzamento ADX apresentou a melhor diversificação da população quando comparados aos crossovers SBX e ARX com e sem o a utilização do ajuste correlacional, no gráfico da Figura 20 é possível ver que o ADX nas duas versões apresenta uma população bem diversa em relação as outras configurações.

Tabela 15 Níveis médios de SPD de todas as configurações para o problema WBD1

Crossover	CAM	Média SPD
ADX	Sim	0,111847917
SBX	Sim	0,014271732
ARX	Sim	0,0171753
ADX	Não	0,074437229
SBX	Não	0,007802155
ARX	Não	0,006849786

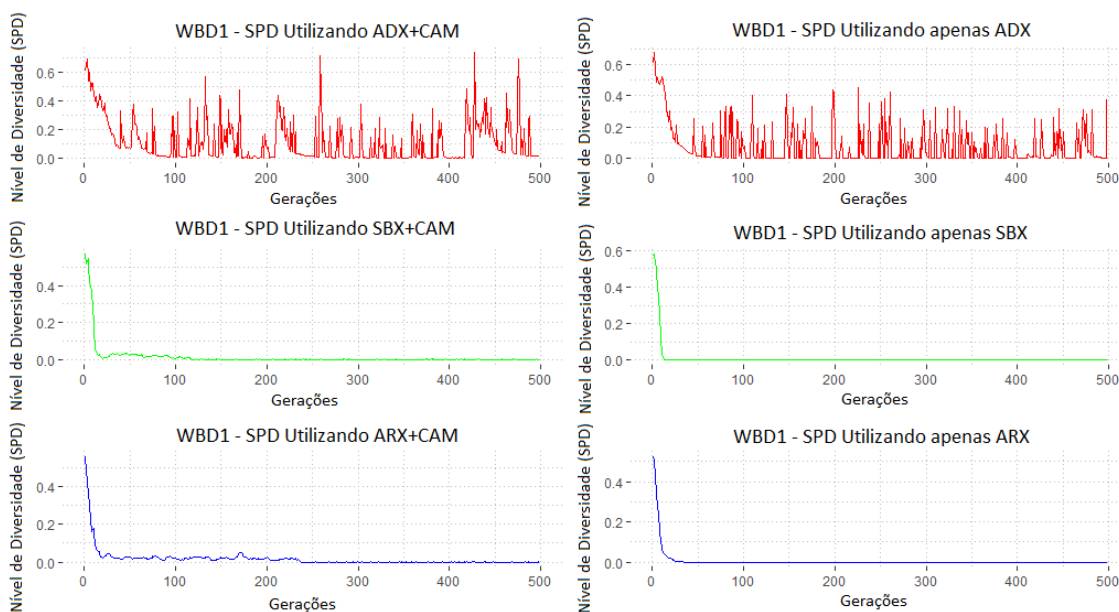


Figura 20 Gráficos de Diversidade de cada crossover para o problema WBD1. Os gráficos à esquerda utilizam o CAM, enquanto os gráficos da direita utilizam a mutação convencional (Fonte: o autor)

Com a utilização do CAM, é possível ver que a curva de convergência do AG para os três cruzamentos mostrados na Figura 21 têm um comportamento muito melhor que suas respectivas versões que não utilizam o ajuste correlacional, principalmente o SBX e o ADX, onde ficaram estagnados em algum ótimo local quando utilizaram a mutação convencional.

Na Tabela 16, o ADX+CAM apresentou o melhor e mais confiável resultado dentre os concorrentes, apresentando um desvio padrão muito menor. Estes resultados são reforçados quando observado que no teste de Friedman apresentado na Tabela 17, o ADX+CAM apresenta $p < 0,05$ quando comparado ao restante das configurações. Em negrito, é notado que o SBX+CAM possui performance similar ao ARX+CAM com $p = 0,089105783$, e o mesmo acontece para as versões com mutação convencional do SBX e ARX, onde obtiveram $p = 1$.

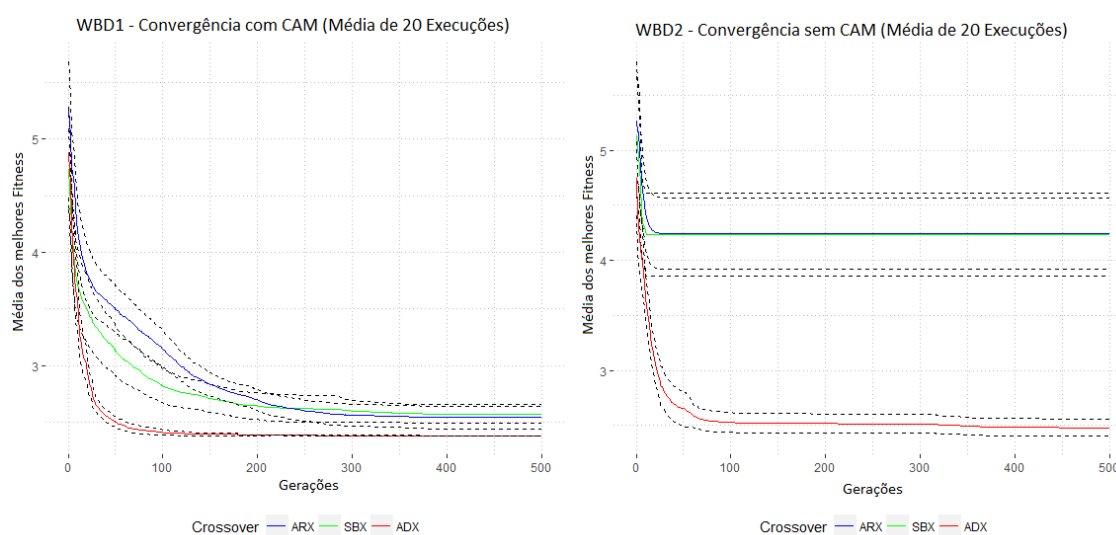


Figura 21 Curvas de convergência das configurações testadas para o problema WBD1. O gráfico à esquerda utiliza o CAM, enquanto o da direita utiliza a mutação convencional (Fonte: o autor)

Tabela 16 Melhores resultados obtidos em cada configuração para o problema do WBD1

Crossover	CAM	Fitness Médio	Desv. Padrão	Melhor Fitness	X1	X2	X3	X4
ADX	Sim	2,381194898	0,000385388	2,38095734	0,244369095	6,217518027	8,291469027	0,244369124
SBX	Sim	2,573268148	0,189580495	2,390155897	0,243032341	6,078524799	8,491649257	0,243032779
ARX	Sim	2,540674969	0,229413567	2,381100828	0,244394728	6,216614302	8,291463407	0,24439473
ADX	Não	2,477468347	0,169429462	2,380957299	0,244368865	6,217508198	8,291487671	0,244368865
SBX	Não	4,233806903	0,867359629	2,996406072	0,337323901	5,667306346	7,026241366	0,343554324
ARX	Não	4,241610503	0,741616166	3,226136684	0,351477463	4,753536657	6,876162573	0,415450468

Tabela 17 Teste Friedman entre as configurações para o problema WBD1

	ADX + CAM	SBX + CAM	ARX + CAM	ADX sem CAM	SBX sem CAM
--	-----------	------------------	-----------	-------------	--------------------

SBX + CAM	3,12E-20	NA	NA	NA	NA
ARX + CAM	4,11E-16	0,089105763	NA	NA	NA
ADX sem CAM	1,14E-06	7,72E-09	3,17E-05	NA	NA
SBX sem CAM	1,45E-41	3,12E-20	2,79E-24	7,62E-33	NA
ARX sem CAM	1,45E-41	3,12E-20	2,79E-24	7,62E-33	1

Para a variante WBD2, na Figura 22 é possível perceber que novamente o ADX mostrou melhor variabilidade da população em relação aos outros tipos de cruzamento. No entanto, analisando o ADX com e sem o CAM houve praticamente a mesma taxa de diversificação da população (SPD) como pode ser visto na Tabela 18 onde o ADX apresenta um $SPD \text{ Médio} = 0,0715422207$ com o ajuste e $SPD \text{ Médio} = 0,079875934$ com a mutação convencional.

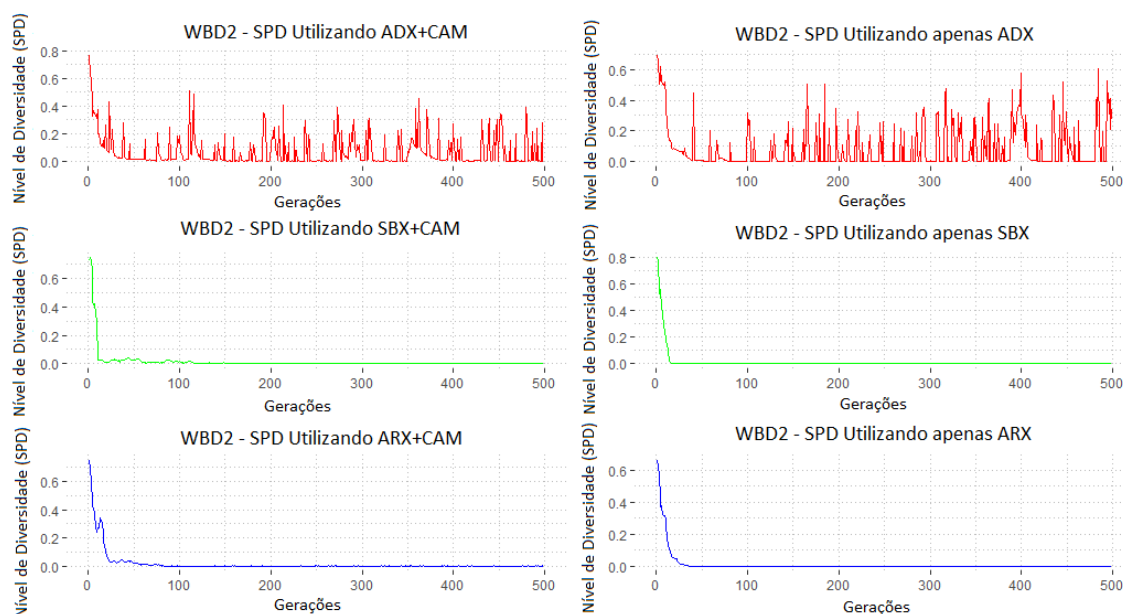


Figura 22 Gráficos de Diversidade de cada crossover para o problema WBD2. Os gráficos à esquerda utilizam o CAM, enquanto os gráficos da direita utilizam a mutação convencional (Fonte: o autor)

Tabela 18 Níveis médios de SPD de todas as configurações para o problema WBD2

Crossover	CAM	Média SPD
ADX	Sim	0,071542207
SBX	Sim	0,013876906
ARX	Sim	0,01669136
ADX	Não	0,079875934
SBX	Não	0,010430594
ARX	Não	0,011155671

As curvas de convergência obtiveram o mesmo comportamento da variante WBD1, onde as configurações que utilizaram o CAM apresentaram melhores otimizações em

relação as suas versões com a mutação convencional. Na Tabela 19 o ADX+CAM apresentou os melhores resultados dentre as outras configurações.

