



UNIVERSIDADE FEDERAL DO MARANHÃO  
CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA – CCET  
CURSO DE PÓS-GRADUAÇÃO EM ENGENHARIA DE ELETRICIDADE

MARK RENATO CAMPOS LIMA

ALGORITMOS GENÉTICOS NA FORMAÇÃO DE GRUPOS PARA  
APRENDIZAGEM COOPERATIVA APOIADA POR COMPUTADOR

São Luís  
2006

MARK RENATO CAMPOS LIMA

## ALGORITMOS GENÉTICOS NA FORMAÇÃO DE GRUPOS PARA APRENDIZAGEM COOPERATIVA APOIADA POR COMPUTADOR

Dissertação apresentada ao curso de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica na área de Ciência da Computação.

Orientador: Prof. Dr. Sofiane Labidi

São Luís  
2006

MARK RENATO CAMPOS LIMA

# ALGORITMOS GENÉTICOS NA FORMAÇÃO DE GRUPOS PARA APRENDIZAGEM COOPERATIVA APOIADA POR COMPUTADOR

Dissertação apresentada ao curso de Pós-Graduação em Engenharia de Eletricidade da Universidade Federal do Maranhão como parte dos requisitos para a obtenção do título de Mestre em Engenharia Elétrica na área de Ciência da Computação.

Aprovada em: \_\_\_ / \_\_\_ / \_\_\_\_\_

BANCA EXAMINADORA

---

Prof. Dr. Sofiane Labidi  
Universidade Federal do Maranhão  
(Orientador)

---

Prof. Dr. Hugo Fuks  
Pontifícia Universidade Católica - RJ

---

Prof. Dr. Zair Abdelouahab  
Universidade Federal do Maranhão

*A minha família.*

## AGRADECIMENTOS

Agradeço primeiramente a Deus por sempre ter iluminado meus caminhos tornando-os, cada vez mais, menos tortuosos.

Aos meus pais e a minha família, por terem me proporcionado alcançar mais essa conquista.

Ao professor Labidi, pela valorosa atenção que a mim concedeu durante boa parte de minha formação acadêmica.

Aos meus amigos, por sempre terem me estimulado e apoiado, contribuindo, direta ou indiretamente, na realização deste trabalho.

A todos que me acompanharam durante o curso, professores e alunos, sem os quais não se teria chegado ao desenvolvimento deste trabalho.

## RESUMO

Este trabalho apresenta uma abordagem baseada em algoritmos genéticos para a resolução do problema da formação de grupos para a aprendizagem cooperativa apoiada por computador. Nesta abordagem o problema é definido como uma tarefa de otimização na qual se pretende otimizar a qualidade dos grupos formados para a aplicação do modelo cooperativo de ensino-aprendizagem utilizado em ambientes computadorizados. Esta qualidade está sujeita a fatores pedagógicos que podem ser estabelecidos pelo modelo pedagógico do ambiente de aprendizagem e são levadas em questão pelo método de formação proposto. Avalia-se a utilização da abordagem proposta aplicando-a no ambiente de ensino-aprendizagem *NetClass*, desenvolvido pela Universidade Federal do Maranhão.

Palavras-chave: Algoritmos Genéticos, Formação de Grupos, Aprendizagem Cooperativa.

## ABSTRACT

This work presents a boarding based on genetic algorithms for the resolution of the groups formation problem for the computer supported cooperative learning. In this boarding the problem is defined as a optimization task in which it intends to optimize the quality of the groups formed for the application of the cooperative model of teach-learning used in computerized environments. This quality is subjects the pedagogical factors that can be established by the pedagogical model of the learning environment and is taken in question for the considered method of formation. It is evaluated use of the boarding proposal applying it in the environment of *NetClass* teach-learning, developed for the Federal University of the Maranhão.

Keywords: Genetic Algorithms, Groups Formation, Cooperative Learning.

## SUMÁRIO

1	INTRODUÇÃO .....	12
1.1	Contexto.....	13
1.2	Objetivos.....	14
1.3	Justificativa.....	14
1.4	Estrutura da Dissertação .....	15
2	APRENDIZAGEM COOPERATIVA APOIADA POR COMPUTADOR .....	17
2.1	Aprendizagem Cooperativa .....	17
2.2	Aprendizagem Cooperativa Apoiada por Computador.....	21
2.3	Formação de Grupos em Ambientes de Aprendizagem Cooperativa .....	23
2.3.1	Formação Oportunista de Grupos.....	23
2.3.2	AulaNet .....	25
2.3.3	TelEduc.....	25
2.3.4	Formação de Grupos Baseada no Perfil do Aluno e no Contexto .....	26
2.3.5	Formação Otimizada de Grupos.....	27
3	ALGORITMOS GENÉTICOS .....	29
3.1	Busca e Otimização .....	30
3.2	Um Breve Histórico Sobre Algoritmos Genéticos .....	30
3.3	Características Principais dos Algoritmos Genéticos .....	32
3.4	Definições Básicas.....	35
3.4.1	Indivíduo.....	35
3.4.2	Codificação .....	36
3.4.3	Função de Avaliação.....	36
3.5	Parâmetros Genéticos .....	37
3.6	Operadores Genéticos .....	38
3.6.1	Operador de Seleção.....	38
3.6.1.1	Seleção Proporcional .....	39
3.6.1.2	Seleção por Torneio.....	39
3.6.1.3	Seleção por Truncamento .....	40
3.6.1.4	Seleção por Ordenação.....	40
3.6.1.5	Elitismo .....	41
3.6.2	Operador de Cruzamento .....	41
3.6.2.1	Operador <i>Order Crossover</i> #1 (OX) .....	43



3.6.2.2	Operador <i>Order Crossover</i> #2.....	44
3.6.2.3	Operador <i>Partially Mapped Crossover</i> .....	45
3.6.2.4	Operador <i>Cycle Crossover</i> .....	46
3.6.2.5	Operador <i>Position Based Crossover</i> .....	47
3.6.3	Operador de Mutação .....	48
3.7	Critério de Convergência .....	50
3.8	Estratégias de Reprodução .....	51
3.9	Conclusão.....	51
4	FORMAÇÃO OTIMIZADA DE GRUPOS UTILIZANDO AG.....	53
4.1	Estratégias de Implementação .....	53
4.1.1	Codificação dos Indivíduos .....	53
4.1.2	População Inicial .....	55
4.1.3	Tamanho da População.....	55
4.1.4	Função de Avaliação.....	56
4.1.5	Operador de Seleção.....	57
4.1.6	Operador de Cruzamento .....	57
4.1.7	Operador de Mutação .....	57
4.1.8	Critério de Parada.....	58
4.1.9	Estratégia de Reprodução .....	58
4.2	Modelagem e Implementação .....	59
4.2.1	Diagrama de Classes.....	60
4.2.1.1	Classe <i>Algorithm</i> .....	61
4.2.1.2	Classe <i>GroupProblem</i> .....	62
4.2.1.3	Classe <i>Population</i> .....	63
4.2.1.4	Classe <i>Individual</i> .....	64
4.2.1.5	Classe <i>Chromosome</i> .....	65
4.2.1.6	Classe <i>Learner</i> .....	65
4.2.2	Diagrama de Interação.....	65
4.3	Conclusão.....	68
5	ESTUDO DE CASO: AMBIENTE NETCLASS .....	69
5.1	NetClass .....	69
5.2	Modelo Pedagógico do NetClass .....	71
5.3	Formação Otimizada de Grupos no NetClass .....	72

5.3.1	Perfil Aceitável de um Grupo .....	73
5.3.2	Coesão Sociométrica.....	75
5.3.2.1	Benefícios e Limitações .....	77
5.3.2.2	Teste Sociométrico.....	79
5.3.2.3	Função de Coesão Sociométrica.....	80
5.3.3	A Função de Avaliação.....	81
5.4	Ferramenta para Formação de Grupos no <i>NetClass</i> .....	82
5.4.1	Classe <i>GroupProblemNetClass</i> .....	82
5.4.2	Classe <i>LearnerNetClass</i> .....	83
5.4.3	Classe <i>Choice</i> .....	84
5.4.4	Aplicação .....	84
5.5	Resultados.....	87
6	CONCLUSÃO .....	93
	REFERÊNCIAS.....	95

## LISTA DE FIGURAS

Figura 2.1 Curva de Desempenho de Grupos .....	20
Figura 2.2 Formação Oportunista de Grupos .....	24
Figura 3.1 Fluxograma básico de um AG .....	33
Figura 3.2 Exemplo de um cromossomo .....	35
Figura 3.3 Exemplo de cruzamento de um ponto .....	42
Figura 3.4 Exemplo de cruzamento multiponto .....	43
Figura 3.5 Exemplo de <i>Order Crossover</i> #1 .....	44
Figura 3.6 Exemplo de <i>Order Crossover</i> #2 .....	45
Figura 3.7 Exemplo de <i>Partially Mapped Crossover</i> .....	46
Figura 3.8 Exemplo de <i>Cycle Crossover</i> .....	47
Figura 3.9 Exemplo de <i>Position Based Crossover</i> .....	48
Figura 3.10 Exemplo de Mutação .....	49
Figura 3.11 Operador de Mutação <i>Swap</i> .....	49
Figura 3.12 Operador de Mutação <i>Shift</i> .....	50
Figura 4.1 Codificação do cromossomo para $n = 3$ .....	54
Figura 4.2 Indivíduos Gêmeos .....	54
Figura 4.3 Diagrama de Classes .....	60
Figura 4.4 Diagrama de Sequência .....	66
Figura 4.5 Método <i>formGroup</i> .....	67
Figura 4.6 Método <i>goOneStep</i> .....	68
Figura 5.1 Ambiente NetClass .....	70
Figura 5.2 Modelo Pedagógico do NetClass .....	71
Figura 5.3 Classificação dos Grupos quanto ao Perfil .....	74
Figura 5.4 Teste sociométrico do <i>NetClass</i> .....	79
Figura 5.5 Exemplo de sociograma .....	80
Figura 5.6 Ferramenta de Formação de Grupos – Escolha da Abordagem .....	85
Figura 5.7 Ferramenta de Formação de Grupos – Definição dos grupos ideal e médio .....	86
Figura 5.8 Aplicação Teste .....	88
Figura 5.9 Resultado para execução baseada no perfil dos alunos .....	90
Figura 5.10 Resultado para execução baseada no teste sociométrico .....	91
Figura 5.11 Resultado da execução utilizando as duas abordagens .....	92

## LISTA DE TABELAS

Tabela 4.1 Relação Número de Alunos x Tamanho da População .....	56
Tabela 4.2 Função de avaliação.....	56
Tabela 4.3 Classe <i>Algorithm</i> .....	62
Tabela 4.4 Classe <i>GroupProblem</i> .....	63
Tabela 4.5 Classe <i>Population</i> .....	63
Tabela 4.6 Classe <i>Individual</i> .....	64
Tabela 4.7 Classe <i>Chromosome</i> .....	65
Tabela 5.1 Benefícios e limitações da sociometria.....	79
Tabela 5.2 Classe <i>GroupProblemNetClass</i> .....	83
Tabela 5.3 Classe <i>LearnerNetClass</i> .....	83
Tabela 5.4 Classe <i>Choice</i> .....	84
Tabela 5.5 Turma de alunos para teste.....	89
Tabela 5.6 Definição dos grupos ideal e médio .....	89
Tabela 5.7 <i>Fitness</i> das soluções por abordagem .....	90

## LISTA DE EQUAÇÕES

Equação 3.1 Probabilidade de seleção pelo método proporcional .....	39
Equação 3.2 Probabilidade de seleção pelo método de ordenação linear .....	40
Equação 3.3 Probabilidade de seleção pelo método de ordenação exponencial .....	40
Equação 4.1 Número de possíveis soluções .....	55

## LISTA DE SIGLAS

ACAC	–	Aprendizagem Cooperativa Apoiada por Computador
AG	–	Algoritmos Genéticos
CSCL	–	Computer Supported Collaborative Learning
SIAC	–	Sistema de Instrução Assistida por Computador
STI	–	Sistema Tutor Inteligente
UML	–	Unified Modeling Language

# 1 INTRODUÇÃO

---

---

Desde o advento dos computadores e paralelamente à sua evolução, diversas áreas de pesquisa - dentre elas a Ciência da Computação, a Pedagogia, a Psicologia - têm se esforçado no sentido de desenvolver eficientes sistemas computacionais de ensino-aprendizagem. Todo esse esforço vem, gradativamente, trazendo avanços significativos na aplicação da informática na educação.

As primeiras experimentações nessa área surgiram na década de 60 quando apareceram os Sistemas de Instrução Assistida por Computador (SIAC) considerados os primeiros sistemas de ensino por computador (STEINBERG, 1984). Com uma concepção bem simples, nesses sistemas o computador é somente uma interface entre o aprendiz e o conteúdo a ser estudado, ou seja, o computador disponibiliza o conteúdo aos aprendizes e avalia as suas respostas aos testes. Não há uma preocupação com a forma de apresentação do conteúdo.

Na década seguinte, favorecidos pela evolução da Inteligência Artificial, surgiram os primeiros sistemas capazes de adaptarem-se as necessidades individuais de cada aprendiz. Com isso os Sistemas Tutores Inteligentes – STI, como são chamados tais sistemas, tornaram o processo de ensino-aprendizagem mais eficiente.

Uma das deficiências dos sistemas educacionais vigentes até então era que o processo de ensino-aprendizagem era executado individualmente, ou seja, não havia interação entre aprendizes, apenas uma relação aprendiz-máquina, o que enfraquecia o aprendizado. Para suprir essa deficiência, com o suporte do avanço da tecnologia de redes de computadores, surgiram os ambientes de Aprendizagem Colaborativa Apoiada por Computador (ACAC) – do inglês *Computer Supported Collaborative Learning* (CSCL).

Estes ambientes permitem aos estudantes estarem geograficamente separados, porém juntos nas atividades de aprendizado, encontrando-se e discutindo a qualquer hora, dispensando a necessidade de estarem fisicamente presentes na mesma localidade, encorajando a participação dos envolvidos no processo educacional independentemente de tempo e espaço. A ubiquidade da aprendizagem cooperativa e seu potencial para apoiar aprendizagem em grupo interativamente têm convencido muitos educadores que estes ambientes são o futuro das próximas gerações de ferramentas educativas para ensino a distância (KREIJNS *et al.*, 2002).

Com o suporte à cooperação entre os aprendizes, esses ambientes trouxeram inúmeras vantagens em relação aos sistemas tradicionais, mas também trouxeram consigo uma gama de problemas, tanto sob aspectos pedagógicos quanto sob aspectos computacionais, não existentes nos seus predecessores. Podemos destacar, dentre esses problemas, a representação dos aprendizes e seus grupos, a interação entre os aprendizes e professores e a distribuição dos aprendizes em grupos. Como conseguir, de maneira eficaz, implementar essas tarefas básicas e essenciais para o bom funcionamento desses ambientes? Nesse trabalho desenvolve-se uma solução para o problema da distribuição eficaz dos aprendizes em grupos.

## 1.1 Contexto

Este trabalho, inserido no projeto *NetClass*, envolve principalmente duas áreas de conhecimento: a computação evolutiva e os ambientes ACAC. Para desenvolver este trabalho escolheu-se como foco principal o estudo dos algoritmos genéticos e, como aplicação deste, o desenvolvimento de um método inteligente para formação de grupos em ACAC.

O projeto *NetClass* tem por objetivo utilizar conceitos da Inteligência Artificial na Educação para desenvolver um ambiente ACAC, baseado na *web*, onde o processo de ensino-



aprendizagem se dá através da iteração entre grupos de alunos, professores e componentes computacionais, todos conectados através da Internet. Este trabalho então está integrado ao *NetClass* e pretende neste validar a eficiência do método proposto.

## 1.2 Objetivos

O objetivo deste trabalho é especificar um método inteligente para a formação de grupos cooperativos em ambientes ACAC que inicialmente será utilizado no ambiente *NetClass*.

Especificamente, pretende-se:

1. Modelar o problema como uma tarefa de otimização;
2. Projetar um algoritmo genético capaz de otimizar a solução do problema.
3. Implementar e integrar no *NetClass* uma ferramenta que utiliza o algoritmo projetado.

## 1.3 Justificativa

A cooperação entre aprendizes em ambientes computacionais de ensino-aprendizagem é um dos pontos chave para o sucesso da aplicação desses sistemas, logo, muitos sistemas dessa natureza buscam promover a cooperação para um melhor aprendizado por parte dos aprendizes. Para tanto, uma das formas adotadas é a divisão das classes em pequenos grupos, onde os alunos de cada grupo interagem diretamente uns com os outros no processo de ensino-aprendizagem, pois é através da troca de informações, perguntas e comentários que este processo é complementado e, conseqüentemente, enriquecido (WOLZ *et al.*,1997).

A distribuição dos alunos em grupos pode afetar diretamente nos seus desempenhos. Essa relação, distribuição *versus* desempenho, pode se dar, por exemplo, em função do perfil de cada aluno e/ou dos laços de afinidade existentes entre eles. Sendo assim, a tarefa de dividir a classe em grupos em qualquer sistema de ensino-aprendizagem cooperativa, torna-se de extrema importância.

Do ponto de vista tecnológico, a cooperação entre aprendizes em um ambiente computacional de ensino-aprendizagem depende de duas tarefas: a definição dos grupos e o estabelecimento de interação entre os grupos e o ambiente, o qual se compõe de outros grupos, dos professores e dos componentes computacionais. Portanto, um ambiente de cooperação deve fornecer ferramentas para executar ambas as tarefas.

Este trabalho propõe um método eficiente para executar a tarefa de definição dos grupos. Baseando-se em fatores pedagógicos, esta tarefa foi definida como um problema de otimização ao qual se aplicou um algoritmo genético para solucioná-lo.

#### 1.4 Estrutura da Dissertação

O presente trabalho é composto por seis capítulos. No próximo capítulo faz-se uma introdução aos Ambientes de Ensino-Aprendizagem Cooperativa Apoiado por Computador onde, além de suas características principais, mostra-se a problemática da formação de grupos para a aprendizagem cooperativa onde se destaca a proposta deste trabalho para a resolução do problema da formação de grupos.

Na seqüência faz-se uma introdução aos Algoritmos Genéticos, a terminologia adotada por esta técnica, sua simulação do processo de evolução, elementos e operadores genéticos utilizados e suas principais aplicações.

Considerando o apresentado nos capítulos 2 e 3, o capítulo 4 apresenta a modelagem do algoritmo genético para otimizar a formação de grupos. Nesse capítulo mostra-se detalhadamente cada um dos elementos que compõem o algoritmo bem como diagramas UML que facilitam o entendimento do seu funcionamento e implementação.

No capítulo 5 são apresentados resumidamente e detalhadamente, respectivamente, o ambiente *NetClass* e a ferramenta desenvolvida para a formação de grupos baseada no algoritmo genético modelado. São mostrados ainda os resultados alcançados.

Finalmente, no capítulo 6, apresentam-se as conclusões, discussões e recomendações para trabalhos futuros.

## 2 APRENDIZAGEM COOPERATIVA APOIADA POR COMPUTADOR

---

---

Neste capítulo descrevem-se as características essenciais da Aprendizagem Cooperativa Apoiada por Computador enfatizando-se a importância do tratamento dos grupos para sua efetiva aplicação. Destaca-se ainda a problemática da formação de grupos para aprendizagem cooperativa mostrando ao final a abordagem proposta neste trabalho.

### 2.1 Aprendizagem Cooperativa

A cooperação é um fator bastante desejável, e muitas vezes essencial, quando um grupo visa alcançar um objetivo comum. Porém, nas salas de aula tradicionais, onde se pode tomar como objetivo comum o aprendizado, a cooperação ainda não é amplamente utilizada. O que se vê frequentemente é um comportamento de competição entre os aprendizes onde o sucesso de um não implica no sucesso dos outros.

Já na aprendizagem cooperativa o sucesso dos aprendizes está correlacionado ao sucesso do grupo onde estes cooperam, ajudando-se a terem um maior aproveitamento no processo de ensino-aprendizagem. Os aprendizes trabalham juntos para alcançar um objetivo comum, através da interdependência existente entre eles, ou seja, cada membro é responsável pela realização deste objetivo (ELLIS et al., 1990). Para que um grupo seja considerado cooperativo devem-se considerar os seguintes aspectos (JOHNSON, et al., 1994):

Interdependência positiva: é o que diferencia um grupo de aprendizagem cooperativa de um grupo esporadicamente conectado. Este aspecto é intencionalmente planejado de modo que todos os membros devem participar para que a tarefa seja completada. É a “cola” que gruda todos os membros bem juntos. Há vários tipos de Interdependência Positiva:

*Interdependência Positiva do Alvo:* Os alunos percebem que podem alcançar seus alvos de aprendizagem se, e somente se, todos os membros de seu grupo podem também alcançar os seus próprios alvos. Membros de um grupo de aprendizagem têm um conjunto de alvos que todos estão se esforçando por alcançar.

*Interdependência Positiva de Recursos:* Cada membro possui só uma parte das informações, dos recursos, dos materiais necessários para a tarefa ser completada, e os recursos dos membros devem ser combinados para que o grupo atinja seu alvo.

*Interdependência Positiva de Papéis:* A cada membro se designam papéis complementares e inter-relacionados que especificam as responsabilidades necessárias do grupo para que ele complete uma tarefa conjunta.

*Interdependência Positiva de Identidade:* O grupo estabelece uma identidade mútua através de um nome, de uma bandeira, um mote, ou canção.

Responsabilidade Individual e em Grupo: a responsabilidade individual é a chave para assegurar que cada componente do grupo receba e promova um reforço cooperativo em seu aprendizado (JONHSON et al., 1994), ou seja, o indivíduo será reconhecido pela sua contribuição dada ao sucesso do grupo.

Interação Direta: é essencialmente necessário que haja um auxílio efetivo e eficiente entre os aprendizes. Para isso deve haver entre os membros dos grupos troca de recursos, assistência e cumplicidade mútua.

