

Algoritmos Evolutivos para o Problema de Clusterização de Grafos Orientados: Desenvolvimento e Análise Experimental

Carlos Rodrigo Dias

Dissertação de Mestrado submetida ao Programa de Pós-Graduação em Computação Aplicada e Automação da Universidade Federal Fluminense como requisito parcial para a obtenção do título de Mestre. Área de concentração: Otimização e Inteligência Artificial.

Orientador: Luiz Satoru Ochi

Niterói, Julho de 2004.

*À minha esposa, Karla,
e aos meus pais, Raimundo e Eunice*

Agradecimentos

Como eu acredito que na vida não se constrói nada sozinho, neste momento eu gostaria de compartilhar a minha felicidade em estar concluindo este trabalho com várias pessoas que foram fundamentais para que isto fosse possível.

Inicio meus agradecimentos com Aquele que está presente em todos os momentos da minha vida, guiando as minhas atitudes e atuando como fonte de inspiração: Deus.

À minha amada esposa, Karla Pacheco Dias, que esteve ao meu lado durante todas as etapas do curso de mestrado, com compreensão, apoio integral e me incentivando em todos os momentos difíceis. Agradeço também pelo seu sacrifício ao deixar de realizar coisas que gostaria para que eu pudesse me dedicar aos estudos. Eu também agradeço a Deus por eu ser tão afortunado em ter acertado na escolha da mulher com que eu quero viver por toda a minha vida.

Aos meus pais, Raimundo Rodrigues Dias e Eunice Teresa Lopes Dias, que sempre me apoiaram e torceram por mim, incondicionalmente. Sem o amor e o apoio deles nenhuma conquista da minha vida seria possível. Aproveito para dizer que eles são muito importantes para mim e que eu os amo muito. Agradeço também aos meus irmãos, Gustavo Eugênio Dias e Natália Cristina Dias, pela torcida e pelos momentos de distração que passamos juntos.

Aos meus avós, Vô Dias e Vó Cotinha, agradeço pelo amor e carinho dedicados a mim e gostaria que me perdoassem pelos momentos que não pudemos passar juntos porque eu estava com compromissos relacionados ao mestrado. Aos meus avós, Vô Antenor (*in memoriam*) e Vó Maria (*in memoriam*), que sempre estiveram próximos a mim, mesmo não

sendo uma convivência física. Agradeço também a todos os meus tios e tias, primos e primas. Um agradecimento especial à tia Ilva, que me hospedou com muito carinho durante um bom período do mestrado. Enfim, agradeço a toda a minha família que eu prezo tanto e que é o alicerce da minha vida.

Aproveito esta oportunidade para realizar um forte agradecimento aos meus amigos do DEINFO (Departamento de Informática da Prefeitura de Juiz de Fora) que me ajudaram muito no período do mestrado, tendo em vista todos os dias em que estive ausente para que pudesse me dedicar aos estudos: considerem esta conquista como sendo também de vocês. Um agradecimento muito especial ao Bessa, líder da nossa equipe, que se mostrou um grande amigo e incentivador em todas as situações ao longo do meu período de curso. Estendo os meus agradecimentos também aos demais amigos da Prefeitura de Juiz de Fora.

Agradeço a todos os meus amigos pessoais que, de alguma forma, me incentivaram e contribuíram para esta vitória. Destaco nesta lista de amigos a Eugênia e o Marco Antônio, que me incentivaram a iniciar o mestrado, e todos os meus companheiros de viagem (Juiz de Fora - Niterói - Juiz de Fora).

Durante o curso, conheci várias pessoas na UFF que de alguma forma marcaram este período da minha vida e gostaria de agradecer a todas elas pelo companheirismo e cumplicidade. Um agradecimento especial para Adriana, Cris, Eduardo Fontana, Eduardo Corrêa, Eyder, José, Helder e Stênio. Agradeço também aos professores e funcionários da Pós-Graduação da UFF, Alexandre Plastino, Carlos Martinhon, Cristina Boeres, Simone, Lúcia Drummond, Ângela e Izabela, pela competência e disposição em contribuir para a excelência no aprendizado dos alunos da Pós-Graduação.

Agradeço à Prefeitura de Juiz de Fora e à Faculdade Metodista Granbery pelo apoio financeiro e pelo tempo disponibilizado para as viagens necessárias para a realização do mestrado e para os eventos do qual participei.

Ao meu professor e orientador, Luiz Satoru Ochi, que soube me motivar e orientar de forma única, e necessária, para que este trabalho fosse de encontro aos meus objetivos pessoais, acadêmicos e profissionais. Muito obrigado Satoru.

Resumo

Neste trabalho são apresentadas propostas para aperfeiçoamento de algoritmos genéticos (AGs). Para isto, um algoritmo genético tradicional (AGT), encontrado na literatura para a solução do problema de clusterização automática de vértices de um grafo orientado não ponderado, é utilizado como base para as adaptações propostas.

As propostas de aperfeiçoamento do AGT correspondem à realização de alterações em alguns dos seus parâmetros e à inserção de novos procedimentos ao algoritmo. Para cada proposta são apresentados os resultados experimentais obtidos a partir da sua implementação e execução, utilizando grafos orientados, não ponderados, contendo diferentes quantidades de vértices e arcos.

São realizados experimentos computacionais, utilizando as propostas de forma individual ou através de combinações entre elas. Os resultados mostram que os algoritmos aqui propostos melhoram consideravelmente o desempenho do AGT em relação à qualidade das soluções obtidas, exigindo, em média, tempos computacionais similares, ou menores, que os do algoritmo da literatura.

Abstract

In this work, new proposals to improve genetic algorithms are presented. In order to do that, a classic genetic algorithm (AGT), which was obtained from the literature for the solution of automatic clustering problem applied to a non-weighted directed graph, is used as the basis for the proposed adaptations.

The proposals for the improvement of AGT consider the accomplishment of modifications in its parameters and the insertion of new procedures to the algorithm. For each proposal, experimental results are presented, following its implementation and execution by using non-weighted directed graphs, which contain different amounts of nodes and edges.

Computational experiments were performed by using the proposals in an individual way or combinations of them, and show that the algorithms here proposed considerably improve the performance of AGT as regards the quality of the obtained solutions, demanding, on the average, similar or even smaller running times in comparison to those of the algorithm in the literature.

Glossário

AE	: Algoritmo Evolutivo (<i>Evolutive Algorithm</i>);
AG	: Algoritmo Genético (<i>Genetic Algorithm</i>);
AEH	: Algoritmo Evolutivo Híbrido;
AGT	: Algoritmo Genético Tradicional;
AM	: Algoritmo Memético (<i>Memetic Algorithm</i>);
AGC	: Algoritmo Genético Construtivo;
GGA	: <i>Grouping Genetic Algorithm</i> (Algoritmo Genético de Agrupamento);
MDG	: <i>Module Dependency Graph</i> (Grafo de Dependências de Módulos);
MQ	: <i>Modularization Quality</i> (Qualidade da Modularização);
PGG	: Programa Gerador de Grafos.

Sumário

Resumo	v
Abstract	vi
Glossário.....	vii
Lista de Figuras.....	x
Lista de Tabelas	xvi
Capítulo 1 - Introdução	1
Capítulo 2 - Algoritmos Evolutivos	3
2.1. Algoritmos Genéticos – Versão Clássica	5
2.1.1. Representação dos Indivíduos	8
2.1.2. Função de Aptidão.....	9
2.1.3. Seleção e Reprodução	9
2.1.4. Operadores Genéticos	11
2.1.5. Outros Parâmetros	14
2.2. Algoritmos Genéticos – Versões Aperfeiçoadas.....	14
2.3. Conclusão.....	16
Capítulo 3 - Clusterização de Grafos Orientados com o Uso de Algoritmo Genético	17
3.1. O Problema de Clusterização	18
3.1.1. Métodos Utilizados para Clusterização	20
3.1.2. Clusterização de Grafos	22
3.2. Clusterização com Algoritmos Genéticos	24
3.2.1. Representação dos Indivíduos	25
3.2.2. AG para Clusterização Automática de Grafos Orientados	30
3.3. Resultados Experimentais.....	36