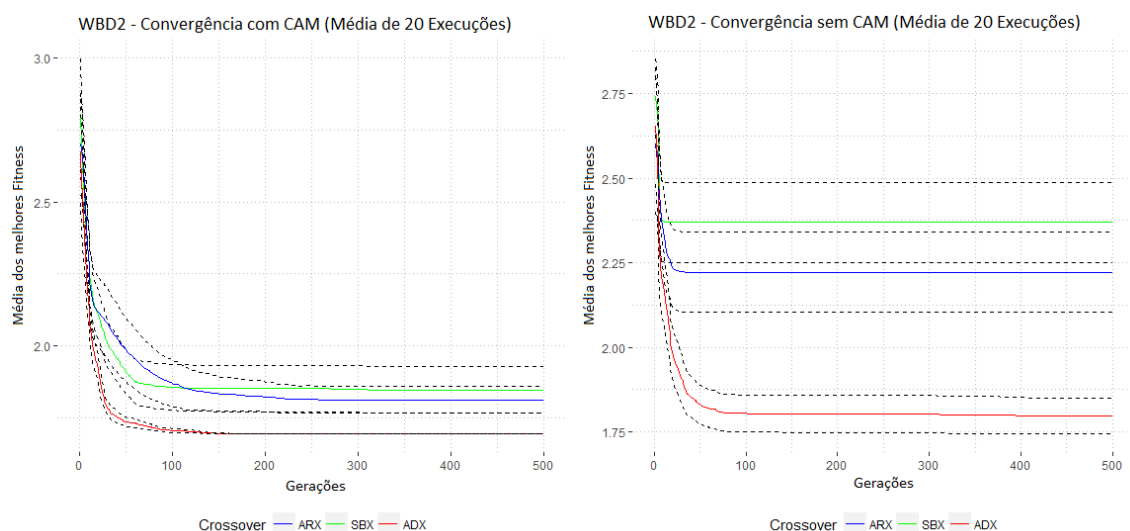


Figura 23 Curvas de convergência das configurações testadas para o problema WBD2. O gráfico à esquerda utiliza o CAM, enquanto o da direita utiliza a mutação convencional (Fonte: o autor)

Tabela 19 Melhores resultados obtidos em cada configuração para o problema do WBD2

Crossover	CAM	Fitness Médio	Desv. Padrão	Melhor Fitness	X1	X2	X3	X4
ADX	Sim	1,695264346	5,26477E-05	1,695247207	0,205729633	3,253120167	9,036624068	0,205729641
SBX	Sim	1,847086425	0,184058862	1,698384005	0,205575207	3,247453715	9,067656663	0,205575319
ARX	Sim	1,812662126	0,106905684	1,696550871	0,206090135	3,248859826	9,028716833	0,206090142
ADX	Não	1,79672566	0,118298108	1,69525163	0,205730867	3,253105548	9,036596843	0,205730873
SBX	Não	2,36868628	0,272667791	1,86184256	0,213648108	3,29484441	8,738486184	0,233217299
ARX	Não	2,22187678	0,27141054	1,785544788	0,201140668	3,282486118	9,266044486	0,212715924

Analisando os testes de Friedman para o WBD2 na Tabela 20, o ADX+CAM apresentou o $p < 0,05$ para todas as outras configurações confirmando o melhor potencial de otimização dentre os outros. Já SBX+CAM apresentou valores maiores que 0,05 para o ARX+CAM e o ADX sem o CAM, concluindo que estas configurações irão ter praticamente a mesma performance. O mesmo acontece com o ARX+CAM quando comparado com o ADX sem o ajuste, onde p atingiu 0,58061184.

Tabela 20 Teste Friedman entre as configurações para o problema WBD2

	ADX + CAM	SBX + CAM	ARX + CAM	ADX sem CAM	SBX sem CAM
SBX + CAM	3,57E-20	NA	NA	NA	NA
ARX + CAM	1,90E-16	0,197547421	NA	NA	NA
ADX sem CAM	4,94E-18	0,580611184	0,580611184	NA	NA
SBX sem CAM	1,32E-41	3,57E-20	6,88E-24	2,55E-22	NA
ARX sem CAM	3,54E-37	2,53E-14	4,94E-18	1,90E-16	0,017266559

4.4.3 MWTCS

No problema MWTCS, a média de diversidade foi maior para o cruzamento ADX conforme mostrado na Tabela 21, onde os níveis médios de SPD foram maiores em relação aos outros tipos de cruzamento. Analisando o gráfico de diversidade presente na Figura 24 é possível ver que houve uma variação maior para o ADX+CAM, onde após a 300ª geração, houve um aumento significativo na diversidade. Indicando que em determinado momento a população pode ter saído de um ótimo local, ou que haviam muitos indivíduos com fitness penalizados por violação de restrição nos genes.

Tabela 21 Níveis médios de SPD de todas as configurações para o problema MWTCS

Crossover	CAM	Média SPD
ADX	Sim	1,667126838
SBX	Sim	0,008153023
ARX	Sim	0,020345447
ADX	Não	0,622638332
SBX	Não	0,005587842
ARX	Não	0,004934748

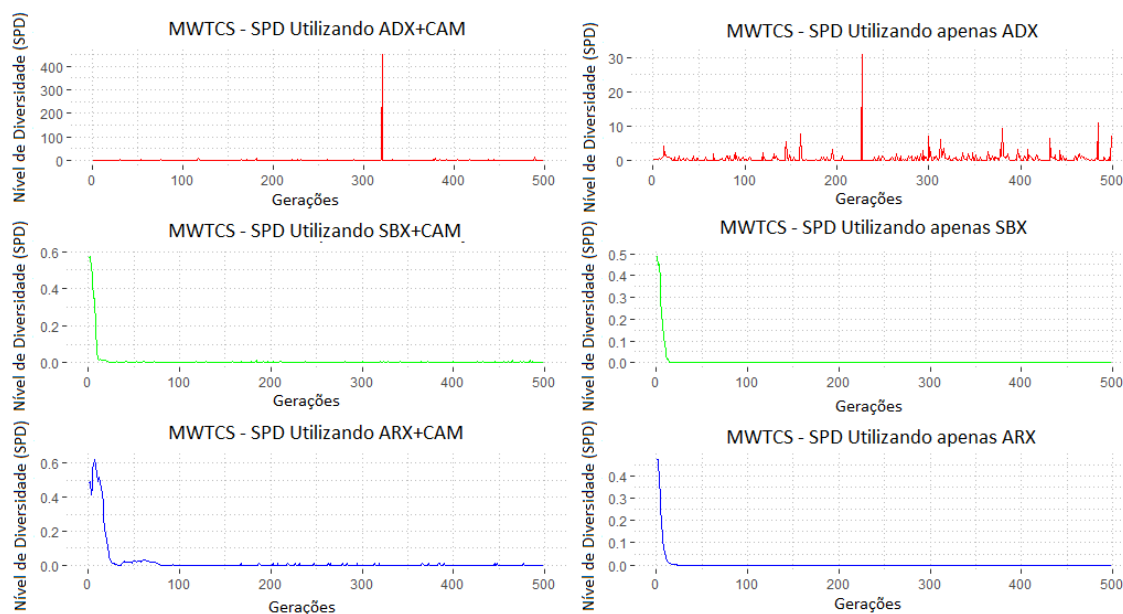


Figura 24 Gráficos de Diversidade de cada crossover para o problema MWTCS. Os gráficos à esquerda utilizam o CAM, enquanto os gráficos da direita utilizam a mutação convencional (Fonte: o autor)

As configurações que utilizaram o CAM mostrados à esquerda da Figura 25, mostram uma convergência mais rápida e otimizada em relação às suas respectivas versões com a mutação convencional. Comparando os resultados mostrados na Tabela 22, o ADX+CAM obteve o melhor resultado com a melhor média geral e desvio padrão dentre as configurações testadas. Este resultado é confirmado pelo teste de

Friedman na Tabela 23, onde o ADX+CAM obteve um valor de $p < 0,05$ quando comparados com os outros cruzamentos.

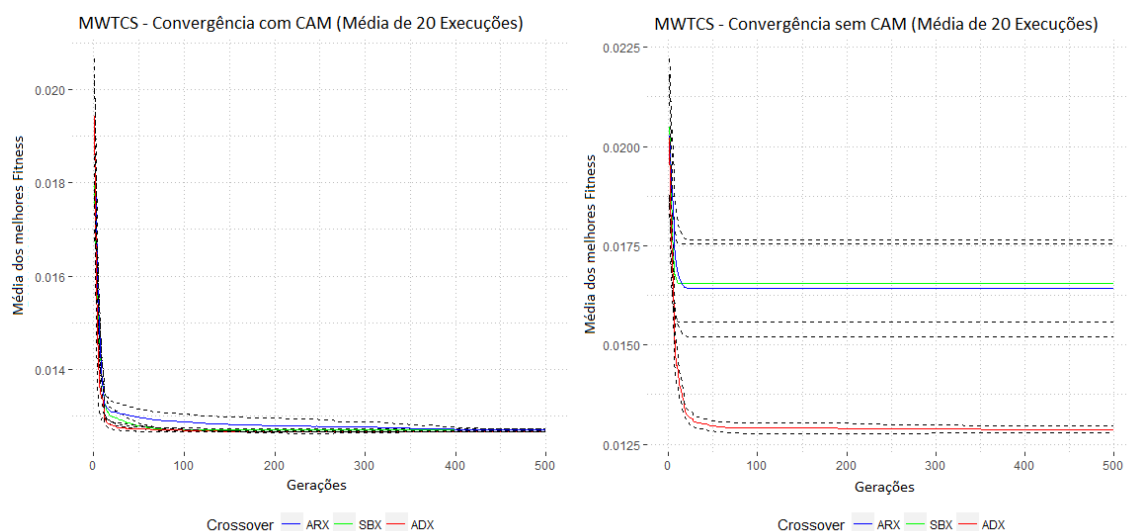


Figura 25 Curvas de convergência das configurações testadas para o problema MWTCs. O gráfico à esquerda utiliza o CAM, enquanto o da direita utiliza a mutação convencional (Fonte: o autor)

Tabela 22 Melhores resultados obtidos em cada configuração para o problema do MWTCs

Crossover	CAM	Fitness Médio	Desv. Padrão	Melhor Fitness	X1	X2	X3
ADX	Sim	0,012665402	2,06789E-07	0,012665234	0,051696686	0,356901202	11,27821854
SBX	Sim	0,012700556	4,62379E-05	0,012665246	0,051663186	0,35609556	11,32553644
ARX	Sim	0,012702066	3,58667E-05	0,01266527	0,051644215	0,355639818	11,35244193
ADX	Não	0,012872378	0,000220028	0,012678427	0,052545669	0,377677494	10,15823144
SBX	Não	0,016561342	0,002225919	0,013559403	0,055650662	0,450778116	7,712635362
ARX	Não	0,016425844	0,002798739	0,013403294	0,05593711	0,456646993	7,380606059

Tabela 23 Teste Friedman entre as configurações para o problema MWTCs

	ADX + CAM	SBX + CAM	ARX + CAM	ADX sem CAM	SBX sem CAM
SBX + CAM	1,79E-12	NA	NA	NA	NA
ARX + CAM	9,13E-21	0,000226533	NA	NA	NA
ADX sem CAM	1,35E-31	3,05E-15	3,92E-07	NA	NA
SBX sem CAM	1,46E-50	3,15E-39	9,87E-33	4,36E-22	NA
ARX sem CAM	3,09E-47	5,64E-35	5,78E-28	1,35E-16	0,008551604

4.4.3 SRD

Para o problema do redutor de velocidade (SRD), novamente o ADX mostrou maior taxa de variação da população, apresentando um gráfico com vários picos de variação na Figura 26. O ADX+CAM mostrou o maior nível médio de SPD entre as outras configurações (Tabela 24).