Processamento de Grupo: é necessário saber se as ações dos membros dos grupos foram ou não bem sucedidas a fim de decidir que ações deverão persistir, quais devem ser aprimoradas e ainda quais devem ser descartadas. O objetivo é verificar a participação ativa dos membros do grupo. Através do processamento de grupo é possível (FERREIRA, 1998):

§ Avaliar a qualidade da interação entre os membros do grupo;

- § Examinar o processo pelo qual o grupo trabalha;
- § Determinar os objetivos e prover efetividade no grupo;
- § Verificar a real situação do grupo no que se refere ao aprendizado.

Habilidade Social: são evidenciadas todas as formas de interação entre os membros dos grupos para a realização de determinada atividade. Em (ELLIS et al., 1990), os autores sugerem algumas etapas para o desenvolvimento dessas habilidades, são elas:

- § Definir a habilidade em função do nível do aprendiz.
- § Ajudar o aprendiz a reconhecer a importância da habilidade ensinada;
- § Solicitar aos aprendizes uma descrição das habilidades;
- § Praticá-las;
- § Discutir e reconhecer o esforço do aprendiz;
- § Repetir a prática quantas vezes forem necessárias.

Considerados esses aspectos percebe-se que os grupos cooperativos extrapolam o conceito de grupo como um simples agrupamento de aprendizes. Não basta apenas reunir equipes para se estabelecer um processo de ensino-aprendizagem cooperativo. Algumas pesquisas buscaram verificar o desempenho entre os grupos tradicionais e os grupos cooperativos. Pode-se ver os resultados dessas pesquisas no gráfico da figura 2.1, que mostra a diferença das performances de cada um dos tipos de grupos (JONHSON et al., 1994).

O gráfico mostra o desempenho de quatro tipos de grupos de aprendizagem: o pseudo-grupo, o grupo de sala de aula tradicional, o grupo de aprendizagem cooperativa e o grupo de aprendizagem cooperativa de alto desempenho. Os membros individuais mostram o desempenho relativo desses aprendizes em relação ao pseudo-grupo.

- § Pseudogrupo: Caracteriza um grupo sem o compromisso de compartilhar tarefas, trocar conhecimentos e experiências.

- § Grupo de Sala de Aula Tradicional: Caracteriza um grupo que aceita trabalhar junto, porém a interdependência dos membros é baixa.
- § Grupo de Aprendizagem Cooperativa: Caracteriza um grupo que trabalha junto compartilhando informações. Nesse grupo é dada ênfase tanto à responsabilidade do grupo quanto individual (JONHSON et al., 1994).
- § Grupo de Aprendizagem Cooperativa de Alto Desempenho: Caracteriza um grupo que atenda a todos os critérios de um grupo de aprendizagem cooperativa e adicionalmente possuem um alto grau de interdependência entre seus membros.

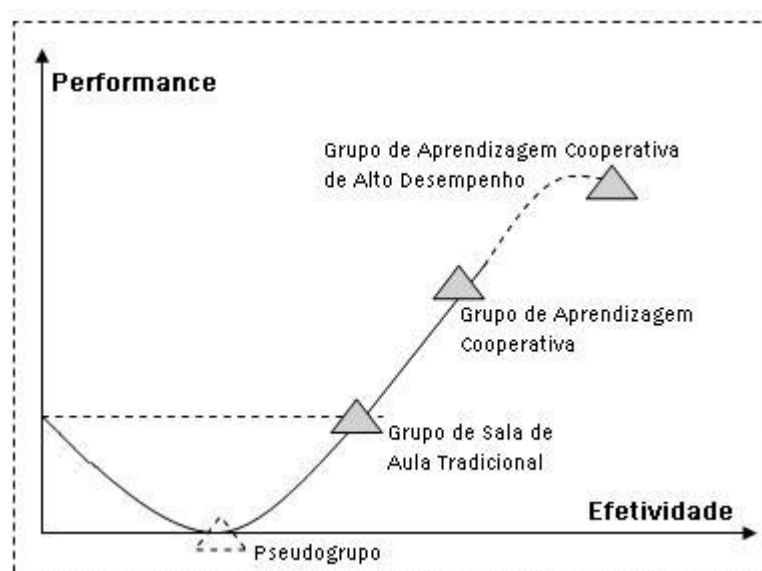


Figura 2.1 Curva de Desempenho de Grupos

Vistos estes conceitos pode-se apresentar a definição de *Green* para aprendizagem cooperativa citada em (FERREIRA, 1998) que segundo ele é a organização da classe em pequenos grupos que trabalham construindo o seu próprio conhecimento, sob a orientação do professor e desenvolvendo várias habilidades de interação, aprendizado e cooperação de acordo com instruções específicas. O sucesso da aprendizagem cooperativa se baseia em alguns elementos principais: a interdependência positiva, a responsabilidade individual, o

*desenvolvimento das habilidades e valores, a participação igualitária, a dinâmica do grupo e a interação entre seus membros.*

Vale ressaltar que esta tendência independe do uso de novas tecnologias, exigindo basicamente uma postura pedagógica inovadora e sem preconceitos na qual a educação deve suportar uma aprendizagem baseada na cooperação, colaboração e descobertas (JOHNSON et al., 1994). Várias estratégias pedagógicas para a aprendizagem cooperativa podem ser vistas em (PASCAL et al., 1996), (ESMA et al., 1995), (LABIDI, et al., 1998).

## 2.2 Aprendizagem Cooperativa Apoiada por Computador

Com o intuito de pesquisar e desenvolver sistemas de apoio à aprendizagem cooperativa, levando em consideração as particularidades desta proposta pedagógica, surgiu uma área de estudo denominada *Computer Supported Collaborative Learning* (aprendizagem cooperativa apoiada por computador). É importante salientar aqui a discussão existente na utilização dos termos “colaboração” e “cooperação”. Estes termos são às vezes usados como sinônimos, enquanto alguns pesquisadores usam-nos distintamente de acordo com o grau de divisão do trabalho. Na cooperação, os membros de um grupo dividem o trabalho, resolvem sub-tarefas separadamente e então unem os resultados individuais em um resultado final, contendo os resultados de todos. Na colaboração, o trabalho é realizado de forma conjunta. Apesar de que uma divisão espontânea pode ocorrer quando se trabalha em grupo: por exemplo, algum membro pode responsabilizar-se pelos aspectos operacionais da tarefa enquanto outro pode focalizar seus esforços nos aspectos estratégicos (DILLENBOURG, 1999). Esclarece-se então que o termo “cooperação” empregado neste trabalho abrange os conceitos de colaboração e cooperação colocados por DILLENBOURG.



O principal benefício na utilização desses sistemas, alvos de estudos da CSCL, é a possibilidade de cooperação sem restrições de tempo e espaço (MACEDO, *et al.* 1999). A eliminação dessas restrições, possibilitada pelo avanço tecnológico, permite a escola expandida: aquela que amplia os espaços de convivência e aprendizagem, que quebra as paredes da escola em direção à comunidade, ao mesmo tempo em que sinaliza a importância da superação das barreiras existentes entre escola e comunidade, entre aluno e professor, entre escola e escola (MORAES, 1995).

De fato, quando o grupo de alunos se localiza geograficamente distribuído, a construção e a implementação de um ambiente de aprendizagem cooperativa traz inúmeras vantagens como a disponibilização imediata do conteúdo do curso, a integração e a cooperação entre os integrantes, o aumento da motivação dos participantes e a diminuição das barreiras culturais ao ensino a distância. Para isso, os ambientes de aprendizagem devem abranger características que visem: otimizar o trabalho de seus participantes; disponibilizar um ambiente agradável de interação; e alcançar a conquista e o compartilhamento do conhecimento.

Além de prover essas características visando alcançar bons resultados é também de fundamental importância para ambientes CSCL fornecerem métodos eficientes que distribuam os aprendizes em grupos de aprendizagem cooperativa de alto desempenho, já que, como apresentado no início do capítulo, a composição dos grupos é fator primordial na aplicação da aprendizagem cooperativa. A seguir apresenta-se a questão da formação de grupos e como ela é resolvida em alguns ambientes de aprendizagem cooperativa. Na sequência apresenta-se o método para formação de grupos proposto neste trabalho.

## 2.3 Formação de Grupos em Ambientes de Aprendizagem Cooperativa

É grande, e cada vez maior, o interesse de pesquisadores no campo da aprendizagem cooperativa apoiada por computador. Em contrapartida, existem poucos trabalhos que se preocupam com a formação de grupos. A distribuição dos grupos é de crucial importância para o processo de ensino-aprendizagem cooperativa e é ainda mais crítica em ambientes virtuais onde os membros podem nem mesmo se conhecer, cooperando à distância. A formação de grupos nesses ambientes é dificultada principalmente por dois aspectos: (1) os integrantes do ambiente em questão, na maioria das vezes, não se conhecem pessoalmente e (2) os integrantes encontram-se em locais geograficamente dispersos. Estes aspectos contribuem para que a formação dos grupos cooperativos seja realizada, geralmente, de forma aleatória, não sendo considerado nenhum critério específico.

A divisão em pequenos grupos pode vir a constituir um excelente recurso para promover e incrementar a aprendizagem e as interações sociais. Segundo Piaget, citado por MINICUCCI (1997), a formação intelectual é privilegiada quando existe a livre cooperação entre os participantes de um grupo. Porém, a eficiência e o sucesso de um grupo estão relacionados a uma série de variáveis durante seu ciclo de vida. Para tornar os grupos efetivos deve-se ter o cuidado de se observar os seguintes aspectos: como caracterizar a organização de um grupo, quais são as funções do grupo, em que situações a formação de grupos é apropriada. A seguir destacam-se algumas abordagens para a formação de grupos efetivos e na sequência apresenta-se a abordagem desenvolvida nesse trabalho.

### 2.3.1 Formação Oportunista de Grupos

A Formação Oportunista de Grupos (Opportunistic Group Formation – OGF), proposta por (IKEDA et al., 1997), é uma função para formar, dinamicamente, grupos de

aprendizagem colaborativa. Quando detectada uma situação favorável à mudança de um aprendiz do modo de aprendizagem individual para a aprendizagem colaborativa, é formado um grupo onde para cada membro é determinado um objetivo e um papel social consistentes com o objetivo do grupo como um todo.

O funcionamento básico da OGF consiste em monitorar os aprendizes em um ambiente de aprendizado individual através de agentes inteligentes e, a partir daí, identificar objetivos individuais de cada um e estabelecer um momento oportuno para a formação dos grupos. Quando o agente detecta uma situação ideal para migrar o aprendiz de um paradigma de aprendizagem individual para o colaborativo ele determina um papel desejável para o aprendiz em um grupo e informa a todos os outros agentes monitorando requisitando uma formação de grupo heterogênea. Os papéis designados pelos agentes podem ser: aprendiz, observador, tutor passivo e tutor ativo. Os agentes então iniciam um processo de negociação visando estabelecer a formação ideal do grupo. A figura 2.2 mostra os passos da Formação Oportunista de Grupos.

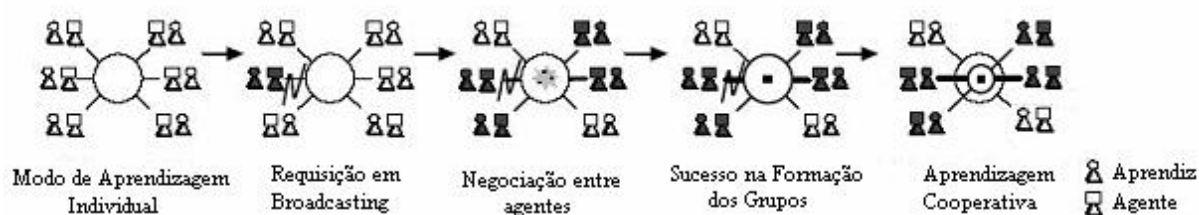


Figura 2.2 Formação Oportunista de Grupos (IKEDA et al., 1997)

Quando a negociação entre os agentes termina em sucesso, cada um dos participantes do grupo é informado do objetivo de aprendizagem do grupo bem como do seu papel. Os papéis determinados a cada um dos alunos não são fixos durante a sessão de aprendizagem colaborativa, podendo mudar de acordo com sua interação com o ambiente. Podem ser propostas várias formações de grupos, cada uma contendo uma justificativa para tal formação.

### 2.3.2 AulaNet

O AulaNet (LUCENA *et al.*, 1999) é um ambiente para ensino e aprendizagem na *Web*, cujo desenvolvimento vem se realizando desde Junho de 1997 no Laboratório de Engenharia de Software do Departamento de Informática da Pontifícia Universidade Católica do Rio de Janeiro (PUC-Rio). O AulaNet se baseia nas relações de trabalho cooperativo que se manifestam nas interações dos aprendizes com seus instrutores, com outros aprendizes e com os conteúdos didáticos.

Para dar suporte ao trabalho em grupo, o AulaNet dispõe de um sistema multiagente que auxilia na formação de grupos (CUNHA *et al.*, 2002).

### 2.3.3 TelEduc

O trabalho desenvolvido por (SILVEIRA, 2005) aborda a formação de grupos no ambiente colaborativo de aprendizagem *TelEduc*. Em sua pesquisa foram utilizadas técnicas de Inteligência Artificial, entre elas os Algoritmos Genéticos e os Sistemas Multiagentes. Os grupos são formados a partir de critérios definidos pelo professor e, além de formar os grupos, são determinados papéis para cada um dos integrantes baseados no estilo cognitivo de cada um, informação mantida pela arquitetura de agentes.

Na arquitetura de agentes do *TelEduc* o agente *forma grupo colaborativo* é o responsável por organizar os grupos, estabelecendo os papéis que os integrantes deverão desempenhar.

Através da utilização de um algoritmo genético o agente implementado define a melhor combinação de alunos para um grupo baseado nos estilos cognitivos e nos critérios definidos pelo professor, que pode escolher se o grupo formado deve ser homogêneo ou

heterogêneo e ainda pode apontar os três papéis preferenciais que o aluno pode assumir, sexo, faixa etária e localização geográfica.

#### 2.3.4 Formação de Grupos Baseada no Perfil do Aluno e no Contexto

Em (MUEHLENBROCK, 2006) além de considerar o perfil dos alunos como parâmetro, a formação de grupos baseia-se também em informações sobre o contexto do aluno tais como localização, atividade e disponibilidade. Utilizando-se uma infraestrutura de sensores de contexto foi desenvolvida uma aplicação que oportunamente forma os grupos baseados nessa informação contextual. Os sensores de contexto podem ser disponibilizados em muitos ambientes, destacam-se:

- Uso de computador: detecção de atividade em teclados e mouses de computadores;
- Uso de telefone: detecção de uso de telefone;
- Localização de PDA;
- Captação de som ambiente com PDA;
- Interface direta entre usuário e PDA;

Estes vários sensores enviam informações para uma base de dados que pode ser acessada tanto por rede convencional ou sem fio. Esta base de dados contém dados estáticos do perfil bem como dados de contexto, além de associar usuários com dispositivos. Então a combinação de informações do perfil com as de contexto possibilita a formação de grupos para a resolução imediata de problemas além de possibilitar resolução de problemas face-a-face.

### 2.3.5 Formação Otimizada de Grupos

Na abordagem aqui proposta para a formação de grupos efetivos de aprendizagem toma-se o problema como sendo uma tarefa de otimização, onde se pretende obter uma distribuição de grupos ótima para uma determinada classe de aprendizes. Essa tarefa de otimização baseia-se em fatores pedagógicos para determinar quão ótima é uma dada distribuição de grupos.

Declarando o problema formalmente tem-se: seja  $C$  uma classe de  $n \times k$  estudantes e  $F_1, F_2, \dots, F_m$  uma lista de  $m$  fatores pedagógicos. Suponha que o professor deseja uma “boa divisão” de  $C$  em  $n$  grupos de  $k$  estudantes sujeitos aos fatores  $F_1, F_2, \dots, F_m$ . Define-se um *grupo-k* em  $C$  como um subconjunto  $\{I_1, I_2, \dots, I_k\}$  de  $C$  com  $k$  elementos. Define-se uma *divisão-k x n* de  $C$  como um subconjunto  $D = \{g_1, g_2, \dots, g_n\}$  com  $n$  elementos, onde cada elemento é um *grupo-k* diferente em  $C$ . Define-se o *fitness* (quão bom é) de uma *divisão-k x n* de  $C$  levando-se em conta os fatores  $F_1, F_2, \dots, F_m$  como uma função definida em termos de  $F_1, F_2, \dots, F_m$  que mapeia  $D$  em um número  $f^3$  que mede o *fitness*. Em símbolos:

$$f^3 = \text{FITNESS}(F_1, F_2, \dots, F_m)(D).$$

O problema da formação de grupos proposta é: *Encontrar uma divisão-k x n D de C que maximize a função  $\text{FITNESS}(F_1, F_2, \dots, F_m)(D)$* . Em símbolos:

$$\max_D \text{FITNESS}(F_1, F_2, \dots, F_m)(D)$$

A abordagem proposta, diferentemente das supra apresentadas e de outras encontradas na literatura pesquisada, não estabelece critérios ou regras pré-definidas para a formação dos grupos. Os critérios são estabelecidos em função da definição dos fatores pedagógicos o que a torna bastante flexível. Nos capítulo 4 e 5 onde apresentam-se, respectivamente, a implementação e uma aplicação do método aqui proposto ficará mais claro

e detalhado o processo de formação de grupos utilizando-se esta abordagem. A seguir são apresentados os conceitos relacionados aos algoritmos genéticos, os quais foram aplicados para desenvolver o método aqui proposto.

### 3 ALGORITMOS GENÉTICOS

---

---

Neste capítulo apresentam-se detalhadamente todas as definições e procedimentos que envolvem a conceituação dos Algoritmos Genéticos (AGs). Serão mostrados com maior profundidade os conceitos que estão diretamente ligados a este trabalho, mas sem deixar de colocar, ainda que em linhas gerais, todos os aspectos que dizem respeito à modelagem e à execução de um AG.

Atualmente os AGs vêm sendo largamente utilizados nas mais diversas áreas de aplicação. Isso decorre de sua potencialidade como uma técnica de otimização de características particulares, combinando procedimentos de busca direcionada e aleatória, de modo a obter o ponto ótimo, ou pontos ótimos, de determinada função. Segundo GOLDBERG (1989), os AGs diferem da maioria dos métodos de busca e otimização por quatro motivos:

- i) trabalham em um espaço de soluções codificadas e não diretamente no espaço de busca;
- ii) trabalham com um conjunto de pontos (população) e não com um único ponto;
- iii) utilizam informações de custo ou recompensa (função de avaliação) e não derivadas ou outro conhecimento auxiliar;
- iv) usam regras de transição probabilísticas ao invés de determinísticas.

Nas seções seguintes ficará clara a potencialidade dos AGs como solução para problemas de otimização e portanto estará justificada a sua aplicação nesse trabalho.



### 3.1 Busca e Otimização

Em toda tarefa que necessite de busca e otimização teremos como principais componentes o espaço de busca, onde são consideradas todas as possibilidades de solução para o problema dado, e a função de avaliação (ou função de custo) a qual avalia cada uma das possibilidades do espaço de busca.

Nos vários métodos de busca e otimização tradicionais existentes o processo inicia-se a partir de uma única solução inicial a qual é iterativamente melhorada até que se chegue à solução ótima. Já as técnicas de computação evolutiva operam sobre um conjunto de soluções (população) em paralelo. Dessa forma a busca é realizada em diferentes áreas do espaço de soluções, onde para cada área é alocado um número apropriado de possibilidades em cada região.

Algoritmos Genéticos, que se enquadram na segunda categoria apresentada acima, são aplicáveis a uma inúmera variedade de problemas de busca de soluções ótimas, ou aproximadamente ótimas, pois, ao contrário dos métodos tradicionais, não impõem muitas particularidades intrínsecas ao problema.

Referimo-nos a “algoritmos genéticos” ou a “um algoritmo genético”, como fazem os pesquisadores da área, e não “ao algoritmo genético” porque AGs são um conjunto de procedimentos com uma série de passos distintos e cada um destes passos pode ser executado de várias formas. A seguir será feito um breve histórico sobre os AGs antes de seguir com a revisão de suas características.

### 3.2 Um Breve Histórico Sobre Algoritmos Genéticos

As idéias, até então revolucionárias, publicadas em 1859 por Charles Darwin no livro *On the Origin of Species by Means of Natural Selection* (em português publicado como

*A Origem das Espécies*) causaram bastante polêmica no mundo científico da época. Segundo Darwin, as espécies evoluem de acordo com o meio em que vivem a fim de adaptarem-se a este.

Com isso, foi confrontada a idéia criacionista de que a Terra, e tudo que nela habita, seja uma criação divina estática e perfeita desde a sua criação, com a idéia evolucionista onde os indivíduos estão em constante competição e evolução para perpetuar sua espécie durante várias gerações. A existência das espécies passa a ser dinâmica; as espécies evoluem, se adaptadas, caso contrário desaparecem.

A Natureza faz o papel de um filtro, onde para cada espécie somente os melhores, os mais aptos, os que melhor se adaptam ao meio conseguem sobreviver o suficiente para chegar em idade de reprodução, perpetuando assim suas características que lhe deram esta condição.

Se esse mecanismo for visto com os olhos da computação percebe-se um claro processo de otimização. Os indivíduos mais adaptados (melhores soluções para um dado problema de otimização) se reproduzem gerando novos indivíduos (novas soluções) que conterão parte do código genético (a informação) de seus antecessores e, portanto, ainda que o novo indivíduo gerado não esteja mais bem adaptado, a média de adaptação da população é melhorada, já que tendem a perpetuarem as características dos mais aptos e extinguem-se as características dos menos aptos. Portanto, ao final de várias gerações, teremos uma população formada por indivíduos bem adaptados ao meio (soluções ótimas do problema).

Com a publicação de *Adaptation in Natural and Artificial Systems* em 1975, John Holland, então pesquisador da Universidade de Michigan, levantou a possibilidade de incorporar a semântica da evolução natural a processos de otimização e dessa forma começaram os estudos sobre algoritmos genéticos. Desde então, estes algoritmos vêm sendo

melhorados e aplicados com sucesso nas mais diversas áreas de aplicação em problemas reais de otimização.