3.3.1. Programa Gerador de Grafos	36
3.3.2. Resultados Obtidos com o Algoritmo Genético Tradicional	38
3.4. Conclusão.....	46
Capítulo 4 - Desenvolvimento e Análise Experimental de Novas Versões do AGT.....	48
4.1. Estudo do operador mutação.....	50
4.1.1. Alteração da Taxa de Aplicação do Operador Mutação	50
4.1.2. Calibração do Operador Mutação.....	61
4.2. Novo Processo de Seleção de Indivíduos	63
4.3. Estudo do Operador de Cruzamento	66
4.4. Procedimento de Busca Local.....	70
4.5. Diversificação da população	75
4.6. Comparação entre as Diferentes Propostas.....	87
4.7. Conclusão.....	92
Capítulo 5 - Versões Híbridas do AGT	93
5.1. Resultados Computacionais	94
5.2. Conclusão.....	101
Capítulo 6 - Conclusão e Trabalhos Futuros	103
Publicações Associadas	106
Referências Bibliográficas	107

Lista de Figuras

2.1	Algoritmo Genético tradicional	7
2.2	Exemplo de aplicação do operador de cruzamento de um ponto	12
2.3	Exemplo de aplicação do operador de cruzamento de múltiplos pontos	12
2.4	Exemplo de aplicação do operador de cruzamento uniforme	12
2.5	Exemplo de aplicação do operador mutação.....	13
3.1	Exemplo de árvore de <i>clusters</i> na clusterização hierárquica	21
3.2	Exemplo de bi-particionamento balanceado de grafos	23
3.3	Exemplos de clusterização de um grafo orientado com 8 vértices.....	24
3.4	Exemplos das diferentes formas de representação de indivíduos em AGs para problemas de clusterização	27
3.5	Exemplo de simetria na representação <i>group-number</i>	28
3.6	Exemplo de repercussão da simetria no operador de cruzamento.....	28
3.7	Exemplo de representação de indivíduos no GGA.....	29
3.8	Clusterização do MDG para o Sistema Operacional Mini-Tunis [13]	30
3.9	Exemplo de codificação <i>group-number</i> para clusterização de um grafo orientado	31
3.10	Exemplos de valores para a função <i>MQ</i> associada a diferentes clusterizações	34
3.11	Exemplo de aplicação do operador de cruzamento no AGT.....	35
3.12	Exemplo de aplicação do operador mutação no AGT	35
3.13	Exemplo de formação de <i>cluster</i> no PGG, com $A = 0,5000$ e topologia anel	37
3.14	Exemplo de formação de <i>cluster</i> no PGG, com $A = 0,5000$ e topologia estrela.....	38
3.15	Gráfico comparativo dos resultados do AGT para cada grafo com topologia anel.....	41

3.16	Gráfico comparativo dos resultados do AGT para cada grafo com topologia estrela	41
3.17	Percentual do valor de MQ médio obtido nas execuções em comparação com o valor de MQ da solução <i>best</i> para cada grafo com topologia anel	41
3.18	Percentual do valor de MQ médio obtido nas execuções em comparação com o valor de MQ da solução <i>best</i> para cada grafo com topologia estrela	42
3.19	Tempo médio de execução do AGT para cada grafo com topologia anel.....	43
3.20	Tempo médio de execução do AGT para cada grafo com topologia estrela	43
3.21	Comparativo entre os tempos médios de execução do AGT e AGTa.....	44
3.22	Detalhes dos gráficos apresentados na Figura 3.21	44
3.23	Comparativo da qualidade das melhores soluções obtidas pelo AGT e AGTa em relação ao MQ <i>best</i> para os grafos com topologia anel	45
3.24	Comparativo da qualidade das melhores soluções obtidas pelo AGT e AGTa em relação ao MQ <i>best</i> para os grafos com topologia estrela.....	45
3.25	Acompanhamento do melhor indivíduo obtido a cada geração do AGT para o grafo A60	46
3.26	Acompanhamento do melhor indivíduo obtido a cada geração do AGT para o grafo A100	46
4.1	AGT aplicado ao grafo A60, com diferentes valores para a taxa de mutação e MQ -alvo = 0,5.....	52
4.2	Acompanhamento da evolução do melhor indivíduo e do MQ médio dos indivíduos ao longo das gerações, para o grafo A60 e valor para a taxa de mutação igual a 0,0240	53
4.3	Acompanhamento da evolução do melhor indivíduo e do MQ médio dos indivíduos ao longo das gerações, para o grafo A60 e valor para a taxa de mutação igual a 0,0160	53
4.4	Acompanhamento da evolução do melhor indivíduo e do MQ médio dos indivíduos ao longo das gerações, para o grafo A60 e valor para a taxa de mutação igual a 0,0040	53

4.5	Acompanhamento da evolução do melhor indivíduo e do MQ médio dos indivíduos ao longo das gerações, para o grafo A60 e valor para a taxa de mutação igual a 0,0002	54
4.6	Acompanhamento da evolução do melhor indivíduo ao longo de 20.000 gerações, para o grafo A60 e valor para a taxa de mutação igual a 0,0040	55
4.7	Acompanhamento da evolução do melhor indivíduo ao longo de 20.000 gerações, para o grafo A60 e valor para a taxa de mutação igual a 0,0002	56
4.8	AGT aplicado ao grafo A10, com diferentes valores para a taxa de mutação (MQ -alvo=0,5)	56
4.9	AGT aplicado ao grafo A20, com diferentes valores para a taxa de mutação (MQ -alvo=0,5)	57
4.10	AGT aplicado ao grafo A40, com diferentes valores para a taxa de mutação (MQ -alvo=0,5)	57
4.11	AGT aplicado ao grafo A80, com diferentes valores para a taxa de mutação (MQ -alvo=0,5)	58
4.12	AGT aplicado ao grafo A100, com diferentes valores para a taxa de mutação (MQ -alvo =0,5).....	58
4.13	Comparação de resultados obtidos pelo AGT e AGT+M(0,004) para grafos A10 a A100	60
4.14	Comparação de resultados obtidos pelo AGT e AGT+M(0,004) para grafos E10 a E100	60
4.15	Comparação de resultados obtidos pelo AGT e AGT+C para grafos A10 a A100.....	62
4.16	Comparação de resultados obtidos pelo AGT e AGT+C para grafos E10 a E100	62
4.17	Comparação de resultados obtidos pelo AGT, AGT+T(2) e AGT+T(4) para os grafos A10 a A100.....	64
4.18	Comparação de resultados obtidos pelo AGT, AGT+T(2) e AGT+T(4) para os grafos E10 a E100.....	64
4.19	Acompanhamento da evolução do melhor indivíduo ao longo de 2000 gerações do AGT+T(2) e AGT+T(4) utilizando o grafo A100.....	65
4.20	Algoritmo do operador de cruzamento do AGT+G.....	67
4.21	Exemplo de utilização do operador de cruzamento do AGT+G	68
4.22	Comparação de resultados obtidos pelo AGT e AGT+G para grafos A10 a A100	69

4.23	Comparação de resultados obtidos pelo AGT e AGT+G para grafos E10 a E100	69
4.24	Acompanhamento da evolução do melhor indivíduo ao longo de 2.000 gerações do AGT+G utilizando o grafo A100.....	70
4.25	Algoritmo de busca local aplicado a uma clusterização C_0	71
4.26	Comparação de resultados obtidos pelo AGT e AGT+B para grafos A10 a A100.....	72
4.27	Comparação de resultados obtidos pelo AGT e AGT+B para grafos E10 a E100	73
4.28	Comparação de resultados obtidos pelo AGT e AGT+B para grafos A160 a A1000.....	74
4.29	Comparação de resultados obtidos pelo AGT e AGT+B para grafos E160 a E1000....	74
4.30	Algoritmo de diversificação da população.....	75
4.31	Exemplo de aplicação do procedimento de diversificação da população.....	76
4.32	Comparação de resultados obtidos pelo AGT, AGT+D(10) e AGT+D(25) para grafos A10 a A100.....	77
4.33	Comparação de resultados obtidos pelo AGT, AGT+D(10) e AGT+D(25) para grafos E10 a E100.....	78
4.34	Acompanhamento da evolução do melhor indivíduo ao longo de 2.000 gerações do AGT+D(10), para o grafo A100.....	78
4.35	Acompanhamento da evolução do valor médio para a função de aptidão dos indivíduos de cada uma das 2.000 gerações do AGT+D(10), para o grafo A100	79
4.36	Detalhes das Figuras 4.34 e 4.35 referentes às evoluções do melhor indivíduo e do valor médio para a função de aptidão dos indivíduos entre as gerações 1.000 e 1.100.	79
4.37	Acompanhamento da evolução do melhor indivíduo ao longo de 2.000 gerações do AGT+D(10), para o grafo E100	79
4.38	Acompanhamento da evolução do valor médio para a função de aptidão dos indivíduos de cada uma das 2.000 gerações do AGT+D(10), para o grafo E100.....	80
4.39	Detalhes das Figuras 4.37 e 4.38 referentes às evoluções do melhor indivíduo e do valor médio para a função de aptidão dos indivíduos entre as gerações 1.000 e 1.100	80
4.40	Acompanhamento da evolução do melhor indivíduo ao longo de 2.000 gerações do AGT+D(25), para o grafo A100.....	80