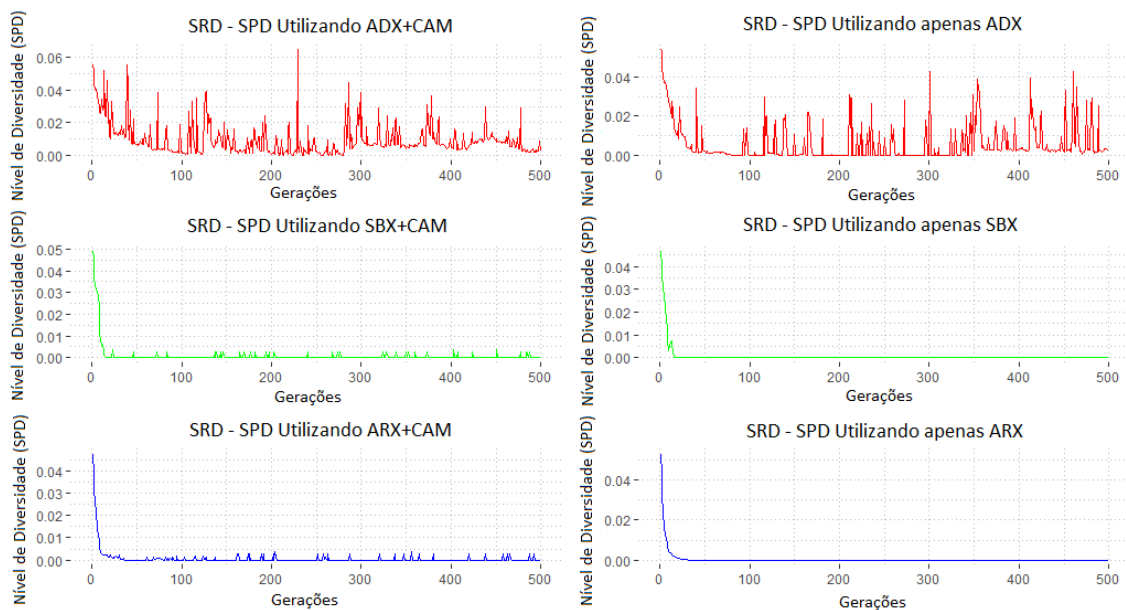


Figura 26 Gráficos de Diversidade de cada crossover para o problema SRD. Os gráficos à esquerda utilizam o CAM, enquanto os gráficos da direita utilizam a mutação convencional (Fonte: o autor)

Tabela 24 Níveis médios de SPD de todas as configurações para o problema SRD

Crossover	CAM	Média SPD
ADX	Sim	0,009205031
SBX	Sim	0,000829004
ARX	Sim	0,000696936
ADX	Não	0,005598273
SBX	Não	0,000545229
ARX	Não	0,000481946

A convergência foi visivelmente melhor quando foi utilizado o CAM para todos os tipos de crossovers, enquanto que nas versões com a mutação convencional, o ADX obteve a melhor otimização (Figura 27). As versões com a utilização do ajuste correlacional apresentaram o melhor resultado de fitness, com valores muito semelhantes, no entanto o ADX+CAM apresentou o melhor fitness médio e melhor confiabilidade com o desvio padrão próximo de zero (Tabela 25). No teste de Friedman, o ADX+CAM apresentou $p < 0,05$ comparado com as outras configurações. Tanto o SBX+CAM e o ARX+CAM apresentaram o mesmo comportamento, indicando que ambos têm o mesmo poder de otimização.

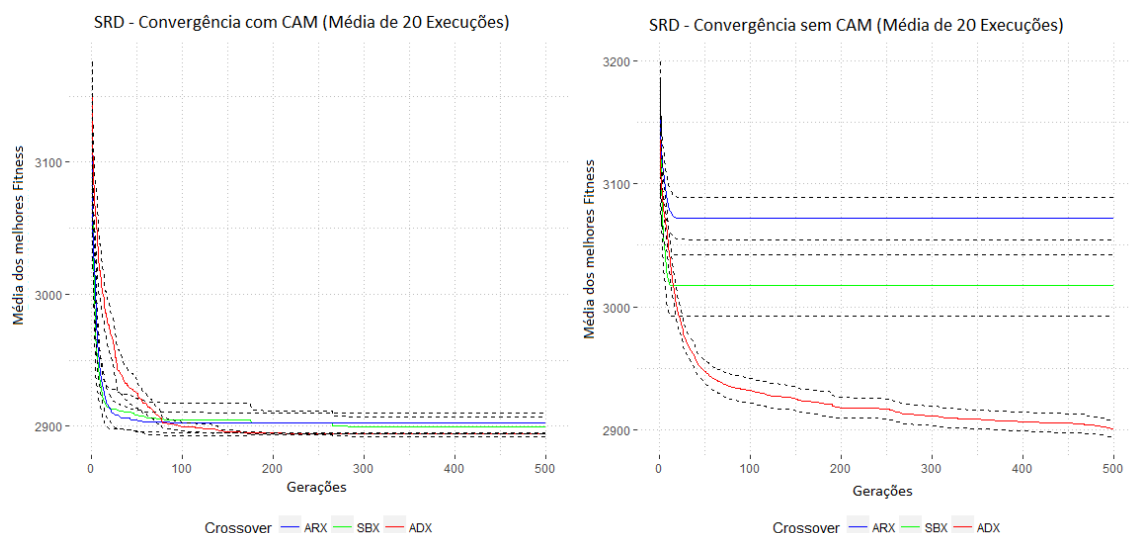


Figura 27 Curvas de convergência das configurações testadas para o problema SRD. O gráfico à esquerda utiliza o CAM, enquanto o da direita utiliza a mutação convencional (Fonte: o autor)

Tabela 25 Melhores resultados obtidos em cada configuração para o problema do SRD

Crossover	CAM	Fitness Médio				Desv. Padrão			
ADX	Sim	2894,382754				0,00123656			
SBX	Sim	2899,508371				17,28414998			
ARX	Sim	2902,467239				17,57607257			
ADX	No	2901,142232				15,37929146			
SBX	No	3017,023008				56,76004703			
ARX	No	3071,630338				38,70176863			
Crossover	CAM	Melhor Fitness	X1	X2	X3	X4	X5	X6	X7
ADX	Sim	2894,382347	3,500000016	0,700000003	17,000000032	7,300000023	7,715320312	2,900000011	5,286654557
SBX	Sim	2894,382284	3,500000031	0,700000004	17,00000003	7,300000035	7,715320026	2,900000049	5,286654469
ARX	Sim	2894,382304	3,500000075	0,700000001	17,000000014	7,3000000352	7,715320183	2,900000028	5,286654472
ADX	No	2894,415945	3,500017471	0,700000334	17	7,300005315	7,715418416	2,9	5,286690856
SBX	No	2952,238809	3,564102727	0,701753597	17,00676682	7,465309122	7,845560158	2,94216127	5,303896827
ARX	No	3006,706731	3,555847898	0,702628428	17,23014312	7,799095756	8,048648956	2,974610109	5,305003096

Tabela 26 Teste Friedman entre as configurações para o problema SRD

	ADX + CAM	SBX + CAM	ARX + CAM	ADX sem CAM	SBX sem CAM
SBX + CAM	2,59E-05	NA	NA	NA	NA
ARX + CAM	0,005236215	0,09508197	NA	NA	NA
ADX sem CAM	7,83E-19	5,98E-09	2,60E-12	NA	NA
SBX sem CAM	1,12E-33	4,92E-25	3,16E-28	3,43E-11	NA
ARX sem CAM	2,41E-40	9,58E-33	1,59E-35	5,87E-20	0,00017079

4.4.4 TBT

Para o problema TBT, o ADX com e sem o CAM conseguiu atingir os melhores níveis de SPD em relação aos outros crossovers, os valores da Tabela 27 mostram que as duas variantes do ADX conseguiram praticamente o mesmo nível de diversificação da população. A Figura 28 mostra que para o ADX com e sem o ajuste correlacional, houve uma grande variabilidade quando comparado as outras configurações.

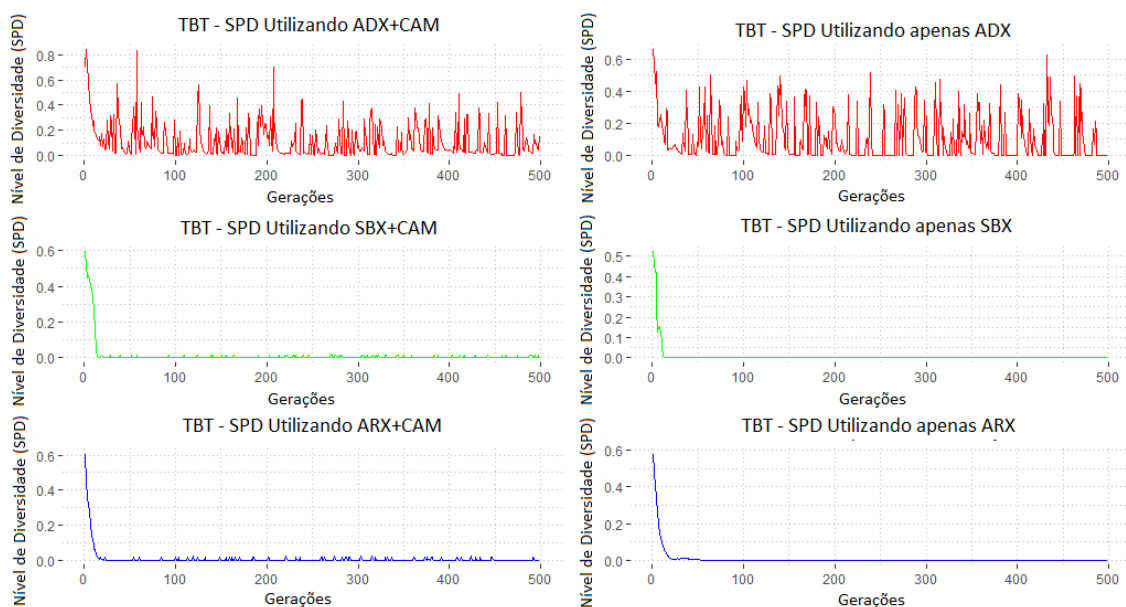


Figura 28 Gráficos de Diversidade de cada crossover para o problema TBT. Os gráficos à esquerda utilizam o CAM, enquanto os gráficos da direita utilizam a mutação convencional (Fonte: o autor)

Tabela 27 Níveis médios de SPD de todas as configurações para o problema TBT

Crossover	CAM	Média SPD
ADX	Sim	0,109022386
SBX	Sim	0,011466528
ARX	Sim	0,008174441
ADX	Não	0,102900506
SBX	Não	0,005544339
ARX	Não	0,007183454

Analisando os resultados da otimização na Tabela 28, o ADX+CAM conseguiu o melhor e mais confiável resultado devido ao seu desvio padrão ser menor que os demais concorrentes. No gráfico de convergência da Figura 28, é possível perceber que com a utilização do CAM, todos os crossovers conseguiram convergir para um ótimo melhor em relação a utilização da mutação convencional. A Tabela 29 com os resultados do teste de Friedman reforçam o ADX+CAM como o melhor candidato entre os outros testados ($p < 0,05$), enquanto que o SBX+CAM e o ARX sem o CAM demonstraram ter o mesma performance ($p = 1$).

