

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

**ESTUDO DA PARAMETRIZAÇÃO DO ALGORITMO HÍBRIDO BASEADO
NO ALGORITMO CULTURAL COM ALGORITMO GENÉTICO EM UMA
ABORDAGEM MULTIPOPULACIONAL**

JOAQUIM ALBERTO LEITE DA SILVA JUNIOR

DM 34/2015

**UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2015**

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

**ESTUDO DA PARAMETRIZAÇÃO DO ALGORITMO HÍBRIDO BASEADO
NO ALGORITMO CULTURAL COM ALGORITMO GENÉTICO EM UMA
ABORDAGEM MULTIPULACIONAL**

JOAQUIM ALBERTO LEITE DA SILVA JUNIOR

DM 34/2015

**UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2015**

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

JOAQUIM ALBERTO LEITE DA SILVA JUNIOR

**ESTUDO DA PARAMETRIZAÇÃO DO ALGORITMO HÍBRIDO BASEADO
NO ALGORITMO CULTURAL COM ALGORITMO GENÉTICO EM UMA
ABORDAGEM MULTIPOPULACIONAL**

Dissertação de Mestrado
submetida à Banca Examinadora
do Programa de Pós-Graduação
em Engenharia Elétrica da UFPA
para a obtenção do Grau de
Mestre em Engenharia Elétrica na
área de Computação Aplicada.

**UFPA / ITEC / PPGEE
Campus Universitário do Guamá
Belém-Pará-Brasil
2015**

Dados Internacionais de Catalogação-na-Publicação (CIP)
Sistema de Bibliotecas da UFPA

Silva Junior, Joaquim Alberto Leite da, 1980-
Estudo da parametrização do algoritmo híbrido
baseado no algoritmo cultural com algoritmo genético em
uma abordagem multipopulacional / Joaquim Alberto Leite
da Silva Junior. - 2015.

Orientador: Roberto Célio Limão de Oliveira.
Dissertação (Mestrado) - Universidade
Federal do Pará, Instituto de Tecnologia,
Programa de Pós-Graduação em Engenharia
Elétrica, Belém, 2015.

1. Otimização combinatória. 2. Algoritmos
genéticos. 3. Computação evolucionária. I.
Título.

CDD 22. ed. 519.64

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

**“ESTUDO DA PARAMETRIZAÇÃO DO ALGORITMO HÍBRIDO
BASEADO NO ALGORITMO CULTURAL COM ALGORITMO GENÉTICO
EM UMA ABORDAGEM MULTIPOPULACIONAL”**

AUTOR: JOAQUIM ALBERTO LEITE DA SILVA JUNIOR

DISSERTAÇÃO DE MESTRADO SUBMETIDA À BANCA EXAMINADORA APROVADA PELO
COLEGIADO DO PROGRAMA DE PÓS-GRADUAÇÃO EM ENGENHARIA ELÉTRICA, SENDO
JULGADA ADEQUADA PARA A OBTENÇÃO DO GRAU DE MESTRE EM ENGENHARIA
ELÉTRICA NA ÁREA DE COMPUTAÇÃO APLICADA.

APROVADA EM: 01/10/2015

BANCA EXAMINADORA:



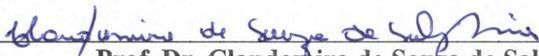
Prof. Dr. Roberto Célio Limão de Oliveira

(Orientador – PPGEE/UFPA)



Prof.ª Dr.ª Adriana Rosa Garcez Castro

(Avaliadora Interna – PPGEE/UFPA)



Prof. Dr. Claudomiro de Souza de Sales Junior

(Avaliador Externo ao Programa – PPGCC/UFPA)



Prof. Dr. Deam James Azevedo da Silva

(Avaliador Externo – UFOPA)

VISTO:

Prof. Dr. Evaldo Gonçalves Pelaes

(Coordenador do PPGEE/ITEC/UFPA)

DEDICATÓRA

Dedico este trabalho a DEUS, meu Senhor que me sustentou em todos os momentos para o desenvolvimento deste trabalho. Dedico também a minha família, esposa e filhos, ao qual estimo muito. Estendo essa dedicatória aos meus pais e irmãos e por fim ao meu orientador Prof. Dr. Roberto Limão que me orientou na elaboração deste intento.

AGRADECIMENTOS

Agradeço em primeiro lugar ao meu Senhor Jesus Cristo por ter me dado entendimento e sabedoria na persistência e concretização deste trabalho.

A minha esposa Kamille Rodrigues, aos meus filhos Gustavo Rodrigues e Guilherme Rodrigues por terem me apoiado e compreendido nos momentos mais difíceis e pela compreensão e amor.

A minha querida mãe Dilma de Souza Silva – in memory, meu pai Joaquim Leite e irmãos Jean Carlo e Carlos Américo que me incentivam em todos os momentos da minha vida.

Aos Professores Dr. Roberto Célio Limão de Oliveira e Dr. Deam James Azevedo da Silva, pela orientação e condução com excelência durante o curso.

Ao professor Dr. Jandecy Cabral Leite, do Instituto de Tecnologia e Educação Galileo da Amazônia - ITEGAM e a todos os professores do Programa de Pós-Graduação em Engenharia Elétrica do ITEC/PPGEE da Universidade Federal do Pará - UFPA, por ter creditado em nós, mestrandos, a confiança pela oportunidade de estar no seio dessa Instituição de ensino agregando valores de qualificação e pesquisa e desenvolvimento dos estudantes da Amazônia.

E a todos que, diretamente e indiretamente, contribuíram para o desenvolvimento e conclusão deste trabalho.

Este trabalho foi desenvolvido com o apoio do Governo do Estado do Amazonas por meio Fundação de Amparo à Pesquisa do Estado do Amazonas, com a concessão de bolsa de estudo.

SUMÁRIO

LISTA DE FIGURAS	X
LISTA DE TABELAS.....	XII
LISTA DE SIGLAS	XIII
RESUMO	XIV
ABSTRACT	XV
CAPÍTULO 1. INTRODUÇÃO.....	1
1.1. CONSIDERAÇÕES INICIAIS	1
1.2. JUSTIFICATIVA.....	1
1.3. RELEVÂNCIA DO TRABALHO	3
1.4. OBJETIVOS.....	4
1.5. DELIMITAÇÃO DA PESQUISA	4
1.6. ESCOPO DO TRABALHO.....	4
CAPÍTULO 2. ALGORITMO GENÉTICO (AG) E ALGORITMO CULTURAL (AC)	6
2.1. BREVE HISTÓRICO DOS ALGORITMOS GENÉTICOS.....	6
2.1.1. CARACTERÍSTICAS DOS ALGORITMOS GENÉTICOS	7
2.1.2. CONSIDERAÇÕES SOBRE OS ALGORITMOS GENÉTICOS	10
2.2. ALGORITMO CULTURAL	12
2.2.1. INTRODUÇÃO	12
2.2.2. CARACTERÍSTICAS DOS ALGORITMOS CULTURAIS	16
2.2.3. MICROEVOLUÇÃO X MACROEVOLUÇÃO	19
2.3. ALGORITMO HÍBRIDO	23
CAPÍTULO 3. ALGORITMO CULTURAL COM MODELO DE ILHAS BASEADO EM ALGORITMO GENÉTICO MULTIPOPULACIONAL (AC-MI) .	26
CAPÍTULO 4. METODOLOGIA E RESULTADOS.....	33
4.1. PROBLEMA DA MOCHILA MULTIDIMENSIONAL (PMM).....	33
4.2. METODOLOGIA.....	35

4.3. EXPERIMENTOS NO ALGORITMO CULTURAL COM MODELO DE ILHAS BASEADO EM ALGORITMO GENÉTICO MULTIPOPULACIONAL (AC-MI)	37
4.4. EXPERIMENTOS DE COMPARAÇÃO DE DESEMPENHO DO AC-MI COM O ALGORITMO GENÉTICO SIMPLES (AG) E COM O ALGORITMO CULTURAL PADRÃO (AC)	42
CAPÍTULO 5. CONCLUSÃO	53
5.1. TRABALHOS FUTUROS.....	54
REFERÊNCIAS BIBLIOGRÁFICAS	55
ANEXOS.....	62
A.1. VISÃO GERAL DO FRAMEWORK UTILIZADO (JAVA EVOLUTIONARY FRAMEWORK - JEF)	62
A.1.1. ESTRUTURA PRINCIPAL DO FRAMEWORK	63
A.1.2. EXEMPLO DE UMA ESTRUTURA DE REPRESENTAÇÃO PARA O CROMOSSOMO.....	64
A.1.3. ESTRUTURA DE REPRESENTAÇÃO DOS ALGORITMOS EVOLUTIVOS.....	65
A.1.4. ARQUIVO DE PARÂMETROS DO FRAMEWORK EM XML.....	66
ARTIGOS PUBLICADOS.....	68

LISTA DE FIGURAS

Figura 1 : Pseudocódigo do Algoritmo Genético (REYNOLDS, 2003).	8
Figura 2 : Funcionamento Básico de um Algoritmo Cultural (REYNOLDS et al., 2010).....	16
Figura 3 : Pseudocódigo do Algoritmo Cultural (REYNOLDS, 2003).	17
Figura 4 : Representação do Conhecimento Normativo (IACOBAN; REYNOLDS; BREWSTER, 2003).	21
Figura 5 : Representação do Conhecimento Situacional (IACOBAN; REYNOLDS; BREWSTER, 2003).	21
Figura 6 : Representação do Conhecimento Topográfico (BECERRA; COELLO, 2005).....	22
Figura 7 : Representação do Conhecimento Histórico (BECERRA; COELLO, 2005).....	23
Figura 8 : Modelo de Ilhas (WHITLEY, 1993).	27
Figura 9 : Topologia de Comunicação +1+2 (SILVA, 2012).	28
Figura 10 : Estratégia Populacional Mestre-Escravo (STACH, 2007).	29
Figura 11 : Modelo de Comunicação do Algoritmo Cultural Multipopulacional (SILVA, 2012).	31
Figura 12 : Possíveis Variações dos Parâmetros do AC-MI Utilizadas nos Experimentos.	36
Gráfico 1 : Função <i>fitness</i> (Hp1).....	48
Gráfico 2 : Função <i>fitness</i> (Hp2).....	48
Gráfico 3 : Função <i>fitness</i> (Pb2).	48
Gráfico 4 : Função <i>fitness</i> (Petersen2).	49
Gráfico 5 : Função <i>fitness</i> (Petersen3).	49
Gráfico 6 : Função <i>fitness</i> (Sento1).	49
Gráfico 7 : Função <i>fitness</i> (Sento2).	50
Gráfico 8 : Função <i>fitness</i> (Weing1).	50
Gráfico 9 : Função <i>fitness</i> (Weing2).	50

Gráfico 10 : Função <i>fitness</i> (Weish1).	51
Gráfico 11 : Função <i>fitness</i> (Weish2).	51

LISTA DE TABELAS

Tabela 1 : Os Melhores Resultados para o Problema da Mochila em Configurações <i>Benchmarks</i> pelo Método AC-MI Padrão.	38
Tabela 2 : Resultados Obtidos para o Problema da Mochila em Configurações <i>Benchmarks</i> pelo Método AC-MI com Variação dos seus Parâmetros.	41
Tabela 3 : Problemas Utilizados nos Experimentos.	42
Tabela 4 : Variação dos Parâmetros para os Problemas Utilizados nos Experimentos.	43
Tabela 5 : Resultados para Todos os <i>Benchmarks</i> da Mochila pelo Método AG.	44
Tabela 6 : Resultados para Todos os <i>Benchmarks</i> da Mochila pelo método AC.	45
Tabela 7 : Resultados para Todos os <i>Benchmarks</i> pelo Método AC-MI com a sua Melhor Parametrização para cada uma das Instâncias Utilizadas do Problema da Mochila Multidimensional.	46

LISTA DE SIGLAS

AC	Algoritmo Cultural
AC-MI	Algoritmo Cultural com Modelo de Ilhas
AEs	Algoritmos Evolutivos
AG	Algoritmo Genético
AMs	Algoritmos Meméticos
CE	Computação Evolutiva
CM	Computação Memética
CN	Computação Natural
EEs	Estratégias Evolucionárias
PMM	Problema da Mochila Multidimensional

RESUMO

A finalidade deste trabalho é analisar a aplicação de um algoritmo cultural híbrido, com população gerada pelo algoritmo genético, de característica multipopulacional, ou ainda, mais precisamente, desenvolver uma parametrização do algoritmo híbrido baseado no algoritmo cultural com algoritmo genético para o problema da mochila multidimensional. Tem-se como objetivo encontrar os melhores parâmetros do algoritmo cultural híbrido, com modelo de ilhas (característica multipopulacional), aplicado ao problema de otimização combinatório denominado de “Mochila Multidimensional”. São executados vários experimentos para efetuar uma avaliação em relação ao desempenho desses mecanismos híbridos com outros algoritmos disponíveis na literatura.

Palavras-Chave: Algoritmos Genéticos, Algoritmos Culturais, Problema da Mochila Multidimensional, Multipopulação.

ABSTRACT

The purpose of this paper is to analyze the application of a hybrid cultural algorithm with population generated by multipopulation feature of the genetic algorithm, or, more specifically, to develop a parameter of the hybrid algorithm based on cultural algorithm with genetic algorithm for multidimensional knapsack problem in areas of computer science and computational intelligence. The aim of this work is to find the best parameters for hybrid cultural algorithm and over genetic algorithm, with model of islands (multipopulation characteristic) applied to combinatorial optimization problem called "Multidimensional Knapsack". Several experiments are performed to make an assessment regarding of these mechanisms hybrids with other algorithms available in the literature.

Keywords: Genetic Algorithms, Cultural Algorithms, Multidimensional Knapsack Problem, Multipopulation.

CAPÍTULO 1. INTRODUÇÃO

1.1. CONSIDERAÇÕES INICIAIS

Em muitos problemas de otimização há dificuldades em alcançar um resultado ótimo ou mesmo um resultado próximo ao valor ótimo em um tempo viável, principalmente quando se trabalha em grande escala. Por isso muitos desses problemas são abordados por heurísticas ou meta heurísticas que executam buscas por melhores soluções dentro do espaço de busca definido (YANG, 2010). Na Ciência da Computação existe a área denominada de Computação Natural onde se encontra um grande conjunto de meta heurísticas adequadas a resolver esses problemas de otimização complexos. A computação natural, linha de pesquisa que, baseada ou inspirada na natureza, permite o desenvolvimento de novas ferramentas de computação para a solução de problemas complexos; resulta na síntese de fenômenos (formas e comportamentos) naturais; e utiliza matéria-prima natural para o desenvolvimento de novas formas de computar (DE CASTRO, 2006). Dentro da computação natural estão os Algoritmos Culturais e os Algoritmos Genéticos, que são considerados meta heurísticas evolutivas que se complementam devido ao mecanismo dual de herança cultura e genética.

1.2. JUSTIFICATIVA

Esse trabalho tem como justificativa o fato de que a Otimização é uma ciência que está sempre em demanda e é empregada em todos os campos de aplicações tais como: engenharia civil, mecânica, automobilística, aérea, econômica, eletrônica, química, etc., (LINDEN, 2008). A otimização está em toda parte, envolvendo desde projetos de engenharia até aplicações para os mercados financeiros (NABHOLZ, 2006; GEN e CHENG, 2000).

Uma organização sempre deseja maximizar seus lucros, minimizar custos e maximizar o desempenho. Mesmo no cotidiano a otimização está presente quando, por exemplo, uma pessoa pretende maximizar sua satisfação com um menor custo no planejamento de férias. Portanto, a busca por soluções ideais é um processo realizado constantemente para cada problema encontrado, embora possa ocorrer incapacidade de se encontrar essas possíveis soluções.

A otimização pode ser descrita como o ato de obter o melhor resultado possível em determinadas circunstâncias. A palavra “otimização” tem origem latina, e significa ótimo final. Similarmente, “*optimus*” significa o melhor. Portanto, otimização refere-se à tentativa de trazer o melhor resultado de um problema em questão (ANDRÉASSON, EVGRAFOV e PATRIKSSON, 2005).

Em termos mais gerais, a teoria de otimização é um conjunto de resultados matemáticos e métodos numéricos para encontrar e identificar o melhor candidato de uma coleção de alternativas, sem ter que explicitamente enumerar e avaliar todas as alternativas possíveis. O processo de otimização está na raiz da engenharia, onde a função clássica do engenheiro é projetar novos sistemas cada vez melhores, mais eficientes e menos dispendiosos, bem como elaborar planos e procedimentos para melhorar o funcionamento dos sistemas existentes.

O poder dos métodos de otimização em determinar o melhor caso, sem realmente testar todos os possíveis casos, vem por meio do uso de um nível modesto de matemática e da realização de cálculos numéricos iterativos usando procedimentos lógicos claramente definidos ou Algoritmos implementados em máquinas de computação (RAVINDRAN, RAGSDELL e REKLAITIS, 2007).

Os métodos de otimização modernos, também, por vezes, chamados métodos de otimização não tradicionais, surgiram como métodos populares para resolver problemas de otimização complexos de engenharia nos últimos anos (RAO, 2009). Seu surgimento tem relação com a estratégia de solução baseada em heurísticas. A heurística engloba estratégias, procedimentos, métodos de aproximação de solução por tentativa e erro, sempre na procura da melhor forma de chegar a um determinado fim para produzir soluções aceitáveis para um problema complexo.

A complexidade do problema de interesse torna impossível a busca de cada solução possível ou combinação das mesmas. O objetivo é encontrar boas soluções viáveis em uma escala de tempo aceitável mesmo que não exista garantia de que as melhores soluções possam ser encontradas (RAO, 2009).

Uma classe de otimizadores não tradicionais, chamados de Algoritmos Evolutivos (AEs), ou simplesmente técnicas de Computação Evolutiva (CE), que é

uma sub área da Computação Natural (CN), tem sido estudados (BRAUN, 1991; CHUNG e REYNOLDS, 1996; BECERRA, 2002; ASTOLFI, 2004; HE e WANG, 2007; YU e GEN, 2010). Os AEs são técnicas de Computação Bioinspirada, fundamentadas em conceitos biológicos (YU e GEN, 2010). As técnicas evolutivas apresentam conceitos cuja origem está em diversos campos da Biologia, especialmente em ideias evolucionistas e na Genética (IBA e NOMAN, 2012). E então os Algoritmos Evolutivos cumprem seu papel dentro do contexto de Otimização.

1.3. RELEVÂNCIA DO TRABALHO

A seguir, tem-se a relevância do trabalho, na qual trata-se do Problema da Mochila Multidimensional (PMM), problema de otimização a ser estudado. Os problemas da mochila são uns dos mais importantes em programação linear inteira e têm sido estudados nos últimos anos por vários investigadores (MARTELLO e TOTH, 1990; PISINGER, 1995; KRAUSE *et al.*, 2013; LEÃO *et al.*, 2014).

Os problemas da mochila são aplicáveis em problemas de embalagem, de carregamento de equipamentos, de corte de materiais, de controle orçamentário e de seleção de projetos de investimento. O problema da mochila também é interessante por constituir um subproblema de modelos mais vastos, como por exemplo, os de constituição de tripulações de voo, de planejamento da produção, de problemas de partição e de concepção de circuitos eletrônicos.

A primeira resolução do problema da mochila, por técnicas mais inteligentes, data dos anos 50, por aplicação da função recursiva (Programação Dinâmica) de Bellman (BELLMAN, 1957). A partir de então, foram propostos inúmeros melhoramentos (MARTELLO e TOTH, 1990; GOMES DA SILVA *et al.*, 2003). Toda a classe de problemas da mochila pertence à família dos problemas NP - Difícil, porém, utilizando Algoritmos Evolutivos, diversos problemas desta classe podem ser resolvidos em tempo pseudo-polinomial (GORDON *et al.*, 1994; KHURI *et al.*, 1994; FUKUNAGA e TAZOE, 2009; GONG *et al.*, 2007; JOHN *et al.*, 2014).