4.41 Acompanhamento da evolução do valor médio para a função de aptidão dos indivíduos de cada uma das 2.000 gerações do AGT+D(25), para o grafo A100	81
4.42 Detalhes das Figuras 4.40 e 4.41 referentes às evoluções do melhor indivíduo e do valor médio para a função de aptidão dos indivíduos entre as gerações 1.000 e 1.100	81
4.43 Acompanhamento da evolução do melhor indivíduo ao longo de 2.000 gerações do AGT+D(25), para o grafo E100	81
4.44 Acompanhamento da evolução do valor médio para a função de aptidão dos indivíduos de cada uma das 2.000 gerações do AGT+D(25), para o grafo E100.....	82
4.45 Detalhes das Figuras 4.43 e 4.44 referentes às evoluções do melhor indivíduo e do valor médio para a função de aptidão dos indivíduos entre as gerações 1.000 e 1.100	82
4.46 Melhor indivíduo obtido pelas versões AGT+D(10) e AGT+D(25), com total de gerações padrão e com total de gerações igual a 32.000, para os grafos A20 a A100.....	84
4.47 Melhor indivíduo obtido pelas versões AGT+D(10) e AGT+D(25), com total de gerações padrão e com total de gerações igual a 32.000, para os grafos E20 a E100	85
4.48 Acompanhamento da evolução do melhor indivíduo ao longo de 32.000 gerações do AGT+D(10), para o grafo A100.....	86
4.49 Acompanhamento da evolução do melhor indivíduo ao longo de 32.000 gerações do AGT+D(25), para o grafo A100.....	86
4.50 Acompanhamento da evolução do melhor indivíduo ao longo de 32.000 gerações do AGT+D(10), para o grafo E100	86
4.51 Acompanhamento da evolução do melhor indivíduo ao longo de 32.000 gerações do AGT+D(25), para o grafo E100	87
4.52 Comparação dos melhores indivíduos obtidos pelo AGT e pelas versões propostas, para os grafos A10 a A100.....	88
4.53 Comparação dos melhores indivíduos obtidos pelo AGT e pelas versões propostas, para os grafos E10 a E100.....	89
4.54 Comparação dos valores de MQ médio referentes às 10 execuções do AGT e de cada versão proposta, para os grafos A10 a A100.....	90

4.55	Comparação dos valores de MQ médio referentes às 10 execuções do AGT e de cada versão proposta, para os grafos E10 a E100	91
------	---	----

Lista de Tabelas

3.1	Grafos utilizados nos experimentos com o AGT	39
3.2	Parâmetros de execução dos experimentos com o AGT.....	40
3.3	Resultados obtidos para os experimentos com o AGT	40
4.1	Grafos utilizados nos experimentos com as adaptações propostas	50
4.2	Comparação entre as soluções obtidas para diferentes versões do AGT com o grafo A60 ($MQ_{best} = 0,7402$)	55
4.3	Valores de MQ para soluções obtidas com as versões do AGT e AGT+M(0,004)	59
4.4	Valores de MQ para soluções obtidas com as versões do AGT e AGT+C	61
4.5	Valores de MQ para soluções obtidas com as versões do AGT, AGT+T(2) e AGT+T(4)	63
4.6	Valor médio de MQ ao longo das gerações e desvio padrão para as versões com e sem torneio utilizando o grafo A100	65
4.7	Valores de MQ para soluções obtidas com as versões do AGT e AGT+G	68
4.8	Valor médio de MQ e desvio padrão para versões AGT e AGT+G para grafo A100.....	70
4.9	Valores de MQ para soluções obtidas com as versões do AGT e AGT+B.....	72
4.10	Valores de MQ para soluções obtidas com as versões do AGT original e AGT+B para instâncias grandes	73
4.11	Valores de MQ para soluções obtidas com as versões do AGT original, AGT+D(10) e AGT+D(25).....	77

4.12	<i>MQ</i> médio dos melhores indivíduos obtidos e valor médio das iterações em que o melhor indivíduo foi obtido nas 10 execuções das versões AGT+D(10) e AGT+D(25).....	83
4.13	<i>MQ</i> dos melhores indivíduos obtidos e as gerações em que o melhor indivíduo foi obtido para uma execução das versões AGT+D(10) e AGT+D(25), com 32.000 gerações.....	84
4.14	Tempo de execução utilizado pelas versões AGT+D(10) e AGT+(25) com número de gerações padrão e com 32.000 gerações	85
4.15	Valores de <i>MQ</i> para os melhores indivíduos obtidos com o AGT original, e com cada uma das alterações propostas	88
4.16	Valores de <i>MQ</i> médios obtidos com o AGT original, e com cada uma das alterações propostas	91
4.17	Tempo médio de execução (em segundos) utilizado para cada versão aplicada a cada grafo	92
5.1	Resultados dos testes realizados com as versões híbridas, para os grafos de 10 a 100 vértices	94
5.2	Versões híbridas mais eficientes	96
5.3	Valores médios para a função <i>MQ</i> dos melhores indivíduos obtidos após a conclusão das 10 execuções de cada uma das melhores versões híbridas	96
5.4	Número médio das iterações em que a solução <i>best</i> foi obtida para cada versão híbrida	98
5.5	Tempo médio para a execução completa de cada versão híbrida (em segundos)	99
5.6	Valores médios de <i>MQ</i> para as soluções obtidas com as versões híbridas do AGT para grafos com 160 e 200 vértices	100
5.7	Tempo médio de execução completa para cada versão aplicada a cada grafo com 160, 200, 500 e 1000 vértices (no formato horas:minutos:segundos)	101

Capítulo 1

Introdução

Os Algoritmos Evolutivos (AEs), que utilizam mecanismos inspirados na Teoria da Evolução de Darwin e nos princípios da herança genética de Mendel, vêm sendo cada vez mais utilizados para a solução de problemas de otimização combinatória. Dentre os diferentes grupos de AEs encontrados na literatura, destacam-se os Algoritmos Genéticos (AGs), que evoluem uma população de soluções ao longo de várias iterações. Nos AGs, a investigação do espaço de busca do problema é realizada através de operadores aplicados sobre as soluções mantidas a cada iteração.

Um dos problemas de otimização que têm sido resolvidos de forma eficiente via AEs é o *problema de clusterização*. O problema de clusterização consiste na obtenção de um agrupamento (clusterização) de elementos de um conjunto em subconjuntos disjuntos, de forma que sejam mantidos em um mesmo subconjunto os elementos com maior similaridade entre si e que seja minimizada a similaridade entre elementos de subconjuntos diferentes. A modelagem de um AG para este problema deve ser realizada de forma a tratar corretamente os atributos a serem considerados na medida de similaridade utilizada.

A partir da verificação do desempenho limitado de um algoritmo genético disponível na literatura para a solução do problema de clusterização de grafo orientado não ponderado (grafo sem peso nos arcos ou com peso constante), apresentado por Doval, Mancoridis e Mitchell [13], surgiu o desafio: como melhorar o desempenho deste AG, no que diz respeito à obtenção de soluções de qualidade, sem que haja um aumento sensível do tempo computacional necessário? Esta questão motivou o desenvolvimento deste trabalho,

que se propõe a melhorar o desempenho deste AG da literatura através de alterações nos seus parâmetros e na inserção de novos procedimentos.

A principal contribuição deste trabalho refere-se, portanto, ao estudo experimental de diferentes propostas desenvolvidas para o aperfeiçoamento de um AG para o problema de clusterização, e que podem ser estendidas para outras aplicações. Desta forma a análise experimental de todas as propostas deste trabalho permite a obtenção de conclusões importantes sobre a influência de diferentes características de um AG na solução de problemas de otimização.

Esta dissertação está organizada em seis capítulos. No Capítulo 2 são apresentadas as principais características de um AE, com ênfase nos AGs, através de uma revisão bibliográfica sobre a utilização destas técnicas computacionais para a solução de problemas de otimização. No Capítulo 3 são tratados o problema de clusterização genérico e o problema de clusterização de grafos, sendo discutidas propostas de utilização dos AGs na solução destas classes de problemas. No Capítulo 4 são apresentadas as propostas de alterações em um AG para a clusterização de grafos orientados não ponderados, com o intuito de melhorar o seu desempenho, e é realizada uma análise experimental de cada proposta utilizada separadamente. Uma análise experimental das combinações das propostas apresentadas no Capítulo 4 é realizada no Capítulo 5. A conclusão deste trabalho é tratada no Capítulo 6.