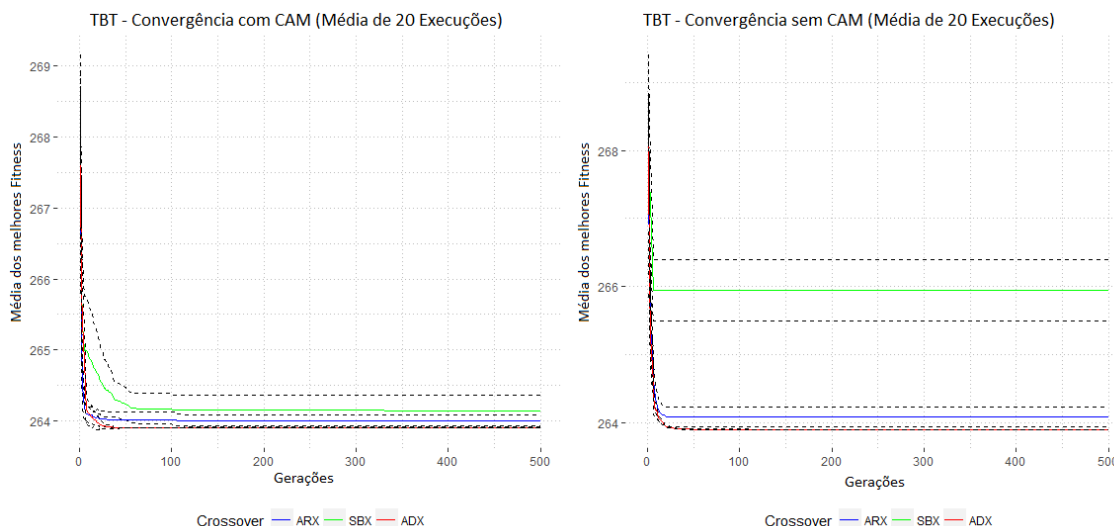


Figura 29 Curvas de convergência das configurações testadas para o problema TBT. O gráfico à esquerda utiliza o CAM, enquanto o da direita utiliza a mutação convencional (Fonte: o autor)

Tabela 28 Melhores resultados obtidos em cada configuração para o problema do TBT

Crossover	CAM	Fitness Médio	Desv. Padrão	Melhor Fitness	X1	X2
ADX	Sim	263,8958436	3,77285E-07	263,8958434	0,788671319	0,408259083
SBX	Sim	264,1438833	0,486676773	263,8958641	0,788840351	0,407781195
ARX	Sim	263,9961037	0,199284018	263,8958434	0,788675419	0,408247487
ADX	Não	263,8968527	0,002043782	263,8958434	0,788676232	0,408245186
SBX	Não	265,9423265	1,033586631	264,1378635	0,798778025	0,382093203
ARX	Não	264,0825426	0,320391861	263,8959226	0,788347079	0,409176962

Tabela 29 Teste Friedman entre as configurações para o problema TBT

	ADX + CAM	SBX + CAM	ARX + CAM	ADX sem CAM	SBX sem CAM
SBX + CAM	6,58E-28	NA	NA	NA	NA
ARX + CAM	3,49E-23	0,04635531	NA	NA	NA
ADX sem CAM	3,66E-08	2,25E-15	2,73E-10	NA	NA
SBX sem CAM	1,33E-44	1,96E-16	1,48E-21	3,90E-35	NA
ARX sem CAM	6,58E-28	1	0,04635531	2,25E-15	1,96E-16

4.4.5 DGT

No problema DGT, o gráfico de diversidade apresentou comportamento semelhante ao do problema TBT, onde o ADX apresentou melhores índices de SPD tanto com o CAM quanto com a mutação convencional (Tabela 29). A Figura 30 também mostra que o comportamento durante a execução do algoritmo foi semelhante.

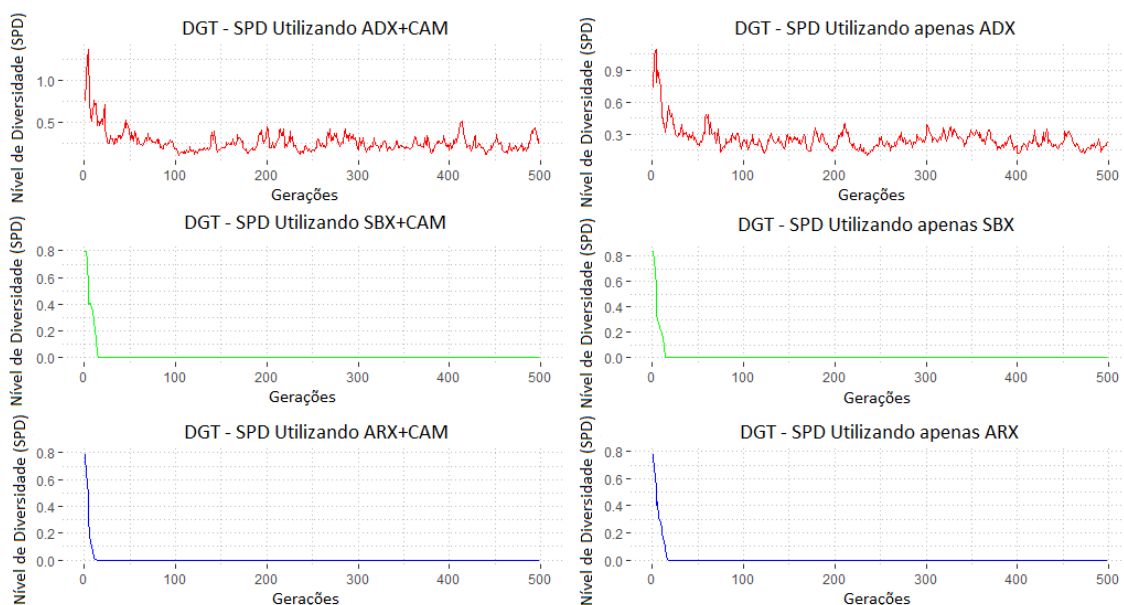


Figura 30 Gráficos de Diversidade de cada crossover para o problema DGT. Os gráficos à esquerda utilizam o CAM, enquanto os gráficos da direita utilizam a mutação convencional (Fonte: o autor)

Crossover	CAM	Média SPD
ADX	Sim	0,260730253
SBX	Sim	0,01137375
ARX	Sim	0,006671197
ADX	Não	0,250778074
SBX	Não	0,009769822
ARX	Não	0,010337964

O gráfico de convergência para este problema ficou parecido para todas as configurações (Figura 31), isso provavelmente aconteceu pelo fato de o DGT utilizar apenas valores inteiros para as suas variáveis e ser relativamente de fácil otimização. Na Tabela 30 pode-se ver que o ADX+CAM e com a mutação convencional alcançaram melhores resultados em relação ao SBX e o ARX. No teste de Friedman da Tabela 31 é possível notar que ambas das configurações do ADX obtiveram $p > 0,05$ indicando que a utilização do ajuste correlacional ou não para este problema, não irá afetar os resultados do ADX de maneira efetiva.

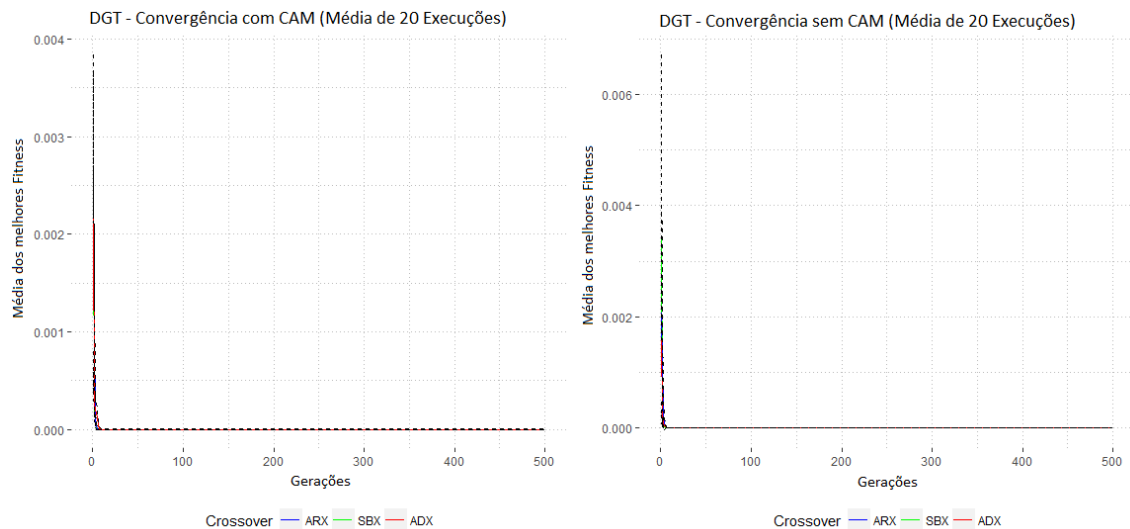


Figura 31 Curvas de convergência das configurações testadas para o problema DGT. O gráfico à esquerda utiliza o CAM, enquanto o da direita utiliza a mutação convencional (Fonte: o autor)

Tabela 30 Melhores resultados obtidos em cada configuração para o problema do DGT

Crossover	CAM	Fitness Médio	Desv. Padrão	Melhor Fitness	X1	X2	X3	X4
ADX	Sim	1,5085E-10	3,01889E-10	2,70086E-12	19	16	49	43
SBX	Sim	3,06999E-06	6,848E-06	2,35764E-09	12	30	48	52
ARX	Sim	2,91529E-08	7,23451E-08	2,70086E-12	19	16	49	43
ADX	Não	2,46004E-10	4,03594E-10	2,70086E-12	16	19	49	43
SBX	Não	2,0689E-06	5,57602E-06	9,92158E-10	24	13	47	46
ARX	Não	2,35903E-08	5,66741E-08	2,70086E-12	19	16	49	43

Tabela 31 Teste Friedman entre as configurações para o problema DGT

	ADX + CAM	SBX + CAM	ARX + CAM	ADX sem CAM	SBX sem CAM
SBX + CAM	1,24E-32	NA	NA	NA	NA
ARX + CAM	3,31E-11	1,45E-17	NA	NA	NA
ADX sem CAM	0,59012357	1,01E-31	3,54E-10	NA	NA
SBX sem CAM	1,54E-35	0,246627372	3,76E-21	1,13E-34	NA
ARX sem CAM	1,22E-13	5,02E-15	0,453916626	1,48E-12	1,16E-18

4.5 Comparação do ADX+CAM com outros Algoritmos Evolutivos Clássicos

Para verificar a performance do AG proposto (ADX+CAM) com outras técnicas evolutivas clássicas presentes na literatura, foram conduzidos testes com um enxame de partículas (*Particle Swarm Optimization* - PSO) e o *Simulated Annealing* (SA). Cada algoritmo evolutivo foi configurado para 30000 verificações de função, no caso do AG proposto, foi designado uma população com 50 indivíduos evoluindo por 500 gerações.