1.4. OBJETIVOS

O tempo de execução, nestes casos, está diretamente relacionado com a dimensão das instâncias, com número de critérios e com o tamanho da mochila. No trabalho de SILVA (2012) o Problema da Mochila Multidimensional (PMM) foi abordado com uma solução por meio do Algoritmo Cultural com Modelo de Ilha Baseado em Algoritmo Genético Multipopulacional (AC-MI). O AC-MI resolve o PMM com eficiência, entretanto fica aberto a escolha de melhores parâmetros para o AC-MI. Essa lacuna é preenchida com o objetivo desse trabalho atual, onde mostra-se a melhor parametrização do Algoritmo Cultural com Modelo de Ilhas Baseado em Algoritmo Genético Multipopulacional (AC-MI) para resolver o Problema da Mochila Multidimensional (PMM).

1.5. DELIMITAÇÃO DA PESQUISA

Neste trabalho serão utilizadas algumas instâncias do problema da mochila multidimensional, disponíveis no repositório OR-Library (BEASLEY, 1990), que tenham resultados já publicados na literatura, para fins de comparação de desempenho. Dadas as instâncias, será realizada uma grande bateria de testes para medir o desempenho do algoritmo estudado. Estes testes serão realizados com 50 ensaios para cada uma das 50 populações iniciais geradas. Este conjunto de testes será feito com a variação de cada um dos parâmetros do Algoritmo Cultural com Modelo de Ilhas Baseado em Algoritmo Genético Multipopulacional (AC-MI).

1.6. ESCOPO DO TRABALHO

No **Capítulo I**, introdução: são apresentados de forma sucinta, os elementos que motivaram a realização deste trabalho juntamente com a justificativa, relevância e contribuição da dissertação. Apresenta ainda os objetivos, delimitação da pesquisa e o escopo do trabalho.

O **Capítulo II**, refere-se à revisão bibliográfica e o estado da arte por meio de pesquisas correlatas no acervo literário técnico nacional e internacional sobre o tema de Algoritmo Genético (AG) e Algoritmo Cultural (AC).

O **Capítulo III**, é orientado ao estudo, conceitos e as definições do Algoritmo Cultural com Modelo de Ilhas Baseado em Algoritmo Genético Multipopulacional

(AC-MI) e suas características.

No **Capítulo IV**, abordam-se os resultados, apresenta-se o conceito e as definições do Problema da Mochila Multidimensional (PMM) e metodologia da pesquisa enfocando a parametrização do Algoritmo Cultural com Modelo de Ilhas Baseado em Algoritmo Genético Multipopulacional (AC-MI) para o Problema da Mochila Multidimensional (PMM).

Por último, o **Capítulo V**, apresentam-se as conclusões proporcionadas pelos resultados da dissertação e as recomendações para possíveis desdobramentos do Algoritmo Cultural com Modelo de Ilhas Baseado em Algoritmo Genético Multipopulacional (AC-MI) para o Problema da Mochila Multidimensional (PMM).

CAPÍTULO 2. ALGORITMO GENÉTICO (AG) E ALGORITMO CULTURAL (AC)

2.1. BREVE HISTÓRICO DOS ALGORITMOS GENÉTICOS

Conforme descrito pela Biologia, a teoria da evolução diz que o meio ambiente seleciona, em cada geração, os seres vivos mais aptos de uma população. Como resultado, somente os mais aptos conseguem se reproduzir, já que normalmente os menos aptos tendem a ser eliminados antes de gerarem descendentes. Durante a reprodução, ocorre, entre outros, fenômenos como mutações e cruzamentos, que atuam sobre o material genético armazenado nos cromossomos. O resultado destes fenômenos levam à variabilidade dos seres vivos da população. Sobre esta população diversificada age a seleção natural, permitindo a sobrevivência apenas dos indivíduos melhor adaptados.

Em meados do século XIX surgiu uma das mais importantes teorias no campo da biologia sobre os aspectos do surgimento e evolução da vida, a Teoria da Seleção Natural.

Tal teoria foi proposta por Charles Darwin no livro *The Origin of Species* em 1859. Darwin defendia que os indivíduos mais bem adaptados ao meio no qual estão inseridos seriam os que teriam maior chance de sobrevivência, em contra partida, os menos adaptados tenderiam a extinguir-se. Com suporte na teoria da seleção natural, as décadas de 1930 e 1940 foram fundamentais para as primeiras tentativas de representação da teoria através de um modelo matemático. Os estudos se deram inicialmente por matemáticos e biólogos que desenvolveram um princípio que determinava como a variabilidade de indivíduos de uma mesma população se reproduz através de dois fatores: mutação e recombinação genética.

Ainda, nos anos 1950 e 1960 cientistas já começavam a estudar sistemas evolucionários como a ideia de que a evolução poderia ser usada como princípio para o desenvolvimento de ferramentas de otimização para problemas de engenharia. Foi na década de 1970 que o professor norte-americano John Henry Holland começou a pesquisar a possibilidade de se incorporar os princípios da evolução natural em um programa de computador que fosse capaz, através de simulações, de resolver problemas complexos assim como ocorre na natureza. A

partir de então, Holland acompanhado de seus alunos e colegas da Universidade de Michigan, dedicaram-se ao estudo dos Algoritmos Genéticos.

O algoritmo proposto por eles era capaz de resolver problemas complexos, mesmo não tendo o conhecimento do tipo de problema que estava sendo resolvido. Em 1975, John Holland, lançou seu primeiro livro “*Adaptation in Natural and Artificial Systems*”, hoje considerado a Bíblia de Algoritmos Genéticos. Ainda nos anos 1980, o ex aluno de Holland, David Goldberg (GOLDBERG, 1989), editou o livro “*Genetic Algorithms in Search Optimization, and Machine Learning*” tornando a técnica de Algoritmos Genéticos mais detalhada e popular.

A computação evolutiva é um dos maiores avanços científicos na computação bioinspirada. O seu passo mais importante foi o desenvolvimento de Algoritmos Evolutivos na década de 1960 e 1970, durante a qual John Holland e seus colaboradores da Universidade de Michigan, desenvolveram o Algoritmo Genético (HOLLAND, 1975). Já em 1962, na Holanda, se estudavam os sistemas adaptativos e foi onde primeiro se usou manipulações do tipo recombinação, frequentemente referenciado como *crossover* para a modelagem de tais sistemas (YANG, 2010). O resumo do desenvolvimento de Algoritmos Genéticos foi publicado em 1975. No mesmo ano, Kenneth De Jong terminou sua importante tese mostrando o potencial e o poder dos Algoritmos Genéticos para uma ampla gama de funções objetivo, com características multimodais ou mesmo descontínuos (YANG, 2011).

2.1.1. CARACTERÍSTICAS DOS ALGORITMOS GENÉTICOS

Um Algoritmo Genético é uma técnica de busca utilizada na ciência da computação para achar soluções aproximadas em problemas de otimização e busca, fundamentado principalmente pelo americano John Henry Holland. Os Algoritmos Genéticos são uma classe particular de Algoritmos Evolutivos que usam técnicas inspiradas pela Biologia evolutiva, como a hereditariedade, mutação, seleção natural e recombinação genética (*crossing over*).

Um Algoritmo Genético é composto por várias etapas, entre as principais estão: Inicialização da População, Avaliação da População, Operadores Genéticos, Substituição e Critério de Parada. Após inicializar a população, a cada geração o Algoritmo Genético irá selecionar os indivíduos mais aptos, para então, aplicar sobre

estes os operadores de recombinação e mutação com isso gerando a prole que irá servir para compor a geração seguinte. Este processo será repetido iterativamente até que a condição de parada seja alcançada (normalmente quando o número máximo de gerações for atingida ou qualquer condição desenvolvida à critério do usuário).

Em um Algoritmo Genético a população inicial é criada após a execução do operador denominado inicialização. A partir deste operador uma população com n indivíduos é criada, sendo que cada um desses indivíduos serão considerados como cromossomos dentro desta população. Cada indivíduo deverá apresentar um conjunto de genes (conhecidos também como genótipo) e um conjunto de características observáveis (conhecida como fenótipo do indivíduo). Logo, o fenótipo corresponde à interação do conteúdo genético com o ambiente, sendo que esta interação é representada pelo conjunto de parâmetros do Algoritmo Genético.

O pseudocódigo do Algoritmo Genético é apresentado na Figura 1, a seguir :

Algoritmo Genético (REYNOLDS, 2003)		
01:	INÍCIO	
02:	t=0	<i>;primeira geração</i>
03:	inicializar população P(t)	<i>;população inicial aleatória</i>
04:	avaliar população P(t)	<i>;calcula f(i) para cada indivíduo</i>
05:	ENQUANTO (não condição_fim) FAÇA	
06:	t=t+1	<i>;próxima geração</i>
07:	selecionar P(t) de P(t-1)	
08:	altera P(t)	<i>;crossover e mutação</i>
09:	avaliar população P(t)	<i>;calcula f(i) para cada indivíduo</i>
10:	FIMENQUANTO	
11:	FIM	

Figura 1 : Pseudocódigo do Algoritmo Genético (REYNOLDS, 2003).

O processo inicial de criação pode ser feito de diversas formas. No entanto, o mais usual é a criação de populações geradas aleatoriamente. Em alguns casos ainda, a introdução de heurísticas pode ser um fator importante para a resolução do problema. Nestes, costuma-se usar cromossomos com soluções aproximadas em meio ao restante da população. Quando usada tal técnica, deve-se atentar para a possibilidade de que o algoritmo possa vir a convergir prematuramente, fazendo com que em um curto espaço de tempo a população tenha indivíduos muito semelhantes.

A avaliação da população é realizada através da função de aptidão, esta, deve indicar a qualidade de cada indivíduo dentro da população, ou seja, o quão cada indivíduo está próximo da resolução do problema. De forma análoga ao que ocorre na natureza, em um algoritmo genético indivíduos com maior grau de adaptação tenderão a multiplicar suas características, enquanto indivíduos menos bem adaptados tenderão a extinguir-se.

Durante a avaliação de cada indivíduo tem-se a geração de um valor denominado *fitness* (ou adequação), este será responsável por ordenar e verificar o grau de aptidão do indivíduo para a solução do problema.

Um exemplo básico de *fitness* é : Supondo uma população inicial $X_0 = \{ X_{10}, X_{20}, \dots, X_{n0} \}$; Para cada solução individual (x_i) da população X , determinar o valor F_i , tal que $F_i = F(x_i) \rightarrow F(x) - \text{fitness function}$.

Em um Algoritmo Genético, vale lembrar que esta etapa é, para a maioria das aplicações, a fase mais crítica do processo, já que deve ser realizada para cada cromossomo de cada população durante todo o processo evolutivo.

Os Algoritmos Genéticos diferem dos algoritmos tradicionais de otimização em basicamente quatro aspectos:

- (A) Baseiam-se em uma codificação do conjunto das soluções possíveis, e não nos parâmetros da otimização em si;
- (B) Os resultados são apresentados como uma população de soluções e não como uma solução única;
- (C) Não necessitam de nenhum conhecimento derivado do problema, apenas de uma forma de avaliação do resultado;
- (D) Usam transições probabilísticas e não regras determinísticas.

O Algoritmo Genético pode convergir em uma busca de azar, porém sua utilização assegura que nenhum ponto do espaço de busca tem probabilidade zero de ser examinado. Toda tarefa de busca e otimização possui vários componentes, entre eles: o espaço de busca onde são consideradas todas as possibilidades de

solução de um determinado problema. As técnicas de busca e otimização tradicionais iniciam-se com um único candidato que, iterativamente, é manipulado utilizando algumas heurísticas (estáticas) diretamente associadas ao problema a ser solucionado.

2.1.2. CONSIDERAÇÕES SOBRE OS ALGORITMOS GENÉTICOS

Os Algoritmos Genéticos (AGs) formam a parte da área dos Sistemas Inspirados na Natureza, simulando os processos naturais e aplicando-os à solução de problemas reais. São métodos generalizados de busca e otimização que simulam os processos naturais de evolução, aplicando a ideia Darwiniana de seleção. De acordo com a aptidão e a combinação com outros operadores genéticos, são produzidos métodos de grande robustez e aplicabilidade. Esses algoritmos estão baseados nos processos genéticos dos organismos biológicos, codificando uma possível solução para, por exemplo, um problema de “cromossomo” composto por cadeia de bits e caracteres.

Esses cromossomos representam indivíduos que são levados ao longo de várias gerações, na forma similar aos problemas naturais, a evoluir de acordo com os princípios de seleção natural e sobrevivência dos mais aptos, descritos pela primeira vez por Charles Darwin em seu livro “Origem das Espécies” (publicado em 1859¹), acrescido da herança genética. Emulando estes processos, os Algoritmos Genéticos são capazes de “evoluir” sobre problemas do mundo real codificados, para promover soluções cada vez melhores em cada geração obtida.

De acordo com a teoria de Charles Darwin, o princípio da evolução favorece indivíduos melhores adaptados ao ambiente, proporcionando com isso que estes tenham maior possibilidade de longevidade e reprodução. Com isso, indivíduos mais

¹ A Origem das Espécies (em inglês: *On the Origin of Species*), do naturalista britânico Charles Darwin, apresenta a Teoria da Evolução. O nome completo da primeira edição (1859) é *On the Origin of Species by Means of Natural Selection, or the Preservation of Favoured Races in the Struggle for Life* (Sobre a Origem das Espécies por Meio da Seleção Natural ou a Preservação de Raças Favorecidas na Luta pela Vida). Somente na sexta edição (1872), o título foi abreviado para *The Origin of Species* (A Origem das Espécies), como é popularmente conhecido.

bem adaptados, tem maior possibilidade de perpetuação do seu código genético nas próximas gerações.

Nos algoritmos genéticos, os cromossomos são então submetidos a um processo que inclui avaliação, seleção e recombinação sexuada (*crossover*) e mutação. Na natureza os indivíduos competem entre si por recursos como: comida, água e refúgio. Adicionalmente, entre os animais de uma mesma espécie, aqueles que não obtêm êxito tendem provavelmente a ter um número reduzido de descendentes, tendo, portanto menor probabilidade de seus genes serem propagados ao longo de sucessivas gerações. A combinação entre os genes dos indivíduos que perduram na espécie, podem produzir um novo indivíduo muito melhor adaptado às características de seu meio ambiente.

Os Algoritmos Genéticos utilizam uma analogia direta deste fenômeno de evolução na natureza. A cada indivíduo se atribui uma pontuação de adaptação, dependendo da resposta dada ao problema por este indivíduo. Aos mais adaptados é dada a oportunidade de reproduzirem-se mediante cruzamentos com outros indivíduos da população, produzindo descendentes com características de ambas as partes.

Se um Algoritmo Genético for desenvolvido corretamente, a população (conjunto de cromossomos) convergirá a uma solução ótima para o problema proposto. O processo que mais contribui para a evolução é a seleção, que escolhe os mais aptos e direciona a busca do algoritmo, dado que possuem uma probabilidade maior de serem escolhidos. Já o *crossover* serve para gerar descendentes próximos e/ou melhores do que os pais selecionados. A mutação também tem um papel significativo na evolução das soluções, sendo necessária para manter a diversidade da população e assim garantir a existência da evolução das soluções, mesmo que eventualmente essas soluções, apresentem baixa aptidão na solução do problema.

Desde então, os Algoritmos Genéticos tornam-se bem sucedidos na solução de uma ampla gama de problemas de otimização com vários artigos de pesquisa e com uma grande quantidade de livros escritos. Os Algoritmos Genéticos são capazes de desenvolver soluções para problemas do mundo real, tais como

problemas de busca e otimização (BEASLEY *et al.*, 1993; HAUPT R., HAUPT S. 2004). A abordagem genética fornece uma alternativa de busca e movimentação no espaço de solução. Essa abordagem identifica as soluções que estão mais próximas da solução ideal (solução ótima), conforme determinado por algumas medidas de distância. Algoritmos Genéticos são utilizados para resolverem problemas que são difíceis de serem resolvidos através da aplicação de métodos convencionais. Em termos gerais, os algoritmos genéticos representam a técnica mais popular da Computação Evolutiva (CE) (MOHAMED *et al.*, 2010).

2.2. ALGORITMO CULTURAL

2.2.1. INTRODUÇÃO

Assim como os Algoritmos Genéticos (AGs), os Algoritmos Culturais (ACs) são Algoritmos Evolutivos (AEs) e tornaram-se métodos eficazes para a solução de problemas tradicionais de otimização complexos, devido às suas características, como o paralelismo implícito e a busca aleatória.

No entanto, os algoritmos evolutivos frequentemente convergem prematuramente produzindo baixa eficiência evolutiva porque as informações implícitas incorporadas no processo de evolução e na área de conhecimento correspondente aos problemas de otimização não são plenamente aproveitadas (GUO *et al.*, 2011). Com a finalidade de aproveitar eficazmente as informações na evolução implícita, REYNOLDS propôs os Algoritmos Culturais que são derivados do processo de evolução cultural humano (REYNOLDS, 1994).

A evolução cultural permite que as sociedades evoluam ou adaptem-se a seus ambientes em taxas que excedem a da evolução biológica baseada apenas na herança genética. Os Algoritmos Culturais possuem três componentes principais: um espaço populacional, um espaço de crença e um protocolo que descreve como o conhecimento se move entre os dois primeiros componentes (ZHANG, 2011). Nos Algoritmos Culturais (JIN e REYNOLDS, 1999) o espaço populacional pode suportar qualquer modelo computacional de base populacional, tais como Algoritmos Genéticos e Estratégias Evolucionárias (EEs). No espaço de crenças, o conhecimento implícito é extraído dos melhores indivíduos provenientes da população e armazenado em diferentes formas. Em seguida, eles são utilizados

para direcionar o processo de evolução do espaço populacional, de modo a induzir a população a fugir de soluções de ótimos locais.

Os algoritmos culturais podem efetivamente melhorar o desempenho do processo evolutivo, além de fornecerem um modelo universal para a extração e utilização das informações evolutivas (GUO *et al.*, 2011; ZHANG, 2011). A pesquisa sobre Algoritmos Culturais foi conduzida por observações sobre a cultura e até que ponto sua influência sobre os indivíduos pode ser levada em consideração para se criar um sistema que utilize tal hereditariedade. A tomada de decisão tanto humana quanto animal é fortemente influenciada pelo conhecimento adquirido através da observação do comportamento dos outros, e quando os padrões de comportamento entre os indivíduos estão espalhados ao longo de gerações passa a ser considerada como uma forma de evolução cultural (DANCHIN *et al.*, 2004).

Em REYNOLDS (1994) foram desenvolvidos alguns modelos para investigar as propriedades de Algoritmos Culturais. Nestes modelos, um espaço de crença é usado para restringir a combinação de traços que os indivíduos possam assumir aceitando ou não um comportamento em um determinado nível. Para compreender o processo de evolução cultural, é importante conhecer o conceito de meme. O “meme” é uma adaptação do grego “mimema” (que significa algo imitado). Dawkins introduziu a noção de memes definindo-os como homólogos de genes no mundo cultural (DAWKINS, 2001).

Memes são, portanto, a unidade de transmissão cultural ou imitação tal como um pedaço de pensamento, um fragmento de música e assim por diante. Sugere-se que ao longo dos tempos o ser humano evoluiu um conjunto único de capacidades que permitiu a formação, codificação e transmissão de informações culturais. O conhecimento deve ser codificado de forma a ser acessível a todos os indivíduos de uma sociedade. Uma vez codificado, esse conhecimento é assimilado por cada indivíduo da sociedade sob o prisma das suas experiências anteriores, podendo este indivíduo incorporar ou não um novo conhecimento àqueles presentes na sua sociedade.