### 3.3 Características Principais dos Algoritmos Genéticos

Os algoritmos genéticos são algoritmos de busca, baseados na teoria da evolução das espécies de Charles Darwin onde os indivíduos de uma determinada população evoluem de acordo com os princípios de seleção natural e sobrevivência dos mais aptos. Estes algoritmos simulam os processos de evolução da natureza, onde cada indivíduo representa uma possível solução para um problema dado (GOLDBERG, 1989).

Holland, representando o processo evolutivo a partir de um modelo de cromossomos, encontrou grande aplicação prática para a determinação de máximos e mínimos de funções matemáticas, o que facilitou a aceitação de seus trabalhos no meio acadêmico (GOLDBERG *et al.*, 2000).

A idéia básica dos AGs consiste, em analogia direta às idéias darwinianas, na sobrevivência dos indivíduos mais aptos (VIANA, 1998). Os indivíduos com maior capacidade de adaptação ao seu meio ambiente terão maiores possibilidades de sobreviverem e reproduzirem-se do que os indivíduos menos adaptados. Depois de muitas gerações, os indivíduos da população adquirem características que lhe conferem uma maior adaptabilidade ao ambiente que indivíduos das gerações anteriores. Diz-se que determinada população evoluiu se é possível mensurar a diferença entre uma população atual e sua antecessora, onde essa diferença é medida em função da aptidão dos indivíduos (KOZA, 1992).

A execução de um AG começa com a geração, aleatória, de possíveis soluções para o problema dado. Estas soluções serão os indivíduos que irão compor a população inicial. No decorrer do processo de evolução, cada indivíduo é avaliado para determinar seu valor de aptidão (*fitness*). Com base nesses, e somente nesses, valores uma população de

indivíduos evolui para uma nova população mediante a aplicação dos operadores genéticos – seleção, *crossover* e mutação. Este processo é repetido até que seja satisfeito um critério de parada, que, por exemplo, pode ser definido em termos de número máximo de gerações (KOZA, 1992). A figura 3.1 mostra um fluxograma que descreve o funcionamento básico dos AGs.

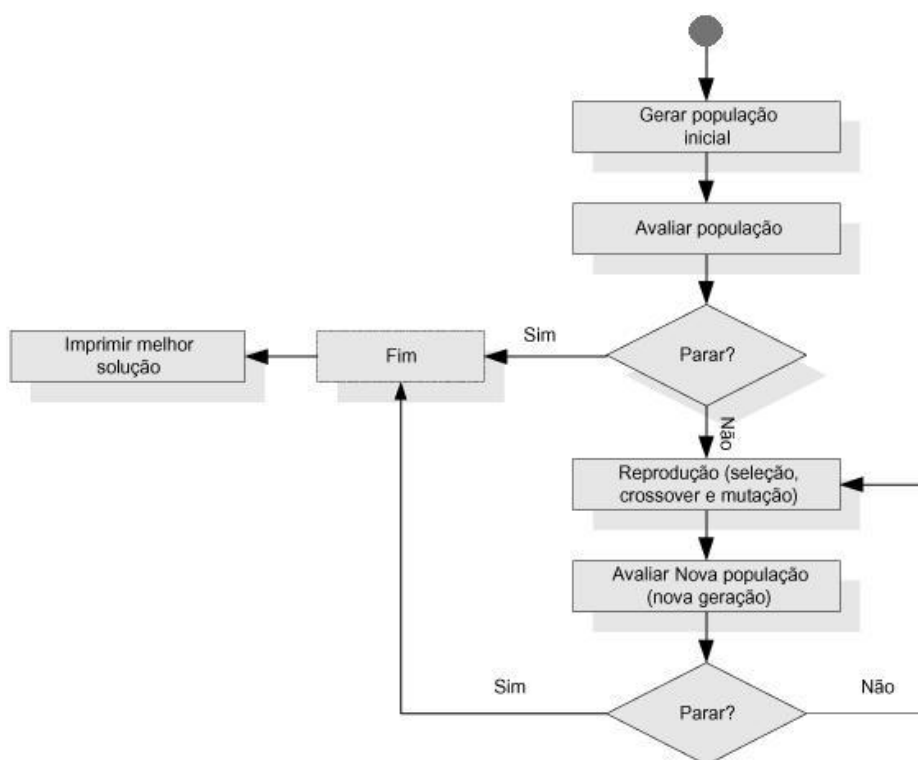


Figura 3.1 Fluxograma básico de um AG

Como já dito anteriormente, os algoritmos genéticos apresentam vantagens sobre os métodos tradicionais de busca e otimização. Aqui, apontam-se algumas características que evidenciam essas vantagens.

- § São algoritmos estocásticos. Duas execuções distintas poderão dar resultados também distintos, ou seja, podem apresentar duas soluções distintas. Isto é bastante útil em problemas onde há uma grande quantidade de soluções

válidas, pois nesses casos é interessante que execuções distintas possam dar resultados distintos.

§ São algoritmos de busca múltipla, assim encontram várias soluções para um mesmo problema. Ainda que, geralmente, os indivíduos da população final sejam similares, são distintos entre si. Assim podemos escolher qual a solução que mais nos convenha segundo a natureza do problema. Veremos mais adiante que essa característica é bem interessante para o nosso problema.

§ Diferentemente dos métodos tradicionais de otimização, cuja convergência e resultado final dependem fortemente da solução inicial, nos algoritmos genéticos a convergência do algoritmo é fracamente influenciada pela população inicial.

§ A otimização se dá em função da representação dos dados. Este é o conceito chave dentro dos algoritmos genéticos, já que uma boa codificação torna a programação bem simples. Além disso, a velocidade da convergência está diretamente relacionada com a representação. Segundo BANZHAF *et al* (1998), o sucesso da execução de um AG depende da codificação adequada do problema.

§ São intrinsecamente paralelos. Independentemente se implementarmos de forma paralela ou não, os algoritmos genéticos buscam em distintos pontos do espaço de soluções de forma paralela. Isto permite que pode-se adequar o código para que execute simultaneamente em vários processadores.

Por tudo isso, os algoritmos genéticos são aplicáveis para a resolução do problema da formação de grupos, produzindo resultados bastante significativos.

### 3.4 Definições Básicas

Desde a sua concepção, é grande o esforço em aperfeiçoar os AG's. Isto gerou uma grande variação de implementações possíveis e aceitáveis para eles. Mas, independentemente da sofisticação de um AG, existem cinco componentes básicos que sempre devem ser incluídos:

- I. A representação do problema em um modelo de cromossomos;
- II. Um método para a geração aleatória da população inicial.
- III. Uma função de avaliação que associa um valor numérico à aptidão de um determinado indivíduo;
- IV. Operadores genéticos, e;
- V. Parametrização genética

A seguir são definidos os conceitos chave para o entendimento de cada um destes itens para, nas seções seguintes, serem detalhados cada um deles.

#### 3.4.1 Indivíduo

O termo *indivíduo* aplica-se a cada membro da população. Em analogia à genética, cada indivíduo possui um material genético que se encontra no cromossomo. O material genético presente no cromossomo é uma representação de uma possível solução para um dado problema (BACK *et al.*, 2000).

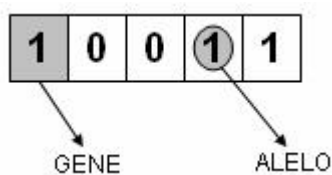


Figura 3.2 Exemplo de um cromossomo

### 3.4.2 Codificação

Um dos pontos chave para uma execução satisfatória de um AG é a codificação adequada do problema (BANZHAF *et al.*, 1998). A codificação dos cromossomos de um AG deve ser feita por meio de uma cadeia finita em algum alfabeto definido (GOLDBERG, 1989). Geralmente esta codificação é feita utilizando-se um alfabeto binário  $\{0,1\}$  (KOZA, 1992). Nesse caso, o espaço de busca apresenta  $2^k$  possibilidades, onde  $k$  é o número de genes do cromossomo. O comprimento do cromossomo é determinado em função do número de variáveis em estudo e da precisão desejada para cada uma delas.

O valor de aptidão de um indivíduo deveria depender apenas do valor dos alelos. Porém, em muitos problemas, o valor de aptidão não depende apenas do valor dos alelos na cadeia de genes do cromossomo. O valor de aptidão pode depender também de alguma combinação entre o valor do alelo e a ordem dos genes. Em outros casos o valor de aptidão pode depender somente da ordem dos genes, a estes dá-se o nome de problemas de permutação (GOLDBERG, 1989). Claramente nestes problemas cada alelo aparece apenas uma vez (DEB, 2000). O espaço de busca para problemas representados por permutação possuem  $K!$  possibilidades, sendo  $K$  o número de genes a serem ordenados (GOLDBERG, 1989).

### 3.4.3 Função de Avaliação

A definição apropriada da função de avaliação é uma das tarefas principais na modelagem dos AGs, já que é através dela que será avaliada a qualidade de cada indivíduo na população (KNOSALA, 2001). A qualidade de cada indivíduo é medida pelo seu valor de

aptidão (*fitness*) que é a propensão de um indivíduo a sobreviver e se reproduzir em um determinado ambiente.

Além da preocupação com uma avaliação correta dos indivíduos é necessária também um cuidado especial com a performance da função de avaliação, pois ela será aplicada para cada indivíduo de cada população e das sucessivas gerações; devido a este fato, grande parte do tempo gasto por um algoritmo genético se aplica à função de avaliação.

### 3.5 Parâmetros Genéticos

Para a utilização de algoritmos genéticos na resolução de problemas de otimização deve ser estabelecida adequadamente uma série de parâmetros que irão dirigir a execução do algoritmo. Isso é feito em função das necessidades do problema e dos recursos disponíveis. São eles:

§ *Tamanho da população*: Uma população pequena pode resultar em uma convergência prematura (solução local) do AG, pois uma pequena parte do espaço de busca é avaliada. Por outro lado, uma população grande pode cobrir melhor o espaço de busca, obtendo uma solução global, mais os custos computacionais são mais elevados (CASTRO, 2001).

§ *Probabilidade de Cruzamento*: Indica a probabilidade de atuação do operador de cruzamento, com variação de 0 a 100%. Quanto maior esta probabilidade, maior será a inserção de novos indivíduos na população. Isto pode trazer um resultado indesejado, pois com boa parte da população sendo substituída pode ocorrer perda de indivíduos com alta aptidão. Por outro lado, com um valor baixo o processamento do algoritmo pode tornar-se demasiadamente lento.

§ *Probabilidade de Mutação*: Indica a probabilidade de atuação do operador de mutação, com variação de 0 a 100%. Uma pequena probabilidade de mutação



possibilita que se chegue em qualquer ponto do espaço de busca, enquanto que uma alta probabilidade de mutação pode tornar a busca essencialmente aleatória.

§ *Intervalo de Geração*: Indica a porcentagem da população que será substituída durante a próxima geração.

§ *Critério de Convergência*: Determina quando o algoritmo chegou a uma solução aceitável.

### 3.6 Operadores Genéticos

Os operadores genéticos são os responsáveis pela transformação da população ao longo das sucessivas gerações, ampliando a busca para atingir o resultado desejado. Submetidas a aplicação dos operadores genéticos, a população se diversifica mantendo as características adquiridas pelas gerações anteriores.

#### 3.6.1 Operador de Seleção

O objetivo do operador de seleção é direcionar o processo para as melhores regiões do espaço de busca. Para tanto esse operador busca os melhores indivíduos da população para reprodução, dando preferência para os indivíduos mais adaptados ao ambiente (MITCHELL, 1996). A seguir são descritos os métodos mais comumente encontrados na literatura.

### 3.6.1.1 Seleção Proporcional

Desenvolvido por John Holland, usa o valor de aptidão dos indivíduos dispostos em uma “roleta”, sendo que cada um dos indivíduos ocupa uma porção da roleta de tamanho proporcional ao seu valor de aptidão. Simula-se o giro da roleta, gerando-se um número aleatório entre 0 e 1. Este número representa a posição da “agulha” da roleta. Apesar de seu grande sucesso, devido sua simplicidade, um dos problemas encontrados pode ser o tempo de processamento, já que o método exige a normalização dos valores de aptidão. Segundo BANZHAF *et al.* (1998), em problemas de maximização, a probabilidade de um indivíduo ser selecionado por esse método é dada pela equação 3.1, onde  $f_i$ , é o valor de aptidão do indivíduo  $i$ .

$$p_i = \frac{f_i}{\sum_{j=1}^N f_j}, \quad i \in \{1, \dots, N\} \text{ e } j \in \{1, \dots, N\}$$

Equação 3.1 Probabilidade de seleção pelo método proporcional

### 3.6.1.2 Seleção por Torneio

A seleção por torneio é baseada em um subconjunto de indivíduos da população (BANZHAF *et al.*, 1998). A operação de seleção por torneio consiste em escolher aleatoriamente um número  $n$  de indivíduos da população, passar o melhor indivíduo dessa sub-população para uma população intermediária e repetir este procedimento  $N$  vezes. Os torneios são frequentemente realizados somente entre dois indivíduos.

### 3.6.1.3 Seleção por Truncamento

Baseado em um valor de  $T$  entre 0 e 1, a seleção é feita aleatoriamente entre os  $T$  melhores indivíduos (MUHLENBEIN, 1993). Por exemplo, se  $T = 0.4$ , então a seleção é feita entre os 40 % melhores indivíduos e os outros 60 % são descartados.

### 3.6.1.4 Seleção por Ordenação

Para suprimir as desvantagens da seleção proporcional, BAKER (1989) sugeriu este método de seleção onde, os indivíduos são ordenados de acordo com os valores de *fitness*. A posição  $N$  é atribuída para o melhor indivíduo e a posição 1 para o pior indivíduo.

Na seleção por ordenação linear, a probabilidade de seleção é linearmente distribuída entre os indivíduos da população em função das posições que lhe foram atribuídas (cf. Equação 3.2).

$$p_i = \frac{1}{N} \left[ \eta^- + (\eta^+ - \eta^-) \left( \frac{i-1}{N-1} \right) \right], \text{ onde } i \in \{1, \dots, N\}, \eta^- \geq 0 \text{ e } \eta^- + \eta^+ = 2$$

Equação 3.2 Probabilidade de seleção pelo método de ordenação linear

O valor de  $\eta^-/N$  é a probabilidade de seleção do pior indivíduo e  $\eta^+/N$  é a probabilidade do melhor indivíduo ser selecionado.

A seleção por ordenação exponencial se diferencia da linear somente pelo fato de que as probabilidades são exponencialmente ponderadas (BAKER, 1989). A base do expoente é o parâmetro  $0 < c < 1$  e a probabilidade do indivíduo ser selecionado é dada pela equação 3.3.

$$p_i = \frac{c-1}{c^N - 1} c^{N-i} \text{ onde } i \in \{1, \dots, N\}$$

Equação 3.3 Probabilidade de seleção pelo método de ordenação exponencial

### 3.6.1.5 Elitismo

Por suas características probabilísticas, os algoritmos genéticos podem a qualquer momento de sua execução encontrar o melhor indivíduo, a solução ótima para o problema. Mas este indivíduo pode ser perdido com a aplicação dos operadores genéticos durante o processo evolutivo.

O uso de elitismo visa contornar esse problema, preservando os melhores indivíduos encontrados durante a evolução. Existem duas formas de aplicação de elitismo: o elitismo simples (GOLDBERG, 1989) e o elitismo global (VASCONCELOS *et al.*, 2001).

O elitismo simples preserva sempre o melhor indivíduo que surge na população. Para tanto, salva o indivíduo e, na geração seguinte, se ainda é melhor que os indivíduos daquela nova população, é inserido diretamente nela.

No elitismo global verifica-se a aptidão dos filhos gerados pela aplicação dos operadores genéticos. Só serão inseridos na nova população aqueles que melhorarem a aptidão média da população.

### 3.6.2 Operador de Cruzamento

É mediante o operador de cruzamento ou recombinação que se obtém indivíduos filhos. Com a aplicação deste operador, os cromossomos dos pais são combinados e o resultado dessa combinação serão dois novos indivíduos (GOLDBARG *et al.*, 2000).

O operador de cruzamento é um dos principais componentes do algoritmo genético, ele é um dos maiores responsáveis pela evolução da população. É através de sua aplicação que se consegue o aumento da diversidade genética da população. Portanto,

estabelecer corretamente o método de cruzamento é fator determinante para o sucesso na aplicação dos algoritmos genéticos.

A forma mais simples de cruzamento é o operador de um ponto. Utilizado em codificações binárias, consiste em selecionar-se aleatoriamente um ponto de cruzamento e, a partir deste, trocar-se o material genético dos pais, formando assim dois indivíduos filhos (MITCHELL, 1996).

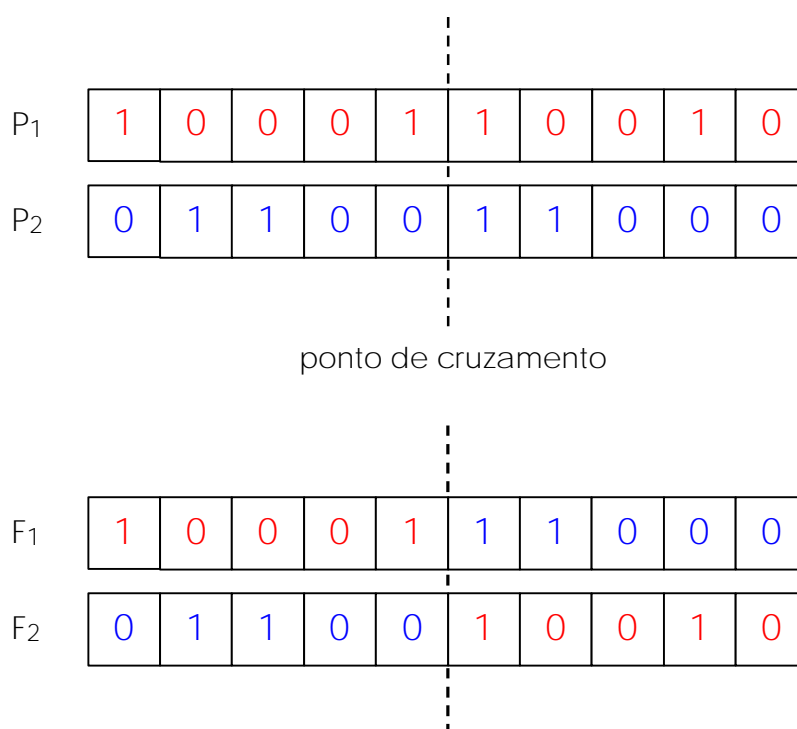


Figura 3.3 Exemplo de cruzamento de um ponto

Em extensão a este método, o operador multiponto estabelece, também aleatoriamente, mais de um ponto de cruzamento (cf. Figura 3.4).

Para problemas de permutação, que não permitem uma codificação binária, existe uma família de cruzamentos específicos descritos em GOLDBERG (1989), STARKWEATHER *et al.* (1991) e WHITLEY *et al.* (1995). A seguir são destacados os principais destes cruzamentos.

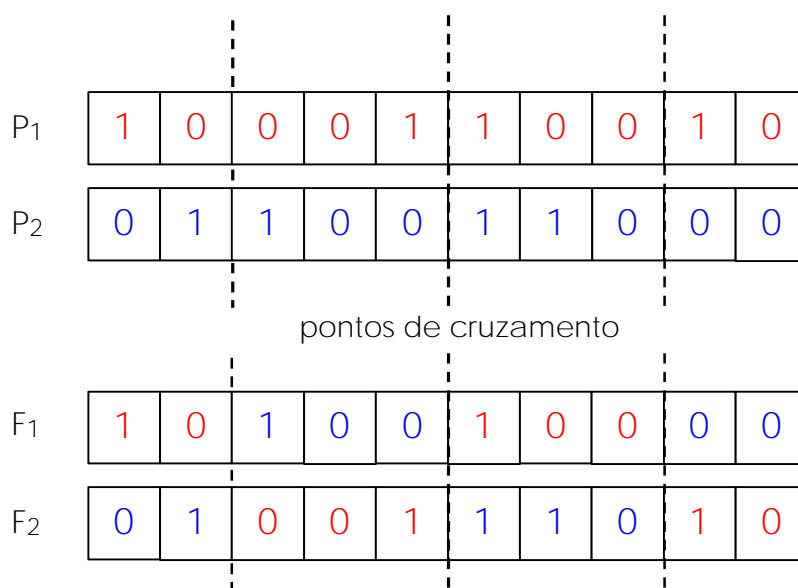


Figura 3.4 Exemplo de cruzamento multiponto

### 3.6.2.1 Operador *Order Crossover* #1 (OX)

Neste método de cruzamento, o filho herda de um dos pais os genes entre dois pontos de cruzamento na mesma seqüência em que se encontram no pai. Os genes fora da faixa entre os dois pontos de cruzamento são herdados do outro pai, também na mesma seqüência em que estão dispostos, porém não ocuparão no filho a mesma posição que estavam no pai, pois, como os genes entre os pontos de cruzamento já foram herdados, pode ocorrer que eles já estejam no filho. Esses genes serão alocados no filho a partir da primeira posição após o segundo ponto de cruzamento até o final do cromossomo e, quando alocado o último gene, a alocação continua da primeira posição do cromossomo até o primeiro ponto de cruzamento (STARKWEATHER *et al.*, 1991).

No exemplo da figura 3.5 é ilustrado o funcionamento deste operador. Os genes 1, 10, 8, 4, 9 são herdados do pai  $P_1$  na mesma seqüência em que nele aparecem. Os genes do pai  $P_2$  que não foram selecionados de  $P_1$ , na mesma seqüência em que aparecem, são, 2, 3, 5, 6 e 7. Assim, no filho, as posições depois do segundo ponto de cruzamento são ocupadas

pelos genes 2, 3 e 5. As posições que estão antes do primeiro ponto são ocupadas pelos genes 5 e 6.