Capítulo 2

Algoritmos Evolutivos

A expressão “Algoritmos Evolutivos” (ou a sua variação, “Algoritmos Evolucionários”) corresponde à classe de algoritmos para a solução de problemas de otimização que utilizam modelos computacionais baseados na teoria da evolução das espécies, proposta por Charles Darwin, e nos princípios básicos da herança genética, descritos por Gregor Mendel.

A partir de muitas evidências colhidas em suas viagens a bordo do navio Beagle, Charles Darwin publicou em 1859 a sua teoria sobre a evolução dos seres vivos através da seleção natural. No seu trabalho, intitulado “*The Origin of Species*”, Darwin propõe um modelo de evolução em que uma população de indivíduos sofre um processo de evolução natural e estes são capazes de se adaptarem ao ambiente em que vivem através de processos de seleção natural, reprodução, recombinação sexual e mutação, onde os indivíduos mais adaptados têm maiores chances de sobreviverem e gerarem descendentes. Segundo Bäck [3], o processo de seleção natural, que privilegia os indivíduos com alta capacidade de sobrevivência (mais adaptados ao meio ambiente, ou mais aptos), permite que a qualidade média da população melhore ao longo do processo evolutivo, levando à obtenção de um indivíduo totalmente adaptado ao ambiente, o indivíduo “ótimo”.

No processo de evolução natural, a combinação da teoria da evolução com a genética fornece um mecanismo que permite o surgimento e a adaptação de seres vivos ao meio-ambiente, através de uma busca no imenso espaço de todas as possíveis combinações de seqüências de DNA. Os Algoritmos Evolutivos (AEs) utilizam estas idéias através da

manipulação de uma população de indivíduos (soluções) que evoluem ao longo de várias iterações do AE, chamadas de “gerações”. Neste contexto, Mühlenbein [34] lembra que os projetistas de AEs devem apenas se inspirar na natureza sem, contudo, tentar copiá-la. Esta inspiração definirá a capacidade dos AEs em promoverem a “evolução” de soluções para problemas reais, desde que estes problemas tenham sido modelados corretamente.

Os AEs são divididos, de uma forma geral, nos seguintes grupos:

- *Estratégias de Evolução*: na proposta original, apresentada por Schwefel [39], não existe seleção de indivíduos para constituição da população da geração seguinte – cada indivíduo de uma população gera um único filho, através da aplicação do operador mutação.
- *Programação Evolutiva*: foi desenvolvida para evoluir máquinas de estados finitos através de um processo semelhante às Estratégias de Evolução [19].
- *Algoritmos Genéticos*: os princípios básicos dos Algoritmos Genéticos (AGs) foram apresentados inicialmente por Holland [24]. Goldberg [23] e Michalewicz [30], dentre muitos outros pesquisadores, tiveram papéis importantes na consolidação dos princípios dos AGs propostos por Holland.
- *Programação Genética*: é um método utilizado para a evolução de programas de computador, inicialmente proposto por Koza [26] como uma aplicação dos AGs na evolução de estruturas em árvore.

Segundo Michalewicz, Xao e Trojanowski [35], “as melhores técnicas de computação evolutiva são os algoritmos genéticos: muito freqüentemente os termos ‘métodos de computação evolutiva’ e ‘métodos baseados em algoritmos genéticos’ são usados de forma alternada”. Neste capítulo serão apresentadas as principais características de um AG tradicional, que pode ser utilizado para resolver um grande número de problemas de otimização. Também serão apresentadas propostas encontradas na literatura para aperfeiçoamento do algoritmo genético tradicional, com o intuito de melhorar a qualidade das soluções obtidas e o seu desempenho global.

2.1. Algoritmos Genéticos – Versão Clássica

Os Algoritmos Genéticos (AGs) foram propostos por Holland [24] como sendo algoritmos de busca de propósito geral, com características de busca estocástica, busca de múltiplos pontos e busca paralela. Em seu trabalho, Holland estava mais interessado na evolução dos indivíduos de uma população, em uma tentativa de explicar os processos adaptativos em sistemas naturais para desenvolver sistemas artificiais baseados nestes processos, do que em resolver problemas de otimização.

O comportamento dos AGs corresponde a uma analogia com o comportamento dos indivíduos de uma população na natureza. Considerando uma população de indivíduos da natureza, estes competem entre si por diferentes recursos disponíveis no seu meio ambiente (*habitat*), como água, comida e abrigo. Cada um destes indivíduos possui características externas (fenótipo), relacionadas à sua constituição genética (genótipo), que os diferem entre si em relação à adaptação ao meio ambiente em que vivem. Esta adaptação afeta diretamente a capacidade de sobrevivência por período suficiente para se reproduzirem pelo acasalamento. Através do acasalamento, as características genéticas dos dois indivíduos envolvidos são combinadas e transmitidas para a prole. Dessa forma as gerações futuras possuem uma grande probabilidade de serem formadas por indivíduos com as características necessárias para um maior tempo de vida, em relação às gerações anteriores – a este processo é dado o nome de evolução natural. Para facilitar a descrição e utilização dos AGs, a terminologia utilizada na Biologia é adotada naturalmente.

Segundo Michalewicz [30], os AGs visam a solução de problemas de otimização complexos, sendo mais robustos do que os métodos tradicionais de busca direta. Uma propriedade importante de um AG é que durante sua execução é mantido um conjunto de soluções potenciais (uma população de indivíduos), enquanto nos métodos tradicionais apenas um único ponto do espaço de busca é processado a cada iteração.

Na população de indivíduos de um AG tradicional cada indivíduo normalmente corresponde à codificação de uma solução para o problema considerado. Para realizar esta codificação, ou representação da solução, usualmente é utilizado um vetor de valores binários, inteiros ou reais, que constitui o próprio indivíduo, que é também chamado de cromossomo.

Holland [24] utiliza o termo “genótipo” para os indivíduos, e considera que o ideal é que a sua representação seja realizada através de uma cadeia de dígitos binários.

O “fenótipo” de um indivíduo é obtido a partir da sua submissão a uma função que irá avaliar a qualidade do seu “código genético” e, dessa forma, corresponde às suas chances de gerar descendentes. Esta função, chamada de função de aptidão, é uma codificação da função-objetivo do problema e define a qualidade de cada indivíduo em relação ao problema modelado. Assim como na evolução natural, num AG deve haver maiores chances de que os códigos genéticos dos indivíduos mais aptos sejam transmitidos para as gerações futuras através do processo seleção “natural” e reprodução.

Uma característica importante de um AG é a utilização dos “operadores genéticos” sobre os indivíduos da população para que possam ser exploradas diferentes áreas do espaço de busca evitando, assim, uma convergência do algoritmo para uma solução ótima local. A combinação entre partes do código genético de diferentes indivíduos (através do operador de cruzamento) e a realização de pequenas alterações genéticas (através do operador mutação) permitem a exploração de novas características, que podem corresponder a uma evolução dos indivíduos. Dessa forma, a população de indivíduos tende a convergir para uma combinação de características dos indivíduos que seja ideal para o problema em questão – a solução ótima. Este mecanismo de evolução natural de soluções permite que os AGs possam ser utilizados para a solução de quase todos os problemas de otimização.

Beasley, Bull e Martin afirmam que

“Se o AG foi implementado corretamente, a população irá evoluir ao longo de sucessivas gerações de tal forma que a aptidão do melhor indivíduo e do indivíduo médio em cada geração será incrementada em direção ao ótimo global. A *convergência* é a progressão em direção à uniformidade crescente.” [6].

Com base nesta afirmação, podemos concluir que a população do AG deve convergir em direção à solução ótima para o problema modelado. Apesar da existência de uma tendência natural de convergência para a solução ótima num AG, devido à utilização e regras de transição probabilísticas e não determinísticas, não é possível garantir que um AG irá conseguir obter a solução ótima para o problema para o qual foi desenvolvido, apesar de obter soluções sub-ótimas utilizando tempo computacional relativamente reduzido se

comparado com métodos exatos. Segundo Goldberg [23], apesar da convergência global da população não ser garantida em um tempo finito, os AGs provaram ser uma técnica de busca robusta para a maioria das aplicações reais. A Figura 2.1 apresenta a estrutura básica de um AG tradicional.