Algoritmos Culturais baseiam-se nas teorias de alguns sociólogos e arqueólogos, que tentaram modelar a evolução cultural sugerindo que a evolução

cultural pode ser vista como um processo de herança, em dois níveis: o nível microevolutivo, que é o material genético herdado dos pais para sua prole e o nível macroevolutivo, que é o conhecimento adquirido por pessoas físicas após seu nascimento, passando através das gerações, e que uma vez codificados e armazenados, são utilizados para orientar o comportamento de indivíduos pertencentes a uma população (BECERRA, 2002).

Em REYNOLDS *et al.* (2010) confirmam esse sistema duplo de herança nos Algoritmos Culturais focando também o processo de herança em dois níveis (microevolutivo e macroevolutivo). No nível microevolutivo tem-se a modelagem da população em si composta por um conjunto de indivíduos, levando-se em consideração qualquer modelo evolutivo baseado em populações. Já no nível macroevolutivo é modelado o conhecimento adquirido pelos indivíduos ao longo das gerações e que codificado e armazenado no espaço de crenças, ajuda a guiar o comportamento dos indivíduos em suas populações.

É importante ressaltar que as informações culturais podem ser transmitidas tanto entre indivíduos de uma população quanto de uma população para outra. Isso permite uma melhor adaptação ao ambiente do que a possível adaptação pela genética (REYNOLDS e ZANONI, 1992, p. 87).

Uma das grandes vantagens da espécie humana na resolução de problemas complexos é a sua possibilidade de agir dentro de padrões de comportamento pré-estabelecidos, dados pelo senso comum e que simplificam o processo de tomada de decisões de um número infinito de possibilidades para alguns caminhos viáveis de comportamento. Já em sistemas artificiais, o problema da criação de um “senso comum” é complicado pela inexistência de um senso comum universal válido em todos os lugares e em todos os tempos. O senso comum passa, portanto, a ser um problema de origem cultural e deve ser tratado apropriadamente.

Segundo o ponto de vista do antropólogo Clifford Geertz (1989², p. 53):

A cultura seria melhor vista como um conjunto de mecanismos de controle (o que os engenheiros de computação chamam de “programas”) para governar o comportamento. A segunda ideia é que o homem é precisamente o animal mais desesperadamente dependente de tais mecanismos de controle, extragenéticos, para ordenar seu comportamento.

Para WILSON (1999), outras abordagens vindas da biologia admitem a interferência da cultura na natureza e vice versa, a chamada coevolução gene cultura, onde regras epigenéticas³ orientariam o comportamento humano. A epigênese, originalmente um conceito biológico, significa o desenvolvimento de um organismo sob influência conjunta da hereditariedade e do ambiente. As regras epigenéticas são regras práticas que permitem aos organismos encontrar soluções rápidas para problemas encontrados no ambiente. Elas predispoem os indivíduos a ver o mundo de certa forma inata e automaticamente fazer certas escolhas em vez de outras.

De acordo com WILSON (1999), tipicamente impelidas pelas emoções, as regras epigenéticas em todas as categorias de comportamento dirigem o indivíduo para as reações relativamente rápidas e exatas mais passíveis de garantir a sobrevivência e a reprodução. Entretanto, deixam em aberto a geração potencial de uma imensa série de variações e combinações culturais. Às vezes, especialmente em sociedades complexas, não mais contribuem para a saúde e o bem-estar. O comportamento que orientam pode dar errado e militar contra os melhores interesses do indivíduo e da sociedade.

Desta forma, segundo REYNOLDS *et. al.* (2010), seja pela vertente antropológica ou pela biológica, admite-se que a cultura aumenta a capacidade de adaptação do ser humano, seja em relação a problemas complexos encontrados em seu meio ambiente, ou a situações culturais novas criadas pela própria sociedade. Esta capacidade dá-se mais rapidamente do que na natureza, que utilizaria o mecanismo de seleção natural. Tem sido frequentemente sugerido, segundo

² GEERTZ, Clifford. **A Interpretação das Culturas**. Rio de Janeiro. Editora Guanabara - Rio de Janeiro (RJ): 1989.

³ Posição superior na genética.

REYNOLDS *et. al.* (2010), que a evolução cultural habilita às sociedades a evoluir ou se adaptar ao seu ambiente em taxas que excedem as da evolução biológica baseada somente na herança genética.

2.2.2. CARACTERÍSTICAS DOS ALGORITMOS CULTURAIS

Algoritmos Culturais são baseados na premissa, de que se pode melhorar a taxa de aprendizado de um Algoritmo Evolutivo adicionando-se mais um elemento de pressão evolutiva – o chamado *belief space*, um mecanismo cultural. Desta forma um sistema de dupla herança, tanto genética individual quanto cultural, poderia adaptar-se melhor e responder com mais eficiência a um grande número de problemas, como demonstrou REYNOLDS.

Os Algoritmos Culturais possuem um funcionamento básico proposto por REYNOLDS (1994) e REYNOLDS *et al.* (2010) onde são descritos dois componentes principais: Espaço Populacional e Espaço de Crenças. A ideia básica dos Algoritmos Culturais é apresentada pela Figura 2, a seguir:

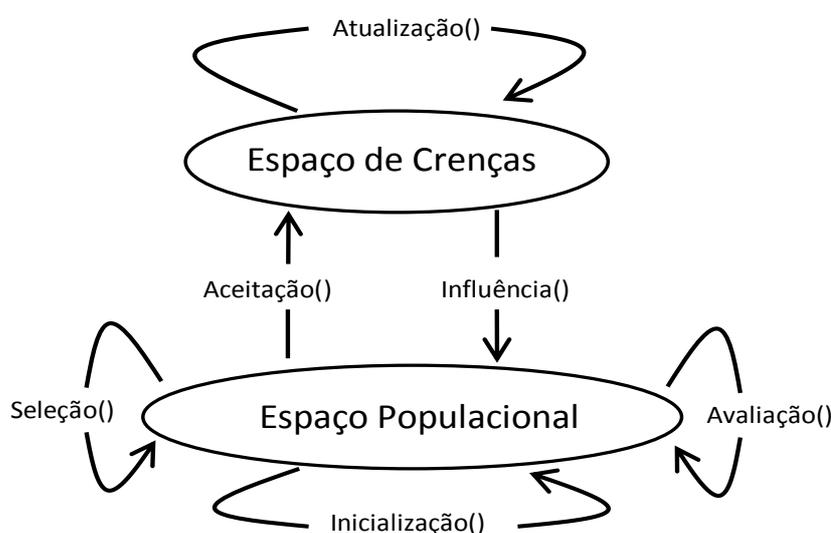


Figura 2 : Funcionamento Básico de um Algoritmo Cultural (REYNOLDS et al., 2010).

O Espaço Populacional é o conjunto de soluções que pode ser modelado utilizando qualquer técnica de Inteligência Computacional que faça uso de uma população de indivíduos, já o Espaço de Crença é o local onde ocorre o armazenamento e representação do conhecimento (experiência ou mapas individuais) adquirido ao longo do processo evolutivo.

No espaço de crença podem se armazenar diferentes tipos de conhecimento como: conhecimento situacional, normativo, do domínio, topográfico e histórico. É importante ressaltar que não é obrigatória a implementação de todos esses tipos de conhecimento no espaço de crença, entretanto as mais utilizadas são o conhecimento situacional e normativo. É a partir desse conhecimento armazenado que os indivíduos são guiados na direção das melhores regiões do espaço de busca.

Os indivíduos são descritos por um conjunto de características e comportamentos e por um mapa que generaliza os conhecimentos e experiências adquiridas por esse indivíduo. Essas características e comportamentos são modificados por operadores genéticos que podem ser influenciados socialmente. Da mesma maneira, os conhecimentos e experiências são unidos e modificados para formar o espaço de crenças.

O pseudocódigo do Algoritmo Cultural é apresentado na Figura 3, a seguir :

Algoritmo Cultural (REYNOLDS, 2003)	
01:	INÍCIO
02:	t=0 <i>;primeira geração</i>
03:	inicializar população P(t) <i>;população inicial aleatória</i>
04:	Inicializar Espaço de Crença EP(t)
05:	avaliar população P(t) <i>;calcula f(i) para cada indivíduo</i>
06:	ENQUANTO (não condição_fim) FAÇA
07:	Comunicação (P(t), EP(t)); <i>;votação(Aceitação)</i>
08:	Atualização EP(t); <i>;uso de operadores culturais</i>
09:	Comunicação (EP(t), P(t)); <i>;promoção (função de influência)</i>
10:	t t+1 <i>;próxima geração</i>
11:	selecionar P(t) de P(t-1)
12:	altera P(t) <i>;crossover e mutação</i>
13:	avaliar P(t) <i>;calcula f(i) para cada indivíduo</i>
14:	FIMENQUANTO
15:	FIM

Figura 3 : Pseudocódigo do Algoritmo Cultural (REYNOLDS, 2003).

A cada nova geração de indivíduos é feita uma avaliação e o conhecimento dos melhores indivíduos pode ou não fazer parte do Espaço de Crença. De acordo com o protocolo de aceitação a nova população a ser gerada é influenciada com o conhecimento anteriormente armazenado através de operadores. Tanto o espaço de crença quanto os indivíduos de uma população podem ser influenciados pelo que se chama de Protocolos de Intercomunicação. A comunicação entre os indivíduos de

uma geração e o Espaço de Crenças é dada por um protocolo dito Função de Aceitação. Já a interação do Espaço de Crenças com a população de indivíduos é chamada de Função de Influência.

Um modelo de busca evolutivo pode ser utilizado na modelagem do componente populacional em Algoritmos Culturais. Os Algoritmos Culturais operam em dois espaços: espaço populacional e espaço de crença.

O espaço populacional consiste num conjunto de soluções do problema, e pode ser modelado através de qualquer técnica de inteligência computacional que utilize uma população de indivíduos.

O espaço de crença é um repositório em que os indivíduos podem armazenar suas experiências para que os demais indivíduos possam aprender indiretamente. Nos Algoritmos Culturais, as informações adquiridas por um indivíduo podem ser compartilhadas com toda a população, diferentemente da maioria das técnicas evolucionárias, onde a informação só pode ser compartilhada com os descendentes.

O protocolo de comunicação é o mecanismo responsável por interligar os espaços, populacional e de crença. Ele estabelece as regras de comunicação, definindo que tipo de informação deve ser trocada entre os espaços (REYNOLDS *et al.*, 2010).

A ideia central dos Algoritmos Culturais é adquirir conhecimento sobre a solução do problema a partir da população em evolução e aplicar este conhecimento para orientar a busca. O conhecimento gerado no espaço populacional é seletivamente aceito ou passado ao espaço de crenças e usado para ajustar as estruturas simbólicas lá existentes. Este conhecimento então pode ser utilizado para influenciar as modificações feitas na próxima geração da população (REYNOLDS e SALEEM, 2005).

Segundo (REYNOLDS (2003), p. 3.591) as principais características dos Algoritmos Culturais são:

- a) Mecanismo Dual de Herança: herda característica tanto do nível micro evolutivo como macro-evolutivo;

- b) Evolução Orientada por Conhecimento: a população é guiada na direção que, segundo o conhecimento armazenado no espaço de crenças, seja a melhor;
- c) Suporte a Hierarquia: tanto a população quanto o espaço de conhecimento podem ser organizados de forma hierárquica;
- d) Conhecimento sobre o domínio separado dos indivíduos: o conhecimento adquirido é armazenado no espaço de crenças e compartilhado entre os indivíduos. Desse modo, quando um indivíduo é eliminado da população, o conhecimento adquirido pelo mesmo permanece;
- e) Suporte a auto-adaptação em vários níveis: permite tanto a auto-adaptação da população quanto do conhecimento e da forma como o conhecimento é adquirido;
- f) Diferentes taxas de evolução: a evolução das populações e do conhecimento não precisa ocorrer na mesma taxa;
- g) Estrutura de funcionamento: permite a modelagem de diversas formas de evolução cultural.

2.2.3. MICROEVOLUÇÃO X MACROEVOLUÇÃO

Os Algoritmos Culturais implementam um mecanismo dual de herança. Esse mecanismo permite que os Algoritmos Culturais explorem tanto a microevolução quanto a macroevolução. A microevolução diz respeito à evolução que acontece no nível populacional. Já a macroevolução é a que ocorre sobre a cultura em si, ou seja, a evolução do espaço de crenças. Nos Algoritmos Culturais a evolução ocorre de forma mais rápida que nas populações sem o mecanismo de macroevolução. Segundo (JIN e REYNOLDS (1999), p. 1.674).

- (1) **Espaço Populacional:** No espaço populacional são representadas as características e comportamentos dos indivíduos. Essa representação pode ser feita através de qualquer técnica que faça uso de uma população de indivíduos, como é o caso dos algoritmos genéticos, programação evolutiva, programação genética, evolução diferencial, sistemas imunes, entre outros.

(2) **Espaço de Crença:** O espaço de crenças é o repositório de símbolos que representam os conhecimentos adquiridos pelo Espaço Populacional ao longo do processo evolutivo. O espaço de crenças permite que os indivíduos sejam removidos da população sem que o conhecimento por eles adquiridos seja perdido, ou seja, se durante o processo de evolução um indivíduo bom é perdido, o seu conhecimento armazenado é propagado para outras gerações. Os Espaços de Crenças guiam os indivíduos em busca de melhores regiões.

(3) **Protocolos de Comunicação:** Os protocolos de comunicação ditam as regras sobre os indivíduos que podem contribuir com conhecimentos para o espaço de crenças (Função de Aceitação) e como o espaço de crenças vai influenciar novos indivíduos (Função de Influência).

(4) **Função de Aceitação:** Na função de aceitação são selecionados indivíduos que irão influenciar o espaço de crenças atual. A função de aceitação pode ser de dois tipos: estática ou dinâmica. Na estática pode-se utilizar do ranking absoluto (uma porcentagem da população é selecionada) ou do ranking relativo (os indivíduos com aptidão acima da média são selecionados). Já na dinâmica o percentual dos indivíduos selecionados varia ao longo do processo evolutivo. Inicialmente o processo é menos seletivo, e se torna mais restrito ao longo da evolução. A função de aceitação é a que mais influência o desempenho de um algoritmo cultural.

(5) **Função de Influência:** Na função de influência é que se estabelece como o conhecimento armazenado no espaço de crenças vai interferir nos operadores do espaço populacional. Geralmente é utilizada uma função de influência para cada tipo de conhecimento armazenado. A função de influência pode ser vista como um mecanismo de auto-adaptação do processo evolutivo, já que ela adapta os operadores de acordo com o conhecimento adquirido.

O conhecimento macroevolutivo pode ser dividido em cinco categorias, segundo (REYNOLDS *et al.*, 2005):

(1) **Conhecimento Normativo:** introduzido por (REYNOLDS e SALEEM, 2005), essa categoria de conhecimento é representada como um conjunto de

intervalos de variáveis, e cada um é visto como uma série promissora de soluções boas ou socialmente aceitáveis para um parâmetro. A Figura 4 contém um exemplo da estrutura onde são armazenada cada variável do cromossomo correspondente a n variáveis V_1, V_2, \dots, V_n com seus respectivos intervalos. Esses intervalos correspondem a mínimos e máximos das variáveis (l e u) e mínimos e máximos das suas respectivas aptidões (L e U). O ajuste do intervalo do conhecimento normativo varia de acordo com o melhor indivíduo. Ou seja, se o indivíduo passou pela função de aceitação e seu intervalo é menor que o intervalo armazenado no espaço de crença, o intervalo é reajustado e vice-versa.

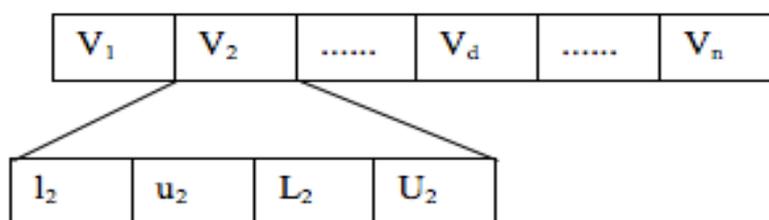


Figura 4 : Representação do Conhecimento Normativo (IACOBAN; REYNOLDS; BREWSTER, 2003).

(2) **Conhecimento Situacional:** contém um conjunto de exemplos que são úteis para a interpretação da experiência dos indivíduos. A Figura 5 contém um exemplo da estrutura utilizada para representar esse tipo de conhecimento. Cada indivíduo aqui considerado como um exemplar (E_1, E_2, \dots, E_n) é armazenado junto com suas características ou variáveis (X_1, X_2, \dots, X_n) e com sua aptidão $f(x)$. O conhecimento situacional é atualizado sempre que é encontrado um indivíduo cuja aptidão supere a aptidão do pior indivíduo armazenado.

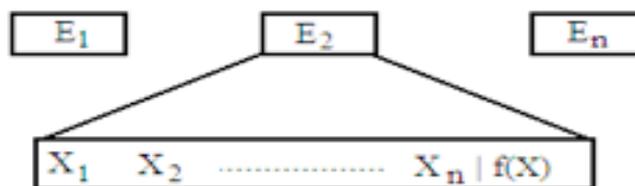


Figura 5 : Representação do Conhecimento Situacional (IACOBAN; REYNOLDS; BREWSTER, 2003).

(3) **Conhecimento Topográfico:** proposto com o objetivo de extrair padrões de comportamento do espaço de busca. Esse tipo de conhecimento identifica regiões promissoras dentro do espaço de busca e faz com que novos indivíduos as explorem. Coello e seus alunos representam esse conhecimento através de uma árvore k dimensões. Segundo (BECERRA; COELHO, 2005) essa representação é mais eficiente do ponto de vista da memória utilizada para armazená-la, conforme ilustrada na Figura 6. A atualização do conhecimento topográfico é dada quando se encontra um novo indivíduo melhor que o indivíduo da célula. Então, essa célula é dividida em células menores.

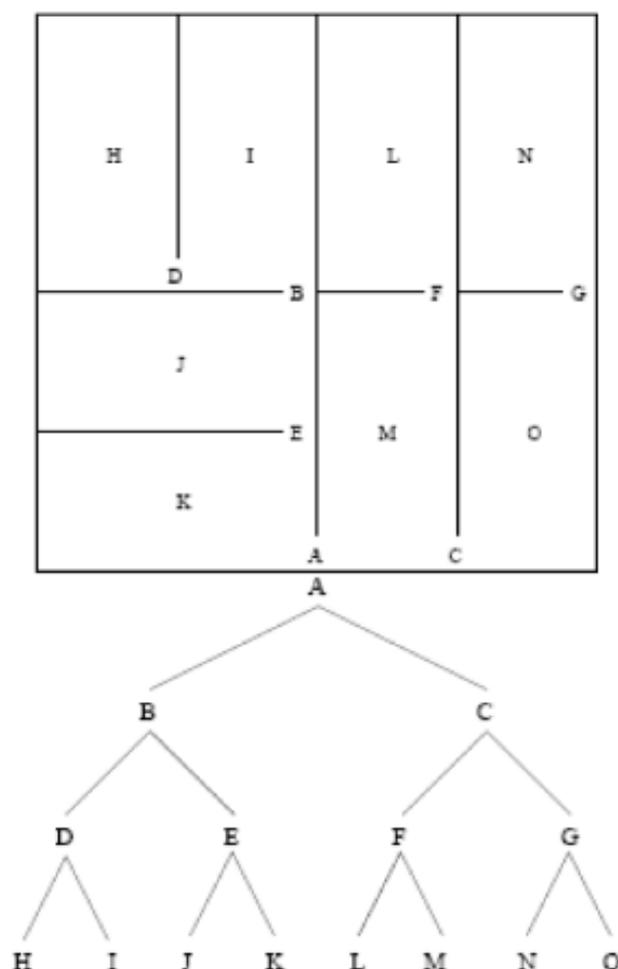


Figura 6 : Representação do Conhecimento Topográfico (BECERRA; COELLO, 2005).