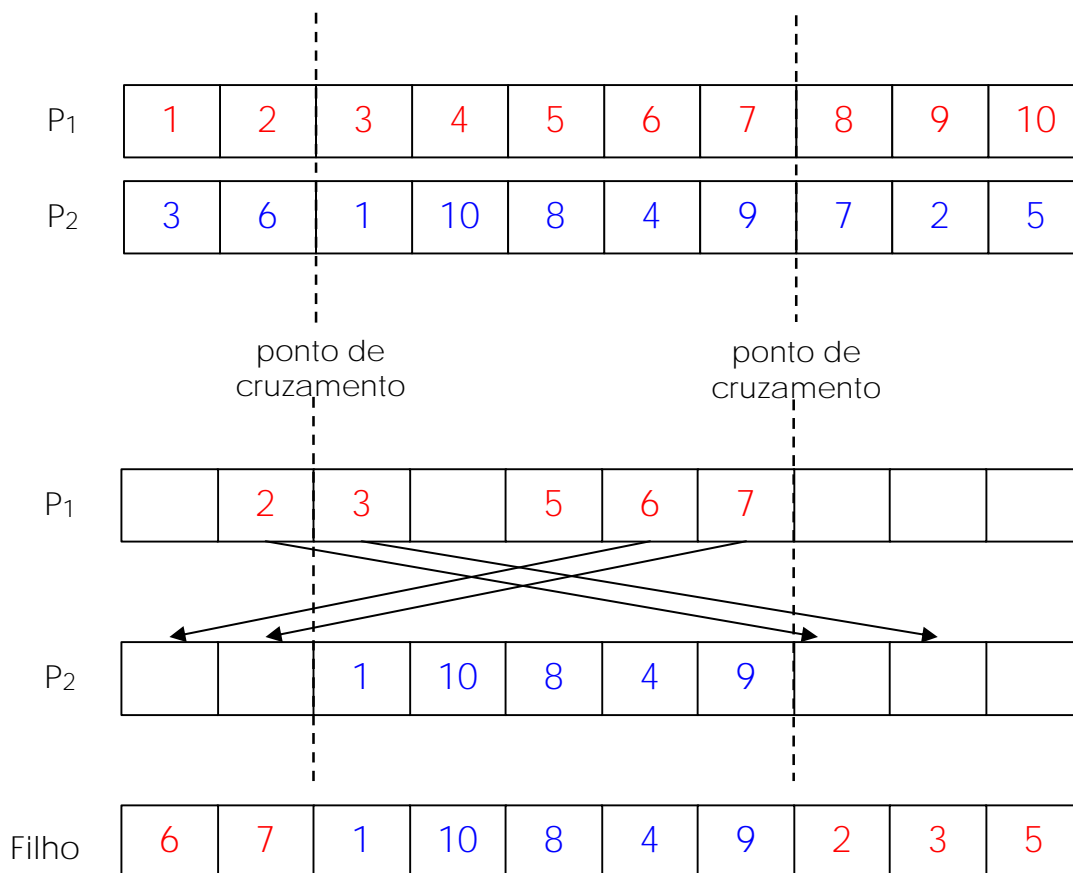


Figura 3.5 Exemplo de *Order Crossover* #1

### 3.6.2.2 Operador *Order Crossover* #2

A aplicação deste operador começa pela seleção aleatória de  $n$  posições chave no segundo pai. Os genes dessas posições no segundo pai são localizados no primeiro pai. O filho herdará, nessas mesmas posições, os genes das posições chave do segundo pai, na mesma seqüência em que neste aparecem. Os genes do primeiro pai que ainda não foram selecionados são herdados na mesma seqüência em que ocorrem (WHITLEY *et al.*, 1995).

Vê-se um exemplo da aplicação deste operador na figura 3.6. As posições dos genes 1, 10, 9, 2 do pai  $P_2$  foram selecionadas como posições chave. As posições correspondentes destes genes em  $P_1$  são, respectivamente a primeira, a décima, a nona e a

segunda. Então o filho herdará, nestas mesmas posições, os genes das posições chave de  $P_2$ , na mesma seqüência em que aparecem neste. Dessa forma, no filho, a primeira, a segunda, a nona e a décima posições serão ocupadas, respectivamente, pelos genes 1, 10, 9 e 2. Seus genes remanescentes serão ocupados pelos genes de  $P_1$ , na mesma seqüência, que ainda não foram selecionados, ou seja descarta-se os genes 1, 10, 9 e 2.

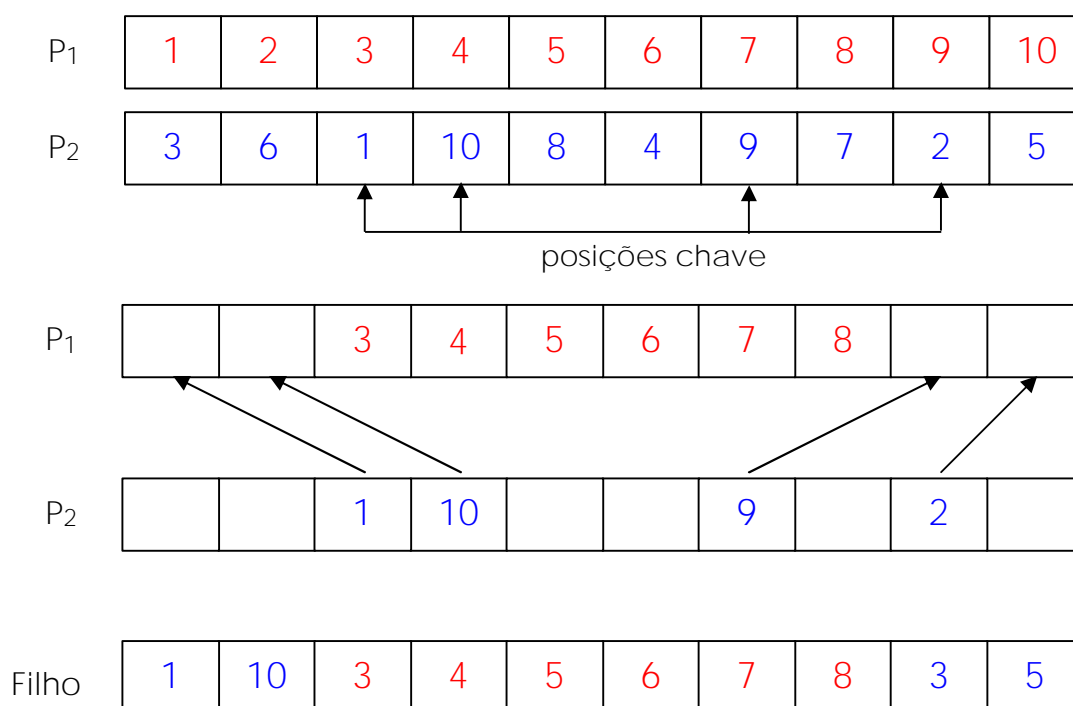


Figura 3.6 Exemplo de *Order Crossover* #2

### 3.6.2.3 Operador *Partially Mapped Crossover*

Para aplicar este operador, inicialmente, seleciona-se aleatoriamente dois pontos de cruzamento. Os genes do primeiro pai, entre essas posições, são herdados pelo filho e são alocados nas mesmas posições e seqüência que aparecem no pai. Os genes fora da faixa entre os pontos de cruzamento são alocados como segue. Deve-se encontrar no primeiro pai as posições que cada um dos genes do segundo pai, entre os pontos, ocupam. Se cada uma dessas posições, encontradas no primeiro pai, estão livres no filho então elas serão ocupadas pelos genes correspondentes do segundo pai (WHITLEY *et al.*, 1995).



Na figura 3.7, os genes 3, 4, 5, 6, do pai  $P_1$  são herdados diretamente pelo filho. O gene 1 do pai  $P_2$  está na mesma posição do gene 3 de  $P_1$ . Assim, deve-se encontrar o gene 3 em  $P_2$  e ocupar essa posição com 1 no filho. O processo é repetido para todos os genes dentro da faixa entre os pontos de cruzamento. As posições que ainda não tiverem sido alocadas no filho serão ocupadas pelos genes de  $P_2$  na mesma seqüência.

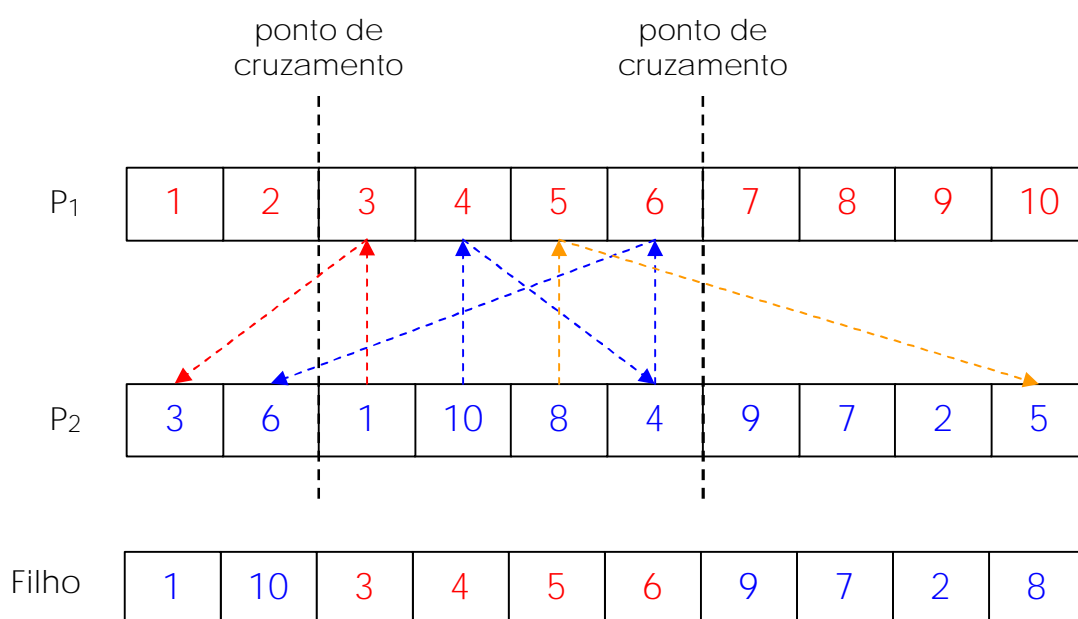


Figura 3.7 Exemplo de *Partially Mapped Crossover*

#### 3.6.2.4 Operador *Cycle Crossover*

Neste operador uma posição é aleatoriamente selecionada para dar início ao ciclo. O gene do primeiro pai nesta posição é herdado para o filho. O gene do segundo pai dessa posição não pode ser alocado no filho, pois esta já fora ocupada pelo gene do primeiro pai. Então este gene do segundo pai é localizado no primeiro pai e é herdado para o filho na posição em que se localizou no primeiro pai. Este processo é repetido até que se encontre o gene inicial no segundo pai. As posições que não foram ocupadas no filho são ocupadas com os genes remanescentes do segundo pai (STARKWEATHER *et al.*, 1991).

No exemplo da figura 3.8, foi selecionada a terceira posição como ponto de partida para o ciclo, logo o gene 3 é herdado pelo filho nessa mesma posição. O gene 1, da terceira posição do pai  $P_2$  é encontrado na primeira posição do pai  $P_1$ , então o gene é herdado pelo filho nesta posição. O gene 5 da primeira posição de  $P_2$  está na quinta posição de  $P_1$ , então o gene 5 é herdado na quinta posição pelo filho. Isto completa o ciclo, pois o gene inicial 3 foi alcançado em  $P_2$ . As posições que ainda estão livres no filho são herdadas de  $P_2$  na mesma seqüência.

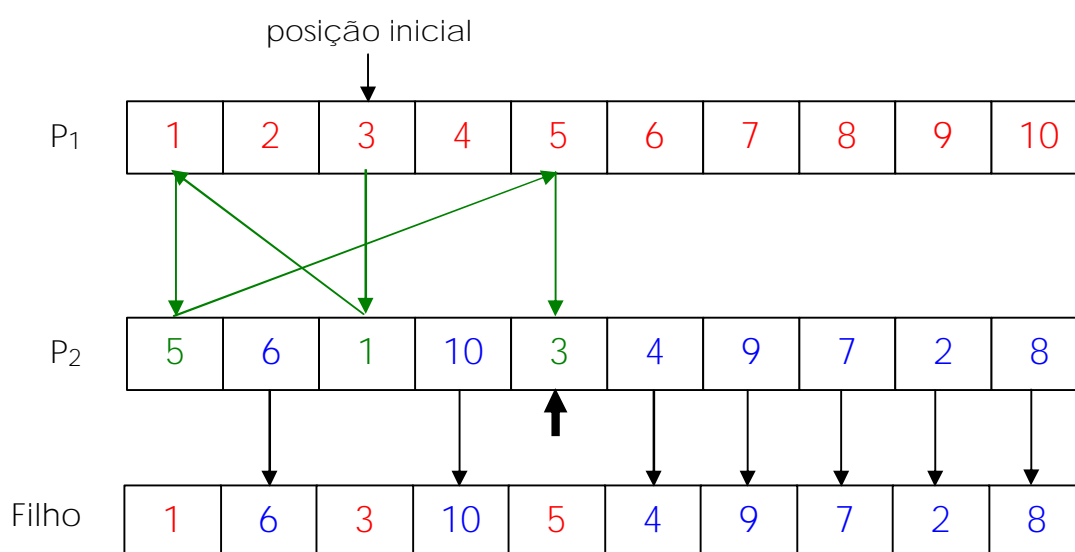


Figura 3.8 Exemplo de *Cycle Crossover*

### 3.6.2.5 Operador *Position Based Crossover*

Em relação aos outros operadores para permutação já apresentados, o *position based crossover* é o de aplicação mais simples. Seleciona-se aleatoriamente várias posições de um dos pais e os genes dessas posições são herdados para o filho. Os genes restantes são herdados na seqüência em que aparecem no segundo pai. Dessa forma, a aplicação desse operador visa preservar a informação durante o processo de recombinação (STARKWEATHER, 1991).

Na figura 3.9, os genes das posições selecionadas - segunda, terceira, sexta e nona - são herdados do pai  $P_1$  diretamente para o filho. Os genes restantes são herdados do pai  $P_2$  na ordem em que aparecem.

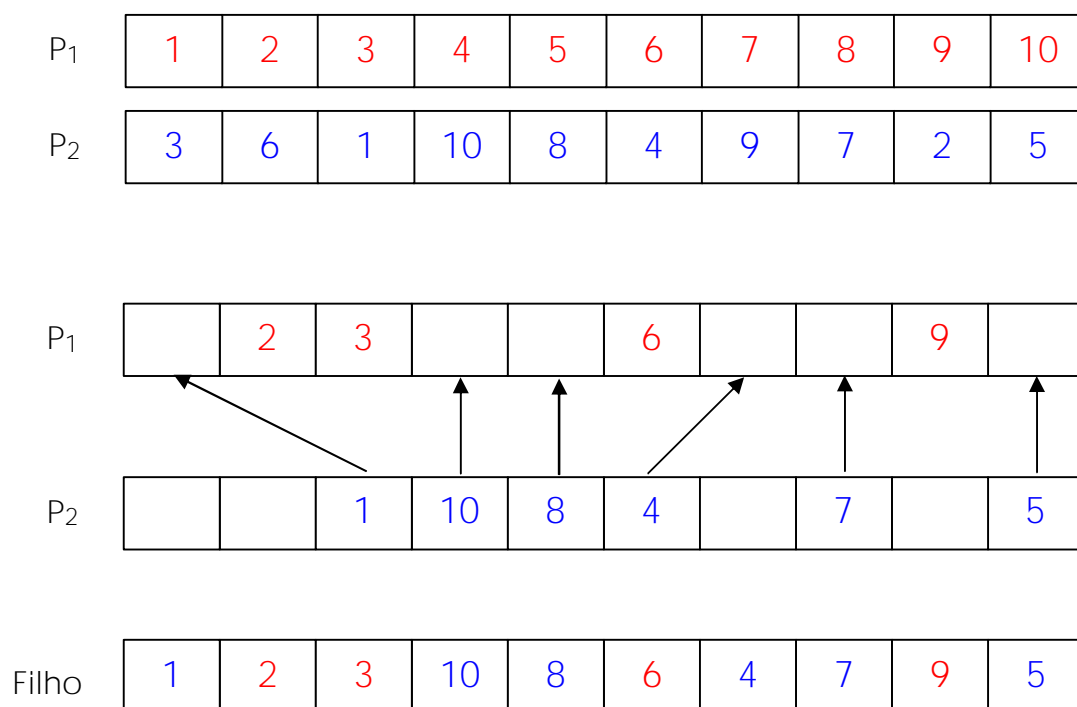


Figura 3.9 Exemplo de *Position Based Crossover*

### 3.6.3 Operador de Mutação

Além de possibilitar a busca em regiões do espaço ainda não exploradas, a aplicação do operador de mutação restaura a diversidade genética eventualmente perdida durante o processo evolutivo (GOLDBERG, 1989).

A mutação, em termos biológicos, consiste na mudança das informações genéticas de um indivíduo. Essa mudança pode ser suficiente para provocar o aparecimento de uma nova característica no organismo. Nos AGs, isto é alcançado alterando-se a informação codificada nos cromossomos dos indivíduos. Nas codificações binárias, a mutação consiste na simples troca de um valor pelo outro (MITCHELL, 1996).

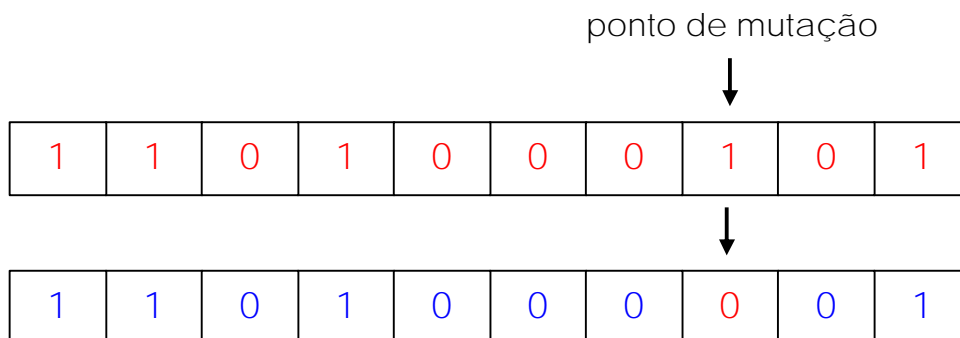


Figura 3.10 Exemplo de Mutação

Nas codificações para permutação, a mutação se baseia na troca de posições entre os genes dos cromossomos, podendo ocorrer de duas formas. A mutação do tipo *swap* consiste na troca aleatória de posições entre dois genes dos cromossomos. Já a mutação *shift* consiste em escolher dois genes aleatoriamente, retirar o primeiro dos genes escolhidos, deslocar todos os genes entre a posição vazia do gene retirado até o segundo dos genes escolhidos. A posição do segundo gene, vazia depois do deslocamento, é então ocupada pelo gene retirado (TUSON, 1995).

REEVES (1995) obteve em seus experimentos um desempenho melhor do operador de mutação *shift* do que do operador *swap*.

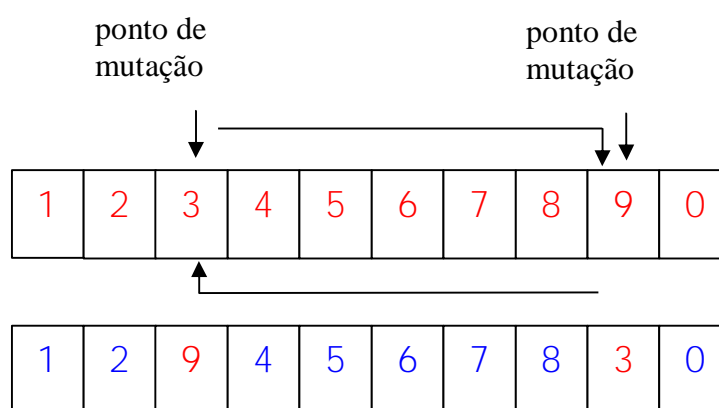


Figura 3.11 Operador de Mutação *Swap*

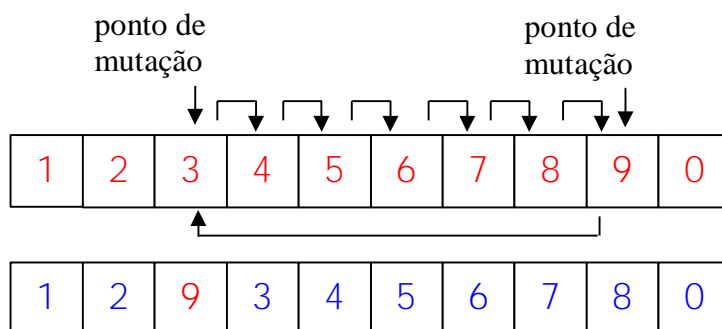


Figura 3.12 Operador de Mutação *Shift*

### 3.7 Critério de Convergência

Os algoritmos estocásticos não possuem métodos para determinar se a solução global para um dado problema foi encontrada. Dessa forma, os AGs continuariam sua busca, indefinidamente, ainda que já tivesse encontrado a solução ótima para o problema. Assim, definir corretamente o critério de término de um AG é de grande valor para uma execução satisfatória.

Se a aptidão requerida é conhecida pode-se determinar o critério de parada em função dela, estabelecendo um máximo erro admissível. Assim, quando determinado indivíduo possuir a aptidão cujo erro é menor que o estabelecido, finaliza-se o processo.

Um outro método que pode ser aplicado é quanto à variação genética da população. Se os indivíduos estão geneticamente muito parecidos, a função de avaliação vai retornar valores muito próximos para eles o que pode significar que eles estejam na mesma região do espaço de busca. Isto caracteriza a existência de um mínimo ou máximo da função.

Deve-se ter o cuidado de estabelecer um critério de parada sempre alcançável, já que os supra citados podem nunca ser alcançados. Este critério comumente é definido em função de um número máximo de gerações.

Todos esses métodos estão sujeitos a falhas. A convergência por variação genética falha quando os AGs convergem para um máximo local, o que caracteriza a convergência prematura. O uso de um número máximo de gerações pode finalizar o processo antes que o algoritmo seja capaz de atingir um ponto ótimo para o problema. Uma forma de contornar esses problemas seria adotar um critério de parada que combine os métodos aqui apresentados.

### 3.8 Estratégias de Reprodução

Desde a aceitação e sua conseqüente popularização, os AGs têm sido empregados freqüentemente de duas formas distintas em função da estratégia de reprodução dos indivíduos: o AG geracional (*generational*) e o AG *steady-state*. (ROGERS *et al.*, 1999).