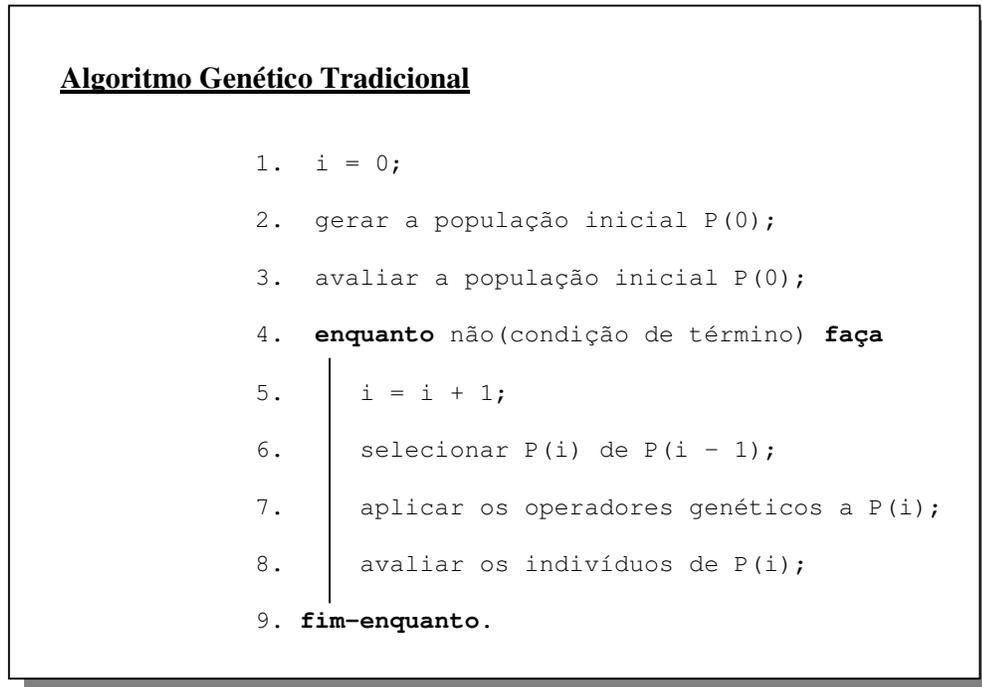


Figura 2.1 – Algoritmo Genético tradicional

Com base no exposto, para que a utilização do AG alcance sucesso na obtenção da solução para um problema específico, na modelagem do AG é necessário considerar:

1. representação dos indivíduos: como representar as possíveis soluções para o problema;
2. função de aptidão: de que forma a função de aptidão pode representar, de forma precisa, a qualidade de cada solução obtida;
3. seleção e reprodução: como será realizada a seleção dos indivíduos de uma geração para constituírem a população da geração seguinte;

4. operadores genéticos: quais operadores genéticos devem ser aplicados, e de que forma;
5. outros parâmetros: quais os valores que devem ser utilizados para o tamanho da população, taxa de aplicação dos operadores genéticos, critério de parada, etc.

Nas próximas sub-seções, serão realizadas algumas considerações relativas às características de um AG tradicional indicadas acima.

2.1.1. Representação dos Indivíduos

De uma forma geral, uma solução potencial para um problema de otimização pode ser representada como um conjunto de valores de parâmetros para o problema modelado. A representação, ou codificação, de um indivíduo em um AG deve concatenar todos estes parâmetros (chamados de genes), formando uma cadeia de valores que corresponde a uma solução para o problema.

Segundo Cole [10] alguns aspectos relacionados à representação dos indivíduos devem ser considerados:

1. a representação dos indivíduos deve ser completa: a representação deve permitir que todos os parâmetros da solução possam assumir todos os valores possíveis no domínio do problema. Caso o AG não consiga representar uma determinada solução, ele nunca poderá obtê-la. Dessa forma, o “espaço de busca” de um AG corresponde a todas as configurações possíveis que um indivíduo pode assumir e deve ser igual ao espaço de busca do problema;
2. a representação deve ser válida: deve-se evitar que possam ser geradas soluções fora do espaço de busca do problema. Caso isto ocorra, deve haver algum mecanismo no AG para tornar a representação válida;
3. evitar a simetria na codificação: a simetria gera redundância, ou seja, indivíduos diferentes podem representar uma mesma solução. Isto pode trazer problemas para a convergência da população. Um exemplo de codificação

com simetria é a codificação *group-number*, utilizada em problemas de clusterização e tratada na Seção 3.2.1.

Os tipos de representação de indivíduos utilizados para AGs aplicados a problemas de clusterização são apresentados em detalhes no Capítulo 3.

2.1.2. Função de Aptidão

A função de aptidão normalmente corresponde à função-objetivo do problema modelado e fornece um valor que permite avaliar a qualidade da solução representada pelo indivíduo em relação às demais soluções do espaço de busca. Por utilizar uma função que avalia a aptidão dos indivíduos de uma população durante a sua evolução, os AGs podem ser aplicados a vários problemas de otimização, bastando utilizar uma função de aptidão específica para o problema.

A função de aptidão deve ser implementada de forma que seja executada de maneira relativamente rápida. Segundo Whitley [41], a execução da função de aptidão nos AGs pode necessitar de quantidade de tempo considerável tendo em vista que, a cada iteração, é necessário avaliar uma população inteira de soluções potenciais (os indivíduos), e não apenas uma solução, como acontece em outras técnicas de otimização.

2.1.3. Seleção e Reprodução

A cada iteração, ou geração, de um AG, partindo-se de uma população de p indivíduos da geração anterior deve ser obtida uma nova população de p indivíduos, sobre a qual serão aplicados os operadores genéticos. Este processo envolve os passos de seleção e reprodução dos indivíduos. A partir dos indivíduos escolhidos no processo de seleção, a reprodução consiste em copiar o código genético dos indivíduos selecionados para a nova população.

Tendo em vista que, ao término de uma iteração, todos os indivíduos são avaliados (Figura 2.1), é possível saber quais deles possuem qualidades superiores aos demais, como solução para o problema modelado. Dessa forma, a seleção de indivíduos deve

corresponder a um mecanismo que permita a sobrevivência dos melhores indivíduos para que estes possam compartilhar suas características com as gerações seguintes.

Existem várias estratégias para a seleção. Beasley, Bull e Martin [6] afirmam que não existe um método considerado absoluto, tendo em vista que os ajustes nas estratégias podem levar a desempenhos semelhantes.

As duas estratégias mais utilizadas para a seleção são descritas a seguir:

1. *seleção proporcional à aptidão*: criada por Holland [24], esta estratégia é inspirada na seleção natural, que envolve a seleção dos indivíduos conforme o valor da sua aptidão. Uma implementação desta estratégia é conhecida como “método da roleta”, que utiliza-se de uma analogia com o jogo de roleta encontrado em cassinos. No método da roleta cada indivíduo possui uma região da roleta proporcional ao valor da sua aptidão e assim uma determinada probabilidade de ser selecionado. Cada vez que a roleta é girada um indivíduo é selecionado. O número total de vezes que a roleta é girada correspondente ao tamanho da população, podendo um mesmo indivíduo ser selecionado mais de uma vez;
2. *seleção por torneio*: existem muitas variações desta estratégia. A mais simples corresponde a, para a seleção de cada indivíduo, inicialmente escolher de forma aleatória t indivíduos da população. Em seguida os indivíduos escolhidos competem entre si e o indivíduo selecionado será aquele que possuir o melhor valor para a aptidão. Normalmente t é utilizado com o valor 2, e um aumento neste valor irá acelerar a convergência da população.

Outro aspecto a ser considerado no processo de seleção é o elitismo, que é uma estratégia utilizada em conjunto com as outras já descritas para a seleção. O elitismo é utilizado para garantir que o melhor indivíduo da população de uma geração seja reproduzido na população da geração seguinte.

Segundo Beasley, Bull e Martin [6] o processo de seleção influencia muito o comportamento de um AG, sendo um dos aspectos críticos para que a evolução da população

ocorra. O tipo de seleção utilizado pode produzir problemas como a convergência prematura para ótimos locais distantes de um ótimo global ou o caso oposto, onde a convergência é muito lenta.

A convergência prematura ocorre quando as características de indivíduos com uma elevada aptidão (mas que não correspondem à solução ótima) dominam rapidamente a população, que converge para um ótimo local de baixa qualidade. A convergência lenta, ou a não convergência, é um problema oposto à convergência prematura, em que ótimos locais de boa qualidade, ou um ótimo global, podem também não serem alcançados. Neste caso o valor médio da aptidão normalmente é alto e a diferença entre o melhor indivíduo e o valor médio é muito pequena. Dessa forma, não existe diversificação suficiente na população para que o AG consiga continuar a sua evolução em direção à solução ótima.