(4) **Conhecimento Domínio:** introduzida por (REYNOLDS e SALEEM, 2005) para resolver os problemas dinâmicos de otimização. Ele foi projetado para atuar sobre locais dinâmicos, especialmente nos termos da predição dos

gradientes crescentes ou decrescentes. Como o próprio nome pressupões, é específico de cada aplicação.

(5) **Conhecimento Histórico:** desenvolvido por (REYNOLDS e SALEEM, 2005) motivado pela necessidade de desenvolver aprendizado em ambientes dinâmicos. Ele contém informações sobre mudanças de sequências ambientais em termos dos deslocamentos na distância e no sentido das tendências conhecidas no espaço da busca. Sua origem cognitiva advém de episódios da memória como ocorre nos seres humanos e animais. A estrutura utilizada para representar o Conhecimento Histórico é demonstrada na Figura 7 onde e_i representa o melhor indivíduo (exemplar) encontrado antes da i -ésima alteração do ambiente. A distância média das mudanças para a característica i é representada por ds_i . Se existem mudanças para a característica i , dr_i representa a direção média dessas mudanças.

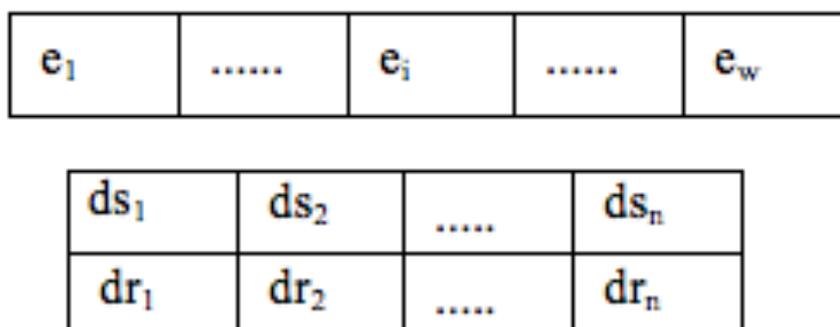


Figura 7 : Representação do Conhecimento Histórico (BECERRA; COELLO, 2005).

2.3. ALGORITMO HÍBRIDO

Os métodos de otimização modernos, também, por vezes, chamados métodos de otimização não tradicionais, surgiram como métodos poderosos e populares para resolver problemas complexos de engenharia nas últimas décadas. Este surgimento tem relação com a estratégia de solução baseada em heurísticas, conforme citado no Capítulo 1, seção 1.2.

Usualmente uma heurística é identificada com um problema específico (WEI e SHUZHUO, 2009). O grau de dificuldade em resolver um problema determinado, com um algoritmo dedicado, está estreitamente relacionado com a sua

complexidade computacional, ou seja, a quantidade de recursos como tempo e memória necessária para fazê-lo. O uso do Algoritmo Genético, normalmente, leva a um ótimo local e causa a estagnação do processo de otimização. Ou seja, não se consegue alcançar outras soluções ótimas naquele momento. O uso destes métodos de otimização, com informação heurística, permite uma melhora no desempenho do algoritmo (DELIN *et al.*, 2011; BADILLO *et al.*, 2013).

Uma abordagem para a obtenção de soluções quase ótimas para os problemas NP em tempo razoável é a aplicação de meta-heurística, guiadas por funções objetivo. Sendo normalmente, utilizadas funções objetivo que são consideradas difíceis, do ponto de vista matemático, por não serem contínuas, não diferenciáveis ou que tenham múltiplos valores máximos e mínimos. Mais ainda, em muitas aplicações no mundo real de otimização meta heurística, as características das funções objetivo não são conhecidas antecipadamente.

No início dos estudos sobre Computação Memética (CM) a mesma era sinônimo do Algoritmo Genético com busca local. Mais tarde, ficou evidente que a área de Computação Memética era mais ampla e envolvia, de uma maneira geral, o uso de métodos gerais de otimização (meta heurísticas) com informações heurísticas de busca local ou de vizinhança, que grosseiramente pode se denominar de métodos híbridos (NERI e COTTA, 2012).

Dentre as técnicas evolutivas híbridas, tinha-se o uso do Algoritmo Genético com busca local, depois se passou a utilizar o Algoritmo Genético combinado com o *simulated annealing* para tratar a vizinhança (JIAO e WANG, 2000). Mais recentemente, o Algoritmo Genético vem sendo utilizado em combinação com o *simulated annealing*, com o acréscimo de uma heurística de busca local (XU *et al.*, 2011). Em outra abordagem, o Algoritmo Genético vem sendo utilizado junto com o Algoritmo Cultural (XUE e GUO, 2007; YAN *et al.*, 2012).

A hibridização é uma alternativa de melhorar o desempenho e eficácia dos algoritmos genéticos. A forma mais comum de hibridização é agrupar os algoritmos com técnicas de busca local e incorporar conhecimentos específicos do domínio ao processo de busca. Utilizar algoritmos híbridos com alguma técnica eficiente para um problema específico, faz com que o melhor dos dois mundos seja aproveitados.

As técnicas que utilizam um modo híbrido entre meta-heurísticas e heurísticas de problemas específicos têm sido criadas para fornecer técnicas de otimização mais eficientes para os problemas mais complexos. Técnicas que empregam meta-heurísticas como busca global e heurísticas específicas do problema como a busca local, são comumente referidos como Algoritmos Meméticos (AMs). Com o desenvolvimento de um projeto adequado, os Algoritmos Meméticos não só podem apresentar uma boa capacidade exploratória, similar ao que faz um algoritmo com base populacional de busca global, mas também proporciona um bom desempenho de exploração durante a busca, semelhante ao que faz um algoritmo de busca local.

Como resultado, algoritmos meméticos tem um desempenho melhor do que algoritmos sem hibridização (YEW-SOON *et al.*, 2010).

CAPÍTULO 3. ALGORITMO CULTURAL COM MODELO DE ILHAS BASEADO EM ALGORITMO GENÉTICO MULTIPOPULACIONAL (AC-MI)

Algoritmos Genéticos Multipopulação ou Modelo de Ilhas é uma extensão do tradicional Algoritmo Genético de População Simples, dividindo a população em várias subpopulações em que o produto da evolução e os indivíduos estão autorizados a migrar de uma subpopulação para outra. Valores diferentes para os parâmetros tais como taxa de cruzamento e de mutação podem ser escolhidas para cada subpopulação, ou seja, o modelo baseado em ilha divide a população em subpopulações parcialmente isoladas, gerando um conjunto com possibilidades de várias populações concorrentes poderem gerar uma melhor solução do que a conseguida com apenas uma população. Isto simula o afastamento geográfico entre as populações de uma espécie.

Normalmente, a abordagem multipopulacional que utiliza o modelo básico de ilhas, utiliza os mesmos valores para estes parâmetros em todas as subpopulações. Com a finalidade de controlar a migração dos indivíduos, vários outros parâmetros podem ser definidos, tais como: (i) a topologia de comunicação que define as conexões entre as subpopulações, (ii) uma taxa de migração que controla a quantidade de indivíduos durante a migração, e (iii) um intervalo de migração que afeta a frequência de migração. Além disso, a migração deve incluir estratégias para a seleção de migrantes e sua inclusão na sua nova subpopulação (AGUIRRE *et al.*, 2000). O tamanho das subpopulações, a topologia de comunicação (o seu grau de conectividade), a taxa de migração e a frequência de migração são fatores importantes relacionados com o desempenho de Algoritmos Genéticos. No modelo Algoritmo Genético baseado em ilhas, as subpopulações são isoladas durante a reprodução, seleção e avaliação. As ilhas normalmente incidem sobre o processo evolutivo dentro das subpopulações de indivíduos antes de migrar para outras ilhas, ou processadores conceituais, que também realizam um processo evolutivo. Em horários ou momentos pré-determinados, durante o processo de busca, ilhas enviam e recebem migrantes de outras ilhas. Isto permite encontrar melhores soluções internas nas ilhas, como também permite a migração destas soluções para outras ilhas existentes, bem como a criação de novas ilhas, desta forma ampliando o número de soluções distintas e melhores para os problemas.

Um detalhe interessante neste modelo está relacionado ao critério de parada escolhido. Ele deve ser baseado em uma condição que envolva todas as ilhas que compõem o algoritmo genético, evitando que em algum momento, uma ilha com maior capacidade computacional chegue ao critério de parada antes que as demais. Podendo assim deixar alguma ilha parte do tempo ociosa (CANTU-PAZ, 2000).

A ideia principal do algoritmo genético em modelo de ilha é a evolução independente (paralela) de subpopulações separadas, que se baseia na crença de que subpopulações múltiplas distribuídas, com regras e interações locais, formam um modelo mais realista das espécies na natureza, com característica fundamental do algoritmo genético em modelo de ilha é a migração de indivíduos com boa avaliação (mais adequados) entre as subpopulações. No resto do tempo, o algoritmo genético em modelo de ilha fica trabalhando para melhorar as soluções internas da ilha.

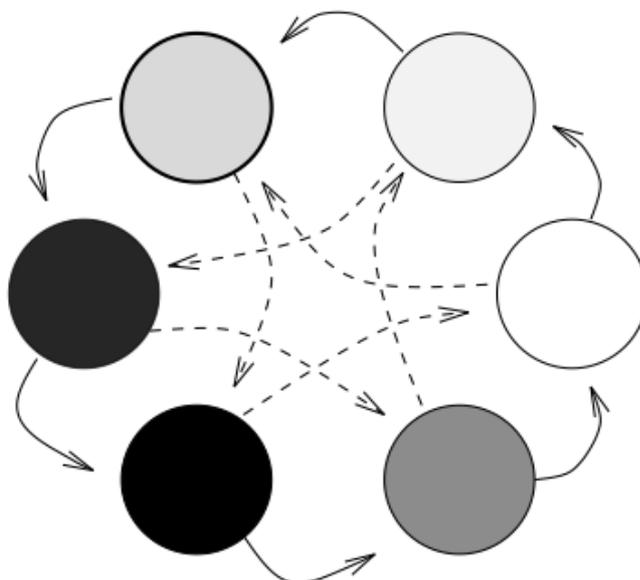


Figura 8 : Modelo de Ilhas (WHITLEY, 1993).

A Figura 8 é uma ilustração do modelo de ilhas, as cores representam a similaridade do material genético, as ilhas mais distantes possuem menor similaridade.

Várias estratégias de migração podem ser escolhidas para serem aplicadas nesse modelo. Duas estratégias frequentemente utilizadas para selecionar os migrantes são a seleção dos melhores e seleção aleatória. Por exemplo, a migração pode implementar uma estratégia de transmissão síncrona elitista ocorrendo a cada geração M . Cada subpopulação transmite uma cópia de seus R melhores indivíduos a todas subpopulações vizinhas. Assim, cada subpopulação em todos os eventos de migração recebem migrantes. A Figura 9 ilustra um modelo de ilhas com uma topologia de comunicação +1+2 em que cada subpopulação está ligada a dois vizinhos ($L = 2$). Neste exemplo, a subpopulação P_0 pode enviar os indivíduos apenas para P_1 e P_2 e receber migrantes apenas a partir de P_4 e P_5 (SILVA, 2012).

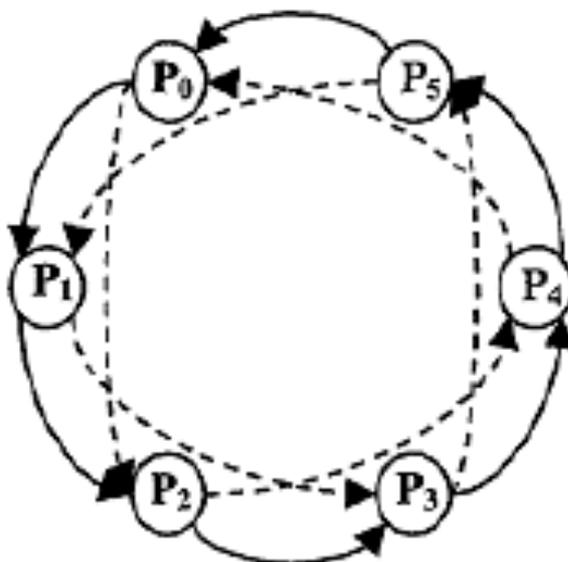


Figura 9 : Topologia de Comunicação +1+2 (SILVA, 2012).

A migração aumenta a pressão seletiva geral pois os indivíduos, que são suficientemente capazes para migrar, possuem novas tentativas adicionais de reprodução na população receptora (LEVINE, 1994). A introdução de indivíduos migrados na população local ajuda a manter a diversidade genética, pois o indivíduo chega de uma subpopulação diferente que evoluiu independentemente.

A estratégia populacional utilizada no Algoritmo Genético Multipopulação ou Modelo de Ilhas foi Master-Slave ou Mestre-Escravo. Neste modelo, uma única população é mantida centralizada e a rotina de avaliação é realizada de forma paralela. O processador mestre se comporta como um algoritmo genético

sequencial, diferenciando-se apenas no cálculo de aptidão (CANTU-PAZ, 2000), etapa na qual distribui os filhos para serem analisados pelos processadores escravos. Os escravos realizam o cálculo de aptidão devolvendo o resultado ao mestre, que dará continuidade à sua execução, ou seja, o processo mestre armazena a população, executa operações de um algoritmo genético, e distribui indivíduos para os processos escravos. Os processos escravos somente avaliam a aptidão dos indivíduos e retorna os valores para o mestre. A Figura 10 ilustra a estratégia populacional.

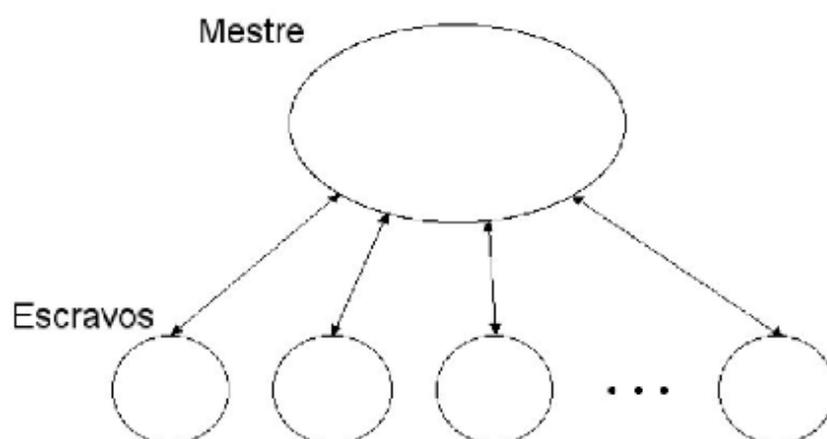


Figura 10 : Estratégia Populacional Mestre-Escravo (STACH, 2007).

O algoritmo usa uma única população e a avaliação dos indivíduos e/ou a aplicação de operadores genéticos é realizada em paralelo. A seleção e a comparação são feitas globalmente, portanto, cada indivíduo pode competir com qualquer outro.

A operação que é mais comumente paralelizada é a avaliação da função de aptidão, porque normalmente exige apenas o conhecimento do indivíduo que está sendo avaliado (e não da população inteira), e assim não há necessidade de se comunicar durante essa fase. Isso geralmente é implementado através de programas mestre-escravo, onde o mestre armazena a população e os escravos avaliam a aptidão. A paralelização de avaliação da aptidão é feita mediante a atribuição uma fração da população para cada um dos processadores disponíveis (no caso ideal um indivíduo por núcleo de processamento).

A comunicação ocorre apenas quando cada escravo recebe o indivíduo (ou um subconjunto de indivíduos) para avaliar e, quando devolvem os valores de aptidão. O algoritmo é dito ser síncrono, se o mestre para e aguarda para receber os valores de aptidão para toda a população antes de prosseguir com a próxima geração. Os algoritmos genéticos síncronos mestre-escravo tem exatamente as mesmas propriedades que um algoritmo genético simples, exceto pela sua velocidade, ou seja, esta forma de algoritmo genético paralelo executa exatamente a mesma busca que um algoritmo genético simples. Uma versão assíncrona do algoritmo genético mestre-escravo também é possível. Neste caso, o algoritmo não para para esperar qualquer resultado de um processador que esteja lento. Por esta razão, o algoritmo genético mestre-escravo assíncrono não funciona exatamente como um AG simples, pois a seleção é realizada com uma fração da população que já foi processada, ou seja, opera sobre a população existente.

O Algoritmo Cultural com Modelo de Ilhas Baseado em Algoritmo Genético Multipopulacional (AC-MI) foi explicitamente referenciado e introduzido no Trabalho de SILVA (2012).

A principal característica no Algoritmo Cultural com Modelo de Ilhas Baseado em Algoritmo Genético Multipopulacional (AC-MI) é a ligação entre o espaço de crenças da população principal e espaço de crença das subpopulações. As transformações culturais ocorrem em paralelo tanto em relação à população principal quanto às subpopulações das ilhas. O elo de comunicação entre os dois espaços de crenças, permite a migração entre os melhores indivíduos armazenados na estrutura do conhecimento cultural. A Figura 11 ilustra o Modelo de Comunicação do Algoritmo Cultural com Modelo de Ilhas Baseado em Algoritmo Genético Multipopulacional (AC-MI).

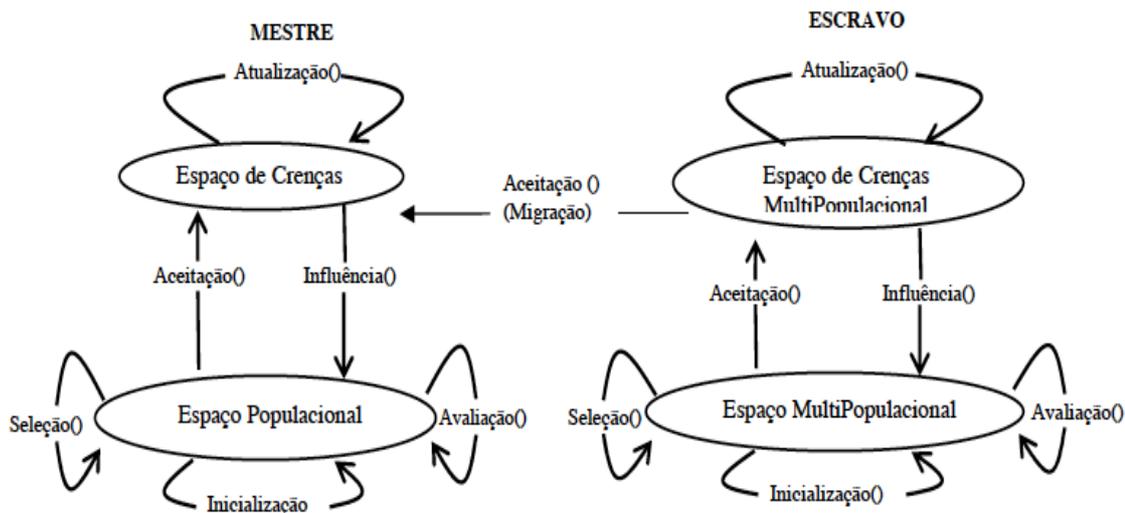


Figura 11 : Modelo de Comunicação do Algoritmo Cultural Multipopulacional (SILVA, 2012).