No AG geracional, toda a população de uma geração anterior é substituída pela nova população obtida através da aplicação dos operadores genéticos (VAVAK *et al.*, 1996). Como toda população é substituída no algoritmo genético geracional, pode ocorrer a perda de bons indivíduos no decorrer do processo evolutivo. Logo, nesse tipo de abordagem, é perfeitamente aplicável o uso de elitismo (CASTRO, 2001).

Já na estratégia *steady-state*, as populações se sobrepõem, ou seja, apenas um ou dois indivíduos se reproduzem a cada geração (ROGERS *et al.*, 1999). Normalmente os indivíduos menos adaptados da população são substituídos pelos filhos resultantes da aplicação dos operadores executadas nos pais (MITCHELL, 1996).

### 3.9 Conclusão

Neste capítulo foram apresentados os principais aspectos relativos aos algoritmos genéticos. Foi possível identificar as potencialidades dos AGs que demonstraram ser uma técnica bastante aplicável ao problema em questão. Um fator importante, e característico dos

AGs, é a busca múltipla, possibilitada pelo uso de uma população de possíveis soluções, o que permite a escolha de uma solução que melhor se adapta ao problema.

A principal desvantagem dos AGs é o alto número de avaliações requerido devido à necessidade de se determinar a aptidão de cada um dos indivíduos de cada uma das populações, a cada geração. Isso, dependendo da complexidade do problema, pode acarretar em um tempo de execução bastante acentuado.

No próximo capítulo será modelado um algoritmo genético capaz de solucionar o problema da formação de grupos.

## 4 FORMAÇÃO OTIMIZADA DE GRUPOS UTILIZANDO AG

Definido o problema da formação de grupos como sendo de otimização, implementou-se um algoritmo genético capaz de encontrar soluções ótimas, especificamente neste caso, capaz de encontrar a melhor divisão dos alunos tendo como critérios os fatores pedagógicos. Neste capítulo apresentam-se quais foram as estratégias ou critérios de implementação utilizados para o desenvolvimento deste algoritmo bem como sua modelagem orientada a objetos.

### 4.1 Estratégias de Implementação

Para a execução eficiente de um AG na busca de uma solução ótima é necessário que seus parâmetros ou estratégias de implementação sejam bem definidos. A parametrização e os métodos aqui adotados estão definidos nas subseções seguintes.

#### 4.1.1 Codificação dos Indivíduos

A primeira decisão que se deve tomar quando da implementação de um programa de computação evolutiva é decidir como a solução será representada. Para o problema em questão, cada cromossomo representará uma classe de alunos, onde cada gene do cromossomo caracteriza um aluno dentro da classe (cf. figura 4.1). Desta forma, o número de genes do cromossomo é definido pelo número  $k$  de alunos da classe. Sequencialmente, cada conjunto de  $n$  genes representa um grupo de alunos dentro da classe, onde  $n$  é o número de alunos que compõem um grupo.



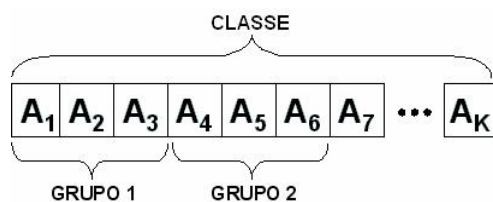


Figura 4.1 Codificação do cromossomo para  $n = 3$

Sendo assim, a diferença entre o código genético dos cromossomos se dará pela ordem dos genes no mesmo, o que caracteriza o problema como sendo de permutação, onde o espaço de busca é constituído por  $k!$  possibilidades. Para o caso da formação otimizada de grupos observa-se uma particularidade que é a distribuição dos genes em grupos. A cada  $n$  genes caracteriza-se um grupo, como dito anteriormente. Isso faz com que se considerem indivíduos gêmeos aqueles que tenham variação genética apenas dentro de determinado grupo. A figura 4.2 exemplifica a ocorrência de gêmeos.

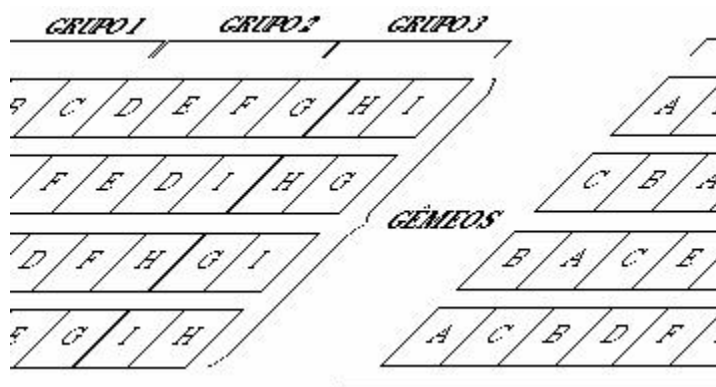


Figura 4.2 Indivíduos Gêmeos

Então, considerando-se essa particularidade, o número de possíveis soluções  $P$  é dado pela equação 4.1, onde  $n$  é o número de alunos na classe,  $k$  é o número de alunos no grupo e  $a$  é dado por  $n/k$ .

$$P = \frac{n!}{a!(k!)^a}$$

Equação 4.1 Número de possíveis soluções

#### 4.1.2 População Inicial

A população inicial utilizada neste trabalho é gerada de forma aleatória. Porém, a “qualidade” da população inicial é um fator crítico para o sucesso do processo evolutivo (DAIDA, 1999). A população inicial deve ser uma amostra significativa do espaço de busca, apresentando uma grande variedade genética de indivíduos, para que seja possível, através da recombinação de seus códigos, convergir para uma solução ótima.

Desta forma, a implementação da geração aleatória da população inicial foi desenvolvida de modo que produza uma grande diversidade genética da população.

#### 4.1.3 Tamanho da População

Não existem estudos conclusivos sobre o tamanho ótimo da população (MITCHELL, 1996). Devido a isso, o tamanho da população foi definido baseado em testes de maneira a se determinar, o mais eficaz possível, o tamanho da população.

Já que na formação otimizada de grupos cada gene do cromossomo é caracterizado por um aluno, o tamanho dos cromossomos varia em função do número de alunos na classe. O tamanho dos cromossomos, nesse caso o número de alunos na classe, influencia na performance do AG. O número de indivíduos na população, onde cada indivíduo é representado por um cromossomo, também é um fator que influencia na execução do AG. Então, a partir de testes com variados tamanhos de cromossomos e população, foi definida

uma relação tamanho do cromossomo versus tamanho da população, conforme mostrado na tabela 4.1, para se chegar ao tamanho ideal da população.

Tamanho do cromossomo (número $n$ de alunos na classe)	Número de cromossomos na população
$n < 10$	128
$9 < n < 20$	256
$19 < n < 30$	512
$n > 30$	756

Tabela 4.1 Relação Número de Alunos x Tamanho da População

#### 4.1.4 Função de Avaliação

A função de avaliação de cada indivíduo da população será determinada pelos fatores pedagógicos definidos para a instância do problema em questão. Estes fatores devem ser aplicados de tal forma que associem uma formação  $D$  a um valor numérico. Tomemos como exemplo a aplicação da função de avaliação para três fatores pedagógicos:  $f_1$ ,  $f_2$  e  $f_3$ .

Cada um destes três fatores aplicados a uma determinada formação  $D$  traduz-se em um valor numérico que indica o grau de aplicação daquele fator para aquela formação. A função de avaliação então será dada em função de cada um destes valores, conforme mostrado na tabela 4.2.

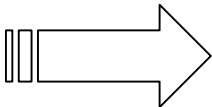
Fatores Pedagógicos		Função de Avaliação
$f_1(D) = X$		
$f_2(D) = Y$		
$f_3(D) = Z$		

Tabela 4.2 Função de avaliação

#### 4.1.5 Operador de Seleção

Não foi encontrado nenhum trabalho conclusivo a respeito da eficiência de determinado operador de seleção em relação aos outros. Então, foram implementados os principais operadores, já citados anteriormente, e por meio de testes o que se mostrou mais eficiente, e conseqüentemente adotado, foi a seleção por torneio binário na qual dois indivíduos são aleatoriamente selecionados e o mais apto (maior *fitness*) é o escolhido para o cruzamento.

#### 4.1.6 Operador de Cruzamento

O operador *position based crossover* é o de melhor desempenho entre os aplicados a problemas de permutação, superando os operadores *order crossover #1*, *order crossover #2*, *partially mapped crossover* e *cycle crossover* (STARKWEATHER *et al.*, 1991). Com base nisto, implementou-se o operador *position based crossover*.

#### 4.1.7 Operador de Mutação

Nos algoritmos genéticos o operador de mutação executa um papel secundário, porém necessário, pois possibilita restaurar a diversidade genética eventualmente perdida durante o processo evolutivo (GOLDBERG, 1989).

Ao utilizar-se o operador *position based crossover*, os operadores de mutação, tanto do tipo *swap* como *shift*, utilizados nas codificações por permutação, já estão implícitos. Ao aplicar-se a recombinação com apenas um ponto de *crossover*, obtém-se um efeito similar ao operador de mutação *swap*, e ao fazê-la com  $K-2$  pontos de *crossover*, o efeito é similar ao

operador de mutação *shift* (SOUZA, 2004). Sendo assim foi suprimido o uso do operador de mutação.

#### 4.1.8 Critério de Parada

Como se pode determinar o *fitness* máximo que um indivíduo pode alcançar no problema da formação otimizada de grupos (conforme será mostrado adiante) o primeiro critério de parada é se este *fitness* for alcançado por algum dos indivíduos da população. Para o caso desse *fitness* não ser alcançado por nenhum dos indivíduos utilizou-se a mesma estratégia adotada por (SOUZA, 2004) em seu trabalho. A execução do algoritmo é encerrada quando um número  $G$  de gerações apresentarem a mesma solução, ou seja, se o AG encontrar uma solução na  $i$ -ésima geração e ela permanecer como a melhor solução durante as  $G$  gerações seguintes, então se considera que o algoritmo convergiu e encerra-se a execução.

#### 4.1.9 Estratégia de Reprodução

Em (VAVAK *et al.*, 1996) foram comparados os desempenhos dos algoritmos genéticos *steady-state* e geracional com codificação binária. Nos testes, onde cada AG foi executado 50 vezes com uma população de 100 indivíduos, o AG *steady-state* se mostrou melhor que o geracional, tanto em função do número de iterações necessárias para atingir o ponto ótimo, quanto em relação à qualidade das soluções. Constatou-se ainda que no AG *steady-state* a estratégia de substituir o pior indivíduo foi melhor que a estratégia de substituir o indivíduo mais antigo.

Já em (TUSON, 1995) temos a comparação das estratégias com uma codificação por permutação. Também se constatou o melhor desempenho do AG *steady-state* sobre o geracional tanto na convergência do algoritmo quanto na qualidade das soluções.

Sendo assim, neste trabalho adotou-se a estratégia *steady-state* para o desenvolvimento do algoritmo genético.

## 4.2 Modelagem e Implementação

A solução implementada pode ser observada, simplificadamente, na seguinte seqüência de passos:

1. Gera-se aleatoriamente uma população de  $n$  indivíduos com  $k$  genes, onde  $n$  e  $k$  são dados em função do número de alunos;
2. Calcula-se o valor de aptidão de cada indivíduo da população;
3. Seleciona-se um par de cromossomos da população com a aplicação do operador de seleção. A seleção é feita de modo que o mesmo cromossomo não pode ser selecionado mais de uma vez;
4. Faz-se o cruzamento do material genético dos pais selecionados mediante a aplicação do operador de *crossover* com uma probabilidade  $P_C = 1$  gerando um indivíduo filho.
5. Se o novo indivíduo for melhor que o pior indivíduo da atual população, substitui-se o pior pelo melhor;
6. Se o critério de parada não for satisfeito retorna-se ao passo 3.

Ao fim deste processo teremos uma população composta por indivíduos que evoluíram de geração em geração e apresentam-se mais aptos que os das gerações anteriores e, como o critério de parada foi atingido, espera-se que pelo menos um desses indivíduos tenha atingido o ponto ótimo da função, ou que ao menos esteja próximo deste.

Nesta seção serão definidas, de maneira clara e concisa, as classes que representam o AG e como estas implementam o processo acima enumerado, assim como os

relacionamentos estáticos existentes entre as mesmas. Continuando a modelagem, será definida ainda a colaboração entre as classes para a realização da tarefa de formar grupos efetivos. Este processo de modelagem foi baseado na abordagem orientada a objetos e descrito na linguagem UML (*Unified Modeling Language*) (RUMBAUGH *et al.*, 2000), uma linguagem de modelagem de sistemas bastante madura e conhecida, tanto no meio comercial quanto acadêmico, que utiliza o paradigma orientado a objetos, permitindo a representação, abstrata e simplificada, de um sistema real.

#### 4.2.1 Diagrama de Classes

A fim de dar uma visão geral de como as classes estão organizadas e facilitar a visualização de seus relacionamentos, apresenta-se na Figura 4.3 um diagrama de classes, o qual demonstra a estrutura estática das classes. A seguir descreve-se cada uma das classes que compõem o diagrama.

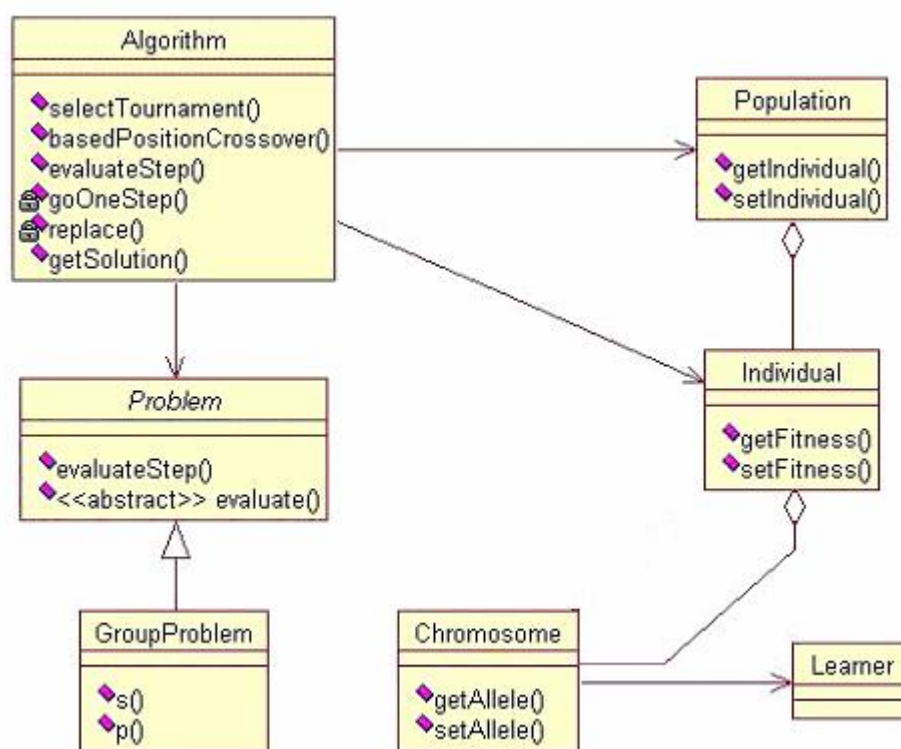


Figura 4.3 Diagrama de Classes

4.2.1.1 Classe *Algorithm*

A classe *Algorithm* encapsula todos os estados e comportamentos inerentes aos critérios de implementação do algoritmo genético, ou seja, é nela que estão mantidas as informações e implementados os métodos que representam a modelagem do algoritmo. Na tabela 4.3 tem-se uma descrição de cada um dos seus atributos e métodos principais. Para cada uma das classes subseqüentes utilizaremos o mesmo padrão de tabela para descrevê-las.

Algori thm	
<i>Atri butos</i>	
int chromLength	Número de alelos por cromossomo. Refletirá o número de alunos na classe.
int popSize	Tamanho da população.
double pc	Probabilidade de aplicação do operador de <i>crossover</i> .
GroupProblem problem	O problema a ser resolvido.
Population population	A população de indivíduos.
<i>Opera ções</i>	
Individual selectTournament()	
Seleciona um indivíduo para a aplicação do operador de <i>crossover</i> . De dois indivíduos selecionados aleatoriamente na população, retorna o de maior <i>fitness</i> .	
Individual basedPositionCrossover(Individual p1,Individual p2)	
Implementação do operador de <i>crossover Based Position Crossover</i> . Recebe como parâmetro os dois pais previamente selecionados e retorna o indivíduo filho, resultante do cruzamento de p1 e p2.	
void replace(Individual newIndividual)	
Substitui o pior indivíduo da população por <i>newI ndi vi dual</i> .	
double evaluateStep(Individual individual)	



Retorna o <i>fitness</i> de <i>individual</i> .
<code>void goOneStep()</code>
Executa um passo na execução do algoritmo que consiste em selecionar dois indivíduos para cruzamento, aplicá-los o operador de <i>crossover</i> , calcular o <i>fitness</i> do indivíduo filho e, se for o caso, substituir o pior indivíduo da população por este.
<code>Individual getSolution()</code>
A solução é o melhor indivíduo da população.

Tabela 4.3 Classe *Algorithm*4.2.1.2 Classe *GroupProblem*

A classe *GroupProblem* trata de manter informações relativas especificamente ao problema da formação dos grupos. É nela que deve ser implementada a função de avaliação o que conseqüentemente acarreta também na implementação da aplicação dos fatores pedagógicos sobre os grupos.

GroupProblem	
<i>Atributos</i>	
<code>int numLearnersGroup</code>	Número de alunos por grupo.
<code>int targetFitness</code>	Valor ótimo do <i>fitness</i> .
<code>boolean tfKnown</code>	O valor ótimo do <i>fitness</i> é conhecido?
<code>long fitnessCounter</code>	Número de avaliações já executadas.
<i>Operações</i>	
<code>double evaluateStep(Individual individual)</code>	
Calcula o <i>fitness</i> de <i>individual</i> e incrementa o atributo <i>fitnessCounter</i> . O cálculo do <i>fitness</i> é feito com uma chamada ao método <i>evaluate</i> .	
<code>abstract double evaluate(Individual individual)</code>	

Definição do método abstrato *evaluate* que deve ser implementado em uma classe filha em função dos fatores pedagógicos específicos do cliente.

Tabela 4.4 Classe GroupProblem

#### 4.2.1.3 Classe *Population*

A população dos indivíduos do algoritmo modelado é representada por esta classe. Os indivíduos, instâncias da classe *Individual*, são armazenados em um vetor. Logo, existe um relacionamento de agregação entre essas classes. Agregação é um tipo especial de associação no qual as duas classes participantes fazem um relacionamento todo-parte. Uma agregação descreve como a classe que possui a regra do todo é composta (contém) de outras classes que possuem a regra das partes. No caso, a classe *Population* (todo) é composta pela classe *Individual* (parte).

Popul ati on	
<i>Atri butos</i>	
int popSize	Tamanho da população.
Individual pop[]	Vetor de indivíduos.
double bestFitness	Melhor <i>fi tness</i> da população.
double avgFitness	Média do <i>fi tness</i> da população.
double worstFitness	Pior <i>fi tness</i> da população.
double betterFitness	Melhor <i>fi tness</i> encontrado durante todo o processo.
<i>Opera ções</i>	
void computeStats()	
Computa as estatísticas da população que consistem em atualizar os atributos <i>bestFi tness</i> , <i>avgFi tness</i> , <i>worstFi tness</i> e <i>betterFi tness</i> .	

Tabela 4.5 Classe *Population*

#### 4.2.1.4 Classe *Individual*

Os indivíduos, como já explicado anteriormente, são representados por uma cadeia de genes, os cromossomos. Assim, na classe *Individual* tem-se como estrutura principal uma instância da classe *Chromosome*, a implementação da cadeia de genes. Mais uma vez aparece o relacionamento de agregação, aqui, com a classe *Individual* sendo o todo e a classe *Chromosome*, a parte.

Individual	
<i>Atributos</i>	
Chromosome chrom	Cromossomo do indivíduo
double fitness	Valor de aptidão do indivíduo.
<i>Operações</i>	
construtor: Individual(int length)	
É instanciado o objeto <i>chrom</i> com o tamanho <i>length</i> e ao <i>fitness</i> é atribuído o valor 0.	
Void assign(Individual individual)	
Copia no indivíduo os atributos de <i>individual</i> .	
Learner getAllele(int index)	
Retorna o alelo (aluno) da posição <i>index</i> do cromossomo.	
Learner setAllele(int index, Learner learner)	
Atribui ao alelo da posição <i>index</i> o aluno <i>learner</i> .	

Tabela 4.6 Classe *Individual*

#### 4.2.1.5 Classe *Chromosome*

A classe *Chromosome* mantém um vetor de alunos, instâncias da classe *Learner*, que por sua vez representam os alelos do cromossomo.

Chromosome	
<i>Atributos</i>	
Learner alleles[]	Vetor de alelos (alunos).
int length	Tamanho do cromossomo.
<i>Operações</i>	
construtor: Chromosome(int length)	
<p>É instanciado o vetor de alleles com o tamanho <i>length</i>. A cada um dos <i>length</i> alelos, é atribuído um aluno, ou seja, um objeto da classe <i>Learner</i>. Aqui se deve ter uma implementação que busque no sistema de ensino-aprendizagem os alunos que deverão compor os grupos. Feitas as atribuições, as posições dos alelos são aleatoriamente trocadas para se garantir a diversidade genética dos indivíduos.</p>	

Tabela 4.7 Classe *Chromosome*

#### 4.2.1.6 Classe *Learner*

A classe *Learner* deve ser implementada pelo cliente da formação otimizada de grupos, sendo que deve conter, obrigatoriamente, atributos e operações que permitam a aplicação dos fatores pedagógicos.

### 4.2.2 Diagrama de Interação

Na seção 4.2 foram mostrados, através do diagrama de classes, os relacionamentos estáticos entre as classes. Nesta seção apresenta-se o diagrama de interação

(cf. Figura 4.4), que possibilita uma visão detalhada da colaboração entre as classes. Na seqüência detalham-se os passos principais da interação entre as classes.