Cada algoritmo foi executado 20 vezes, a Tabela 32 mostra os resultados estatísticos obtidos após todas as execuções. Para grande parte dos problemas, o ADX+CAM

mostrou melhores resultados em relação aos outros algoritmos evolutivos, exceto no problema TBT, onde o PSO obteve valores semelhantes.

Foi aplicado o teste T pareado para assegurar que os resultados de um algoritmo são realmente melhores que os concorrentes, ou seja, quando $p < 0,05$ confirma-se que uma configuração é melhor que a outra.

Comparando a performance do ADX+CAM com o PSO e o SA mostrado na Tabela 33, é possível constatar que o GA obtém valores de $p < 0,05$, indicando a confiabilidade dos resultados mostrados para o algoritmo proposto neste trabalho, portanto confirmando que o ADX+CAM é melhor.

As curvas de convergências mostrados na Figura 32 e Figura 33 apontam que o AG (em azul) proposto alcança a melhor otimização em relação ao PSO (em vermelho) e o SA (em verde), tendo SA com a pior performance em relação aos outros.

Tabela 32 Resultados comparativos para os algoritmos evolutivos, os valores são a média de 20 execuções para cada algoritmo

Problema/Algoritmo	Melhor	Mediana	Média	Pior	Desv. Padrão	Variância	
DPV1	AG	5804,3762565	5804,3764295	5804,3767727	5804,3831679	0,0015110	2,28E-06
	PSO	5959,9438064	6573,7801216	6988,1912812	13756,7604985	1675,9313051	2808745,7395116
	SA	6853,3672792	8153,4648054	8207,8582150	9831,4218680	756,1727858	571797,2820582
DPV2	AG	5885,3329659	5885,3331945	5885,3331955	5885,3334250	0,000136126	1,85E-08
	PSO	5972,4022431	6780,3205959	6751,9703029	7524,3494835	369,9394966	136855,2311764
	SA	6922,0566095	8394,2598434	8406,9928462	9867,8898331	829,7893463	688550,3592198
WBD1	AG	2,3809573	2,3810040	2,3811949	2,3822941	0,0003854	1,49E-07
	PSO	2,4504108	2,5845788	2,7518595	4,0361274	0,3888061	0,1511702
	SA	2,5189233	2,7189795	2,7218511	2,8752662	0,0974808	0,0095025
WBD2	AG	1,6952472	1,6952478	1,6952643	1,6954760	5,26E-05	2,77E-09
	PSO	1,6983537	1,8166167	1,9595216	2,5739594	0,2889512	0,0834928
	SA	1,7916778	1,8814996	1,8795096	1,9565442	0,0409617	0,0016779
MWTCS	AG	0,0126652	0,0126653	0,0126654	0,0126660	2,07E-07	4,28E-14
	PSO	0,0126671	0,0136205	0,0136291	0,0159951	7,67E-04	5,89E-07
	SA	0,0128295	0,0131474	0,0131655	0,0138139	2,09E-04	4,37E-08
TBT	AG	263,8958434	263,8958435	263,8958436	263,8958450	3,77E-07	1,42E-13
	PSO	263,8958434	263,8958434	263,8958437	263,8958466	9,09E-07	8,27E-13
	SA	263,9150830	263,9679676	263,9857913	264,1543707	0,0676148	0,0045718
DGT	AG	2,70086E-12	2,30782E-11	1,5085E-10	9,92158E-10	3,01889E-10	9,11368E-20
	PSO	2,30782E-11	2,35764E-09	3,86926E-05	0,000773602	0,00017298	2,9922E-08
	SA	2,30782E-11	3,43047E-09	2,58858E-08	1,61223E-07	4,5131E-08	2,0368E-15
SRD	AG	2894,3823473	2894,3824608	2894,3827541	2894,3879893	1,24E-03	1,53E-06
	PSO	2914,1360462	3010,6239206	3023,0994604	3147,6440018	59,1938779	3503,9151795
	SA	2945,5810119	2966,0943175	2965,7906103	2985,4235394	11,3439139	128,6843820

Tabela 33 Teste T-pareado assumindo variâncias iguais entre o ADX+CAM e os outros dois algoritmos testados (PSO e SA)

Problema	PSO	SA
DPV1	1,37E-15	2,34E-89
DPV2	1,64E-59	3,67E-142
WBD1	3,52E-88	1,87E-19

WBD2	1,98E-185	9,55E-35
MWTCS	1,09E-137	2,60E-07
TBT	1,98E-39	0,000220391
DGT	5,66E-11	0,005822203
SRD	0	1,49E-83

Curva de Convergência para cada Problema (Média de 20 Execuções)

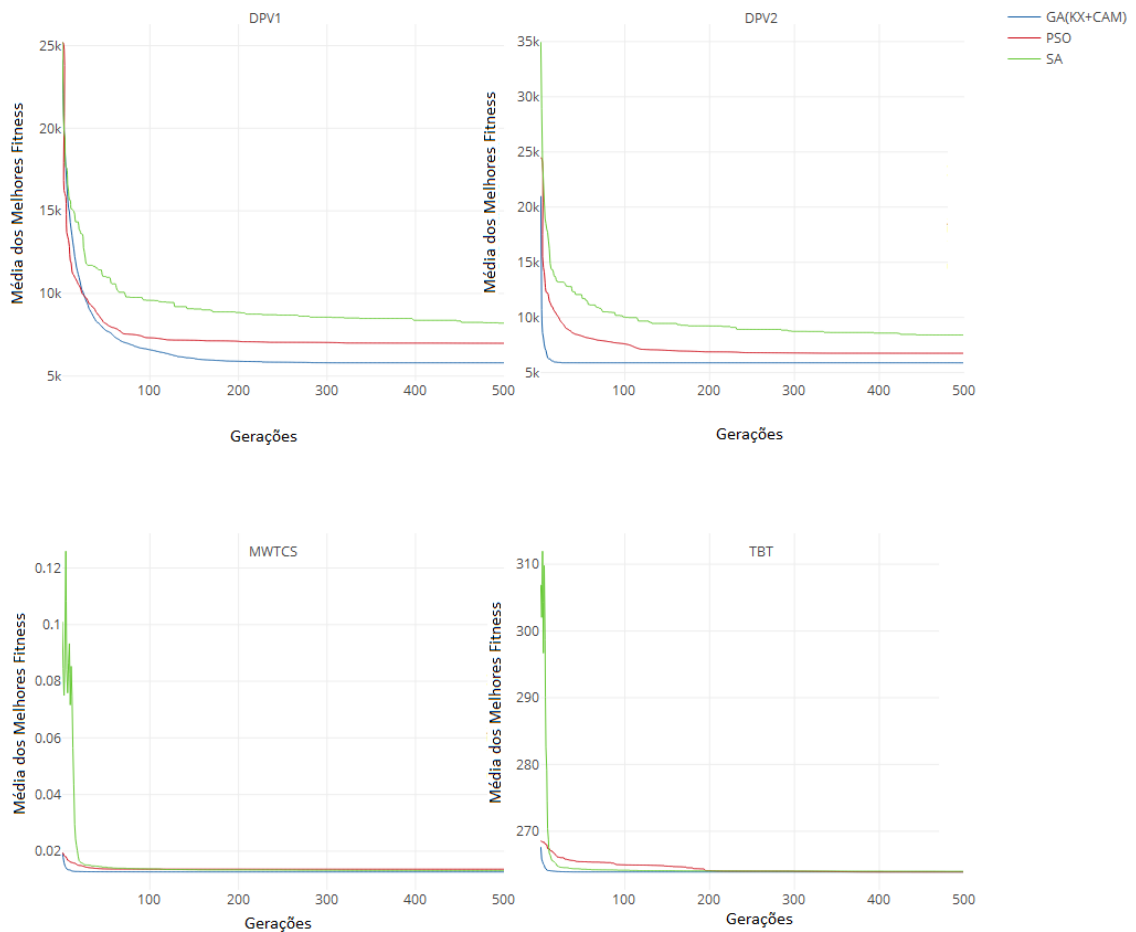


Figura 32 Curvas de Convergência para os problemas DPV1, DPV2, MWTCS e TBT (Fonte: o autor)

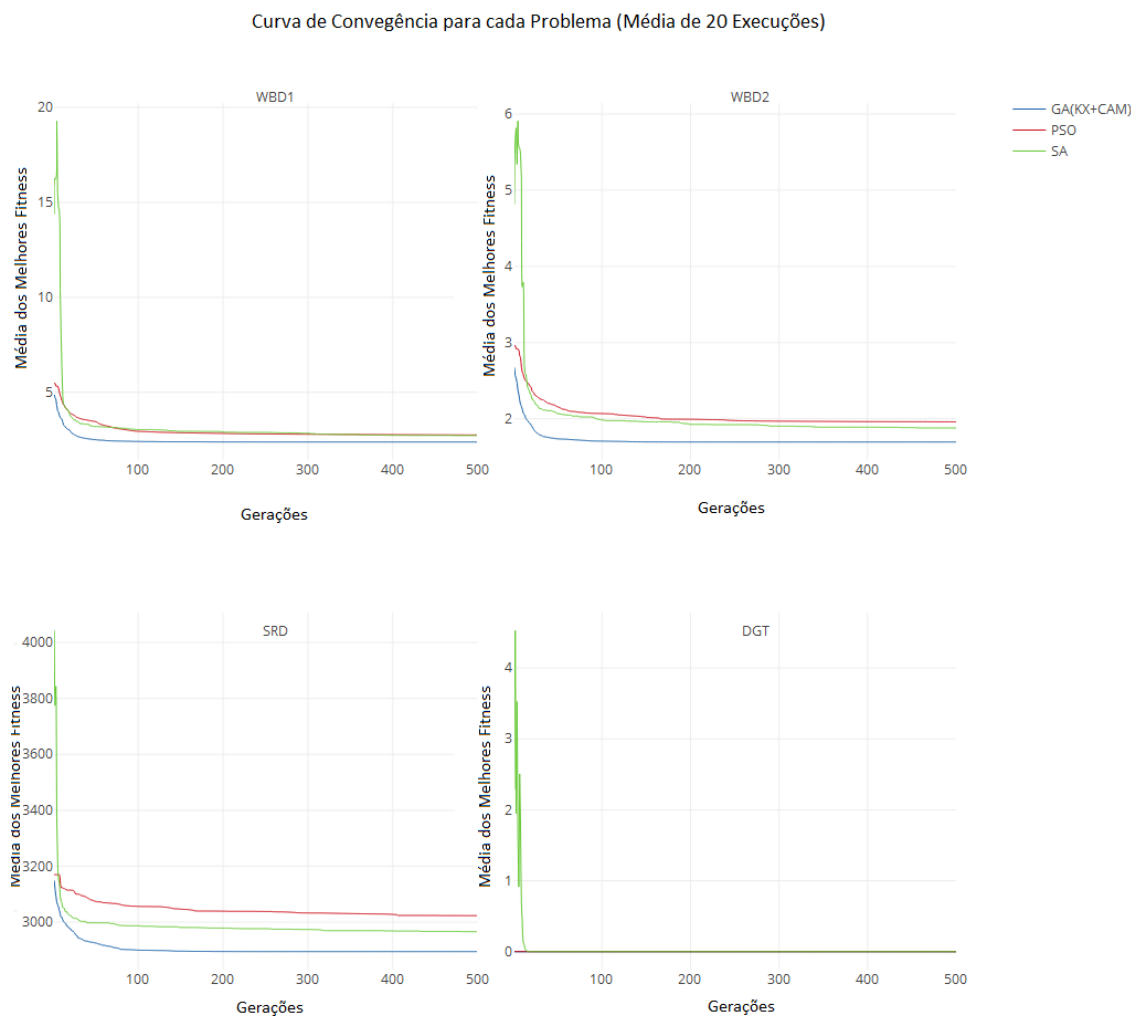


Figura 33 Curvas de Convergência para os problemas WBD1, WBD2, SRD e DGT (Fonte: o autor)

4.6 Comparação dos Resultados com outros Trabalhos da Literatura

Comprando os resultados do ADX+CAM com outros algoritmos presentes na literatura, o AG proposto atingiu melhores otimizações para os diversos problemas testados. Para o DPV1, DPV2, WBD1, DGT e TBT, esta pesquisa conseguiu os melhores resultados médios, desvio padrão e melhor fitness. Os resultados são mostrados na Tabela 34, Tabela 35, Tabela 36, Tabela 37, Tabela 38, Tabela 39, Tabela 40 e Tabela 41, os melhores resultados estão em negrito.