As migrações entre as ilhas ocorrem através da estrutura do espaço de crença Espaço MultiPopulacional (escravo) que realiza o processo de comunicação entre as subpopulações e envia os melhores indivíduos do Espaço MultiPopulacional para o Espaço Populacional (mestre) através da função de Aceitação (), inserida no modelo somente para a migração dos indivíduos. Ela ocorre em um intervalo predefinido cujo parâmetro é M (a cada geração M), onde os indivíduos são avaliados pela função de aceitação e atualizados no espaço de crenças principal. A migração do Espaço MultiPopulacional para o Espaço Populacional é realizada utilizando-se um número de indivíduos que são considerados como um conjunto de exemplos para o resto da população (Conhecimento Situacional). O uso do AC-MI surgiu pela maior capacidade de convergência dos Algoritmos Culturais diante dos Algoritmos Genéticos e pela constatação no trabalho de SILVA (2012) de que na maioria dos experimentos era possível encontrar o valor ótimo desde que houvesse maior diversidade populacional e que por pouco não se conseguia baixar o desvio padrão, ou seja, os ensaios realizados, pois na maioria das vezes se fizesse 50 ensaios, obtinham-se 47 ou 48 ensaios com valores ótimos encontrados na literatura. Ou seja, por pouco não se atingiam 100%, o que significava que por pouco o desvio padrão não era zero). A ideia de utilização de subpopulações com Modelo de Ilhas se mostrou eficaz em relação ao aumento da diversidade (SILVA, 2012), uma vez que o modelo de ilhas apresenta subpopulações geograficamente separadas e podem efetuar intercâmbio de informações ao longo do tempo, permitindo que alguns indivíduos migrem de uma subpopulação para outra de

acordo com diversos padrões. A principal razão para esta abordagem é re-injetar periodicamente diversidade em outras subpopulações. Considera-se, nesse caso, que as diferentes subpopulações tendem a explorar diferentes partes do espaço de busca.

CAPÍTULO 4. METODOLOGIA E RESULTADOS

Nesta seção serão mostrados o problema de otimização delimitado neste trabalho que é o Problema da Mochila Multidimensional (PMM), bem como a metodologia utilizada nos experimentos e os resultados da aplicação do Algoritmo Cultural com Modelo de Ilhas Baseado em Algoritmo Genético Multipopulacional (AC-MI). Para verificar o desempenho do algoritmo utilizado, foram avaliados os resultados obtidos no problema abordado bem como a comparação desses resultados com outros algoritmos já publicados, ou seja, são realizadas comparações com outros algoritmos na literatura. Os parâmetros do AC-MI utilizados foram escolhidos de forma seletiva e fazendo pequenas variações em torno de um valor central. Ressalta-se que o tamanho dos problemas da mochila varia de 10 a 105 objetos de 2 a 30 mochilas que podem ser encontrados em OR-Library (BEASLEY, 1990). Os subsídios para a melhor escolha dos parâmetros do AC-MI proposta neste trabalho foi obtida por meio de vários ensaios previamente planejados.

4.1. PROBLEMA DA MOCHILA MULTIDIMENSIONAL (PMM)

O PMM é definido em (MARTELO e TOTH, 1990) como um problema de n objetos e m mochilas de capacidades específicas c_j ($j = 1, \dots, m$). As variáveis binárias x_i ($i = 1, \dots, n$) representam os itens selecionados para (1=SIM, 0=NÃO) serem levados nas m mochilas. Todo objeto possui um lucro p_i e um peso w_{ij} para cada mochila m . O objetivo é encontrar a melhor combinação de n objetos maximizando a somatória dos lucros p_i multiplicados pela variável binária x_i , representado matematicamente pela Equação (1). Suas restrições são as capacidades c_j de cada mochila. Portanto, o somatório dos valores de w_{ij} multiplicados por x_i deve ser menor ou igual a c_j , representado matematicamente pela Inequação (2).

$$\sum_{i=1}^n p_i \cdot x_i \quad (1)$$

$$\sum_{i=1}^n w_{ij} \cdot x_i \leq c_j (j = 1, \dots, m) \quad (2)$$

Todos os valores de p_i , w_{ij} e c_j são definidos como inteiros positivos e obedecem as Inequações (3) e (4).

$$w_{ij} \leq \max \{c_j\} (i = 1, \dots, n)(j = 1, \dots, m) \quad (3)$$

$$c_j \geq \min \{w_{ij}\} (i = 1, \dots, n)(j = 1, \dots, m) \quad (4)$$

A Inequação (3) pode ser interpretada como o limite máximo que a variável w_{ij} pode assumir, sendo menor ou igual a c_j , ou seja, menor ou igual a qualquer c_j das m mochilas. Esta mesma analogia pode ser feita para a Inequação (4), onde cada c_j é definida como maior ou igual ao mínimo dos pesos w_{ij} , limitando a capacidade mínima das mochilas ao menor peso dos n itens. A dimensão ou espaço de busca do PMM depende diretamente dos valores de n e m . Uma função exponencial binária com expoente n monta todas as possibilidades de arranjo dos n elementos respeitando as capacidades de cada mochila m . Consequentemente para buscar a solução ótima, pode-se testar todas as 2^n possibilidades para cada mochila m pelo método da força bruta, ou seja, varrer o espaço de busca que cada instância cria de $m \times 2^n$ possibilidades.

Como mencionado, este trabalho tem como foco uma das variantes binárias mais conhecidas do PMM, o Problema 0-1 de Múltiplas Mochilas (PMM), também conhecido na literatura como: *0-1 Multi-Knapsack Problem* (MKP), *Multiconstraint Knapsack Problem*, *Multiple Knapsack Problem* ou *0/1 Multidimensional Knapsack Problem* (KHURI et al., 1994).

4.2. METODOLOGIA

Para validar os resultados apresentados abaixo e melhorar a compreensão dos experimentos no AC-MI (SILVA, 2012), segue os parâmetros utilizados, índices de desempenho e símbolos utilizados:

- o parâmetro P é o tamanho da população principal;
- o número de ilhas é K (número de subpopulações);
- o símbolo N representa o número de vezes que o ótimo global foi encontrado no total de todas as execuções;
- *média de gerações* é a média de gerações onde as melhores soluções foram encontradas nos ensaios (execuções);
- *média* é a média das melhores soluções e *DesvPd* é o desvio padrão em torno da *média*;
- o *Tempo* representa o tempo de execução, em segundos, para que o algoritmo obtivesse esses resultados;
- o parâmetro PM é a Probabilidade de Mutação e PR a Probabilidade de Recombinação.

A parametrização do PMM para o AC-MI consistiu em utilizar as melhores parametrizações encontradas no AC-MI, levando-se em conta um conjunto de possíveis configurações, tais como: torneio = 3, P representando o tamanho da população com valor igual a 400 ou 800, K representando o número de ilhas com valor igual a 4 ou 8, PR representando a probabilidade de recombinação ou cruzamento, que variavam entre os valores 50%, 75% e 90%, e PM representando a probabilidade de mutação, que variavam entre os valores 2,5%, 5%, 7,50% e 10%, conforme ilustrado na Figura 12. Cumpre informar que todos os experimentos aqui relatados foram executados em uma máquina com processador Intel Core i7, CPU 3.5 GHz, 8 GB memória RAM e Windows 7 Professional de 64 Bits.

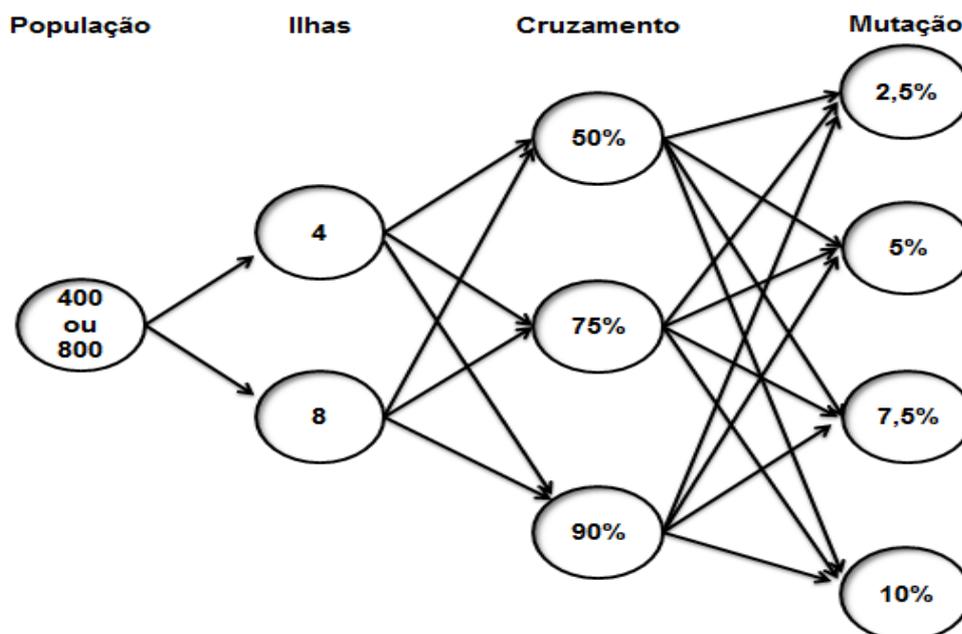


Figura 12 : Possíveis Variações dos Parâmetros do AC-MI Utilizadas nos Experimentos.

O Espaço populacional de um algoritmo cultural é o componente responsável pela microevolução do algoritmo e geralmente corresponde a algum algoritmo de Computação Evolutiva. Neste trabalho o espaço populacional é implementado na forma de um algoritmo genético baseado no modelo de ilha, conforme citado no Capítulo 3.

Como em todo algoritmo genético, o espaço populacional deste trabalho possui uma população de cromossomos (chamados de indivíduos neste trabalho por causa da terminologia dos algoritmos culturais), operadores genéticos, um método de seleção dos pais e um método de seleção dos indivíduos da próxima população (política de substituição na terminologia dos algoritmos genéticos).

Como dito anteriormente, a seleção dos pais ocorre através de um torneio.

O Espaço de crenças é o componente responsável pela macroevolução do algoritmo cultural. A sua função é a de armazenar as experiências extraídas dos indivíduos nos conhecimentos.

Neste trabalho foi utilizado o Conhecimento Situacional, conforme citado no Capítulo 2, seção 2.2, subseção 2.2.3.

Como descrito no Capítulo 2, seção 2.2, subseção 2.2.3, uma característica

importante do espaço de crenças é a forma como seus protocolos de comunicação (Função de Aceitação e Funções de Influência) são implementados. A Função de aceitação é o parâmetro que diz quantos indivíduos serão aceitos para contribuir com o seu conhecimento no espaço de crenças. O controle desse parâmetro pode ser classificado como determinístico e populacional. Neste trabalho, a Função de Aceitação foi de 20%. A Função de Influência é utilizada de acordo com qual dos tipos de conhecimento será utilizado para influenciar a geração dos indivíduos. Neste trabalho, a Função de Influência utilizada foi baseado em cima do conhecimento situacional.

O ajuste de parâmetros de Algoritmos Evolutivos é uma tarefa complexa e bastante controversa, visto que é muito difícil provar que um determinado conjunto de parâmetros é ótimo para um determinado problema.

Como dito anteriormente, o controle da taxa de cruzamento e mutação estão correlacionados, visto que na implementação do presente trabalho eles foram feitos sendo um complemento do outro.

Para todos os casos foram utilizados as variações no conjunto de parâmetros determinados de maneira empírica. Os parâmetros são ajustados manualmente ao longo da evolução e possuem influência nos resultados finais obtidos pelo algoritmo, conforme pode ser observado nas seções seguintes.

4.3. EXPERIMENTOS NO ALGORITMO CULTURAL COM MODELO DE ILHAS BASEADO EM ALGORITMO GENÉTICO MULTIPOPULACIONAL (AC-MI)

Para fazer as comparações e avaliar o desempenho do AC-MI, foi realizada uma comparação de vários testes com o Algoritmo Genético Simples e com o Algoritmo Cultural Convencional, utilizando os mesmos problemas da mochila.

Os resultados para o problema da mochila em configurações *benchmarks* são mostrados na Tabela 1, com o AC-MI parametrizado no trabalho de SILVA (2012).

**Tabela 1 : Os Melhores Resultados para o Problema da Mochila em Configurações
Benchmarks pelo Método AC-MI parametrizado no trabalho de SILVA (2012).**

Problema (n,m)		P	K	N	Média de Gerações	Média	DesvPd	Tempo (m.ss)
H. M. Weingartner and D. N. Ness	Weing7 (105,2)	400	8	100	44.49	1095445	0.00	0.50
	Weing7 (105,2)	100	7	100	68.87	1095445	0.00	0.36
C. C. Petersen	Petersen6 (39,5)	400	8	100	30.22	10618	0.00	0.35
	Petersen6 (39,5)	400	4	100	26.29	10618	0.00	0.37
	Petersen7 (50,5)	400	8	100	78.49	16537	0.00	0.33
	Petersen7 (50,5)	400	4	100	71.51	16537	0.00	0.26
S. Senyu and Y. Toyada	Sento1 (60,30)	400	4	100	87.44	7772	0.00	1.11

Para os ensaios realizados com o AC-MI da Tabela 1, foram utilizados os seguintes parâmetros de configuração : o método de seleção utilizada foi o de torneio, com o valor de torneio = 3; P = 400 e P = 100 representando o tamanho da população inicial; K = 4, K = 7 e K = 8 representando o número de ilhas; N = 100 representando o numero de vezes que o ótimo global foi encontrado em um total de 100 execuções independentes; PR = 0,6 representando a probabilidade de recombinação e PM = 0,025 representando a probabilidade de mutação. Após as 100 execuções é retirado os resultados das Médias de Gerações, Média, Desvio Padrão e o Tempo.

Para o Problema Weing7(105,2) que representa um Problema de Mochila Multidimensional (PMM) com 105 objetos e 2 mochilas, e afim de simplificar os resultados mostrados na Tabela 2, foram encontrados os seguintes melhores parâmetros de configuração para os ensaios realizados no AC-MI : torneio = 3, P = 400 representando o tamanho da população inicial, K = 4 representando o número de ilhas, PR = 0,9 representando a probabilidade de recombinação e PM = 0,1

representando a probabilidade de mutação, cujo melhor valor médio encontrado é 1095445 (o valor ótimo) com média de gerações de 134,22 no Tempo de 0.31 segundos. Enquanto em (SILVA, 2012) tem-se valor médio encontrado de 1095445 (o valor ótimo) com média de gerações de 44,49 no Tempo de 0.50 segundos, significando que houve uma redução no tempo de execução para o algoritmo encontrar esses resultados, com um aumento na média de gerações. Ou seja, a nova parametrização conseguiu encontrar a melhor solução em um número maior de execuções em um menor tempo.

Para o Problema Petersen6(39,5), que representa um Problema de Mochila Multidimensional (PMM) com 39 objetos e 5 mochilas, e afim de simplificar os resultados mostrados na Tabela 2, foram encontrados os seguintes melhores parâmetros de configuração para os ensaios realizados no AC-MI : torneio = 3, P = 400 representando o tamanho da população inicial, K = 8 representando o número de ilhas, PR = 0,9 representando a probabilidade de recombinação e PM = 0,075 representando a probabilidade de mutação, cujo melhor valor médio encontrado é 10618 (o valor ótimo) com média de gerações de 50,58 no Tempo de 0.12 segundos. Enquanto em (SILVA, 2012) tem-se valor médio encontrado de 10618 (o valor ótimo) com média de gerações de 30,22 no Tempo de 0.35 segundos, significando que houve uma redução no tempo de execução para o algoritmo encontrar esses resultados, com um aumento na média de gerações. Ou seja, novamente, a nova parametrização conseguiu encontrar a melhor solução em um número maior de execuções em um menor tempo.

Para o Problema Petersen7(50,5) que representa um Problema de Mochila Multidimensional (PMM) com 50 objetos e 5 mochilas, e afim de simplificar os resultados mostrados na Tabela 2, foram encontrados os seguintes melhores parâmetros de configuração para os ensaios realizados no AC-MI : torneio = 3, P = 400 representando o tamanho da população inicial, K = 4 representando o número de ilhas, PR = 0,9 representando a probabilidade de recombinação e PM = 0,1 representando a probabilidade de mutação, cujo melhor valor médio encontrado é 16537 (o valor ótimo) com média de gerações de 142,16 no Tempo de 0.25 segundos. Enquanto em (SILVA, 2012) tem-se valor médio encontrado de 16537 (o valor ótimo) com média de gerações de 78,49 no Tempo de 0.33 segundos,

significando que houve uma redução no tempo de execução para o algoritmo encontrar esses resultados, com um aumento na média de gerações. Ou seja, outra vez, a nova parametrização conseguiu encontrar a melhor solução em um número maior de execuções em um menor tempo.

Para o Problema Sento1(60,30) que representa um Problema de Mochila Multidimensional (PMM) com 60 objetos e 30 mochilas, e afim de simplificar os resultados mostrados na Tabela 2, foram encontrados os seguintes melhores parâmetros de configuração para os ensaios realizados no AC-MI : torneio = 3, P = 400 representando o tamanho da população inicial, K = 4 representando o número de ilhas, PR = 0,75 representando a probabilidade de recombinação e PM = 0,1 representando a probabilidade de mutação, cujo melhor valor médio encontrado é 7772 (o valor ótimo) com média de gerações de 173,38 no Tempo de 1.00 segundos. Enquanto em (SILVA, 2012) tem-se valor médio encontrado de 7772 (o valor ótimo) com média de gerações de 87,44 no Tempo de 1.11 segundos, significando que houve uma redução no tempo de execução para o algoritmo encontrar esses resultados, com um aumento na média de gerações. Ou seja, comprova-se que a nova parametrização consegue encontrar a melhor solução em um número maior de execuções em um menor tempo.

Conforme mostra-se na Tabela 2, é possível observar que o AC-MI com uma melhor parametrização obtém resultados superiores, ao variar o número de subpopulações (ilhas) e o tamanho da subpopulação. Ressalta-se ainda que o AC-MI melhor parametrizado permaneceu encontrando sempre todos os valores para N e com o menor *DesvPad*. Sendo que a principal diferença foi em apresentar os maiores valores para *média de gerações*. Estes resultados mostram que o AC-MI melhor parametrizado produz um maior desempenho para todos os índices de desempenho utilizados.

Tabela 2 : Resultados Obtidos para o Problema da Mochila em Configurações *Benchmarks* pelo Método AC-MI com Variação dos seus Parâmetros.

Problema (n,m)		P	K	N	Média de Gerações	Média	DesvPd	Tempo (m.ss)
H. M. Weingartner and D. N. Ness	Weing7 (105,2)	400	8	50	92.78	1095445	0.00	0.33
	Weing7 (105,2)	400	4	50	134.22	1095445	0.00	0.31
	Weing7 (105,2)	800	8	50	55.48	1095445	0.00	0.27
	Weing7 (105,2)	800	4	50	91.86	1095445	0.00	0.33
C. C. Petersen	Petersen6 (39,5)	400	8	50	50.58	10618	0.00	0.12
	Petersen6 (39,5)	400	4	50	49.28	10618	0.00	0.09
	Petersen6 (39,5)	800	8	50	27.74	10618	0.00	0.10
	Petersen6 (39,5)	800	4	50	38.44	10618	0.00	0.10
	Petersen7 (50,5)	400	8	50	120.10	16537	0.00	0.29
	Petersen7 (50,5)	400	4	50	142.16	16537	0.00	0.25
	Petersen7 (50,5)	800	8	50	77.44	16537	0.00	0.25
	Petersen7 (50,5)	800	4	50	103.94	16537	0.00	0.33
S. Senyu and Y. Toyada	Sento1 (60,30)	400	8	50	134.44	7772	0.00	1.07
	Sento1 (60,30)	400	4	50	173.38	7772	0.00	1.00
	Sento1 (60,30)	800	8	50	112.54	7772	0.00	1.15
	Sento1 (60,30)	800	4	50	120.14	7772	0.00	1.07

Como já descrito anteriormente, o item *média de gerações* foi introduzido, a fim de avaliar outro tipo de desempenho cujo valor representa a média de gerações em que o valor ótimo foi encontrado para 50 execuções independentes para cada problema apresentado.