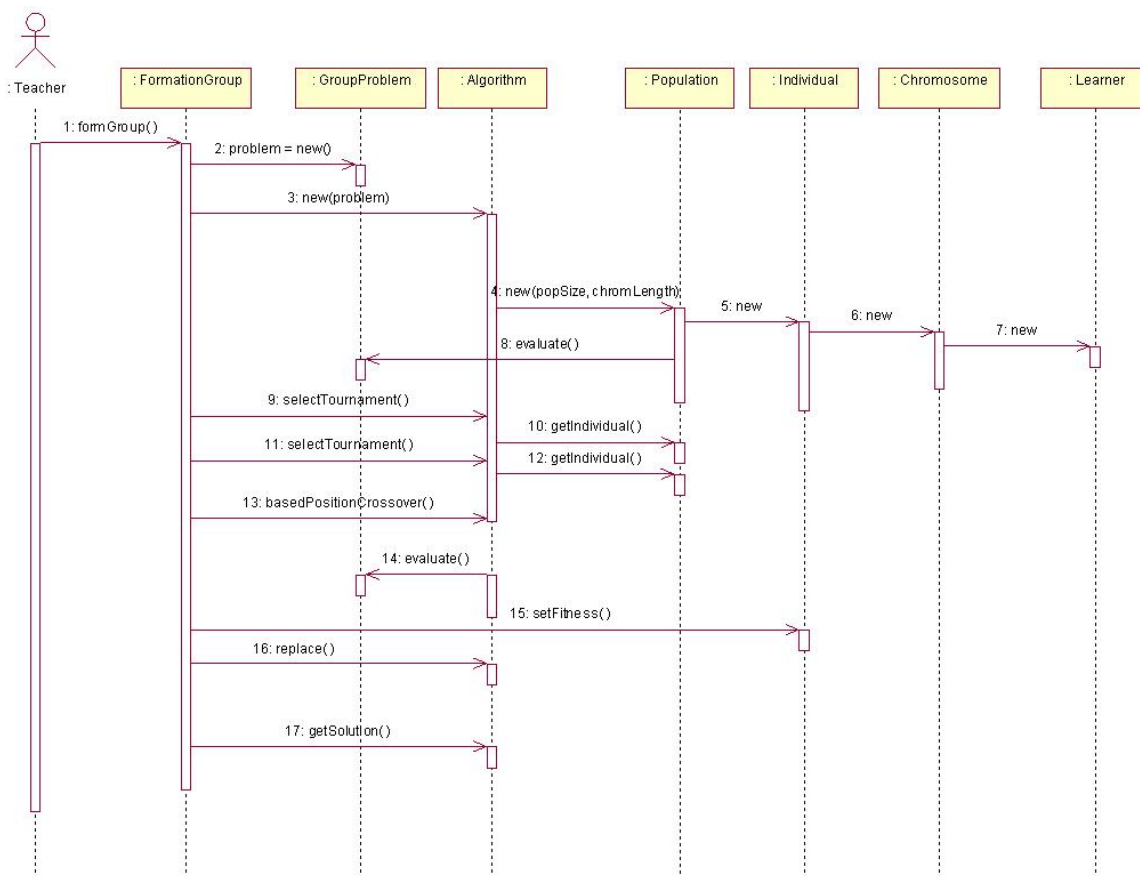


Figura 4.4 Diagrama de Seqüência

A execução começa com uma chamada ao método *formGroup* da classe *FormationGroup*, através de uma interface, pelo usuário responsável por executar a formação de grupos, no caso o professor. Este método, conforme visto na Figura 4.5, encapsula todo o processo de formação dos grupos, retornando ao final o melhor indivíduo da população, ou seja, uma possível solução.

Primeiramente é criada uma instância da classe *ProblemGroup* (mensagem 2) e em seguida são iniciados seus atributos principais com valores dos atributos da classe *FormationGroup*. Na seqüência instancia-se um objeto da classe *Algorithm* (linha 6), passando como parâmetro o objeto *problem* recém criado e os dados para a parametrização genética. No método construtor da classe *Algorithm* é instanciada também a população onde

são criados os indivíduos da população inicial. As mensagens 4, 5, 6 e 7 correspondem a criação da população inicial. Esses indivíduos são criados de forma bastante diversificada para assegurar uma grande variação genética da população. Essa diversidade é em função da posição de cada um dos cromossomos que compõem o indivíduo.

Código do método <i>formGroup</i>
<pre> 1111     public Individual formGroup() { 11111111 problem = new ProblemGroup(); 21111111 problem.setGeneN(gn); 31111111 problem.setGeneL(gl); 41111111 problem.setTargetFitness(tf); 51111111 61111111 ga = new Algorithm(problem, popSize, gn, gl, pc, pm); 71111111 81111111 for (int step = 0; step &lt; MAX_ISTEPS; step++) { 91111111 ga.goOneStep(); 10111111 11111111 if((problem.tfKnown()) &amp;&amp; 12111111 (ga.getSolution()).getFitness() &gt;= problem.getTargetFitness()){ 13111111 break; 14111111 } 15111111 } 16111111 17111111 return ga.getSolution(); 18111111 } </pre>

Figura 4.5 Método *formGroup*

Após a criação da população inicial a população é avaliada (mensagem 8) e o processo evolucionário propriamente dito é iniciado. Consiste basicamente na chamada sucessiva ao método *goOneStep()* (que corresponde as mensagens de 9 a 16) até que se atinja o critério de parada estabelecido. O código da Figura 4.6 mostra a implementação deste método.

Na linha 2 vêm-se dois passos dos mais importantes na execução do algoritmo: são selecionados da população dois indivíduos por meio da seleção por torneio, através do método *selectTournament()*, e a eles é aplicado o operador de cruzamento *basedPositionCrossover()* que retorna um indivíduo filho resultante do cruzamento que é

atribuído ao indivíduo *auxIndiv*. Na seqüência calcula-se o *fitness* do novo indivíduo e com a chamada ao método *replace()* ele é inserido na população em substituição ao pior indivíduo da população atual. Quando o critério de parada for atingido a população terá evoluído e os seus melhores indivíduos muito provavelmente serão soluções ótimas para o problema.

Código do método <i>goOneStep</i>
<pre> 1  public void goOneStep() { 2  auxIndiv.assign(bmx(selectTournament(),selectTournament())); 3  auxIndiv.setFitness(problem.evaluateStep(auxIndiv)); 4  replace(auxIndiv); 5  } </pre>

Figura 4.6 Método *goOneStep*

### 4.3 Conclusão

Neste capítulo foi mostrado como se aplicou os algoritmos genéticos para implementar a formação otimizada de grupos. No capítulo seguinte apresenta-se uma ferramenta integrante do ambiente cooperativo de ensino-aprendizagem *NetClass* que utiliza o algoritmo aqui descrito para formar grupos de aprendizagem cooperativa.

## 5 ESTUDO DE CASO: AMBIENTE NETCLASS

---

---

Com o intuito de aplicar o método de formação de grupos proposto, foi desenvolvido, no ambiente de ensino-aprendizagem *NetClass*, uma ferramenta para formação de grupos que se utiliza deste método. Neste capítulo apresenta-se o processo de desenvolvimento desta ferramenta sem antes se fazer uma breve descrição do ambiente *NetClass*.

### 5.1 NetClass

A proposta do projeto *NetClass* é criar um ambiente onde estudantes são separados em vários grupos organizados para cooperar e aprender através da interação com seu próprio grupo, com o sistema, o professor e outros grupos, tendo a sua disposição vários recursos, desde a multimídia até a tecnologia da Internet, visando alcançar o máximo de estímulo e conseqüente aprendizagem do aluno.

A abordagem de Sistemas Multiagentes (SMA), introduzida pela Inteligência Artificial Distribuída (IAD), adicionou a idéia de Trabalho Cooperativo, onde um grupo de agentes – sociedade de agentes artificiais e humanos – trabalha em conjunto visando alcançar um objetivo comum e onde cada um é responsável por realizar tarefas diferentes. Dessa forma busca-se alcançar um alto grau de flexibilidade e adaptação a mudanças no ambiente e nos problemas a serem resolvidos. O desafio é coordenar o comportamento inteligente nessa sociedade de agentes autônomos, seus conhecimentos, metas e planos para resolver problemas. O projeto *NetClass* faz uso dessa abordagem Multiagente. A sociedade de agentes do *NetClass* consta do agente Tutor, agente de Modelagem do Aprendiz, agente de Domínio e agente Estrategista (pedagogo).



Para dar suporte a toda essa estrutura de software inteligente se faz necessário o uso de um ambiente que incorpore características como escalabilidade, modularidade, interface de fácil manipulação, bem como a utilização na Internet, já que se trata de um sistema de ensino à distância. O *NetClass* incorpora em seu projeto cada uma dessas características. A figura 5.1 mostra a arquitetura básica do ambiente.

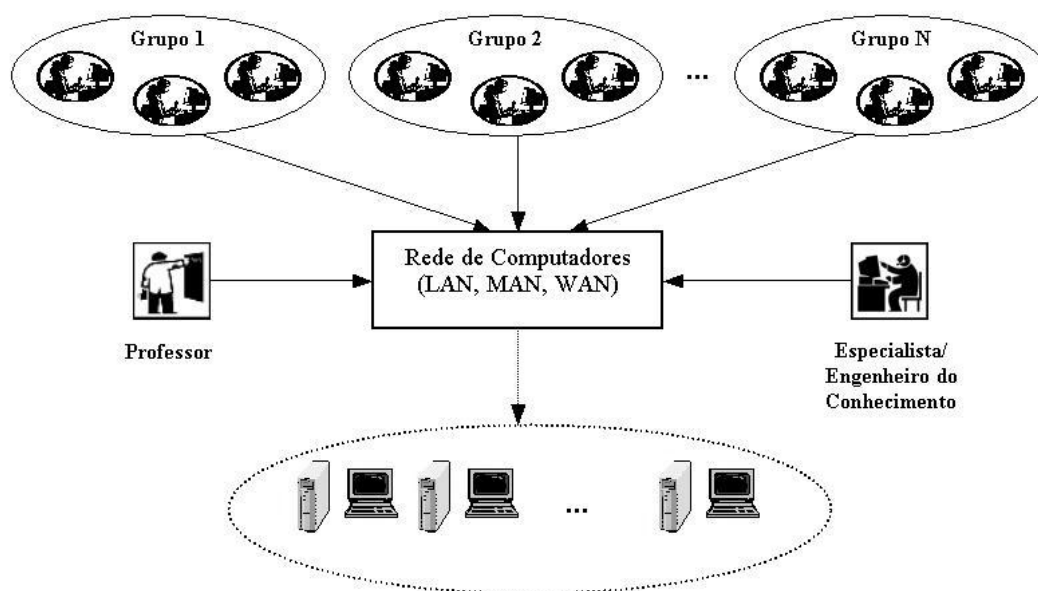


Figura 5.1 Ambiente NetClass

Os grupos, também chamados de áreas cooperativas, são formados por três estudantes interagindo com o sistema através de um terminal. Tal quantidade de alunos foi estabelecida pelo fato de que um número inferior não proporcionaria a quantidade de interações adequadas a um trabalho cooperativo.

Os terminais são as ferramentas utilizadas para possibilitar as interações com o professor e os outros grupos de estudantes. As interações são suportadas através de recursos de comunicação de uma rede de computadores, que pode ser de tipo LAN, MAN ou WAN.

Um professor interagindo com o sistema através de um terminal é considerado como uma área cooperativa específica.

## 5.2 Modelo Pedagógico do NetClass

Uma componente chave para o sucesso de um sistema educacional são o modelo e a fundamentação pedagógica no qual ele se baseia. O *NetClass* tem uma forte fundamentação pedagógica.

A aprendizagem no *NetClass* é um processo que ocorre em virtude da cooperação existente entre todos os elementos que o compõe (aprendiz, professor, especialista e engenheiro do conhecimento e os agentes artificiais). É nesse contexto que o processo de aprendizagem cooperativa ocorre. Esse processo é esquematizado no *NetClass* como sendo composto por seis Atividades Pedagógicas (cf. Figura 6.2).

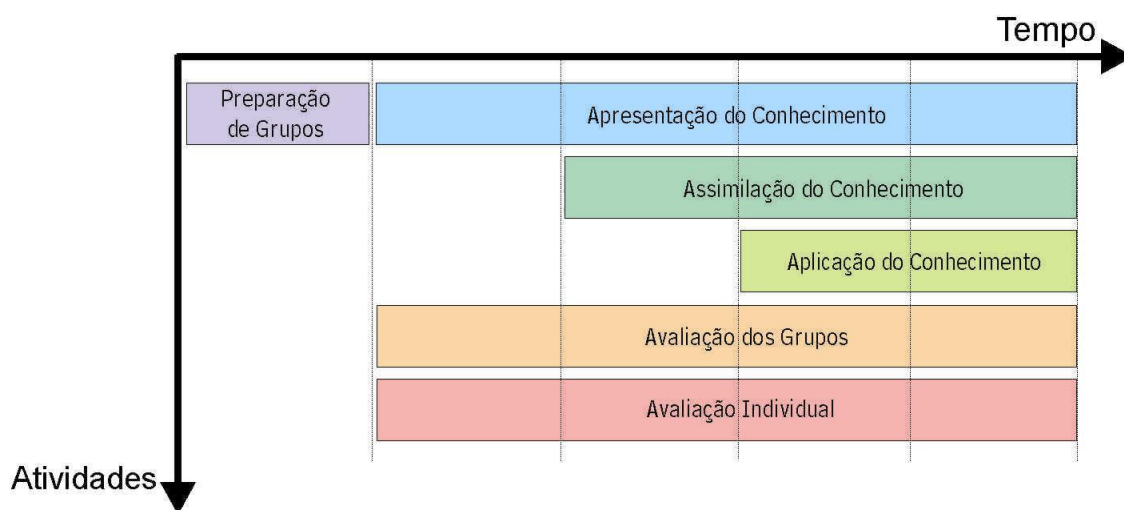


Figura 5.2 Modelo Pedagógico do NetClass

Cada uma dessas atividades é responsável por um determinado aspecto no processo de ensino-aprendizagem do *NetClass*. Deste modo, a cada atividade estão

associadas funções específicas que são desempenhadas com o uso de estratégias pedagógicas apropriadas, escolhidas de acordo com o modelo do aprendiz.

Na figura 5.2, pode-se ainda observar que em determinados momentos algumas atividades são realizadas paralelamente a outras ao longo do tempo, ou seja, na verdade, após a fase de Preparação dos Grupos, em qualquer momento, mais de uma atividade estará sendo realizada ao mesmo tempo durante esse processo.

As estratégias pedagógicas aplicadas no *NetClass* visam alcançar um melhor aproveitamento dos alunos no processo de ensino-aprendizagem de acordo com as diferenças comportamentais dos aprendizes em cada uma dessas fases.

Este trabalho tem como alvo a primeira fase do processo de ensino-aprendizagem, a preparação dos grupos, portanto suprimiram-se aqui aspectos relativos às demais fases e passa-se então, já na seção seguinte, a se apresentar como foi utilizada a formação otimizada de grupos no ambiente *NetClass*.

### 5.3 Formação Otimizada de Grupos no NetClass

Para se adotar a formação otimizada de grupos é necessário, primeiramente, definir os fatores pedagógicos que serão aplicados como critérios para a formação dos grupos. Assim, foram definidos dois fatores pedagógicos a serem aplicados no ambiente *NetClass*: um baseado no perfil dos alunos, onde o professor é responsável por definir quais os agrupamentos aceitáveis em função deste perfil. E o outro baseado na coesão estabelecida pelo uso de testes sociométricos, que levam em conta as preferências de agrupamentos dos alunos. Dessa forma, os critérios definidos visam contemplar a satisfação tanto dos professores quanto dos alunos no que diz respeito aos agrupamentos formados.

Respectivamente, estes fatores foram denominados como “perfil aceitável de um grupo” e “coesão sociométrica”.

### 5.3.1 Perfil Aceitável de um Grupo

O que se chama de perfil do aluno é um resumo das atitudes sociais e psicológicas ou características de um determinado aluno. Geralmente define-se o *perfil do aluno* como uma  $p$ -tupla  $\langle at_1^l, at_2^l, \dots, at_p^l \rangle$ , onde cada  $at_{i \leq p}^l$  é o valor de um atributo do aluno. Estes atributos são inferidos e mantidos pelo agente de modelagem do aprendiz, que é o responsável por manter o perfil de cada um destes. Assim, a partir dessas informações que compõem o perfil, define-se uma função *perfil do aluno* que mapeia estudantes de uma classe  $C$  em perfis. Simbolicamente:

$$l \xrightarrow{P} \langle at_1^l, at_2^l, \dots, at_p^l \rangle$$

No *NetClass* adotou-se inicialmente um perfil baseado em três atributos, todos de grande importância na aplicação da aprendizagem cooperativa. A capacidade de cooperação do aluno ( $at_1^l$ ). A capacidade de liderança ( $at_2^l$ ). E o grau de conhecimento do aluno sobre determinado assunto ( $at_3^l$ ). Todos ( $at_{i=1-3}^l$ ) variam em uma escala que vai de muito baixo para muito alto, passando por baixo, médio e alto. Logo, todo estudante está associado a uma tripla tal como  $l \xrightarrow{P} \langle medio, baixo, alto \rangle$ . Estas informações, como descrito acima, são armazenadas e mantidas pelo agente de modelagem do aprendiz, parte do *NetClass*.

O professor, embasado no perfil dos alunos, pode definir então o que para ele é um agrupamento ideal. Em outras palavras, o professor deve descrever, em função dos atributos – grau de cooperação, grau de conhecimento e grau de liderança – o que seria um grupo ideal para a aplicação da aprendizagem cooperativa. O que se chama aqui descrever o grupo é determinar que graus de atributos cada um dos integrantes dos grupos deveria possuir.

Por exemplo, o professor pode definir o grupo ideal como sendo: grupoIdeal = {a<sub>1</sub>, a<sub>2</sub>, a<sub>3</sub>} onde a<sub>1</sub> = {cooperação: alto, conhecimento: alto, liderança: muito alto}, a<sub>2</sub> = {cooperação: médio, conhecimento: médio, liderança: médio} e a<sub>3</sub> = {cooperação: médio, conhecimento: baixo, liderança: baixo}

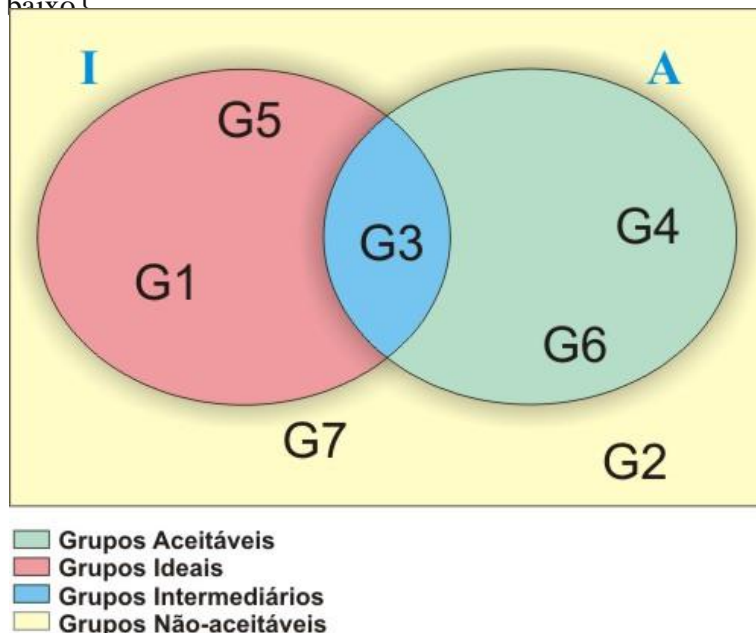


Figura 5.3 Classificação dos Grupos quanto ao Perfil

Considerando-se as diferenças individuais potencialmente existentes entre os aprendizes, nem todos os grupos, ou mesmo nenhum, poderiam ser um espelho do grupo ideal descrito pelo professor. Portanto é necessária a definição de um grupo aceitável que seria como um limite mínimo de aceitação do professor para os agrupamentos. Assim define-se um intervalo no qual estão grupos aceitáveis e grupos não-aceitáveis (cf. Figura 5.3).

Logo, considerando  $P$  o perfil dos alunos na classe  $C$ ,  $I$  o conjunto de perfis do grupo ideal e  $A$  o conjunto de perfis do grupo aceitável, define-se a aceitação de um grupo  $g = \{a_1, a_2, \dots, a_k\}$  pelo professor, baseado em  $P$ ,  $I$  e  $A$  como sendo a função:

$$f(P(g), P(a_1), \dots, P(a_k)) \begin{cases} \in I - A \\ \in I \cap A \\ \in A - I \\ \text{ou contrário} \end{cases} \quad \text{Aceitável} \quad \text{ou}$$

Esta função mapeia os grupos em categorias de aceitação. A categoria 3 representa os grupos mais aceitáveis. Categoria 0 não é um grupo aceitável.

### 5.3.2 Coesão Sociométrica

No ano de 1953 Jacob Levy Moreno, discípulo de Freud, expõe seu *método sociométrico* na *Medical Society* do estado americano de Nova York, onde é bem aceito. A seguir publica *Who shall survive*, sua principal obra e que é considerada a pedra fundamental do movimento sociométrico. A partir de então a sociometria vem sendo largamente difundida e desenvolvida por sociólogos, professores, psicólogos, antropólogos e outros estudiosos de áreas afins.

Inicialmente Moreno definia a sociometria como “*o estudo matemático das características psicológicas de conjuntos sociais, a técnica experimental dos métodos quantitativos e os resultados obtidos de sua aplicação*”. Posteriormente passou a defini-la como “*o estudo da organização e evolução dos grupos e da posição em que neles ocupam seus indivíduos em função da estrutura interna de cada indivíduo*”. Pouco depois dava esta outra definição: “*A sociometria tem por objetivo o estudo matemático das propriedades psicológicas das populações; com este propósito utiliza uma técnica experimental baseada em métodos quantitativos e expõe os resultados obtidos pela aplicação destes métodos. Desta forma, busca um exame metódico sobre a evolução e a organização dos grupos sobre a posição dos indivíduos nos grupos*” (MORENO, 1954).