Para o problema MWTCS, [53] obteve o melhor resultado e [60] a melhor média. Dados a variabilidade dos resultados, o algoritmo de [61] obteve o resultado mais confiável com o menor desvio padrão na Tabela 39. No problema WBD2 na Tabela 37, [18] obteve os melhores resultados gerais. Grande parte dos trabalhos não mostravam o

resultado médio ou desvio padrão para o SRD apresentado na Tabela 39, considerando o melhor resultado obtido, o ADX+CAM alcançou a melhor otimização.

Tabela 34 Resultados Comparativos para o problema DPV1

Algoritmo	Média	Desv. Padrão	Melhor Resultado
Dimmopoulos [22]	NA	NA	5850,3830
Mahdavi et al. [62]	NA	NA	5849,7617
Hedar e Fukushima [61]	6164,585867	257,473670	5868,764836
Gandomi et al. [86]	5937,33790	164,54747	5850,38306
Garg [53]	5805,473914033477	1,411462164114731	5804,448670820886
ADX+CAM	5804,376773	0,001511001	5804,376257

Tabela 35 Resultados Comparativos para o problema DPV2

Algoritmo	Média	Desv. Padrão	Melhor Resultado
Sandgren [52]	N/A	N/A	8129,1036
Kannan and Kramer [87]	N/A	N/A	7198,0428
Deb and Gene [88]	N/A	N/A	6410,3811
Coello [69]	6293,8432	7,4133	6288,7445
Coello e Montes [70]	6177,2533	130,9297	6059,9463
He e Wang [73]	6147,1332	86,4545	6061,0777
Montes e Coello [75]	6850,0049	426	6059,7456
Kaveh e Talatahari [77]	6081,7812	67,2418	6059,7258
Kaveh e Talatahari [89]	6075,2567	41,6825	6059,0925
Gandomi et al. [90]	6447,736	502,693	6059,714
Cagnina et al. [74]	6092,0498	12,1725	6059,714335
Coello [78]	6440,3786	448,4711	6059,7208
He et al. [91]	6289,92881	305,78	6059,7143
Akay and Karaboga [79]	6245,308144	205	6059,714339
Garg [53]	5887,557024	2,74529	5885,403282
Garg [18]	5885,382053	0,04908	5885,332773
ADX+CAM	5885,333195	0,000136126	5885,332966

Tabela 36 Resultados Comparativos para o problema WBD1 (* - houve quebra de restrições)

Algoritmo	Média	Desv. Padrão	Melhor Resultado
Ragsdell & Phillips [63]	NA	NA	2,385937
Rao [64]	NA	NA	2,3860
Deb [54]	NA	NA	2,433116
Deb [92]	NA	NA	2,38119
Ray e Liew [65]	3,2551371	0,9590780	2,3854347
Hwang e He [66]*	2,26	NA	2,25
Mehta e Dasgupta [93]	2,3811786	NA	2,381134
Garg [53]	2,38108932	1,01227E-4	2,38099617
ADX+CAM	2,381194898	0,000385388	2,38095734

Tabela 37 Resultados Comparativos para o problema WBD2

Algoritmo	Média	Desv. Padrão	Melhor Resultado
Coello [69]	1,771973	0,01122	1,748309
Coello and Montes [70]	1,792654	0,07471	1,728226
Dimopoulos [22]	NA	NA	1,731186
He and Wang [94]	1,748831	0,012926	1,728024
Hedar e Fukushima [61]	1,7564428	0,0424175	1,7250022
Montes et al. [76]	1,725	1,00E-15	1,724852
Montes e Coello [75]	1,81329	0,0705	1,7373

Cagnina et al. [74]	2,0574	0,2154	1,724852
Kaveh e Talatahari [77]	1,729752	0,0092	1,724918
Kaveh e Talatahari [89]	1,727564	0,008254	1,724849
Gandomi et al. [86]	1,878656	0,2677989	1,7312065
Mehta e Dasgupta[93]	1,724865	NA	1,724855
Akay e Karaboga [79]	1,741913	0,031	1,724852
Garg [53]	1,69530842	2,84E-05	1,69526388
Garg [18]	1,6952471	2,19E-09	1,6952471
ADX+CAM	1,695264346	5,26477E-05	1,695247207

Tabela 38 Resultados Comparativos para o problema MWTCS

Algoritmo	Média	Desv. Padrão	Melhor Resultado
Belegundu [55]	NA	NA	0,0128334
Arora [68]	NA	NA	0,0127303
Coello [69]	0,01276920	3,9390E-06	0,01270478
Ray e Saini [71]	0,015526	NA	0,0130600
Coello e Montes [70]	0,012742	5,9000E-06	0,0126810
Ray e Liew [65]	0,012922669	5,92E-05	0,01266924934
Hu et al.[72]	0,012718975	6,446E-06	0,0126661409
He e Wang [73]	0,012730	5,1985E-06	0,0126747
Zhang et al. [57]	0,012669366	1,25E-06	0,012665233
Hedar & Fukushima [61]	0,012665299	2,2E-09	0,012665285
Montes et al. [76]	0,012666	2,0E-07	0,012665
Montes e Coello [75]	0,013461	9,6600E-05	0,012698
Cagnina et al. [74]	0,0131	4,1E-05	0,012665
Kaveh e Talatahari [77]	0,012720	3,4888E-06	0,0126432
Omran e Salman [60]	0,0126652642	NA	0,0126652375
Coelho [78]	0,013524	0,001268	0,012665
Akay & Karaboga [79]	0,012709	0,012813	0,012665
Garg [53]	0,0126689724845	9,429426E-07	0,0126652327883
ADX+CAM	0,012665402	2,06789E-07	0,012665234

Tabela 39 Resultados Comparativos para o problema SRD

Algoritmo	Média	Desv. Padrão	Melhor Resultado
Akhtar et al. [80]	NA	NA	3008,197440
Rao e Xiong [81]	NA	NA	3000,959715
Cagnina et al. [74]	NA	NA	2996,347849
Jaberipour e Khorram [82]	NA	NA	2994,477531
Li e Papalambros [83]	NA	NA	2994,553869
Tosserams et al. [84]	NA	NA	2996,645783
Lu e Kim [85]	NA	NA	3019,583365
Lin et al. [43]	NA	NA	2993,738921
Huang [46]	NA	NA	2990,124384
Lin et al. [67]	NA	NA	2994,471921
ADX+CAM	2894,382754	0,00123656	2894,382347

Tabela 40 Resultados Comparativos para o problema TBT

Algoritmo	Média	Desv. Padrão	Melhor Resultado
Mohamed, A [95]	263,895843	0,00E+00	263,8958434
Mohamed e Sabry [96]	263,895843	5,34E-13	263,8958434
Wang e Li [97]	263,895843	4,34E-14	263,8958434
Zhang et al. [57]	263,895844	9,70E-07	263,8958433
Wang et al. [98]	263,90041	1,10E-03	263,8958435
Melo & Carosio [99]	263,895843	2,58E-07	263,8958434
Gong et al. [100]	263,895843	0,00E+00	263,8958434
Sadollah et al. [101]	263,897996	3,93E-03	263,895852

Eskandar et al. [102]	263,895903	8,71E-05	263,895843
ADX+CAM	263,8958436	3,77285E-07	263,8958434

Tabela 41 Resultados Comparativos para o problema DGT

Algoritmo	Média	Desv. Padrão	Melhor Resultado
Sandgren [52]	NA	NA	5,712E-6
Kannan e Kramer [87]	NA	NA	2,146E-8
Deb e Goyal [103]	NA	NA	2,701E-12
Gandomi et al. [90]	1,9841E-09	3,5546E-09	2,701E-12
Garg [53]	1,2149E-09	8,7787E-10	2,70085E-12
ADX+CAM	1,5085E-10	3,01889E-10	2,70086E-12

Capítulo 5. Considerações Finais

Otimização de problemas através de algoritmos evolutivos ainda é uma área de pesquisa bastante estudada devido ao aumento da complexidade de problemas reais e a necessidade do processamento computacional acompanhar a otimização para tais problemas. Como não há uma evolução igual entre estas duas vertentes, a possibilidade da utilização de algoritmos de “força bruta” é descartada. Para heurísticas evolutivas serem eficientes na otimização de problemas, é preciso haver balanceamento de poder exploratório global e local (diversificação e intensificação), e realizar este tipo de balanceado não é uma tarefa trivial. É preciso haver um estudo do comportamento da heurística, tipo de problema a ser resolvido e o ambiente ao qual determinada técnica vai ser utilizada [20][11][12][13][14].

Este trabalho propõe um novo cruzamento e uma nova fase de ajuste correlacional como substituto à mutação clássica do algoritmo genético. O cruzamento ADX eleva o poder de melhorar resultados previamente conhecidos (intensificação) através de um fino ajuste nos genes dos progenitores, criando herdeiros com genes semelhantes. Enquanto o CAM promove uma capacidade exploratória global mais eficiente do AG (diversificação) pois afeta grande parte da população incrementando ou diminuindo o valor dos genes de acordo com um cálculo onde há a correlação do gene com o valor do fitness.

Através da realização dos testes em diferentes categorias de problemas foi possível constatar que o algoritmo proposto consegue ser capaz de alcançar resultados satisfatórios para grande parte das otimizações. Para os problemas do CEC’2015 o AG conseguiu melhores resultados que o humanCog [45] para algumas funções. Apesar da performance pouco relevante nos problemas do CEC’2015, é importante ressaltar que os métodos propostos são relativamente mais simples e com capacidade de implementação relativamente mais fácil.

O uso do operador CAM possibilitou uma varredura maior do espaço de busca. No problema de cinemática inversa, o trabalho proposto alcançou uma otimização melhor que os outros tipos de cruzamento, conseguindo criar um caminho idêntico ao caminho ótimo desejado. Para os problemas de engenharia, o ADX+CAM mostrou os melhores

resultados, conseguindo alcançar níveis de SPD maiores nas configurações que utilizaram o ajuste correlacional (CAM).