2.1.4. Operadores Genéticos

Após a seleção e reprodução dos indivíduos de uma geração, é iniciada uma nova geração utilizando uma população intermediária, sobre a qual serão aplicados os operadores genéticos. Durante a aplicação dos operadores genéticos, ao mesmo tempo em que são gerados novos indivíduos, algumas características de adaptação dos indivíduos das gerações anteriores são mantidas. Os operadores genéticos normalmente utilizados em um AG tradicional são o operador de cruzamento e o operador mutação.

Segundo Goldberg [23], o operador de cruzamento é a principal força direcionadora em um AG. O operador de cruzamento realiza a troca de partes de pares de indivíduos com o objetivo de tentar obter indivíduos melhores a partir dos indivíduos selecionados. Dessa forma, o principal objetivo do cruzamento é utilizar o conhecimento obtido em pontos do espaço de busca visitados previamente. A aplicação do operador de cruzamento a um par de indivíduos normalmente está sujeita a uma taxa de probabilidade de aplicação, definida como parâmetro para a execução do AG.

A partir da seleção de um par de indivíduos, existem diversas formas de utilização do operador de cruzamento. As mais comuns são:

1. *cruzamento de um ponto*: a partir de um ponto de cruzamento dos indivíduos envolvidos, obtido de forma aleatória, os valores constantes dos trechos situados após o ponto de cruzamento são trocados entre os indivíduos do par. Um exemplo é apresentado na Figura 2.2;

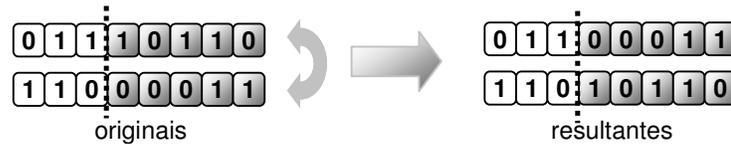


Figura 2.2 – Exemplo de aplicação do operador de cruzamento de um ponto

2. *cruzamento de múltiplos pontos*: é uma generalização do cruzamento de um ponto, em que pares de pontos de cruzamento são obtidos de forma aleatória e os valores dos indivíduos localizados entre cada par de pontos de cruzamento são trocados. Um exemplo é apresentado na Figura 2.3;

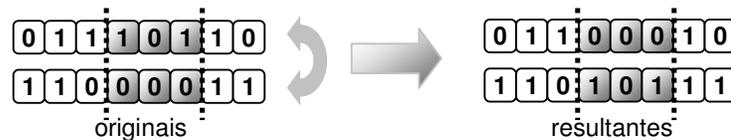


Figura 2.3 – Exemplo de aplicação do operador de cruzamento de múltiplos pontos

3. *cruzamento uniforme*: uma máscara de dígitos binários é obtida de forma aleatória, onde o dígito 1 indica que o valor na respectiva posição dos indivíduos deverá ser trocado e o dígito 0 indica que os valores originais das posições dos indivíduos envolvidos, equivalentes às posições da máscara, devem ser mantidos com os valores originais. Um exemplo é apresentado na Figura 2.4.

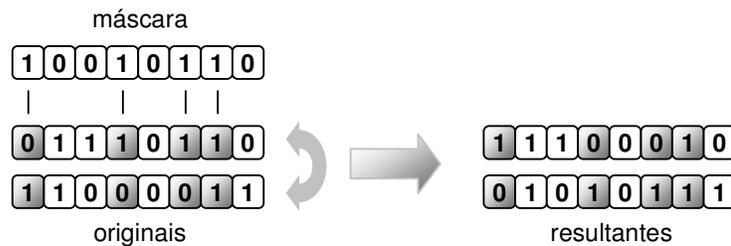


Figura 2.4 – Exemplo de aplicação do operador de cruzamento uniforme

Outros tipos de operadores de cruzamento podem ser necessários dependendo do problema modelado. Davidor [11], por exemplo, utiliza um operador de cruzamento modificado em um AG desenvolvido para a aplicação de planejamento de caminhos em trajetórias de robôs. Neste AG cada indivíduo possui tamanho variável, o que pode gerar ambigüidades ao se combinar dois indivíduos. Portanto, neste novo operador são identificados *a priori* quais pontos de cruzamento irão gerar indivíduos não viáveis e, dessa forma, tais pontos não são utilizados para o cruzamento. É importante observar que este tipo de operador de cruzamento deve agregar conhecimento do domínio do problema.

Diferentemente do operador de cruzamento, o operador mutação realiza trocas aleatórias de alguns valores dos indivíduos, com o intuito de pesquisar novas áreas do espaço de busca, a partir de indivíduos selecionados. Ao permitir a manutenção da diversidade genética da população, o operador mutação evita que a população fique estagnada em uma região de ótimo local. A Figura 2.5 apresenta um exemplo de aplicação do operador mutação em um indivíduo representado por uma cadeia de dígitos binários, em que os valores dos elementos selecionados, em destaque, são trocados.

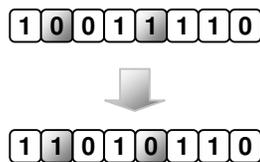


Figura 2.5 – Exemplo de aplicação do operador mutação

O operador mutação também está sujeito a uma taxa de aplicação que deve ser definida conforme características do problema. A taxa de aplicação do operador mutação refere-se à probabilidade de cada um dos elementos de um indivíduo sofrer a mutação. Caso a taxa de aplicação seja próxima a 100%, a busca se torna efetivamente aleatória. O parâmetro taxa de mutação é, portanto, determinante para o grau de convergência da população durante a execução de um AG.

Segundo Beasley, Bull e Martin [6], a definição dos operadores de cruzamento e mutação, bem como as suas respectivas taxas de aplicação, podem ser determinantes para a ocorrência dos problemas de convergência descritos na Seção 2.1.3. Os problemas de

convergência originados da definição incorreta das taxas de aplicação do operadores genéticos podem, por exemplo, eliminar a possibilidade de exploração do espaço de busca pelo operador de cruzamento, ficando apenas o operador mutação responsável pela exploração que, por isto, pode se tornar lenta e aleatória.

2.1.5. Outros Parâmetros

Em um AG tradicional existem vários outros parâmetros e características que podem interferir na qualidade das soluções obtidas. Dentre eles os principais são o tamanho da população, o total de gerações e a forma de geração da população inicial:

1. *tamanho da população*: enquanto populações com poucos indivíduos podem convergir rapidamente para ótimos locais, trabalhar com populações grandes pode exigir a realização de muita computação;
2. *critério de parada*: os dois principais critérios de parada são o número total de gerações ou a convergência da população. Caso seja definido um número fixo de gerações, pode ser que este número não seja suficiente para se obter a solução ótima. Caso o critério de parada seja a convergência da população, pode-se definir que o AG irá terminar quando todos os indivíduos forem iguais, o que pode nunca ocorrer, ou quando exista um percentual de indivíduos iguais na população;
3. *geração da população inicial*: a população inicial pode ser gerada de forma aleatória ou pode ser utilizada uma heurística que utilize algum conhecimento do problema e possibilite um direcionamento da busca a uma região do espaço de busca próxima à solução ótima.

2.2. Algoritmos Genéticos – Versões Aperfeiçoadas

Embora os AGs sejam muito utilizados na literatura para a solução de problemas de otimização, o desempenho destes na sua forma tradicional pode ser pior do que o desempenho de outras metaheurísticas. Com o intuito de melhorar o desempenho dos AGs, novas versões diferentes do AG tradicional têm sido propostas. Dentre as variações de AGs

mais utilizadas, estão as propostas de inclusão de novos operadores genéticos e novos mecanismos para a criação da população inicial, bem como o desenvolvimento de novos processos para a seleção e reprodução de indivíduos.

Nessa “evolução” dos AGs, Glover [22] introduziu a abordagem evolucionária denominada *Scatter Search*, que apresenta características semelhantes e complementares aos AGs, e que tem se mostrado promissora na resolução de problemas de otimização [20, 21, 22, 27, 31]. Na abordagem *Scatter Search*, a idéia principal é trabalhar com uma versão mais determinística de AGs utilizando a idéia da combinação linear de boas soluções (soluções elite) obtidas durante as iterações do algoritmo, com o intuito de obter uma solução intermediária entre elas que seja melhor do que as soluções elite envolvidas.

Em outra linha de pesquisa relacionada à melhora no desempenho dos AGs, Moscato [33] propôs os Algoritmos Meméticos (AMs), cuja representação formal foi apresentada mais tarde por Radcliffe e Surry [38]. Nos AMs é introduzido um procedimento de busca local, que corresponde ao processo de alteração nos valores dos elementos de uma solução para investigar uma área do espaço de busca próxima a ela, e que é aplicado a todos os indivíduos da população de cada geração. Devido ao elevado custo computacional da aplicação de um procedimento de busca local a um grande número de indivíduos, a utilização dos AMs é mais indicada quando a função de aptidão pode ser decomposta. Dessa forma, uma pequena alteração no indivíduo durante a busca local não tornará necessária a avaliação de todo o indivíduo pela função de aptidão, mas será realizada a avaliação apenas da parte afetada pela alteração, o permitirá a atualização da aptidão do indivíduo.