Se a sociometria pretende analisar a estrutura do grupo, é necessário distinguir os dois tipos de estrutura grupal:

- 1) *Estrutura externa ou formal*: é de origem oficial, convencional e externa aos membros do grupo. Os vínculos que unem os indivíduos estão

determinados pelo papel oficial que cada um ocupa no grupo ou pelas tarefas que se devem realizar no mesmo e por isso esses vínculos são fracos e superficiais. Esta estrutura pode ser percebida em uma simples observação (organograma institucional).

- 2) *Estrutura interna ou informal*: é de origem espontânea e se baseia nas atrações pessoais, nos sentimentos, preferências, simpatias, antipatias e repulsas dos membros que formam o grupo. É esta a estrutura que deve ser captada mediante as técnicas sociométricas (sociograma sociométrico).

Sem dúvida nenhuma, a sociometria, tomada em seu aspecto mais interessante, vem a reduzir-se em um conjunto de métodos experimentais, uns terapêuticos, ou pedagógicos, e outros de investigação (OVEREJO, 1988).

Nas técnicas terapêuticas, ou pedagógicas, tenta-se encontrar os possíveis desvios, tanto grupais como pessoais, sendo dessa forma complementares às técnicas de investigação. As três mais importantes e conhecidas, todas criadas por Moreno, são o sociodrama, o psicodrama e o *role-playing*.

Com as técnicas de investigação se descobrem e analisam-se uma série de relações interindividuais e a real situação de uma pessoa no grupo. De acordo com o objetivo buscado podem ser divididas em: teste sociométrico, teste de interação, teste de avaliações coletivas, teste de espontaneidade, teste de rol, teste de eleição espontânea, teste de percepção sociométrica, teste de comparação de pares, teste ordinal, entre outros.

Entre as técnicas sociométricas de investigação supra citadas, o teste sociométrico é a de mais freqüente aplicação para o conhecimento da estrutura informal de uma classe, pois apresenta, além da vantagem de sua precisão, simplicidade e rapidez em seu emprego.

Destacaremos as seguintes aplicações da sociometria (CASANOVA, 1991), entre outras, pois são as que interessam para o desenvolvimento desse trabalho, que visa a formação dos grupos.

- a) Detecção de problemas de adaptação social: detectar a existência de elementos isolados no grupo, não escolhidos por ninguém, de forma a fazer com que seus companheiros de grupo lhes dêem atenção. Detectar a existência de alunos rejeitados pela maioria dos membros onde será necessário examinar as causas dessa marginalização. Detectar a existência de subgrupos onde será conveniente distribuí-lo entre a classe, etc.
- b) Determinação de estratégias metodológicas na aula: especialmente nos momentos para trabalhos em grupos, será um dado fundamental a estrutura informal da classe. Se procurará, a princípio, que os grupos a serem formados sejam os grupos espontaneamente formados pelos próprios alunos, em função de suas preferências. Darão um resultado ótimo, sempre que a classe estiver bem integrada e sem problemas especiais, como os descritos no item anterior. Quando houver algum aluno com dificuldade de integração, o conhecimento estrutural do grupo servirá para incorporá-lo com os companheiros que mais lhe possam motivar e com os que se encontre mais compenetrado. Assim será mais fácil eliminar o isolamento.

#### 5.3.2.1 Benefícios e Limitações

Podem-se dividir os benefícios e limitações da sociometria em três aspectos:



1) Resultados: A sociometria permite oferecer resultados descritivos das relações sociais com bastante precisão. Além disso, com poucas perguntas pode-se obter uma grande quantidade de informação, como se observará com o teste sociométrico. Porém, por outro lado, seus resultados não dão as razões e os porquês dessas relações sociais. Por ser essencialmente quantitativa, escapam aspectos qualitativos de grande importância.

2) Técnicos: Aplicar um teste sociométrico é realmente bastante simples, normalmente o número de perguntas é bem reduzido. Porém a nível técnico encontra-se uma série de dificuldades posto que a análise dos resultados se torna uma tarefa bastante árdua no caso do teste sociométrico. Além disso, devem-se ter cuidados com algumas condições pré-existentes para a aplicação do teste como, por exemplo, a necessidade de se ter um mínimo de conhecimento mútuo entre os membros da classe.

3) Éticos: É imprescindível assegurar a absoluta confidencialidade dos resultados que somente serão conhecidos pelo professor. A aplicação dessas técnicas tem como objetivo melhorar as relações sociais dos grupos e com essa finalidade deverão ser usadas discretamente. Podem-se ter alguns problemas se reveladas questões de aspectos de afinidade que podem causar conflitos entre os membros dos grupos.

Aspectos	Vantagens	Limitações
RESULTADOS	Grande quantidade de informação	Faltam as razões
	Alta precisão numérica	Faltam resultados qualitativos
	Aborda aspectos sociais	Podem ser circunstanciais
TÉCNICOS	Fácil aplicação	Difícil análise dos resultados

	Versatilidade para temas e repetições	Requer conhecimento mínimo entre os membros
ÉTICOS	Confidencial	Questões problemáticas como perguntas sobre repulsa

Tabela 5.1 Benefícios e limitações da sociometria

### 5.3.2.2 Teste Sociométrico

O teste sociométrico, como dito anteriormente, é uma das técnicas de investigação de ordem quantitativa da metodologia sociométrica que permite determinar o grau no qual os indivíduos são aceitos ou rejeitados em um grupo (seu status sociométrico), descobrir as relações entre os indivíduos e revelar a estrutura do grupo.

Um teste sociométrico consiste, basicamente, na aplicação de um questionário aos alunos e sua posterior análise. Devem-se destacar alguns pontos primordiais na elaboração dos questionários que são: os tipos de perguntas (escolha, rejeição e percepção) e o número de escolhas/rejeições a serem feitos. A figura 5.4 mostra o questionário aplicado no *NetClass*.

Teste Sociométrico - Módulo Inteligência Artificial	
<b>INSTRUÇÕES</b>	
Informe, em ordem de preferência, três alunos com os quais gostaria de trabalhar	
<b>OBSERVAÇÕES</b>	
1) As suas respostas nos ajudarão a proporcionar-lhe uma melhor aprendizagem	
2) Somente o professor do módulo saberá suas respostas, os outros alunos não saberão	
3) Responda sinceramente	
	1a. Escolha Ana Maria Souza Nunes
	2a. Escolha Pedro Paulo da Silva
	3a. Escolha Carlos Alberto Santos
<input type="button" value="OK"/> <input type="button" value="Cancelar"/>	

Figura 5.4 Teste sociométrico do *NetClass*

A partir das respostas dadas pelos alunos, podem ser feitas várias aplicações sobre esses resultados como explicado nos itens (a) e (b) da seção anterior. Esse resultado, para facilitar sua leitura e análise, podem ser representados graficamente. Essa representação gráfica do conjunto de relações sociais detectadas entre os membros da classe é o sociograma. A figura 5.5 mostra um exemplo de sociograma. Os círculos representam os alunos e as setas as escolhas de cada um dos alunos. As setas vermelhas caracterizam a primeira escolha, as setas azuis a segunda escolha e as setas verdes representam a terceira escolha.

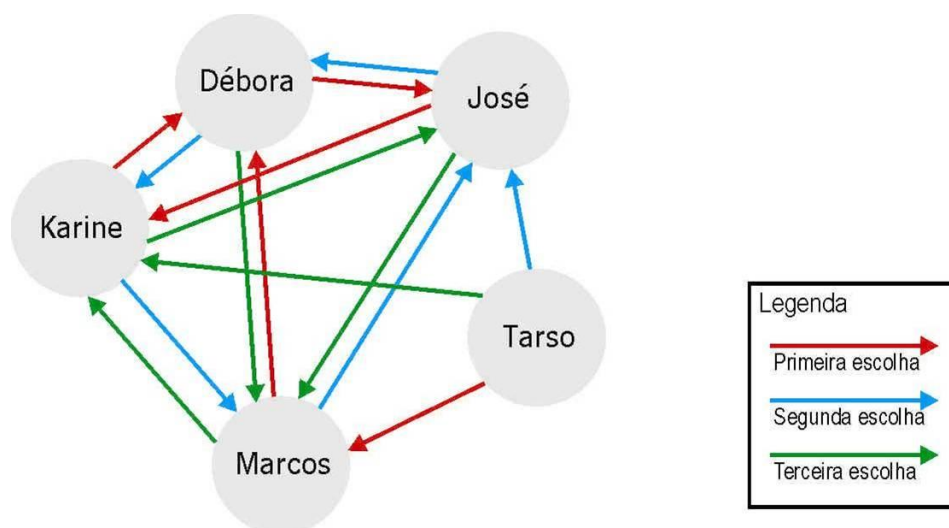


Figura 5.5 Exemplo de sociograma

### 5.3.2.3 Função de Coesão Sociométrica

Baseados na sociometria foram estabelecidas quatro regras para a formação de grupos no *NetClass*:

- § Regra 1: nunca formar grupos quando não há nenhuma relação (escolha) entre seus membros;
- § Regra 2: Para um aluno isolado, sempre colocá-lo em um grupo em que sua primeira escolha se satisfaz;

§ Regra 3: Sempre formar os grupos satisfazendo pelo menos alguma escolha de todos os componentes;

§ Regra 4: Quando possível formar grupos que preservem as preferências mútuas.

Usando as regras  $R$ , a função de coesão sociométrica de um grupo foi definida como sendo:

$$s[g]_R = \left. \begin{array}{l} 0 \text{ if } \neg R_1(g) \vee \neg R_2(g) \\ 1 \text{ if } R_1(g) \wedge R_2(g) \wedge \neg R_3(g) \\ 2 \text{ if } R_1(g) \wedge R_2(g) \wedge R_3(g) \wedge \neg R_4(g) \\ 3 \text{ caso contrário} \end{array} \right\}$$

Esta função mapeia os grupos  $g$  em graus de coesão. Grau 0 indica nenhuma coesão. Categoria 3 indica alta coesão.

### 5.3.3 A Função de Avaliação

Definidos os fatores pedagógicos a serem aplicados na formação otimizada de grupos no ambiente *NetClass*, pôde-se determinar a função objetivo a ser maximizada pelo algoritmo genético. A função de avaliação de cada indivíduo da população é dada pelo cálculo das funções  $\rho$  e  $s$  anteriormente apresentadas. Onde  $\rho$  está relacionada aos perfis dos alunos e  $s$  está associada com a coesão sociométrica dos grupos. A função de avaliação será então definida em termos de:

$$F(\rho[g], s[g]) = x \cdot \rho[g] + y \cdot s[g]$$

onde  $x$  e  $y$  são pesos que podem ser atribuídos pelo professor para favorecer uma das duas abordagens. Então a função de avaliação de cada indivíduo  $D$  é dada por:

$$FITNESS(\rho[\cdot]'_A, s[\cdot]_R)(D) = \sum_{g \in D} F(\rho[g]'_A, s[g]_R)$$

onde  $F(.,.)$  mapeia os graus de aceitação e coesão sociométrica em números que refletem a qualidade do grupo segundo os critérios  $\rho$  e  $s$  estabelecidos. Pode-se então observar que os valores máximo e mínimo de *FITNESS* são dados, respectivamente, por  $3n(x + y)$  e  $n(x + y)$ , onde  $n$  é o número de grupos formados, pois os valores máximos e mínimos de  $\rho$  e  $s$  para cada grupo válido são 3 e 1 respectivamente.

#### 5.4 Ferramenta para Formação de Grupos no *NetClass*

Com a definição dos fatores pedagógicos e, conseqüentemente, da função de avaliação a ser adotada, pôde-se então modelar e implementar uma ferramenta para a formação otimizada de grupos integrada ao *NetClass*. Primeiramente foram implementadas as classes *GroupProblemNetClass* e a classe *LearnerNetClass*, descritas a seguir, que modelam as características específicas da formação de grupos no *NetClass*.

##### 5.4.1 Classe *GroupProblemNetClass*

A classe *GroupProblemNetClass* estende a classe *GroupProblem* implementando o método *evaluate*. Para tanto, implementa também os métodos  $\rho$  e  $s$ , que são, respectivamente, a implementação das funções de aceitação do perfil e da coesão sociométrica.

GroupProblemNetClass extends GroupProblem
<i>Operações</i>
double evaluate(Individual individual)
Avalia o indivíduo, retornando o <i>fitness</i> calculado em função dos métodos $p(i\ n\ d\ i\ v\ i\ d\ u\ a\ l)$ e $s(i\ n\ d\ i\ v\ i\ d\ u\ a\ l)$ .
double p(Individual individual)
Implementação da função de aceitação do perfil.

<code>double s(Individual individual)</code>
Implementação da função de coesão sociométrica.

Tabela 5.2 Classe *GroupProblemNetClass*5.4.2 Classe *LearnerNetClass*

A classe *LearnerNetClass* mantém as informações dos aprendizes necessárias para a aplicação dos fatores pedagógicos. Sendo assim seus atributos refletem o perfil dos alunos utilizados pelo fator de aceitação do perfil e as suas preferências de agrupamento, a serem utilizadas pela função de coesão sociométrica. Fez-se aqui necessária então a criação de uma nova classe que mantenha as escolhas dos aprendizes, a classe *Choice*, que será descrita na seção seguinte.

LearnerNetClass	
<i>Atributos</i>	
String id	Identificador do aluno.
String name	Nome do aluno.
int cooperationLevel	Nível de cooperação.
int knowledgeLevel	Nível de conhecimento.
int leadershipLevel	Nível de liderança.
Choice choices[]	Preferências de agrupamento.
boolean isIsolated	Indica se o aluno é isolado, ou seja, se ninguém o escolheu no teste sociométrico.
<i>Operações</i>	
construtor: <code>LearnerNetClass(String id, String name)</code>	
Cria uma instância da classe, inicializando os atributos id e name.	

Tabela 5.3 Classe *LearnerNetClass*

### 5.4.3 Classe *Choice*

A classe *Choice* é responsável por manter as preferências de agrupamentos dos aprendizes. Como já dito, cada aprendiz escolhe três outros com os quais desejaria agrupar-se, em ordem de preferência. A classe *LearnerNetClass*, que representam os aprendizes, tem um relacionamento com a classe *Choice*, que é implementado como um vetor de escolhas, componente da classe *LearnerNetClass*.

Choice	
<i>Atributos</i>	
String id	Identificador do aluno escolhido.
boolean mutualChoice	Indica se é uma escolha mútua.
int order	Indica a ordem de preferência da escolha.
<i>Operações</i>	
construtor: Choice(String id, boolean mutualChoice, int order)	
Cria uma instância da classe, inicializando os atributos id, name e order.	

Tabela 5.4 Classe *Choice*

Na seção seguinte apresentam-se as funcionalidades da ferramenta necessária para a realização da tarefa de formar grupos no ambiente *NetClass*.

### 5.4.4 Aplicação

A ferramenta de formação de grupos é uma aplicação *web* (Java / JSP) integrada junto ao módulo do professor no ambiente *NetClass* que deve possibilitá-lo formar os grupos

segundo os critérios já definidos. Apresenta-se aqui esta aplicação, passando por cada uma das telas da ferramenta.

Inicialmente o professor deve selecionar qual tipo de abordagem será aplicada na formação dos grupos: (a) baseado no perfil dos alunos, (b) baseado no teste sociométrico ou (c) em ambas (cf. Figura 5.6).

The screenshot shows the 'NETclass - Ambiente de Ensino à Distância' interface. The main content area is titled 'Avaliações ::' and displays the following information:

Os seguintes grupos estão formados para este módulo:

Grupos - Módulo Orientação a Objetos em Java				
Grupo 1	Cooperação	Conhecimento	Liderança	Sociograma
Bysmarck Barros	★★★★★	★★★★★	★★★★★	
Carlos Eduardo	★★★★★	★★★★★	★★★★★	
Daniel Lima	★★★★★	★★★★★	★★★★★	
Grupo 2				
David Cavassana	★★★★★	★★★★★	★★★★★	
Fransislene Oliveira	★★★★★	★★★★★	★★★★★	
Valdeci Ribeiro	★★★★★	★★★★★	★★★★★	

Below the table, there is a section titled 'Qual a abordagem para a redivisão de grupos?' with three radio button options:

- Baseado nos perfis dos alunos
- Baseado no teste sociométrico
- Baseado em ambas abordagens

Navigation buttons include 'Próximo >>' and 'Voltar <<'. The status bar at the bottom shows 'Concluído' and 'Intranet local'.

Figura 5.6 Ferramenta de Formação de Grupos – Escolha da Abordagem

Escolhendo a opção (a), a ferramenta o permite definir os grupos ideal e aceitável que serão os parâmetros utilizados para aplicação do fator pedagógico baseado no perfil dos alunos. Para facilitar a definição destes grupos é apresentado ao professor a classificação de cada um dos atributos dos alunos (cf. Figura 5.7). O professor então define os grupos e inicia o processo de formação de grupos.



**NetClass**  
Sistema de Gerenciamento de Aprendizagem

**Menu ::**

- MEUS MÓDULOS
  - Módulos
  - Lições
  - Avaliações
  - Cadastrar Questões
  - Aluno e Notas
  - Corrigir Avaliação
  - Acompanhamento
  - Lançar Notas
  - Relatório de Notas
- SERVIÇOS
- GRUPOS
- ATIVIDADES
- AVALIAÇÕES
- DADOS PESSOAIS
- LOGOUT

**Avaliações ::**

Escolha os perfis do grupo ideal e do grupo médio para formação:

Perfil dos alunos - Módulo Orientação a Objetos em Java			
Aluno	Cooperação	Conhecimento	Liderança
Bysmarck Barros Sousa	★★★★☆	★★★★☆	★★★★☆
Carlos Eduardo dos Santos Araujo	★★★★☆	★★★★☆	★★★★☆
Daniel Lima	★★★★☆	★★★★☆	★★★★☆
David Cavassana Costa	★★★★☆	★★★★☆	★★★★☆
Fransislene de Oliveira Barros	★★★★☆	★★★★☆	★★★★☆
Valdeci Ribeiro da Silva Jr	★★★★☆	★★★★☆	★★★★☆

**Visão Gráfica do Perfil dos Alunos**

**Grupo Ideal**

Aluno #1 Aluno #2 Aluno #3

Liderança min      max

Cooperação min      max

Conhecimento min      max

Liderança min      max

Cooperação min      max

Conhecimento min      max

Liderança min      max

Cooperação min      max

Conhecimento min      max

**Grupo Médio**

Aluno #1 Aluno #2 Aluno #3

Liderança min      max

Cooperação min      max

Conhecimento min      max

Liderança min      max

Cooperação min      max

Conhecimento min      max

Formar Grupos

::: Voltar :::

Rodapé - LSI netclass@lsi.ufma.br

Figura 5.7 Ferramenta de Formação de Grupos – Definição dos grupos ideal e médio

Se a escolha do professor for a opção (b), a ferramenta checa se já fora aplicado algum teste sociométrico (necessário para a aplicação do fator pedagógico de coesão

sociométrica) naquele módulo, apresentando ao professor o resultado dessa verificação. O professor então, a partir desse resultado, pode optar por aplicar um novo teste ou utilizar o teste já realizado. A realização do teste se dá com uma mensagem enviada a cada um dos alunos do módulo contendo um *link* para a tela apresentada na Figura 5.4. Ao fim do teste, isto é, quando todos os alunos tiverem escolhido suas preferências, a ferramenta então inicia o processo de formação dos grupos.

Escolhendo a opção (c), o professor será guiado, primeiro ao processo da opção (a), depois para o processo da opção (b) e só após isso o processo de formação de grupos será iniciado.

O algoritmo genético é então iniciado com os parâmetros necessários e ao fim de sua execução a ferramenta apresenta ao professor os grupos formados segundo os critérios estabelecidos, mostrando a classificação dos atributos de cada um dos alunos agrupados e comparando com os grupos ideal e aceitável por ele previamente definidos. Mostra também o sociograma de cada um dos grupos. O professor então pode analisar a formação proposta pela ferramenta e então aceitá-la, rejeitá-la ou ainda alterá-la.

## 5.5 Resultados

Para experimentar e avaliar os resultados do método foi desenvolvida uma aplicação (cf. Figura 5.8), também *web*, onde se pôde fazer variados testes. Nesta seção apresenta-se, com base nos testes com essa aplicação, os resultados alcançados com o método de formação de grupos proposto. Para isso foi definida uma turma composta por 21 alunos (cf. Tabela 5.5), onde estes foram divididos em sete grupos de três integrantes cada. Definiu-se ainda um grupo ideal e um grupo médio e, além disso, definiu-se as preferências de agrupamento de cada um dos alunos, simulando a aplicação de um teste sociométrico.



Formação de Grupos		
Parâmetros Genéticos		
	Tamanho da População:	512
	Taxa de Crossover:	100 %
	Taxa de Mutação:	0 %
	Operador de Crossover:	BASED POSITION CROSSOVER
	Operador de Seleção:	BINARY TOURNAMENT
 Aguarde...		
<input type="button" value="Formar Grupos"/>		

Figura 5.8 Aplicação Teste

As preferências de agrupamento estão representadas na tabela pela numeração das colunas 1ª, 2ª e 3ª escolhas. Cada aluno da tabela é identificado por um número e é este número que aparece nas colunas de escolha. Assim podemos ver que, por exemplo, Agatha Christie escolheu, em ordem de preferência, os alunos Edgar Allan Poe, Arthur Conan Doyle e Julio Verne.