Levando em conta o aumento exploratório global promovido pelo CAM e local promovido pelo ADX, é notável que houve um melhoramento no resultado e no comportamento populacional do algoritmo para os problemas de minimização testados. Utilizando as duas técnicas em conjunto, houve uma melhora significativa nas otimizações realizadas, salvo em algumas exceções. Quando comparados com outros trabalhos presentes da literatura, o método proposto obteve melhores resultados para a maioria dos problemas testados. Como visto no tópico 3.3, é notável o aumento do custo computacional do ADX e do CAM quando comparados a operadores tradicionais de cruzamento e mutação.

Desta forma, este trabalho contribui com técnicas para aumento dos níveis exploratórios gerais de um algoritmo evolutivos através do aumento da capacidade de exploração e intensificação do espaço de busca.

É necessário a comparações específicas de níveis exploratórios e intensificatórios do trabalho proposto com outros métodos presentes na literatura para apontar as vantagens/desvantagens dos algoritmos testados considerando tempo de execução, diversidade da população, capacidade de otimização, etc. Com isso, será possível apontar com mais exatidão as capacidades do trabalho apresentado.

Referências

- [1] Z. Michalewicz, *Genetic Algorithms + Data Structures = Evolution Programs*, Springer - Verlag, Berlin, 1994.
- [2] Kennedy, J.; Eberhart, R. "Particle Swarm Optimization". *Proceedings of IEEE International Conference on Neural Networks. IV.* pp. 1942–1948, 1995.
- [3] Karaboğa, Derviş. "An Idea Based on Honey Bee Swarm For Numerical Optimization", *TECHNICAL REPORT-TR06*, 2005.
- [4] Storn, R.; Price, K. Differential evolution - a simple and efficient heuristic for global optimization over continuous spaces. *Journal of Global Optimization.* 11 (4): 341–359. 1997.
- [5] Woo, Z., Hoon, J., & Loganathan, G. V. A New Heuristic Optimization Algorithm: Harmony Search. *SIMULATION*, 76(2), 60–68. 2001.
- [6] Xinjie Yu, Mitsuo Gen. *Introduction to Evolutionary Algorithms*, Springer-Verlag, 2010.
- [7] Randy L. Haupt, Sue Ellen Haupt. *PRACTICAL GENETIC ALGORITHMS, SECOND EDITION*, John Wiley & Sons, 2004.
- [8] Hills, Thomas T., Todd, Peter M., Lazer, David, Redish, A. David, Couzin, Iain D. Exploration versus exploitation in space, mind, and society *Trends in Cognitive Sciences* , Volume 19, Issue 1, 46 - 54, Elsevier, 2014
- [9] Hari Mohan Pandeya, Ankit Chaudharyb, Deepti Mehrotrac. A comparative review of approaches to prevent premature convergence in GA. *Applied Soft Computing* 24, 1047–1077, 2014.
- [10] D. Curran, C. O’Riordan, and H. Sorensen, The effects of lifetime learning on the diversity and fitness of populations, in *Proc. 9th Ann. Conf. Genetic Evol. Comput.*, p. 337, 2007.
- [11] Brian Mc Ginley, John Maher, Colm O’Riordan, and Fearghal Morgan. Maintaining Healthy Population Diversity Using Adaptive Crossover, Mutation, and Selection. *IEEE Transactions on Evolutionary Computation*, Vol. 15, No. 5, 2011.
- [12] Gupta, Deepti. Ghafir, Shabina. Overview of Methods to maintain genetic algorithm performance. *International Journal of Emerging Technology and Advanced Engineering.* Volume 2, Issue 5, 2012
- [13] Feng, Zhiwei. Zhang, Qingbin. Zhang, Qingfu. Tang, Qiangang. Yang, Tao. Ma, Yang. A multiobjective optimization-based framework to balance the global exploration and local exploitation in expensive optimization. *Journal of Global Optimization*, Volume 61, Issue 4, pp 677-694. 2015
- [14] Bing Xue, Mengjie Zhang, Will N. Browne, Xin Yao. A Survey on Evolutionary Computation Approaches to Feature Selection. *IEEE Transactions on Evolutionary Computation*, Vol. 20, No.4, 2016
- [15] David E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*, Addison Wesley Longman, 1989.
- [16] Harish Garg, S.P. Sharma, Multi-objective reliability-redundancy allocation problem using particle swarm optimization. *Computers & Industrial Engineering*, Volume 64, Issue 1 (2013), pp. 247-255.
- [17] Harish Garg. Performance analysis of an industrial system using soft computing based hybridized technique. *Journal of the Brazilian Society of Mechanical Sciences and Engineering.* Volume 39 (2016). pp 1441–1451.
- [18] Harish Garg, A hybrid PSO-GA algorithm for constrained optimization problems, *Applied Mathematics and Computation*, Volume 274 (2016), pp. 292-305
- [19] Harish Garg. Performance analysis of an industrial system using soft computing based hybridized technique. *Journal of the Brazilian Society of Mechanical Sciences and Engineering.* Volume 39 (2016). pp 1441–1451.

- [20] Crepinsek, Matej. Liu, Shih-Hsi. Mernik, marjan. Exploration and exploitation in evolutionary algorithms: A survey. *ACM Comput. Surv.* 45, 3, Article 35, 33 pages, 2013
- [21] Wong, Y.-Y., Lee, K.-H., Leung, K.-S., and Ho, C.-W. A Novel approach in parameter adaptation and diversity maintenance for genetic algorithms. *Soft Computing.* 7, 506–515. 2003
- [22] G. G. Dimopoulos, Mixed-variable engineering optimization based on evolutionary and social metaphors, *Computer Methods in Applied Mechanics and Engineering*, 196 (2007), 803–817.
- [23] F. Herrera, M. Lozano, J.L. Verdegay. Tackling Real-Coded Genetic Algorithms: Operators and Tools for Behavioural Analysis, *Artificial Intelligence Review*, 12 (1998), 265–319
- [24] Walaah AbdElrouf, Adil Yousif, Mohammed Bakri Bashir. High Exploitation Genetic Algorithm for Job Scheduling on Grid Computing. *International Journal of Grid and Distributed Computing*, Vol. 9, No. 3 (2016), 221-228.
- [25] Fatemeh Vafaei, Gyorgy Turán, Peter C. Nelson, Tanya Y. Berger-Wolf. Balancing the Exploration and Exploitation in an Adaptive Diversity Guided Genetic Algorithm. *IEEE Congress on Evolutionary Computation* (2014)
- [26] L.J. Eshelman, J.D. Schaffer, Real-coded genetic algorithms and interval-schemata, in: *Proceedings of the Workshop on Foundations of Genetic Algorithms*, Vail, CO, USA, pp. 187–202, 1933.
- [27] Cervantes-Castilho, Adriana. Mezura-Montes, Efren, Coello, Carlos. An Empirical Comparison of Two Crossover Operators in Real-Coded Genetic Algorithms for Constrained Numerical Optimization Problems. *Special Session on Evolutionary Computation - ROPEC*, 2014.
- [28] Deb, R.B. Agrawal, Simulated binary crossover for continuous search space, *Complex Syst.* 9 115–148, 1995.
- [29] Y.C. Chuang, C.T. Chen, C. Hwang, A real-coded genetic algorithm with direction-based crossover operator. *Information Sciences*. Volume 305, Pages 320–348, 2015.
- [30] Ma Li, Li Qianting, Ma Meiqiong, Lv Sicong. Optimization and Application of Single point Crossover and Multi Offspring Genetic Algorithm. *International Journal of Hybrid Information Technology*. Vol.9, No.1, pp. 1-8, 2016.
- [31] Zhu, Qingling. Lin, Qiuzhen. Du, Zhihua. Liang, Zhengping. Wang, Wenjun. Zhu, Zexuan. Chen, Jianyong. Huang, Peizhi. Ming, Zhong. A Novel adaptive hybrid crossover operator for multi objective evolutionary algorithm. *Information Sciences*. Volume 345, Pages 177–198, 2016.
- [32] Chuanga, Yao-Chen. Chena, Chyi-Tsong. Hwangba, Chyi. A simple and efficient real-coded genetic algorithm for constrained optimization. *Applied Soft Computing*. Volume 38, Pages 87–105, 2016.
- [33] Rituraj Singh Patwal, Nitin Narang, Harish Garg, A novel TVAC-PSO based mutation strategies algorithm for generation scheduling of pumped storage hydrothermal system incorporating solar units. *Energy*, Volume 142 (2018), pp. 822-837.
- [34] Deepika Rani, T.R. Gulati, Harish Garg. Multi-objective non-linear programming problem in intuitionistic fuzzy environment: Optimistic and pessimistic view point. *Expert Systems with Applications*, Volume 64 (2016), pp. 228-238.
- [35] Ma Li, Li Qianting, Ma Meiqiong and Lv Sicong. Optimization and Application of Single-point Crossover and Multi-offspring Genetic Algorithm, *International Journal of Hybrid Information Technology* Vol.9, No.1, pp. 1-8, 2016
- [36] Zhang Qiongbing, Ding Lixin. A new crossover mechanism for genetic algorithms with variable-length chromosomes for path optimization problems. *Expert Systems with Applications* 60, pp. 183–189, 2016
- [37] Ajay Shrestha, Ausif Mahmood. Improving Genetic Algorithm with Fine-Tuned Crossover and Scaled Architecture. *Journal of Mathematics* Volume 2016, Article ID 4015845, 10 pages, 2016