Outra variação dos AGs, proposta por Lorena e Furtado [28], corresponde aos Algoritmos Genéticos Construtivos (AGC), que agregam novas características à proposta tradicional de AG. Dentre as principais características agregadas está a associação de um *rank* a cada indivíduo, considerando que um indivíduo pode ser constituído de blocos de sub-soluções ou uma solução completa, e a utilização de uma função de aptidão bi-objetiva, que é utilizada para avaliar tanto as soluções completas quanto os blocos de sub-soluções, os chamados *schemata*.

Para reduzir o tempo computacional exigido pela execução de um AG, uma alternativa que vem sendo muito utilizada é a sua paralelização. Tendo em vista que os AGs

realizam uma busca paralela no espaço de soluções do problema, o desenvolvimento de uma versão paralela pode ser realizado de forma mais direta do que em outros algoritmos desenvolvidos para problemas de otimização [14, 37].

2.3. Conclusão

Neste capítulo foram apresentadas as características básicas de um AE e, mais especificamente, de um AG clássico. Um AG é indicado para a solução de problemas de complexidade computacional elevada, fornecendo soluções sub-ótimas através de uma exploração paralela em diversas áreas do espaço de soluções do problema.

A obtenção do sucesso na aplicação dos AEs, ou AGs, na solução de problemas de otimização depende de uma modelagem cuidadosa do problema a ser tratado. A escolha dos parâmetros do AG também é determinante para o seu desempenho.

Capítulo 3

Clusterização de Grafos Orientados com o Uso de Algoritmo Genético

De uma forma geral, obter a solução para um problema de clusterização corresponde ao processo de agrupar os elementos de um conjunto de tal forma que os grupos formados, ou *clusters*, representem uma configuração em que cada elemento possua uma maior similaridade com qualquer elemento do mesmo *cluster* do que com elementos de outros *clusters*. As técnicas de clusterização vêm sendo tratadas com frequência na literatura para a solução de vários problemas de aplicações práticas em diversas áreas do conhecimento. Em trabalhos recentes, as técnicas de clusterização vêm sendo utilizadas para: Mineração de Dados, na qual o objetivo é identificar grupos de itens relacionados em uma base de dados [7, 18, 15]; Bioinformática, principalmente no que tange à descoberta de padrões de expressão gênica em *microarrays* [8, 36, 25]; Engenharia de Software, para particionar a estrutura modular de sistemas de informação [13], dentre outras.

Neste capítulo será definido o problema de clusterização e serão descritos alguns métodos utilizados para a sua solução. Será tratado também o problema de clusterização aplicado a grafos, conhecido como problema de particionamento de grafos, que é NP-Difícil e cujas opções de algoritmos viáveis para grafos com muitos vértices são, na sua maioria, os algoritmos que utilizam métodos heurísticos. Será também apresentado com detalhes, um algoritmo genético tradicional adaptado ao problema de clusterização de grafos orientados não ponderados [13], que será utilizado como referência no restante deste trabalho.

3.1. O Problema de Clusterização

Dado um conjunto com n elementos $X = \{X_1, X_2, \dots, X_n\}$, o problema de clusterização consiste na obtenção de um conjunto de k *clusters*, $C = \{C_1, C_2, \dots, C_k\}$, tal que os elementos contidos em um *cluster* C_i possuam uma maior similaridade entre si do que com os elementos de qualquer um dos demais *clusters* do conjunto C . O conjunto C é considerado uma *clusterização* com k *clusters* caso as seguintes condições sejam satisfeitas:

$$\bigcup_{i=1}^k C_i = X \quad (3.1)$$

$$C_i \neq \emptyset, \text{ para } 1 \leq i \leq k \quad (3.2)$$

$$C_i \cap C_j = \emptyset, \text{ para } 1 \leq i, j \leq k \text{ e } i \neq j \quad (3.3)$$

O valor de k pode ser conhecido ou não. Caso o valor de k seja fornecido como parâmetro para a solução, o problema é referenciado na literatura como “problema de k -clusterização” [18]. Caso contrário, isto é, caso o k seja desconhecido, o problema é referenciado como “problema de clusterização automática” e a obtenção do valor de k faz parte do processo de solução do problema, como em [13].

Em uma k -clusterização, o número total de diferentes formas de agrupamento de n elementos de um conjunto em k *clusters*, conforme exposto em [9] e [10], equivale à função $N(n, k)$ apresentada em (3.4).

$$N(n, k) = \frac{1}{k!} \sum_{i=0}^k (-1)^i \binom{k}{i} (k-i)^n \quad (3.4)$$

Com o intuito de ilustrar o crescimento exponencial do número de soluções possíveis para um problema de k -clusterização, considerando a equação (3.4), para combinar 10 elementos em 2 *clusters*, 100 elementos em 2 *clusters*, 100 elementos em 5 *clusters* e 1000 elementos em 2 *clusters*, temos $N(10, 2) = 511$, $N(100, 2) = 6,33825 \times 10^{29}$, $N(100, 5) = 6,57384 \times 10^{67}$ e $N(1000, 2) = 5.3575 \times 10^{300}$ formas diferentes, respectivamente.

Para o problema de clusterização automática o número total de combinações sofre um incremento significativo, sendo definido de acordo com a equação (3.5).

$$N(n) = \sum_{k=1}^n \frac{1}{k!} \sum_{i=0}^k (-1)^i \binom{k}{i} (k-i)^n \quad (3.5)$$

Dessa forma, para um conjunto com 10 elementos, a clusterização automática tem que considerar 115.975 diferentes maneiras de combinar os elementos em um número de *clusters* que pode variar de 1 a 10.

Outro aspecto a ser considerado em relação ao problema de clusterização é como medir o quanto um elemento é similar a outro e, assim, identificar se ambos devem estar contidos em um mesmo *cluster* ou não. Para isto deve ser utilizada uma “medida de similaridade”, que é específica para cada problema de clusterização a ser tratado.

Um importante critério utilizado para identificar a similaridade entre dois elementos é a distância entre eles, que trabalha com as diferenças entre os valores de cada atributo dos elementos. Neste caso, quanto menor for a distância entre um par de elementos maior é a similaridade entre eles. Como medidas de distância muito utilizadas podemos citar a distância euclideana e a distância “city-block” [10]:

- *distância euclideana*: considera a distância d entre dois elementos X_i e X_j no espaço p -dimensional:

$$d(X_i, X_j) = \left[\sum_{l=1}^p (x_{il} - x_{jl})^2 \right]^{\frac{1}{2}} \quad (3.6)$$

- *distância “city-block”*: corresponde à soma das diferenças entre todos os p atributos de dois elementos X_i e X_j , não sendo indicada para os casos em que existe uma correlação entre tais atributos:

$$d(X_i, X_j) = \sum_{l=1}^p |x_{il} - x_{jl}| \quad (3.7)$$

Existem problemas de clusterização em que a distância não pode ser utilizada, ou não é conveniente que seja utilizada, como medida de similaridade, tendo em vista que os

valores dos atributos não são escalares. Como exemplo, ao tratar um problema de clusterização que envolve atributos como sexo e endereço, são necessárias outras medidas que demonstrem o grau de similaridade entre as instâncias da base de dados. Outro exemplo, em que a medida de distância não se aplica diz respeito a alguns problemas de clusterização de vértices em estruturas de grafos em que não são considerados os pesos das arestas. Nestes problemas, também referenciados como problemas de particionamento de grafos, são necessárias, portanto, medidas que considerem apenas as conexões entre os seus vértices (esta última classe de problemas será descrita em detalhes na Seção 3.1.2).

3.1.1. Métodos Utilizados para Clusterização

No processo de clusterização, a busca pela melhor solução no espaço de soluções viáveis é um problema NP-Difícil. A partir das equações (3.4) e (3.5), conforme exposto anteriormente, verifica-se que a avaliação exaustiva de todas as configurações de clusterizações possíveis é computacionalmente inviável, restringindo com isso o uso de métodos exatos para a sua solução. Dessa forma, métodos heurísticos ou aproximados têm sido propostos com frequência, os quais fornecem soluções sub-ótimas com significativa redução da complexidade na solução do problema. Entretanto, devido à grande heterogeneidade das aplicações de problemas de clusterização, as heurísticas são normalmente desenvolvidas para determinadas classes de problemas, ou seja, não existe uma heurística que seja genérica a tal ponto que possa obter bons resultados em todas as aplicações de clusterização.