Turma de alunos para teste						
	Cooperação	Liderança	Conhecimento	1ª Escolha	2ª Escolha	3ª Escolha
1. Agatha Christie	★★★★★	★★★☆☆	★★★★☆	9	3	15
2. Aldous Huxley	★★★☆☆	★★★★☆	★★★★★	14	16	5
3. Arthur Conan Doyle	★★★☆☆	★★★★☆	★★★★☆	2	4	5
4. Charles Baudelaire	★★★☆☆	★★★☆☆	★★★★☆	19	13	9
5. Dan Brown	★★★☆☆	★★★☆☆	★★★★★	3	20	18
6. Daniel Defoe	★★★★☆	★★★★★	★★★★☆	12	9	15
7. Dante Alighieri	★★★☆☆	★★★☆☆	★★★★☆	11	16	5
8. Douglas Adams	★★★★☆	★★★☆☆	★★★★★	21	16	15

9. Edgar Allan Poe	★★★★☆	★★★★☆	★★★★☆	6	3	12
10. Erasmo de Rotterdam	★★★★☆	★★★★☆	★★★★☆	7	4	20
11. George Orwell	★★★★☆	★★★★☆	★★★★☆	18	5	15
12. H. P. Lovecraft	★★★★☆	★★★★☆	★★★★☆	3	1	15
13. J. K. Rowling	★★★★☆	★★★★☆	★★★★☆	19	21	10
14. J. R. R. Tolkien	★★★★★	★★★★☆	★★★★★	18	16	3
15. Julio Verne	★★★★☆	★★★★☆	★★★★☆	12	13	2
16. Lewis Carroll	★★★★☆	★★★★☆	★★★★☆	19	18	6
17. Mary Shelley	★★★★☆	★★★★☆	★★★★☆	11	1	3
18. Michael Crichton	★★★★☆	★★★★☆	★★★★☆	1	17	12
19. Oscar Wilde	★★★★☆	★★★★☆	★★★★☆	18	13	21
20. Robert Louis Stevenson	★★★★☆	★★★★☆	★★★★☆	17	16	7
21. Terry Pratchett	★★★★★	★★★★☆	★★★★☆	6	7	9

Tabela 5.5 Turma de alunos para teste

A tabela 5.6 mostra a definição dos grupos ideal e médio, onde para cada um dos alunos do grupo tem-se um determinado grau de cooperação, conhecimento e liderança.

	Grupo Ideal			Grupo Médio		
	Cooperação	Liderança	Conhecimento	Cooperação	Liderança	Conhecimento
Aluno #1	★★★★☆	★★★★★	★★★★☆	★★★★☆	★★★★☆	★★★★☆
Aluno #2	★★★★☆	★★★★☆	★★★★☆	★★★★☆	★★★★☆	★★★★☆
Aluno #3	★★★★★	★★★★☆	★★★★☆	★★★★☆	★★★★☆	★★★★☆

Tabela 5.6 Definição dos grupos ideal e médio

Para se avaliar os resultados foram gerados grupos utilizando as três possíveis abordagens: baseada no perfil dos alunos, baseada no teste sociométrico e ainda baseada em ambos. As figuras 5.9, 5.10 e 5.11 mostram, respectivamente, o resultado da execução para cada uma dessas abordagens. Deve-se destacar que diferentes execuções para uma mesma abordagem geram resultados distintos, porém de mesmo valor de *fitness*.

Grupos				
Grupo 1	Cooperação	Conhecimento	Liderança	Sociograma
George Orwell	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Aldous Huxley	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Douglas Adams	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Grupo 2	Cooperação	Conhecimento	Liderança	Sociograma
Dan Brown	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Julio Verne	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Mary Shelley	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Grupo 3	Cooperação	Conhecimento	Liderança	Sociograma
Arthur Conan Doyle	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
J.R.R. Tolkien	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
J.K. Rowling	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Grupo 4	Cooperação	Conhecimento	Liderança	Sociograma
Erasmus de Rotterdam	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Robert Louis Stevenson	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Lewis Carroll	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Grupo 5	Cooperação	Conhecimento	Liderança	Sociograma
Michael Crichton	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
H. P. Lovecraft	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Terry Pratchett	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Grupo 6	Cooperação	Conhecimento	Liderança	Sociograma
Agatha Christie	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Oscar Wilde	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Daniel Defoe	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Grupo 7	Cooperação	Conhecimento	Liderança	Sociograma
Edgar Allan Poe	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Dante Alighieri	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Charles Baudelaire	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
OK				

Figura 5.9 Resultado para execução baseada no perfil dos alunos

Para a execução baseada em uma única abordagem, aplicando-se a fórmula  $3n(x+y)$  definida na seção 5.3.3, obtem-se um *fitness* máximo de valor igual a 21 já que não foram utilizados pesos na aplicação dos testes. Para a execução baseada nas duas abordagens, perfil e teste sociométrico, tem-se um *fitness* máximo de valor igual a 42. A tabela 5.7 mostra os valores de *fitness* alcançados, para a turma dada, em cada uma das três abordagens.

Abordagem	<i>Fitness</i>	<i>Max Fitness</i>
Perfil	17	21
Teste Sociométrico	10	21
Ambas	25	42

Tabela 5.7 *Fitness* das soluções por abordagem



Grupos				
Grupo	Cooperação	Conhecimento	Liderança	Sociograma
<b>Grupo 1</b>				
Dan Brown	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Daniel Defoe	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Dante Alighieri	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
<b>Grupo 2</b>				
George Orwell	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Charles Baudelaire	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Mary Shelley	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
<b>Grupo 3</b>				
Arthur Conan Doyle	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
J.R.R. Tolkien	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Aldous Huxley	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
<b>Grupo 4</b>				
Erasmus de Rotterdam	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Robert Louis Stevenson	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Julio Verne	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
<b>Grupo 5</b>				
Michael Crichton	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Douglas Adams	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Terry Pratchett	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
<b>Grupo 6</b>				
Oscar Wilde	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Lewis Carroll	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
J.K. Rowling	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
<b>Grupo 7</b>				
H. P. Lovecraft	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Agatha Christie	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
Edgar Allan Poe	☆☆☆☆☆	☆☆☆☆☆	☆☆☆☆☆	
OK				

Figura 5.10 Resultado para execução baseada no teste sociométrico

Analisando-se os valores apresentados na tabela 5.7 nota-se que em nenhum dos casos o valor máximo de *fitness* foi alcançado, mas deve-se fazer a seguinte consideração: os valores máximos para o *fitness* ali apresentados não levam em consideração os critérios definidos para a formação dos grupos, no caso a definição dos grupo ideal e médio e as preferências dos alunos apontadas pelo teste sociométrico. O valor máximo mostrado na tabela se dá somente em função do número de grupos a se formar, ou seja, é a aplicação da fórmula  $3n(x+y)$ . Portanto, para uma real constatação da otimização dos grupos deve-se considerar os critérios estabelecidos. Se analisados os resultados apresentados nas figuras 5.10

a 5.12 juntamente com os critérios definidos chega-se a conclusão de que os grupos formados alcançam o maior valor possível de *fitness*.

Grupos				
Grupo	Cooperação	Conhecimento	Liderança	Sociograma
<b>Grupo 1</b>	★★★★☆	★★★★☆	★★★★☆	
Julio Verne	★★★★☆	★★★★☆	★★★★☆	
Douglas Adams	★★★★☆	★★★★☆	★★★★☆	
Dan Brown	★★★★☆	★★★★☆	★★★★☆	
<b>Grupo 2</b>	★★★★☆	★★★★☆	★★★★☆	
Dante Alighieri	★★★★☆	★★★★☆	★★★★☆	
Charles Baudelaire	★★★★☆	★★★★☆	★★★★☆	
Mary Shelley	★★★★☆	★★★★☆	★★★★☆	
<b>Grupo 3</b>	★★★★☆	★★★★☆	★★★★☆	
Arthur Conan Doyle	★★★★☆	★★★★☆	★★★★☆	
J.R.R. Tolkien	★★★★☆	★★★★☆	★★★★☆	
J.K. Rowling	★★★★☆	★★★★☆	★★★★☆	
<b>Grupo 4</b>	★★★★☆	★★★★☆	★★★★☆	
Aldous Huxley	★★★★☆	★★★★☆	★★★★☆	
Robert Louis Stevenson	★★★★☆	★★★★☆	★★★★☆	
Lewis Carroll	★★★★☆	★★★★☆	★★★★☆	
<b>Grupo 5</b>	★★★★☆	★★★★☆	★★★★☆	
H. P. Lovecraft	★★★★☆	★★★★☆	★★★★☆	
Michael Crichton	★★★★☆	★★★★☆	★★★★☆	
Terry Pratchett	★★★★☆	★★★★☆	★★★★☆	
<b>Grupo 6</b>	★★★★☆	★★★★☆	★★★★☆	
Agatha Christie	★★★★☆	★★★★☆	★★★★☆	
Edgar Allan Poe	★★★★☆	★★★★☆	★★★★☆	
Daniel Defoe	★★★★☆	★★★★☆	★★★★☆	
<b>Grupo 7</b>	★★★★☆	★★★★☆	★★★★☆	
Oscar Wilde	★★★★☆	★★★★☆	★★★★☆	
Erasmus de Rotterdam	★★★★☆	★★★★☆	★★★★☆	
George Orwell	★★★★☆	★★★★☆	★★★★☆	
OK				

Figura 5.11 Resultado da execução utilizando as duas abordagens

## 6 CONCLUSÃO

---

---

Esta dissertação envolveu o estudo sobre os algoritmos genéticos para formação de grupos para aprendizagem cooperativa apoiada por computador. Foi apresentado primeiramente o estado da arte da ACAC tendo sido evidenciada a importância da distribuição dos grupos para a sua aplicação. Foi realizado, ainda, um estudo apresentando alguns trabalhos que propõem métodos para a formação de grupos e ao final deste foi apresentado o método proposto neste trabalho como sendo uma tarefa de otimização.

Definido o problema como de otimização, o passo seguinte foi o de definir uma ferramenta de otimização a ser aplicada para a sua resolução. A opção pelos AGs deveu-se a um trabalho de revisão sobre os métodos de otimização onde foram consideradas as características mais marcantes e os tipos de problemas onde cada família de métodos é mais eficiente. Devido à natureza do problema verificaram-se várias vantagens no uso dos AGs.

Tendo-se escolhido o método de otimização, passou-se a estudá-lo em mais detalhes e ao fim dessa pesquisa foi modelado o algoritmo genético a ser aplicado na formação de grupos para ACAC. A partir dessa modelagem foi desenvolvida uma ferramenta para a formação de grupos no ambiente *NetClass*.

Além das contribuições inerentes aos trabalhos de dissertação, como a análise do estado da arte das áreas que envolvem os temas da pesquisa, destacam-se como as principais contribuições deste trabalho a concepção de um método para a formação eficiente de grupos de aprendizagem cooperativa, a concepção de uma arquitetura funcional para a aplicação deste método e sua implementação.

O trabalho apresentado não teve a pretensão de realizar uma comparação entre os algoritmos genéticos e outras possíveis técnicas de otimização. Os AGs buscam uma solução ótima partindo de várias soluções simultaneamente, enquanto que os métodos clássicos de



otimização efetuam a busca voltados a uma única solução ótima dentre todas as possíveis soluções. Além disso, por utilizarem os conceitos da evolução e da genética, os algoritmos genéticos são capazes de direcionar suas buscas de uma maneira mais eficiente do que um processo aleatório. Os AGs mostraram-se bastante eficientes na otimização do problema proposto, alcançando ótimos resultados a um custo computacional baixo.

Há de se considerar que os resultados alcançados baseiam-se nos critérios estabelecidos para a formação dos grupos, ou seja, nos fatores pedagógicos aplicados. O sucesso efetivo do processo de ensino-aprendizagem aplicado aos grupos depende de uma correta definição desses fatores.

O *NetClass* encontra-se ainda em fase de desenvolvimento, onde se inclui aí o Agente de Modelagem do Aprendiz responsável por inferir o modelo do aprendiz de onde se pode extrair informações a respeito do perfil dos aprendizes, elemento fundamental para a aplicação de um dos fatores pedagógicos propostos. Dessa forma simulou-se a inferência do perfil dos aprendizes para a aplicação da ferramenta de formação de grupos.

Sugere-se para trabalhos futuros a formulação de mais fatores pedagógicos visando abranger um maior número de critérios pré-formulados a serem adotados na formação otimizada de grupos. Isso acarreta em um nível maior de reutilização por parte de sistemas ACAC que não precisariam sequer preocuparem-se em estabelecer esses critérios. Um outro ponto importante é verificar a eficiência dos grupos formados, analisando-se os resultados alcançados com a aplicação dos fatores pedagógicos aplicados. Um terceiro aspecto a se considerar para futuros trabalhos é a utilização de técnicas ainda mais robustas no que diz respeito aos algoritmos genéticos.

Tem-se a convicção que os objetivos propostos foram alcançados. Espera-se que o trabalho aqui desenvolvido seja de grande valia não só para o *NetClass* mas também para outros pesquisadores que trabalhem em áreas afins.

## REFERÊNCIAS

BACK, T.; FOGEL, D. B.; MICHALEWICZ, Z. (Ed.) Evolutionary computation 2: advanced algorithms and operators. Bristol, UK: Institute of Physics Publishing, 2000.

BAKER, J. E. An analysis of the effects of selection in genetic algorithms. Phd thesis, Graduate School of Vanderbilt University, Nashville, Tennessee, 1989.

BANZHAF, W.; NORDIN, P.; KELLER, R. E.; FRANCONI, F.D. Genetic programming: an introduction on the automatic evolution of computer programs and its applications. San Francisco, CA: Morgan Kaufmann, 1998.

CASANOVA, M. A. La sociometría en el aula, La Muralla, Madrid, 1991.

CASTRO, R. E. de. Otimização de estruturas com multi-objetivos via algoritmos genéticos. Rio de Janeiro, 2001. 206 f. Tese (Doutorado em Ciências em Engenharia Civil) – COPPE, Universidade Federal do Rio de Janeiro.

CUNHA, L. M.; FUKS, H.; LUCENA, C.J.P. Formação de Grupos no Ambiente AulaNet Utilizando Agentes de Software, XIII Simpósio Brasileiro de Informática na Educação - SBIE 2002, 12 a 14 de Novembro de 2002, São Leopoldo – RS.

DAIDA, J. M. Challenges with verification, repeatability and meaningful comparisons in genetic programming. Proceedings of the 4<sup>th</sup> Annual Conference in Genetic Programming (GECCO'99). ISBN 1558606114. pp. 1069-1076. Morgan Kauffmann, 1999.

DEB, K. Encoding and decoding functions. In: BACK, T.; FOGEL, D. B.; MICHALEWICZ, Z. (Ed.) Evolutionary computation 2: advanced algorithms and operators. Bristol, UK: IOP, 2000. p. 4-11.

DILLENBOURG, P. What is Collaborative Learning Anyway? In: Collaborative Learning: Cognitive and Computational Approaches. Reino Unido: Elsevier Science Ltda, 1999, p. 1-14.

ELLIS, S.; WHALEN, S. F. Cooperative Learning: getting started. Scholastic, New York, 1990.

ESMA, A.; FRASSON C.; LIBERT G. Towards new learning strategies in intelligent tutoring systems. SBIA-95. Campinas: Springer-Verlag Edition. October 1995.

FERREIRA, J. S. Concepção de um ambiente multi-agentes de ensino inteligente integrando o paradigma de aprendizagem cooperativa. 1998. Dissertação (Mestrado em

Engenharia da Eletricidade) – Programa de Pós-Graduação em Engenharia da Eletricidade - Universidade Federal do Maranhão, São Luís.

GOLDBARG, M. C.; LUNA, H. P. L. Otimização combinatória e programação linear: modelos e algoritmos. Rio de Janeiro: Campus, 2000.

GOLDBERG, D. E. Genetic algorithms in search, optimization and machine learning. Reading, MA: Addison-Wesley, 1989.

IKEDA, M., Go, S.; MIZOGUCHI, R. (1997) Opportunistic Group Formation, Proc. of AI-ED 97, August, Japan, pp.166-174.

JOHNSON, D. W.; JOHNSON, R. T. Learning together and alone, cooperative, competitive and individualistic learning. Allyn and Bacon, Paramount. 1994.

KNOSALA, R.; WAL, T. A production scheduling problem using genetic algorithm. Journal of Materials Processing Technology, v. 109. p. 90-95, 2001.

KOZA, J. R. Genetic programming: on the programming of computers by means of natural selection. Cambridge, MA: MIT, 1992.

KREIJNS, K.; KIRSCHNER, P. A.; JOCHEMS, W. The Sociability of Computer-Supported Collaborative Learning Environments. Educational Technology & Society. Vol. 5, No. 1, 2002.

LABIDI, S.; FERREIRA, J. S. Technology-assisted instruction applied to cooperative learning: the SHIECC project. In: PROCEEDINGS OF THE IEEE INTERNATIONAL CONFERENCE FRONTIERS IN EDUCATION (FIE'98). Tempe, Arizona, november 4-7, 1998.

LIMA, M; LABIDI, S; FILHO, O. C. B.; FONSECA, L. C. C. Aprendizagem Cooperativa e o Problema da Formação de Grupos. RENOTE - Revista Novas Tecnologias na Educação, Porto Alegre, v. 3, n. 1, 2005.

LIMA, M; LABIDI, S; Modelagem do Aprendiz Cooperativo em uma Sessão de Aprendizagem no Ambiente Netclass: Uma Especificação para a Formação Inicial de Grupos Cooperativos. 4a. Conferencia Iberoamericana en Sistemas, Cibernética e Informática (CISCI 2005), Orlando, Florida, USA. 2005.

LUCENA, C. J. P.; FUKS, H.; MILIDIU, R.; LAUFER, C.; BLOIS, M.; CHOREN, R., TORRES, V.; FERRAZ, F.; ROBICHEZ, G.; DAFLON, L. AulaNet: Ajudando professores a fazerem o seu dever de casa. In: Anais do XXVI Seminário Integrado de

Software e Hardware, págs. 105-117. Rio de Janeiro, RJ. 1999.

MACEDO, A. A.; PIMENTEL, M. da G. C.; FORTES, R. P. de M. Studyconf: infraestrutura de suporte ao aprendizado cooperativo na WWW. *Revista Brasileira de Informática na Educação*. Florianópolis, nº 5, p. 77 - 99. Setembro, 1999.

MINICUCCI, A. Dinâmica de Grupo: teorias e sistemas. São Paulo: Atlas, 1997.

MITCHELL, M. An introduction to genetic algorithms. Cambridge, MA: MIT, 1996.

MORAES, M. C. O Paradigma Educacional Emergente. 1996. 520f. Tese - Programa de Educação-supervisão e Currículo, PUC, São Paulo.

MORENO, J. L. Fundamentos de la sociometría, Paidós, Buenos Aires, 1954.

MORENO, J. L. Who shall survive? Foundations of sociometry, group psychotherapy and sociodrama (Student edition). Roanoke, VA: Royal, 1953.

MUHLENBEIN, H.; SCHIERKAMP-VOOSEN, D. Predictive models for the breeder genetic algorithms. *Evolutionary Computation Vol 1*. pp 25-49. MIT Press, 1993.

MUEHLENBROCK, M. Learning Group Formation Based on Learner Profile and Context. *International Journal on E-Learning*. 5 (1), pp. 19-24. Chesapeake, VA: AACE, 2006.

OVEREJO, A. Psicología social de la educación, Herder, Barcelona, 1988.

PASCAL, L.; MARTIAL, V.; PATRICK, B. Cooperation between humans and a pedagogical assistant in a learning environment. In: **INTERNATIONAL CONFERENCE ON THE DESIGN OF COLLABORATIVE KNOWLEDGE-BASED SYSTEMS (COOP'96)**. Juan-les-Pins, France, 1996.

REEVES, C. R. A genetic algorithm for flowshop sequencing. *Computers and Operations Research*, n. 22, p. 5-13, 1995.

ROGERS, A.; PRUGEL-BENNETT, A. Modelling the dynamics of a steady state genetic algorithm. In: BANZHAF, W.; REEVES, C. (Ed.). *Foundations of Genetic Algorithms 5*. San Francisco, CA: Morgan Kaufmann, 1999. p. 57-68.

RUMBAUGH, J.; BOOCH, G.; JACOBSON, I. UML – Guia do Usuário. Rio de Janeiro:

Editora Campus Ltda, 2000.

SILVEIRA, S. R.; BARONE, D. A. C. Formação de Grupos Colaborativos em Cursos a Distância via Web utilizando Técnicas de Inteligência Artificial. *RENOTE - Revista Novas Tecnologias na Educação*, Porto Alegre, v. 3, n. 2, 2005.

SOUZA, D. O. de, Algoritmos Genéticos Aplicados ao Planejamento do Transporte Principal de Madeira. Curitiba, 2004. 184f. Dissertação (Mestrado em Ciências Florestais) - Setor de Ciências Agrárias da Universidade Federal do Paraná.

STARKWEATHER, T.; McDANIEL, S.; MATHIAS, K.; WHITLEY, D.; WHITLEY, C. A comparison of genetic sequencing operators. In: *INTERNATIONAL CONFERENCE ON GENETIC ALGORITHMS*, 1991. Los Altos, CA: Morgan Kaufmann, 1991. p. 69-76.

STEINBERG, E. *Teaching Computers to Teach. Computer-based Education Research*. Laboratory and College of Education University of Illinois at Urbana- Champaign. Hillsdale, N. J.: Lawrence Erlbaum Associates Publishers, 1984.

TUSON, A. L. Adapting operator probabilities in genetic algorithms. Edimburgh, 1995. 137 f. Dissertação (Mestrado em Ciências) - Department of Artificial Intelligence, University of Edinburgh.

VASCONCELOS, J. A.; TAKAHASHI, R. H. C.; SALDANHA, R. R. Improvements in Genetic Algorithms. *IEEE – Transactions on Magnetics*. v. 37. n. 5. pp. 3414-3417. 2001.

VAVAK, F; FOGARTY, T. C.; Comparison of steady-state and generational genetic algorithms for use in nonstationary environments. In: *INTERNATIONAL CONFERENCE ON EVOLUTIONARY COMPUTATION*, 1996. Proceedings... New York, NY; IEEE, 1996. p. 192-195.

VIANA, G. V. R. Meta-heurísticas e programação paralela em otimização combinatória. Fortaleza: EUFC, 1998.

WHITLEY, D. L.; YOO, N. W. Modeling simple genetic algorithms for permutation problems. In: WHITLEY, D. L.; VOSE, M. D. (Ed.). *Foundations of genetic algorithms 3*. San Mateo, CA: Morgan Kaufmann, 1995. p. 163-184.

WOLZ, U.; PALME, J. Computer-mediated communication in collaborative educational settings. *ACM SIGCUE*, 1997.