4.4. EXPERIMENTOS DE COMPARAÇÃO DE DESEMPENHO DO AC-MI COM O ALGORITMO GENÉTICO SIMPLES (AG) E COM O ALGORITMO CULTURAL PADRÃO (AC)

No intuito de testar o AC-MI no problema de mochila multidimensional com diferentes espaços de busca, e comparar o seu desempenho com o Algoritmo Genético Simples (AG) e com o Algoritmo Cultural Padrão (AC), foram selecionadas algumas instâncias da mochila, todas disponíveis em OR-Library (BEASLEY, 1990) e descritas na Tabela 3, com o número m de mochilas, o número n de itens e o espaço de busca ($m \times 2^n$). Os espaços de busca variam entre $m \times 2^4$ até $m \times 2^{30}$, com valores maiores m , com espaços de busca com mais ótimos locais, aumentando a complexidade da instância.

Tabela 3 : Problemas Utilizados nos Experimentos.

Autor(es)	Instância	m	n	Espaço de Busca ($m \times 2^n$)	Ótimo
A. Freville and G. Plateau	Hp1	28	4	28×2^4	3418
	Hp2	35	4	35×2^4	3186
	Pb2	34	4	34×2^4	3186
C. C. Petersen	Petersen2	10	10	10×2^{10}	87061
	Petersen3	15	10	15×2^{10}	4015
S. Senyu and Y. Toyada	Sento1	60	30	60×2^{30}	7772
	Sento2	60	30	60×2^{30}	8722
H. M. Weingartner and D. N. Ness	Weing1	28	2	28×2^2	141278
	Weing2	28	2	28×2^2	130883
W. Shi	Weish1	30	5	30×2^5	4554
	Weish2	30	5	30×2^5	4536

Para o Algoritmo Genético Simples (AG) e Algoritmo Cultural Padrão (AC), foram utilizadas as melhores parametrizações encontradas no AC-MI, tais como: torneio = 3, P representando o tamanho da população com valor igual a 400 ou 800, K representando o número de ilhas com valor igual a 4 ou 8, PR representando a probabilidade de recombinação, que variavam entre os valores 50%, 75% e 90% e PM representando a probabilidade de mutação, que variavam entre os valores 2,5%, 5%, 7,50% e 10%, conforme ilustrado na Tabela 4.

Tabela 4 : Variação dos Parâmetros para os Problemas Utilizados nos Experimentos.

Problema (n,m)		Algoritmo Genético (AG)			Algoritmo Cultural (AC)		
		P	PR	PM	P	PR	PM
A. Freville and G. Plateau	Hp1 (28,4)	400	50%	2,5%	400	50%	2,5%
	Hp2 (35,4)	400	50%	2,5%	400	50%	2,5%
	Pb2 (34,4)	400	50%	2,5%	400	50%	2,5%
C. C. Petersen	Petersen2 (10,10)	400	50%	2,5%	400	50%	2,5%
	Petersen3 (15,10)	400	90%	2,5%	400	90%	2,5%
S. Senyu and Y. Toyada	Sento1 (60,30)	400	75%	10%	400	75%	10%
	Sento2 (60,30)	400	50%	10%	400	50%	10%
H. M. Weingartner and D. N. Ness	Weing1 (28,2)	400	50%	10%	400	50%	10%
	Weing2 (28,2)	400	90%	10%	400	90%	10%
W. Shi	Weish01 (30,5)	400	90%	2,5%	400	90%	2,5%
	Weish02 (30,5)	400	50%	5%	400	50%	5%

Os ensaios aqui apresentados consistem em 50 execuções independentes e cada execução utiliza uma semente (*seed*) diferente para a população inicial aleatória. Na Tabela 5 encontram-se os resultados obtidos pelo Algoritmo Genético Simples (AG) para todos os *benchmarks* utilizados.

Tabela 5 : Resultados para Todos os *Benchmarks* da Mochila pelo Método AG.

Problema (n,m)		Algoritmo Genético (AG)			
		Média	DesvPD	Ótimo Encontrado	Tempo (m.ss)
A. Freville and G. Plateau	Hp1 (28,4)	3393.48	20.08	3404	0.25
	Hp2 (35,4)	3169.52	21.84	3186	0.25
	Pb2 (34,4)	3179.88	12.74	3186	0.24
C. C. Petersen	Petersen2 (10,10)	87005.16	169.35	87061	0.20
	Petersen3 (15,10)	4015.00	0.00	4015	0.19
S. Senyu and Y. Toyada	Sento1 (60,30)	3991.48	742.97	4553	0.38
	Sento2 (60,30)	8266.86	57.76	8243	0.44
H. M. Weingartner and D. N. Ness	Weing1 (28,2)	140653.74	390.33	140568	0.21
	Weing2 (28,2)	129874.52	672.18	130053	0.21
W. Shi	Weish01 (30,5)	4552.52	10.46	4554	0.21
	Weish02 (30,5)	4534.86	4.26	4536	0.23

Na Tabela 6 encontram-se os resultados obtidos pelo Algoritmo Cultural Padrão (AC) para todos os *benchmarks* utilizados.

Tabela 6 : Resultados para Todos os *Benchmarks* da Mochila pelo método AC.

Problema (n,m)		Algoritmo Cultural (AC)			
		Média	DesvPD	Ótimo Encontrado	Tempo (m.ss)
A. Freville and	Hp1 (28,4)	3399.60	13.52	3404	0.23
G. Plateau	Hp2 (35,4)	3178.78	9.31	3186	0.25
	Pb2 (34,4)	3185.28	3.56	3186	0.15
C. C. Petersen	Petersen2 (10,10)	87061.00	0.00	87061	0.01
	Petersen3 (15,10)	4015.00	0.00	4015	0.02
S. Senyu and	Sento1 (60,30)	4756.26	970.61	3428	0.45
Y. Toyada	Sento2 (60,30)	8644.86	28.11	8622	0.53
H. M. Weingartner and	Weing1 (28,2)	141278.00	0.00	141278	0.10
D. N. Ness	Weing2 (28,2)	130883.00	0.00	130883	0.11
W. Shi	Weish01 (30,5)	4554.00	0.00	4554	0.11
	Weish02 (30,5)	4536.00	0.00	4536	0.11

Na Tabela 7 encontram-se os resultados obtidos pelo AC-MI com a sua melhor parametrização para todos os *benchmarks* utilizados.

Tabela 7 : Resultados para Todos os *Benchmarks* pelo Método AC-MI com a sua Melhor Parametrização para cada uma das Instâncias Utilizadas do Problema da Mochila Multidimensional.

Problema (n,m)		Algoritmo Cultural Modelo de Ilha (AC-MI)			
		Média	DesvPD	Ótimo Encontrado	Tempo (m.ss)
A. Freville and	Hp1 (28,4)	3418.00	0.00	3418	0.05
G. Plateau	Hp2 (35,4)	3186.00	0.00	3186	0.17
	Pb2 (34,4)	3186.00	0.00	3186	0.21
C. C. Petersen	Petersen2 (10,10)	87061.00	0.00	87061	0.01
	Petersen3 (15,10)	4015.00	0.00	4015	0.02
S. Senyu and Y. Toyada	Sento1 (60,30)	7772.00	0.00	7772	1.00
	Sento2 (60,30)	8722.00	0.00	8722	1.03
H. M. Weingartner and D. N. Ness	Weing1 (28,2)	141278.00	0.00	141278	0.03
	Weing2 (28,2)	130883.00	0.00	130883	0.04
W. Shi	Weish01 (30,5)	4554.00	0.00	4554	0.15
	Weish02 (30,5)	4536.00	0.00	4536	0.04

As variações dos resultados da Tabela 5, Tabela 6 e Tabela 7 refletem o comportamento de cada algoritmo frente a instâncias com espaço de busca de tamanho e complexidade diferentes.

De uma maneira geral observa-se que o AC-MI é mais rápido que o AG e o AC. Das onze instâncias do PMM, apenas em quatro o AC-MI foi mais lento do que o AC (Pb2(34,4), Sento1(60,30), Sento2(60,30) e Weish01(30,5)), sendo que em

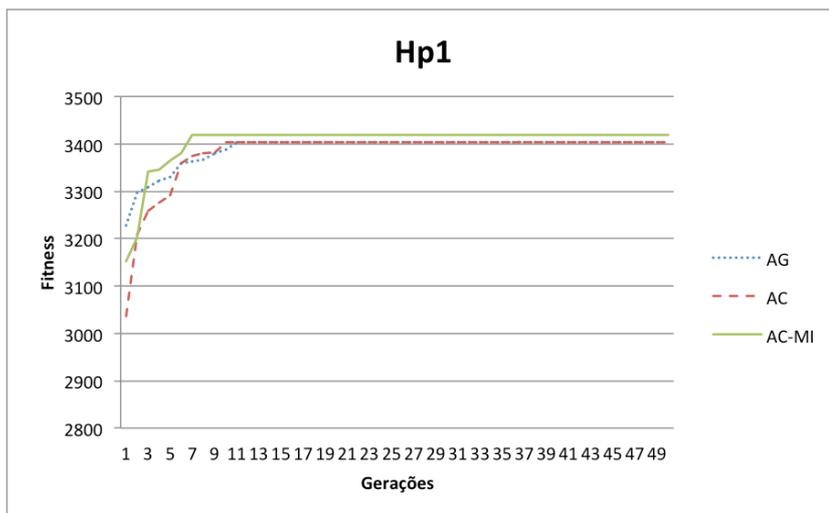
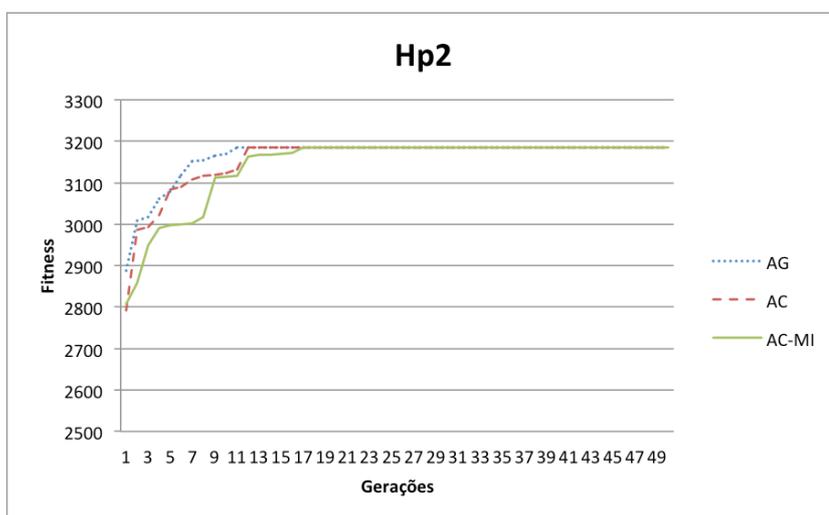
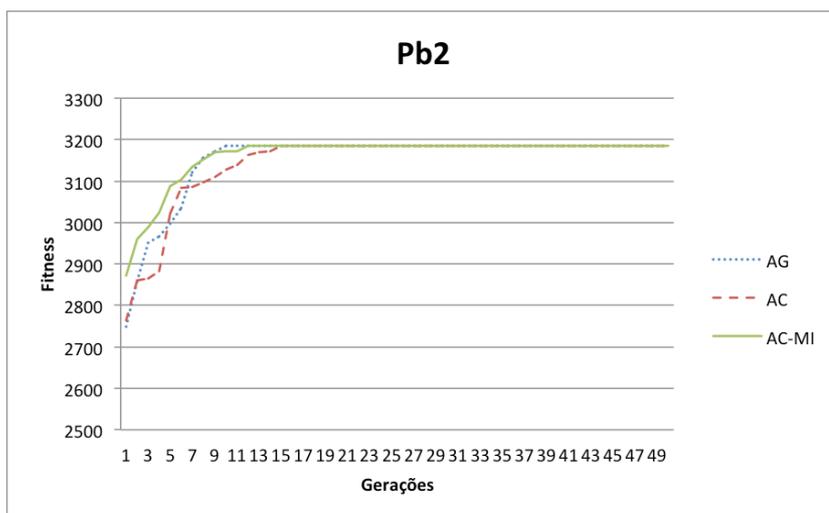
Sento1(60,30) e Sento2(60,30) o AC-MI também foi mais lento do que o AG. Em Petersen2(10,10) e Petersen3(15,10) o AC-MI foi tão rápido quanto o AC.

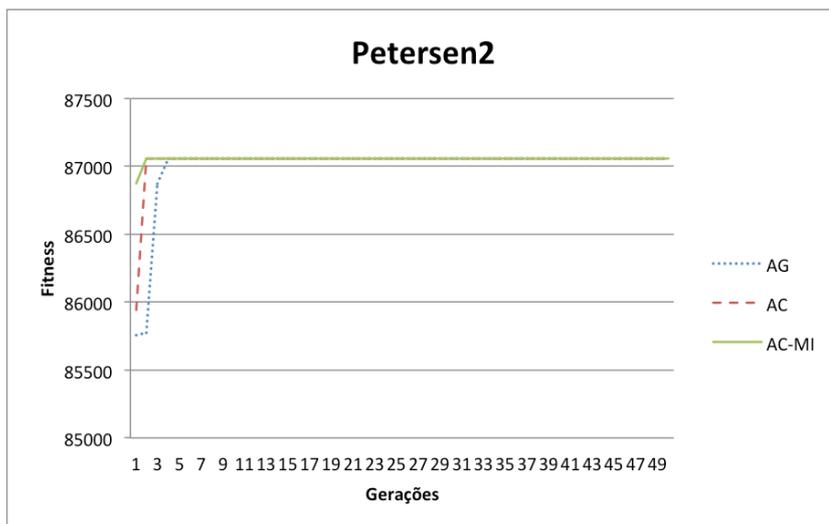
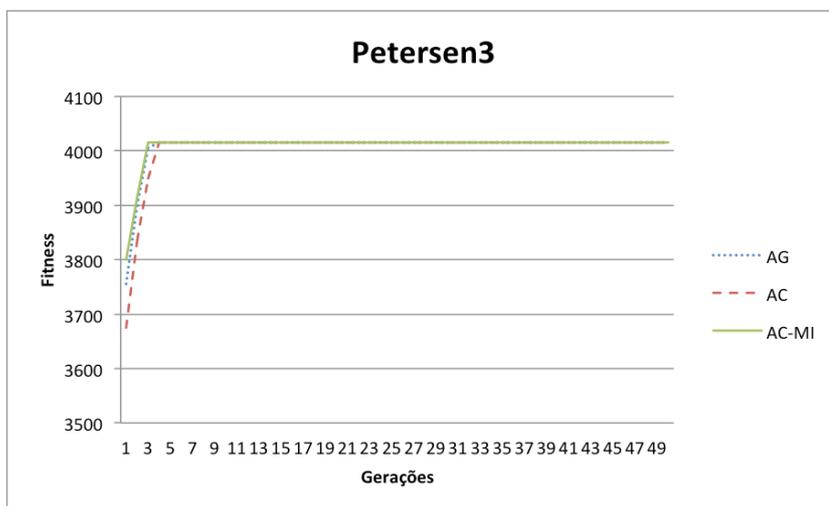
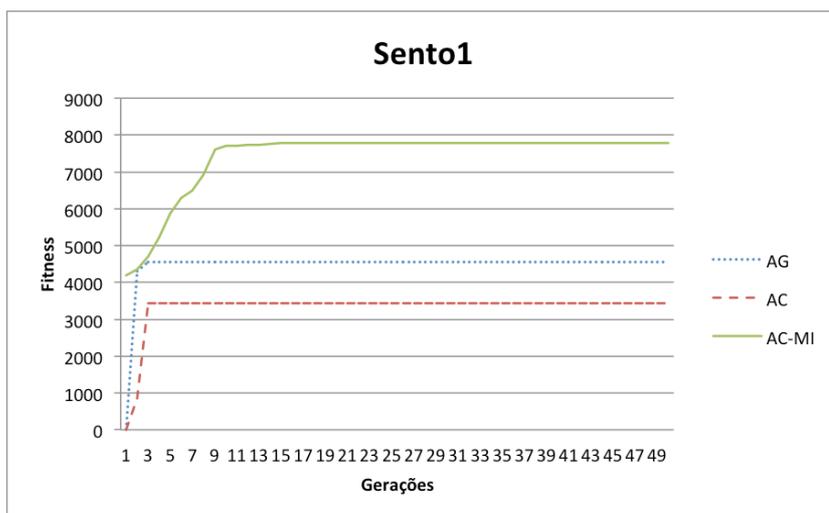
O AC-MI encontrou um ótimo melhor do que o AG e AC em três instâncias (Hp1(28,4), Sento1(60,30) e Sento2(60,30)). Nas outras instâncias o ótimo do AC-MI foi igual ao AG e ao AC, exceto em Weing1(28,2) e Weing2(28,2) onde o AC-MI teve um ótimo melhor do que o AG mas igual ao AC.

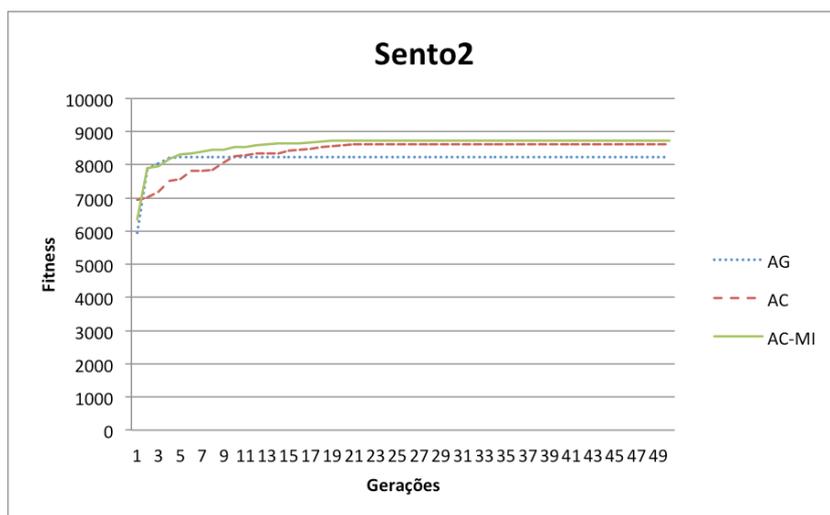
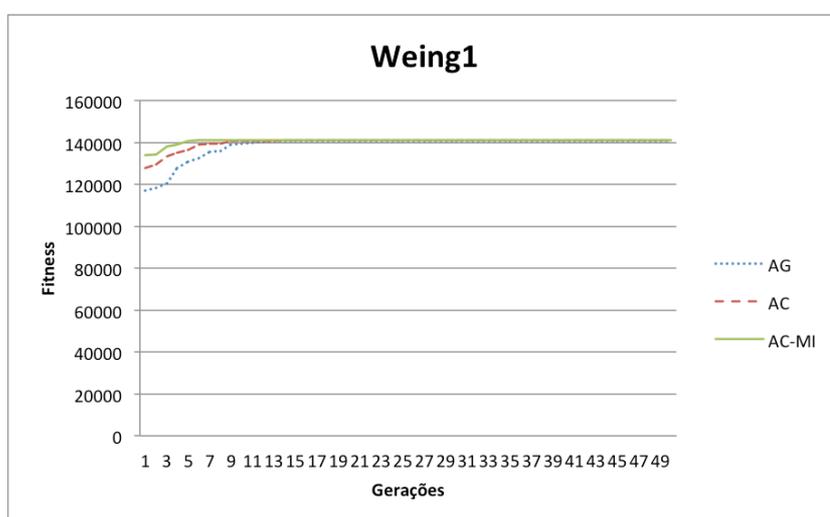
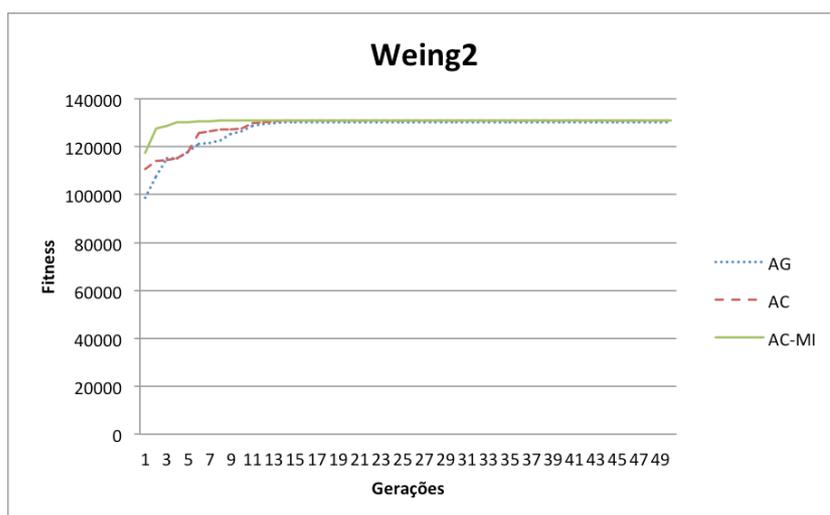
O desvio padrão do AC-MI foi zero em todas as instâncias, enquanto o AC teve zero apenas em seis (Petersen2(10,10), Petersen3(15,10), Weing1(28,2), Weing2(28,2), Weish01(30,5) e Weish02(30,5)). O AG só teve desvio padrão zero em Petersen3(15,10).