- [38] Z.-Q. Chen, Y.-F. Yin, A New Crossover Operator for Real-coded Genetic Algorithm with Selective Breeding Based on Difference Between Individuals, in: 8th International Conference on Natural Computation (ICNC), 2012.
- [39] Kaya, Yilmaz, Uyar, Murat. Tekin, Ramazan. A Novel Crossover for Genetic Algorithm: Ring Crossover. CoRR, <http://arxiv.org/abs/1105.0355>, 2011
- [40] S. H. Ling. Iterated Function System-Based Crossover Operation for Real-Coded Genetic Algorithm. *Journal of Intelligent Learning Systems and Applications*, 7, 37-41 2015.
- [41] Kendall, M.G. Rank Correlation Methods. New York: Hafner Publishing Co. 1955.
- [42] Abdi, Hervé. The Kendall Rank Correlation Coefficient. *Encyclopedia of Measurement and Statistics*. Ed. Neil Salkind. 2007.
- [43] M.-H. Lin, J.-F. Tsai, and P.-C. Wang, Solving engineering optimization problems by a deterministic global optimization approach, *Applied Mathematics & Information Sciences*, vol. 6, No. 3, supplement, pp. 1101–1107, 2012.
- [44] Q. Chen, B. Liu, Q. Zhang, J. J. Liang, P. N. Suganthan, B. Y. Qu. Problem Definitions and Evaluation Criteria for CEC 2015 Special Session on Bound Constrained Single-Objective Computationally Expensive Numerical Optimization, *IEEE Conference on Evolutionary Computation*, (2015).
- [45] A. Al-Dujaili, K. Subramanian, S. Suresh. Humancog: A cognitive architecture for solving optimization problems. *2015 IEEE Congress on Evolutionary Computation*, (2015) pp. 3220-3227.
- [46] C. H. Huang, Engineering design by geometric programming, *Mathematical Problems in Engineering*, vol. 2013, Article ID 568098, 8 pages, 2013.
- [47] J.L Rueda, I. Erlich, MVMO for Bound Constrained Single-Objective Computationally Expensive Numerical Optimization, *2015 IEEE Congress on evolutionary Computation*, (2015) pp. 1011-1017.
- [48] M. Andersson, S. Bandaru, A. Ng, A. Syberfeldt. Parameter tuned CMA-ES on the CEC'15 expensive problems, *2015 IEEE Congress on Evolutionary Computation*, (2015) pp. 1950-1957.
- [49] M. R. Tanweer, S. Suresh, N. Sundararajan. Improved srpso algorithm for solving cec 2015 computationally expensive numerical optimization problems, *2015 IEEE Congress on Evolutionary Computation*, (2015) pp. 1943-1949.
- [50] Momani, Shaher & Abo-Hammour, Zaer & Alsmadi, Othman. Solution of Inverse Kinematics Problem using Genetic Algorithms. *Applied Mathematics & Information Sciences*. (2015) 10. 1-9.
- [51] Za'er S. Abo-Hammour, Othman MK. Alsmadi, Sofian I. Bataineh, Muhannad A. Al-Omari & Nafee' Affach. Continuous Genetic Algorithms for Collision-Free Cartesian Path Planning of Robot Manipulators. *International Journal of Advanced Robotic Systems*. (2011). Vol 8. No 6. 14-36.
- [52] E. Sandgren, nonlinear integer and discrete programming in mechanical design, *Proceeding of the ASME Design Technology Conference*, Kissimmee, FL, pp. 95-105, 1988.
- [53] Garg. Harish. Solving Structural Engineering Design Optimization Problems Using an Artificial Bee Colony Algorithm. *Journal of Industrial and Management Optimization*. Volume 10, Number 3, July 2014.
- [54] K. Deb. Optimal design of a welded beam via genetic algorithms, *AIAA Journal*, Vol. 29, No. 11, pp. 2013-2015, 1991.
- [55] A. D. Belegundu. A Study of Mathematical Programming Methods for Structural Optimization. Department of Civil and Environmental Engineering, University of Iowa, Iowa City, Iowa, 1982.
- [56] Golinski, J. Optimal Synthesis Problems Solved by Means of Nonlinear Programming and Random Methods. *Journal of Mechanisms*, Vol. 5, pp. 287-309, 1970.
- [57] M. Zhang, W. Luo and X. Wang, Differential evolution with dynamic stochastic selection for constrained optimization, *Information Sciences*, 178 (2008), 3043–3074.

- [58] Joaquín Derrac, Salvador García, Daniel Molina, Francisco Herrera. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm and Evolutionary Computation* 1 (2011), 3–18.
- [59] S. A. Amanchi, *Applied Nonparametric Statistical Tests to Compare Evolutionary and Swarm Intelligence Approaches*. Master of Science, Department of Computer Science, North Dakota State University, 2014.
- [60] M. G. H. Omran and A. Salman, Constrained optimization using CODEQ, *Chaos, Solitons & Fractals*, 42 (2009), 662–668.
- [61] A. R. Hedar and M. Fukushima, Derivative - free filter simulated annealing method for constrained continuous global optimization, *Journal of Global Optimization*, 35 (2006), 521–549.
- [62] M. Mahdavi, M. Fesanghary and E. Damangir, An improved harmony search algorithm for solving optimization problems, *Applied Mathematics and Computation*, 188 (2007), 1567–1579.
- [63] K. M. Ragsdell and D. T. Phillips, Optimal design of a class of welded structures using geometric programming, *ASME Journal of Engineering for Industries*, 98 (1976), 1021-1025.
- [64] S. S. Rao, *Engineering Optimization: Theory and Practice*, 3rd edition, John Wiley & Sons, Chichester, 1996.
- [65] T. Ray and K. M. Liew, Society and civilization: An optimization algorithm based on the simulation of social behavior, *IEEE Transactions on Evolutionary Computation*, 7 (2003), 386–396.
- [66] S. F. Hwang and R. S. He, A hybrid real-parameter genetic algorithm for function optimization, *Advanced Engineering Informatics*, 20 (2006), 7–21.
- [67] Ming-Hua Lin, Jung-Fa Tsai, Nian-Ze Hu, Shu-Chuan Chang. *Design Optimization of a Speed Reducer Using Deterministic Techniques*. *Mathematical Problems in Engineering*. Hindawi Publishing Corporation. (2013)
- [68] J. S. Arora, *Introduction to Optimum Design*, McGraw-Hill, New York, 1989.
- [69] C. A. C. Coello, Use of a self -adaptive penalty approach for engineering optimization problems, *Computers in Industry*, 41 (2000), 113–127.
- [70] C. A. C. Coello and E. M. Montes, Constraint- handling in genetic algorithms through the use of dominance-based tournament selection, *Advanced Engineering Informatics*, 16(2002), 193–203.
- [71] T. Ray and P. Saini, Engineering design optimization using a swarm with an intelligent information sharing among individuals, *Engineering Optimization*, 33 (2001), 735–748.
- [72] X. H. Hu, R. C. Eberhart and Y. H. Shi, Engineering optimization with particle swarm, *Proceedings of the 2003 IEEE Swarm Intelligence Symposium*, (2003), 53–57.
- [73] Q. He and L. Wang, An effective co - evolutionary particle swarm optimization for constrained engineering design problems, *Engineering Applications of Artificial Intelligence*, 20 (2007), 89–99.
- [74] L. C. Cagnina, S. C. Esquivel and C. A. C. Coello, Solving engineering optimization problem with the simple constrained particle swarm optimizer, *Informatica*, 32 (2008), 319–326.
- [75] E. M. Montes and C. A. C. Coello, An empirical study about the usefulness of evolution strategies to solve constrained optimization problems, *International Journal of General Systems*, 37 (2008), 443–473.
- [76] E. M. Montes, C. A. C. Coello, J. V. ReSim and L. M. Davila, Multiple trial vectors in differential evolution for engineering design, *Engineering Optimization*, 39 (2007), 567 - 589.
- [77] A. Kaveh and S. Talatahari, An improved ant colony optimization for constrained engineering design problems, *Engineering Computations*, 27 (2010), 155–182.
- [78] L. S. Coelho, Gaussian quantum-behaved particle swarm optimization approaches for constrained engineering design problems, *Expert Systems with Applications*, 37 (2010), 1676–1683.
- [79] B. Akay and D. Karaboga, Artificial bee colony algorithm for large-scale problems and engineering design optimization, *Journal of Intelligent Manufacturing*, 23 (2012), 1001–1014.

- [80] S. Akhtar, K. Tai, and T. Ray, A socio-behavioral simulation model for engineering design optimization, *Engineering Optimization*, vol. 34, No. 4, pp. 341–354, 2002.
- [81] S. S. Rao and Y. Xiong, A hybrid genetic algorithm for mixed discrete design optimization, *Journal of Mechanical Design*, vol. 127, No. 6, pp. 1100–1112, 2005.
- [82] M. Jaberipour and E. Khorram, Two improved harmony search algorithms for solving engineering optimization problems, *Communications in Nonlinear Science and Numerical Simulation*, vol. 15, No. 11, pp. 3316–3331, 2010.
- [83] H. L. Li and P. Papalambros, A production system for use of global optimization knowledge, *Journal of Mechanisms, Transmissions, and Automation in Design*, vol. 107, No. 2, pp. 277–284, 1985.
- [84] S. Tosserams, L. F. P. Etman, and J. E. Rooda, An augmented Lagrangian decomposition method for quasi-separable problems in MDO, *Structural and Multidisciplinary Optimization*, vol. 34, No. 3, pp. 211–227, 2007.
- [85] S. Lu and H. M. Kim, A regularized inexact penalty decomposition algorithm for multidisciplinary design optimization problems with complementarity constraints, *Journal of Mechanical Design*, vol. 132, No. 4, Article ID 041005, 12 pages, 2010.
- [86] A. H. Gandomi, X. S. Yang, and A. H. Alavi, Mixed variable structural optimization using firefly algorithm, *Computers & Structures*, 89 (2011), 2325–2336.
- [87] B.K. Kannan, S.N. Kramer, An augmented lagrange multiplier based method for mixed integer discrete continuous optimization and its applications to mechanical design, *Trans. ASME, J. Mech. Des.* 116 (1994) 318–320.
- [88] K. Deb, A.S. Gene, A robust optimal design technique for mechanical component design, in: D. Dasgupta, Z. Michalewicz (Eds.), *Evolutionary Algorithms in Engineering Applications*, Springer, Berlin, (1997), pp. 497–514.
- [89] A. Kaveh, S. Talatahari, Engineering optimization with hybrid particle swarm and ant colony optimization, *Asian J. Civil Eng. (Build. Hous.)* 10 (2009)
- [90] A. Gandomi, X.S. Yang, A. Alavi, Cuckoo search algorithm: a metaheuristic approach to solve structural optimization problems, *Eng. Comput.* (2011a) 1–19.
- [91] S. He, E. Prempan and Q. H. Wu, An improved particle swarm optimizer for mechanical design optimization problems, *Engineering Optimization*, 36 (2004), 585–605.
- [92] K. Deb, An efficient constraint handling method for genetic algorithms, *Computer Methods in Applied Mechanics and Engineering*, 186 (2000), 311–338.
- [93] V. K. Mehta. B. Dasgupta, A constrained optimization algorithm based on the simplex search method, *Engineering Optimization*, 44 (2012), 537–550.
- [94] Q. He, L. Wang, An effective co-evolutionary particle swarm optimization for constrained engineering design problems, *Engineering Applications of Artificial Intelligence*, 20 (2007) 89–99.
- [95] Mohamed, A.W. A novel differential evolution algorithm for solving constrained engineering optimization problems. *Journal of Intelligent Manufacturing*. 2017.
- [96] Mohamed, A.W., Sabry, H. Z. Constrained optimization based on modified differential evolution algorithm. *Information Sciences*, 194, (2012) 171–208.
- [97] Wang, L., & Li, L. An effective differential evolution with level comparison for constrained engineering design. *Structural and Multidisciplinary Optimization*, 41, (2010) 947–963.
- [98] Wang, Y., Cai, Z., Zhou, Y. Accelerating adaptive trade-off model using shrinking space technique for constrained evolutionary optimization. *International Method for Numerical Methods in Engineering*, 77(11), (2009) 1501–1534.

- [99] Melo, V. V., Carosio, G. L. Investigating multi-view differential evolution for solving constrained engineering design problems. *Expert Systems with Applications*, 40(9), (2013) 3370–3377.
- [100] Gong, W., Cai, Z., Liang, D. Engineering optimization by means of an improved constrained differential evolution. *Computer Methods in Applied Mechanics and Engineering*, 268, (2014) 884–904.
- [101] Sadollah, A., Bahreininejada, A., Eskandar, H., Hamdi, M. Mine blast algorithm: A new population-based algorithm for solving constrained engineering optimization problems. *Applied Soft Computing*, 13, (2013) 2592–261.
- [102] Eskandar, H., Sadollah, A., Bahreininejad, A., & Hamdi, M. Water cycle algorithm—A novel metaheuristic optimization method for solving constrained engineering optimization problems. *Computers & Structures*, 110–111, (2012) 151–166.
- [103] K. Deb, M. Goyal, A combined genetic adaptive search (GeneAS) for engineering design, *Comput. Sci. Inf.* 26 (4) (1996) 30–45.