A média do AC-MI foi maior do que a do AC, em cinco instâncias (Hp1(28,4), Hp2(35,4), Pb2(34,4), Sento1(60,30), Sento2(60,30)). Nas outras seis instâncias (Petersen2(10,10), Petersen3(15,10), Weing1(28,2), Weing2(28,2), Weish01(30,5) e Weish02(30,5)) a média do AC-MI foi igual ao do AC. Em todas as instâncias a média do AC-MI foi maior do que a do AG, exceto em Petersen3(15,10) onde foi igual.

Para continuar comparando o desempenho do Algoritmo Genético Simples (AG), Algoritmo Cultural Padrão (AC) e do AC-MI utilizados nesses experimentos, os Gráficos 1, 2, 3, 4, 5, 6, 7, 8, 9, 10 e 11 mostram a convergência das soluções para o máximo obtido. Eles representam o valor da função *fitness* da melhor solução a cada geração do Algoritmo Genético Simples (AG) ou iteração do Algoritmo Cultural Padrão (AC) e do AC-MI.

Gráfico 1 : Função *fitness* (Hp1).Gráfico 2 : Função *fitness* (Hp2).Gráfico 3 : Função *fitness* (Pb2).

Gráfico 4 : Função *fitness* (Petersen2).Gráfico 5 : Função *fitness* (Petersen3).Gráfico 6 : Função *fitness* (Sento1).

Gráfico 7 : Função *fitness* (Sento2).Gráfico 8 : Função *fitness* (Weing1).Gráfico 9 : Função *fitness* (Weing2).

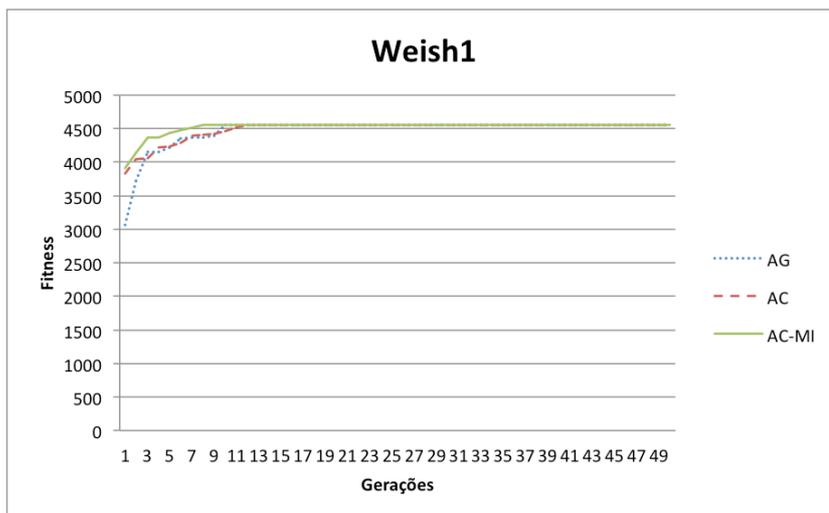


Gráfico 10 : Função *fitness* (Weish1).

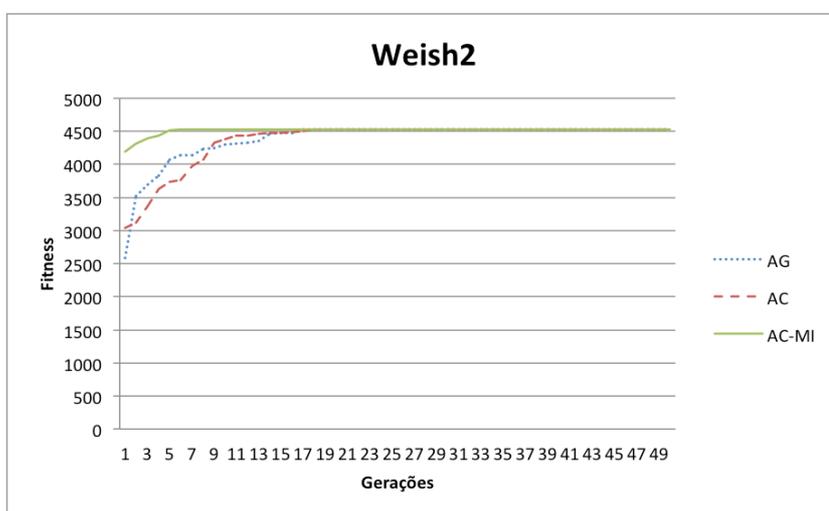


Gráfico 11 : Função *fitness* (Weish2).

Ao analisar os Gráficos 1-11 pode-se perceber como cada algoritmo se comporta durante a evolução da melhor solução encontrada. Nos Gráficos 2, 3, 4, 5, 10 e 11 nota-se que todos os algoritmos (AG, AC e AC-MI) atingiram o valor máximo de *fitness* com o seu valor ótimo conhecido.

Nos Gráficos 1, 8 e 9 os valores da função *fitness* do AC-MI convergem muito rapidamente quando comparados com os valores de AG e AC. Esta característica sugere que o AC-MI encontra mais rapidamente a solução, que é melhor do que as encontradas pelo AG e pelo AC. É importante destacar que nesses gráficos apenas o AC-MI e AC atingiram o valor máximo de *fitness* com o seu valor ótimo conhecido.

Nos Gráficos 6 e 7 evidenciam ainda mais a diferença de evolução do AC-MI para o AG e AC, para os problemas com espaço de busca mais complexos. Nestes casos, a função *fitness* do AC-MI evolui mais suavemente, sugerindo a manutenção da diversidade ao longo das iterações. Esta é uma característica essencial para obtenção de soluções boas para instâncias de grandes complexidades como a instância SENTO1 e SENTO2 (Tabela 3).

CAPÍTULO 5. CONCLUSÃO

O problema da mochila multidimensional e suas aplicações na indústria são motivações para pesquisadores buscarem novos e alternativos métodos para resolver problemas NP-completos. As aplicações do problema da mochila multidimensional em problemas do mundo real criam a necessidade de algoritmos mais rápidos e eficientes. Os experimentos feitos com o Algoritmo AC-MI apresentam resultados muito satisfatórios, com uma vantagem significativa quando comparado com os resultados alcançados com o AG e AC, além de apontarem as meta-heurísticas como alternativas eficazes e interessantes para este problema. O tempo em processamento de cada instância também é um resultado muito relevante para os experimentos. Ele demonstra o esforço computacional que cada método necessita para processar seu algoritmo.

No problema da mochila multidimensional também foi possível observar o mesmo comportamento envolvendo o AG e AC padrão. Ou seja, ocorreram situações em que não foi possível escapar de ótimos locais em alguns ensaios realizados. Na busca de soluções para tal problema, observou-se que a inclusão de multipopulações no Algoritmo AC-MI melhorou a convergência e a velocidade de busca. O Algoritmo AC-MI melhorou as médias de gerações em todos os experimentos quando comparados ao AG e AC padrão.

Os resultados positivos apoiam a ideia de que a hibridização de algoritmos e o uso de multipopulações nos Algoritmos Culturais é uma abordagem altamente desejável para direcionar o espaço de busca para encontrar soluções mais próximas do valor ótimo.

A hibridização dos métodos também é uma tendência nos algoritmos evolutivos, gerando, assim, outras vertentes para trabalhos futuros e passando assim a ser uma diretriz de estudos. Ainda que não se deve utilizar qualquer tipo de hibridização ou qualquer algoritmo de busca local. A hibridização correta para cada problema é o desejável, e que o uso de algoritmos genéticos e algoritmos culturais híbridos facilitam a implementação e produzem bons resultados.

5.1. TRABALHOS FUTUROS

Como principais recomendações para trabalhos futuros propõe-se:

- ✓ **Outros tipos de conhecimento:** Neste trabalho foi utilizado um tipo de conhecimento, o conhecimento situacional, sendo que existem outros tipos de conhecimento que seriam interessantes de serem utilizados na resolução do PMM, tais como o conhecimento normativo, conhecimento histórico e o conhecimento topográfico;
- ✓ **Hibridização:** Apesar dos algoritmos culturais serem híbridos por natureza, seria interessante estudar-se a hibridização com outros métodos;
- ✓ **Outras populações:** apesar dos algoritmos genéticos serem bastantes eficientes e serem comprovadamente bons algoritmos para a resolução do PMM, outras técnicas, como as Estratégia Evolutivas e a Programação Evolutiva, poderiam ser utilizadas no espaço populacional;
- ✓ **Ajustes do AG e AC:** Elaborar uma melhor parametrização do AG Simples e AC Padrão para realizar uma comparação com o melhor do AC-MI.

REFERÊNCIAS BIBLIOGRÁFICAS

AGUIRRE H.; TANAKA K.; SUGIMURA T.; OSHITA S. **Improved Distributed Genetic Algorithm with Cooperative-Competitive Genetic Operators**. In: Proceedings IEEE International Conference on Systems, Man, and Cybernetics (SMC 2000), pp. 3816-3822, Oct. 2000.

ANDRÉASSON N., EVGRAFOV A., PATRIKSSON M. **An Introduction to Optimization: Foundations and Fundamental Algorithms**, 2005. Disponível em: http://www.math.chalmers.se/Math/Grundutb/CTH/tma947/0405/kompendium_sub.pdf. Acessado em março de 2015.

ASTOLFI A. **Optimization: An introduction**, 2004, Disponível em <http://cap.ee.imperial.ac.uk/~astolfi/Courses/outs/Optim.pdf> , acessado em março de 2015.

BADILLO, A. R., RUIZ, J. J., COTTA, C., FERNÁNDEZ-LEIVA, A. J. **On user centric memetic algorithms**. Soft Computing, Vol. 17, Issue 2, pp- 285-300, 2013.

BEASLEY, D.; BULL, D. R.; MARTIN, R. R. **An Overview of Genetic Algorithms: Part 1, Fundamentals**. Inter-University Committee on Computing. University Computing, UK, 1993.

BEASLEY J. E. **OR-Library: distributing test problems by electronic mail**, Journal of the Operational Research Society, 41(11), PP.1069-1072, 1990.

BECERRA, R. L.; COELLO, C. A. C. **Optimization with constraints using a cultured differential evolution approach**. In: Genetic and Evolutionary Computation Conference. [S.l.: s.n.], p. 27-34, 2005.

BECERRA R. L. **Algoritmos Culturales Aplicados a Optimización con Restricciones y Optimización Multiobjetivo**. Tese (Doutorado) - Instituto Politécnico Nacional do México, 2002.

BELLMAN R. E. **Dynamic Programming**. Princeton University, Princeton University Press, Princeton, NJ, 1957.

BRAUN H. **On solving traveling salesman problems by genetic algorithms.** In: H.-P. Schwefel and R. Manner, editors, *Parallel Problem Solving from Nature - Proceedings of 1st Workshop, PPSN 1*, volume 496 of *Lecture Notes in Computer Science*, pages 129-133, Dortmund, Germany: Springer-Verlag, Berlin, Germany, 1991.

CANTU-PAZ E. **Efficient and Accurate Parallel Genetic Algorithms.** Norwell, MA, USA: Kluwer Academic Publishers, 2000.

CHUNG C. J., REYNOLDS R. **A tested for solving optimization problems using culture algorithms.** In *Evolutionary Programming V: Proceedings of the Fifth Annual Conference on Evolutionary Programming*. Cambridge, Massachusetts: MIT Press, 1996.

DANCHIN E.; GIRALDEAU L.; VALONE T.; WAGNER R. **Public information: from nosy neighbors to cultural evolution.** *Science*, 305, 487-491, 2004.

DAWKINS R. **O gene egoísta.** Ed. Itatiaia Ltda, 2001.

DELIN L., LIXIAO Z., ZHIHUI X. **Heuristic Simulated Annealing Genetic Algorithm for Traveling Salesman Problem**, *Computer Science & Education (ICCSE)*, 2011 6th International Conference on, On page(s): 260 - 264, Volume: Issue: 3-5 Aug. 2011.

DE CASTRO, L. N. **Fundamentals of Natural Computing: Basic Concepts, Algorithms and Applications.** Chapman & Hall/CRC, 2006.

FUKUNAGA A. S., TAZOE S. **Combining Multiple Representations in a Genetic Algorithm for the Multiple Knapsack Problem**, 2009 IEEE Congress on Evolutionary Computation (CEC 2009), Trondheim, Norway, pp.2423-2430, May, 2009.

GEERTZ, Clifford. **A Interpretação das Culturas.** Rio de Janeiro (RJ). Editora Guanabara, Rio de Janeiro, 1989.

GEN M., CHENG R. **Genetic Algorithms and Engineering Optimization**, John Wiley & Sons, Inc., 2000.

GOLDBERG D.E. **Genetic Algorithms in Search, Optimization and Machine Learning**, AddisonWesley, 1989.

GOMES DA SILVA C., FIGUEIRA J., CLÍMACO J. **An interactive procedure dedicated to the bi-criteria knapsack problem**. (In Portuguese), Research Report N° 4, INESC -Coimbra, Portugal, 2003.

GONG M., LICHENG J., WENPING M., SHUIPING G. **Solving multidimensional knapsack problems by an immune-inspired algorithm**, *Evolutionary Computation*, Issues.25-28, pp.3385-3391, Sept 2007.

GORDON V. S., BÖHM P. W., and WHITLEY L. D. **A note on the performance of genetic algorithms on zero-one knapsack problems** in Proc. 1994 ACM Symp. Applied Computing, E. Deaton, D. Oppenheim, J. Urban, and H. Berghel, Eds. New York: ACM, pp. 194–195, 1994.

GUO Y.-N., CHENG J., CAO Y.-Y., LIN Y. **A novel multi-population cultural algorithm adopting knowledge migration**. *Soft Comput.* 15(5): 897-905, 2011.

HAUPT R. L; HAUPT S. E. **Practical Genetic Algorithms**. Second Edition John Wiley & Sons, Inc., 2004.

HE Q., WANG L. **An effective co-evolutionary particle swarm optimization for constrained engineering design problem**. *Engineering Applications of Artificial Intelligence*, Elsevier, 20 (1): 88-99, 2007.

HOLLAND J. H. **Adaptation in Natural and Artificial Systems**. University of Michigan Press, 1975.

IACOBAN, R.; REYNOLDS, R.; BREWSTER, J. **Cultural swarms: modeling the impact of culture on social interaction and problem solving**. In: IEEE Swarm Intelligence Symposium. [S.l.: s.n.], p. 205-211, 2003.

IBA, H., NOMAN, N. **New Frontier in Evolutionary Algorithms - Theory and Applications**. Imperial College Press, 2012.

JIAO L. C., WANG L. A. **Novel Genetic Algorithm Based on Immunity**, IEEE Transactions on Systems, Man, and Cybernetics Part A: Systems and Humans, Sept., 30(5): 552-561, 2000.

JIN X.; REYNOLDS R. G. **Using knowledge-based evolutionary computation to solve nonlinear constraint optimization problems: a cultural algorithm approach**. Proceedings of the 1999 IEEE Congress on Evolutionary Computation - CEC 99, Volume 3, July, Washington, DC, USA, p. 1672 - 1678, 1999.

JOHN H. D., MATTHEW H., KHALED I., ENDER O. **A genetic programming hyper-heuristic for the multidimensional knapsack problem**, Kybernetes, Vol. 43 Iss: 9/10, pp.1500 – 1511, 2014.

KRAUSE J., CORDEIRO J. A., LOPES H. S. **Comparação de Métodos de Computação Evolucionária para o Problema da Mochila Multidimensional**. Meta-Heurísticas em Pesquisa Operacional, Lopes H. S., Rodrigues L. C. A., Steiner M. T. A. (Eds.), OmniPax Editora, 2013.

KHURI S., BACK T., and HEITK ÖTTER J. **An evolutionary approach to combinatorial optimization problems**, in Proc. 22nd Annu. ACM Computer Science Conf., D. Cizmar, Ed. New York: ACM, pp. 66–73, 1994.

LEÃO A. A. S., CHERRI L. H., ARENALES M. N. **Determining the K-best solutions of knapsack problems**. Computers & Operations Research, v. 49, p. 71-82, September, 2014.

Levine, D. **A parallel genetic algorithm for set partitioning problem**, Technical report. ANL-94/23 Argonne National Laboratory, 1994.

LINDEN R. **Algoritmos Genéticos**, 2 ed., Rio de Janeiro: Brasport, 2008.

MARTELLO S., TOTH P. **Knapsack problems - Algorithms and computer implementation**, John Wiley & Sons, Inc, 1990.

MOHAMED E. R. et al. **Performance Evaluation of Genetic Algorithm for solving Routing Problem in Communication Network**, International Journal of Computer Science and Information Security, pp 37-43, Vol 8 Number 3, 2010.

NABHOLZ, R. **Seleção Ótima de Ativos Multi-Período com Restrições Intermediárias utilizando o Critério Média-Variância**. Tese de Doutorado, POLI-USP, 2006.

NERI, F., COTTA, C. **Memetic algorithms and memetic computing optimization: A literature review**. *Swarm and Evolutionary Computation*, 2, pp. 1-14, 2012.

PISINGER D. **Algorithms for knapsack problems**, Ph.D. thesis, Dept. of Computer Science, University of Copenhagen, 1995.

RAO S. S. **Engineering Optimization: Theory and Practice**, Wiley, 2009.

RAVINDRAN A., RAGSDELL K. M., REKLAITIS G. V. **Engineering Optimization: Methods and Applications**, 2nd Edition, ISBN: 978-0-471-55814-9, Wiley, 2007.

REYNOLDS, R. G. **Cultural Swarms: Knowledge-driven Problem Solving in Social Systems**. In: *IEEE International Conference on Systems, Man and Cybernetics*, p. 3589-3594 vol.4, 2003.

REYNOLDS, R. G. **An introduction to cultural algorithm**. In: *3rd Annual Conference on Evolutionary Programming*. [S.l.: s.n.], 1994.

REYNOLDS R. G., CHE X. D., ALI M. **Weaving the social fabric: The past, present and future of optimization problem solving with cultural algorithms**. *Int. J. Intell. Comput. Cybern.*, 3(4): 561-592, 2010.

REYNOLDS, R. G.; PENG B.; CHE, X. **Knowledge Swarms: Generating Emergent Social Structure in Dynamic Environments**. Paper extracted from *Proceedings of the Agent 2005. Conference on Generative Social Processes*, Chicago, October 13-15, 2005.

REYNOLDS, R. G.; SALEEM, S. M. **The Impact of Environmental Dynamics on Cultural Emergence. Perspectives on Adaptions in Natural and Artificial Systems**. Oxford University Press, 2005.

REYNOLDS, R. G.; ZANONI, E. **Why cultural evolution can proceed faster than biological evolution**. In: *Proceedings of International Symposium on Simulating Societies*. [S.l.: s.n.], p. 81-93, 1992.

SILVA D. J. A. **Algoritmos Culturais com Abordagem Memética e Multipopulacional Aplicados a Problemas de Otimização**. Tese de Doutorado, PPGEE, Universidade Federal do Pará, 2012.

STACH W. **Group for Intelligent Processing of Data Seminar Series. Speeding Up Genetic Algorithms: Parallel Approach**. University of Alberta, 2007.

WHITLEY D. **A Genetic Algorithm Tutorial**. 1993.

WILSON, E. O. **Consiliência**. São Paulo. Ed. Campus, São Paulo (SP), 1999.

WEI D., SHUZHUO L. **An Artificial Intelligence Algorithm for Multi-dimensional Knapsack Problem Based on Small World Phenomenon**. IEEE Conference Publications, vol. 4, pp 665 – 669, 2009.

XU, Q., ZHANG, G., ZHAO, C., An, A. **A robust adaptive hybrid genetic simulated annealing algorithm for the global optimization of multimodal functions**. Chinese Control and Decision Conference (CCDC), May, Mianyang, China, pp. 7-12, 2011.

XUE, Z., GUO, Y. **Improved Cultural Algorithm based on Genetic Algorithm**. IEEE International Conference on Integration Technology - ICIT '07, March, Shenzhen, China, pp. 117-122, 2007.

YAN, X., WU, Q., ZHANG, C., CHEN, W., LUO, W., LI, W. **An Efficient Function Optimization Algorithm based on Culture Evolution**. International Journal of Computer Science Issues, Vol. 9, Issue 5, No 2, pp. 11-18, 2012.

YANG X. S. **Review of meta-heuristics and generalized evolutionary walk algorithm**, International Journal of Bio-Inspired Computation, v.3 n.2, p.77-84, April 2011.

YANG X. S. **Engineering Optimization: An Introduction with Metaheuristic Applications**, John Wiley and Sons, ISBN 0470582464 , 2010.

YEW-SOON, O., MENG, H. L., XIANSHUN, C. **Memetic Computation—Past, Present & Future**. IEEE COMPUTATIONAL INTELLIGENCE MAGAZINE, Volume: 5, Issue: 2, MAY, pp. 24-31, 2010.

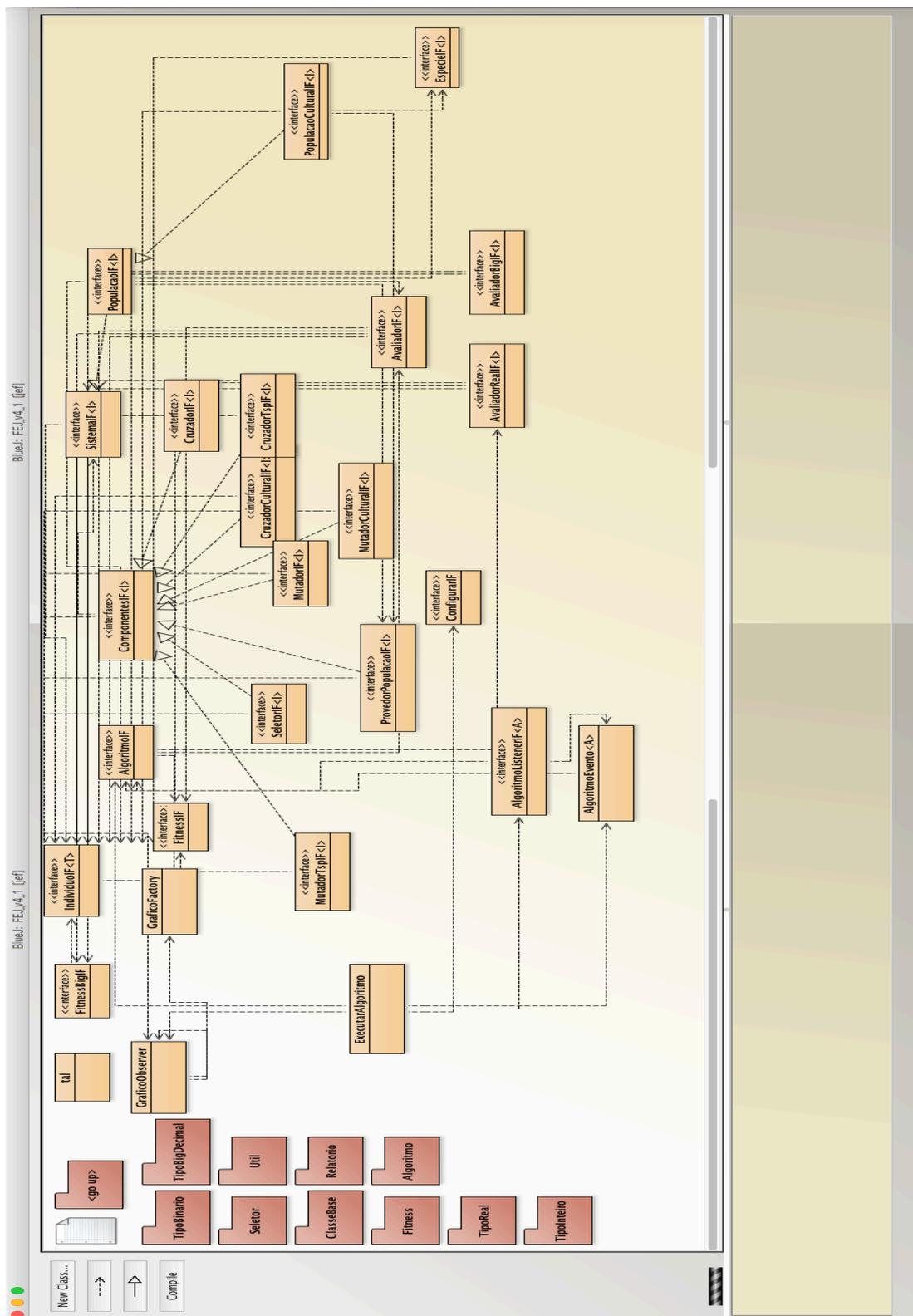
ZHANG Y. **Study on Cultural Algorithm**, International Conference on Future Computer Science and Education, pp.558-560; 2011.

ANEXOS

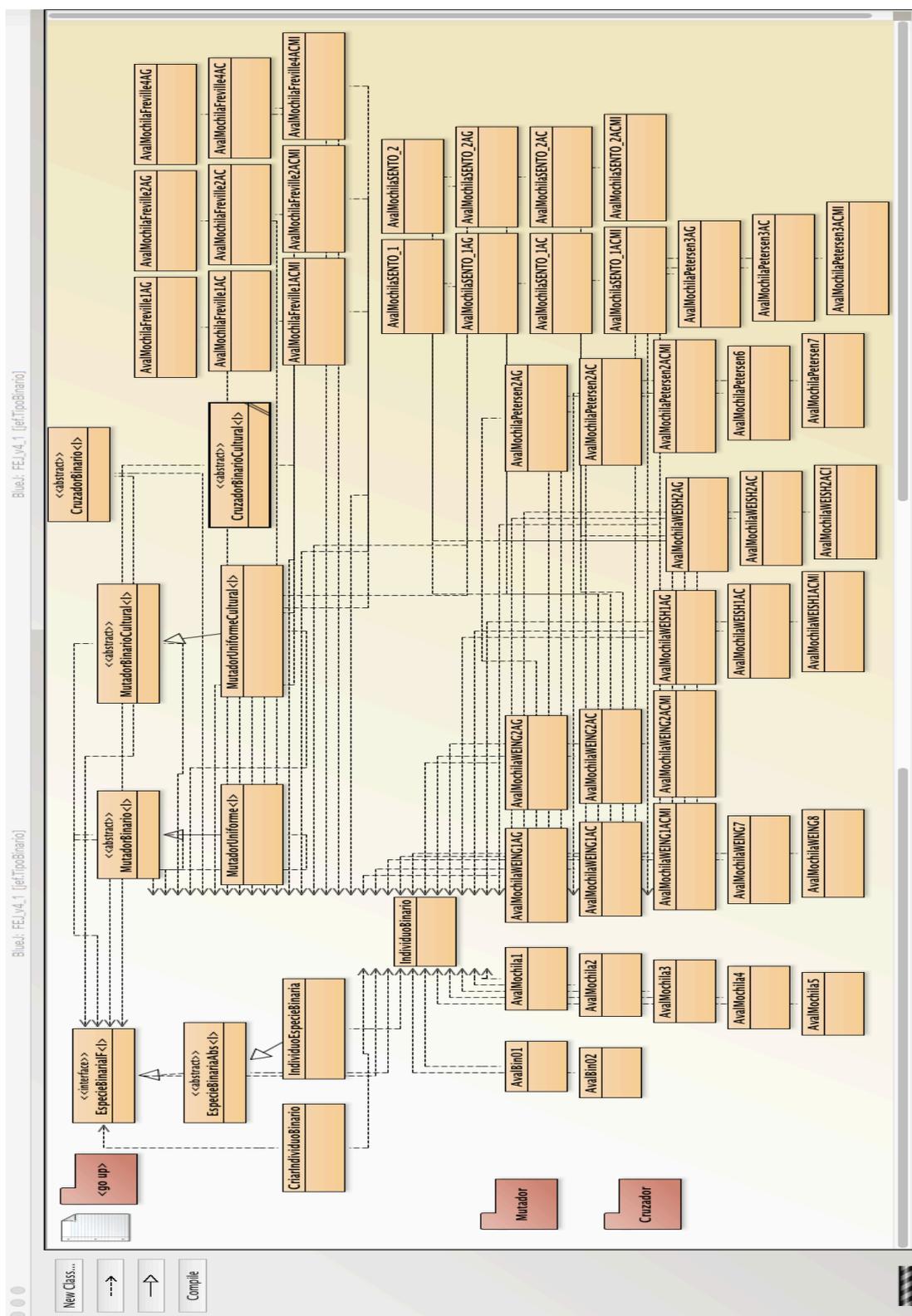
A.1. VISÃO GERAL DO FRAMEWORK UTILIZADO (JAVA EVOLUTIONARY FRAMEWORK - JEF)

Qualquer tipo de algoritmo de Computação Evolutiva (CE) pode ser executado usando o JEF, desde que alguns requisitos mínimos sejam cumpridos. A única condição necessária é ter uma população de indivíduos a que uma sequência de operações evolutivas seja iterativamente aplicada. O sistema JEF foi codificado na linguagem de programação Java, que garante a sua portabilidade entre todas as plataformas que implementam uma JVM (Java Virtual Machine). Declarações de verificação e validação são embutidas no código para assegurar que as operações sejam válidas e para relatar problemas para o usuário. O uso de XML como formato de arquivo também é um aspecto central do JEF, que fornece uma base comum para o desenvolvimento de ferramentas para analisar e gerar arquivos, e para integrar o framework com outros sistemas (SILVA, 2012).

A.1.1. ESTRUTURA PRINCIPAL DO FRAMEWORK

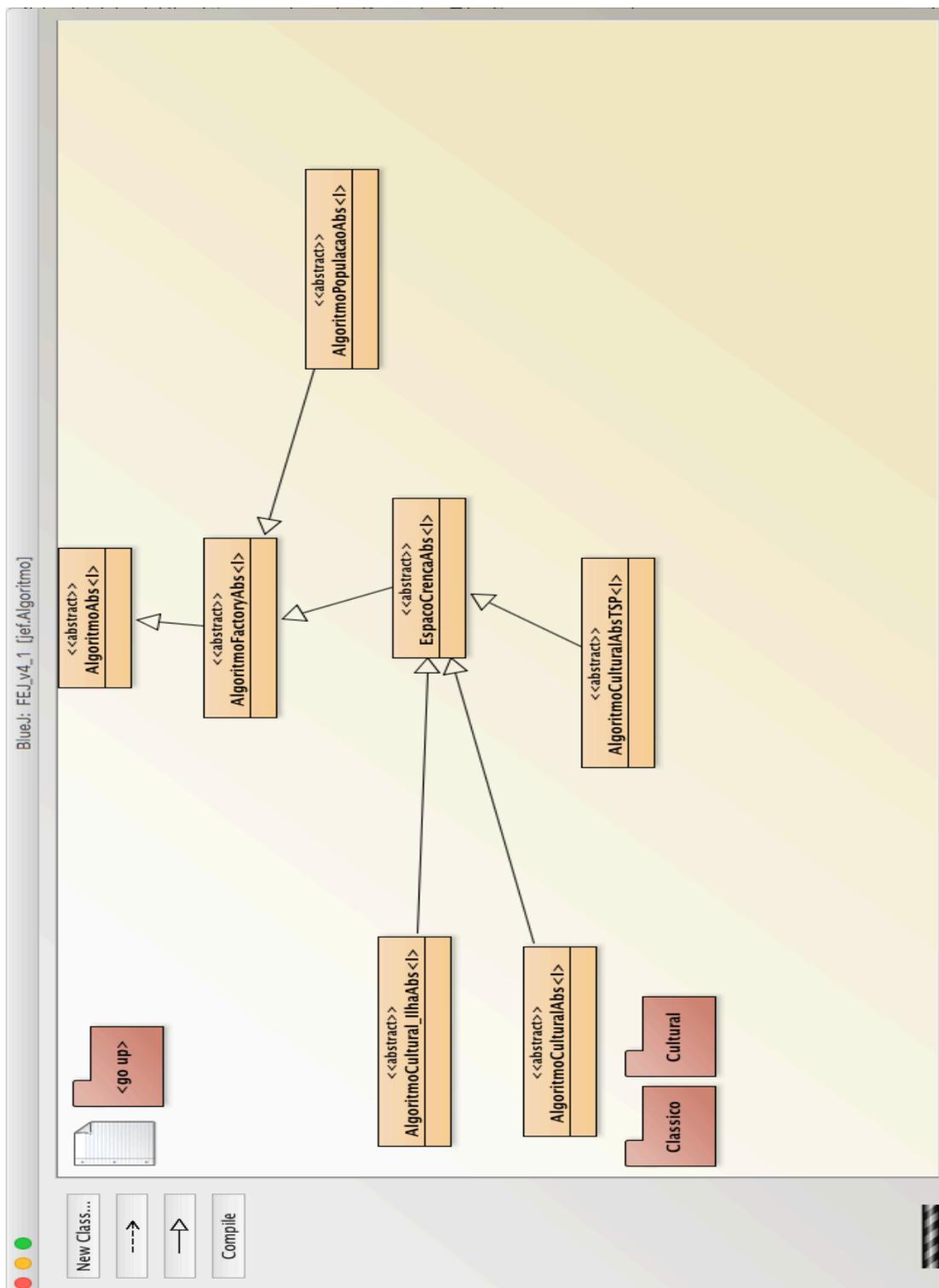


A.1.2. EXEMPLO DE UMA ESTRUTURA DE REPRESENTAÇÃO PARA O CROMOSSOMO



A.1.3. ESTRUTURA DE REPRESENTAÇÃO DOS ALGORITMOS EVOLUTIVOS

A figura abaixo representa a estrutura dos algoritmos evolutivos disponíveis no framework.



A.1.4. ARQUIVO DE PARÂMETROS DO FRAMEWORK EM XML

O arquivo de entrada segue o padrão XML. Desse modo, o framework carrega o arquivo XML e configura sua estrutura de acordo com os parâmetros definidos neste arquivo.

A.1.4 Exemplo:

```

1 <experiment>
2   <max-repeticoes>50</max-repeticoes>
3   <process>
4     <algorithm type="jef.Algoritmo.Cultural.AlgCultural_Ilha" >
5       <tipo-otimizacao type="Max"/>
6       <conhecimento-cultural type="Situacional"/>
7       <population-size>800</population-size>
8       <max-of-generations>500</max-of-generations>
9       <valor-otimo> 8722 </valor-otimo>
10      <!--comentario tipo= "esquema-cromossomo=1***01***1"-->
11      <species type="jef.TipoBinario.IndividuoEspecieBinaria" tamanho-cromossomo="60" />
12      <rand-gen-factory type="jef.Util.Aleatorio.RanMersenneTwistertFactory" noseed="1234567890"/>
13      <provider type="jef.TipoBinario.CriarIndividuoBinario"/>
14      <parents-selector type="jef.Seletor.SeletorTorneio" tam-torneio="3"/>
15      <recombinator type="jef.TipoBinario.Cruzador.CruzadorUniformeCultural" prob-cruz="0.9" prob-cruz-ponto="0.9"/>
16      <mutator type="jef.TipoBinario.MutadorUniformeCultural" prob-mut="0.1" prob-mut-ponto="0.1" number-of-mutation-points="19" />
17      <evaluator type="jef.TipoBinario.AvalMochilaSENTO_2ACMI"/>
18    </algorithm>
19    <listeners>
20      <listener type="jef.Relatorio.RelatorioBasePopulacao">
21        <report-cancela-relatorio-xml> true </report-cancela-relatorio-xml>
22        <report-cancela-grafico>true</report-cancela-grafico>
23        <report-dir-name>relatorio1</report-dir-name>
24        <report-frequency>10</report-frequency>
25        <include-individuals>false</include-individuals>
26      </listener>
27    </listeners>
28  </process>
29 </experiment>

```

São 50 execuções (<max-repeticoes>) utilizando o algoritmo cultural baseado no modelo de ilha (<algorithm type>) e utilizando o conhecimento situacional (<conhecimento-cultural type>). Com uma população de tamanho=800 (<population-size>) com 500 gerações (<max-of-generations>) e o valor ótimo=8722 para o problema é definido em <valor-otimo>. O problema aqui é definido do tipo binário (<species type>) e tamanho do cromossomo=60 (<tamanho-cromossomo>) que representa a dimensão do problema. O item (<rand-gen-factory type>) indica o tipo de algoritmo utilizado para valores aleatórios (utiliza seeds). O tipo de seleção é torneio (<parents-selector type>) com tamanho igual a 3 (<tam-torneio>). O tipo de

cruzamento e mutação são descritos por (<recombinator type>) e (<mutator type>) respectivamente. As probabilidades de cruzamento e mutação são (<prob-cruz>) e (<prob-mut>) respectivamente. A função aqui é definida por (<evaluator type>).

ARTIGOS PUBLICADOS

1. SILVA JUNIOR, J. A. L.; SILVA, D. J. A.; OLIVEIRA, R. C. L.; **PARAMETRIZAÇÃO DE UM ALGORITMO CULTURAL HÍBRIDO PARA O PROBLEMA DA MOCHILA MULTIDIMENSIONAL**. XII Simpósio Brasileiro de Automação Inteligente – SBAI 2015. Natal – RN, Brasil.
2. SILVA, D. J. A.; SILVA, J. A. L.; AFFONSO, C. M.; OLIVEIRA, R. C. L.; **USO DE ALGORITMO CULTURAL COM UMA NOVA ABORDAGEM MEMÉTICA POR MEIO DO SIMULATED ANNEALING PARA O PROBLEMA DO CAIXEIRO VIAJANTE**. XI Simpósio Brasileiro de Automação Inteligente – SBAI 2013. Fortaleza – CE, Brasil.