As heurísticas existentes para a solução de problemas de clusterização podem ser classificadas, de forma geral, em métodos hierárquicos e métodos de particionamento [18]. Nos algoritmos tradicionais para a clusterização hierárquica os *clusters* vão sendo formados gradativamente através de aglomerações ou divisões de elementos/*clusters*, gerando uma hierarquia de *clusters*, normalmente representada através de uma estrutura em árvore, conforme exemplificado na Figura 3.1. Nesta classe de algoritmos, cada *cluster* com tamanho maior que 1 pode ser considerado como sendo composto por *clusters* menores.

Nos algoritmos de aglomeração, que utilizam uma abordagem *bottom-up*, cada elemento do conjunto é, inicialmente, associado a um *cluster* distinto, e novos *clusters* vão

sendo formados pela união dos *clusters* existentes. Esta união ocorre de acordo com alguma medida que forneça a informação sobre quais deles estão mais próximos uns dos outros. Nos algoritmos de divisão, com uma abordagem *top-down*, inicialmente tem-se um único *cluster* contendo todos os elementos do conjunto e, a cada passo, são efetuadas divisões, formando novos *clusters* de tamanhos menores, conforme critérios pré-estabelecidos.

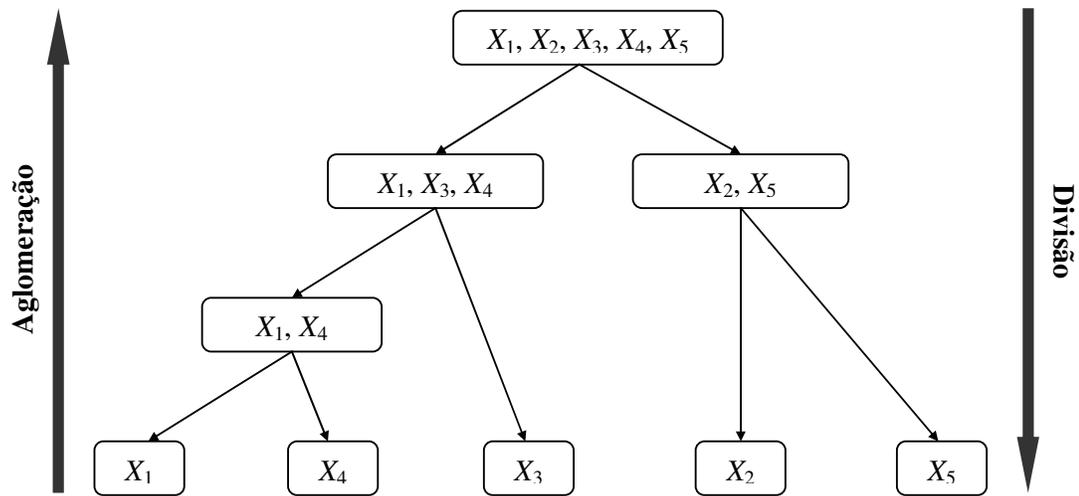


Figura 3.1 – Exemplo de árvore de *clusters* na clusterização hierárquica

Berkhin [7] aponta como vantagens dos algoritmos de clusterização hierárquica a facilidade em lidar com qualquer medida de similaridade utilizada e a sua conseqüente aplicabilidade a qualquer tipo de atributo (numérico ou categórico). As desvantagens relacionam-se à imprecisão do critério de parada e ao fato de que a maioria dos algoritmos desta classe não re-visitam os *clusters* formados ao longo de suas execuções. Este último aspecto está relacionado ao fato dos algoritmos para clusterização hierárquica serem apenas algoritmos construtivos, não permitindo o refinamento de soluções obtidas durante a sua execução. Cole [10] acrescenta como desvantagem da clusterização hierárquica o fato desta necessitar que seja realizado o armazenamento de matrizes de similaridade, o que faz com que a sua utilização seja mais adequada para conjuntos de dados pequenos.

Com relação ao critério de parada nos algoritmos de clusterização hierárquica, a formação dos *clusters* pode ser interrompida quando o número de *clusters* desejado for obtido, no caso de uma *k*-clusterização, ou caso alguma outra condição de parada ocorra. A

falta de refinamento no processo de agrupamento ou desagregação normalmente fornece um caráter guloso ao método hierárquico tradicional.

Nos algoritmos de clusterização que utilizam algum método de particionamento, o conjunto de elementos é dividido em k subconjuntos, podendo k ser conhecido ou não, e cada configuração obtida é avaliada através de uma função-objetivo. Caso a avaliação da clusterização indique que a configuração não atende ao problema em questão, nova configuração é obtida através da migração de elementos entre os *clusters*, e o processo continua de forma iterativa até que algum critério de parada seja alcançado. Neste esquema de migração dos elementos entre os *clusters*, referenciado na literatura como *otimização iterativa* [7], os *clusters* podem ser melhorados gradativamente, o que não ocorre nos métodos hierárquicos.

Os métodos de particionamento para k -clusterização incluem ainda as técnicas *k-medoids* e *k-means*, de acordo com o tipo de representatividade utilizada para os *clusters*: no *k-medoids*, o elemento que melhor representa o *cluster*, é definido de acordo com seus atributos sem que haja muita influência dos valores próximos aos limites do *cluster*; no *k-means* o elemento representativo de um *cluster* é o seu centróide, que possui um valor médio para os atributos considerados, relativo a todos os elementos do *cluster*. A utilização do centróide como elemento representativo de um *cluster* é conveniente apenas para atributos numéricos e possui um significado geométrico e estatístico claro podendo, entretanto, receber muita influência de um único elemento que se encontre próximo à fronteira do *cluster*.

Além dos métodos hierárquicos e de particionamento, é possível observar um crescimento significativo de propostas utilizando metaheurísticas aplicadas a problemas de clusterização, como os algoritmos evolutivos (AEs), com destaque para os algoritmos genéticos (AGs) [10, 9, 32]. A utilização de algoritmos genéticos aplicados ao problema de clusterização será tratada na Seção 3.2.

3.1.2. Clusterização de Grafos

O problema de clusterização aplicado a grafos, também referenciado na literatura como “problema de particionamento de grafos”, consiste em, dado um grafo $G = (V, E)$, com

V sendo o conjunto de vértices e E o conjunto de arestas, particionar os vértices em subconjuntos disjuntos, ou *clusters*, otimizando alguma função-objetivo.

No problema de particionamento *balanceado* de grafo a diferença de cardinalidade entre o maior *cluster* e o menor *cluster* deve ser de, no máximo, uma unidade. Quando o número de *clusters* é igual a dois, o problema é referenciado na literatura como problema de bisseção de grafos ou problema de bi-particionamento de grafos [4]. Em [5] o problema de particionamento de grafos é considerado como consistindo do particionamento do conjunto de vértices do grafo em k *clusters* disjuntos e com a mesma cardinalidade, tal que o total de arestas da qual participam pares de vértices de diferentes *clusters* seja minimizado. A Figura 3.2 apresenta um exemplo de bi-particionamento balanceado de um grafo constituído de 14 vértices e 27 arestas.

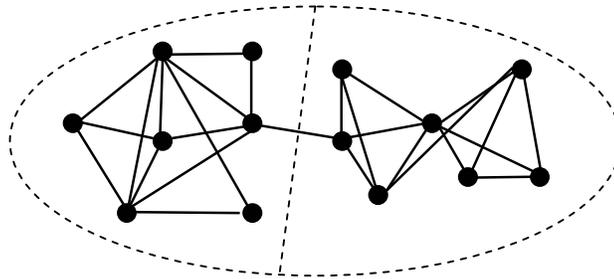


Figura 3.2 – Exemplo de bi-particionamento balanceado de grafos

O problema de particionamento de grafos é NP-Difícil, mesmo com o valor de k igual a dois ou quando algum desbalanceamento é permitido. Segundo Battiti et al. [5], para grafos com mais de 100 vértices, as únicas opções viáveis são os algoritmos heurísticos.

O problema de clusterização de grafos tratado neste trabalho com mais destaque corresponde à clusterização automática considerando grafos orientados e sem valores de peso nos arcos. Neste problema, o objetivo da clusterização é agrupar os vértices do grafo em *clusters* de tal forma que seja maximizado o número total dos arcos internos a cada *cluster*, ao mesmo tempo em que seja minimizado o número total de arcos entre pares de vértices que estejam em diferentes *clusters* [13]. Nesse contexto, a Figura 3.3 exemplifica duas clusterizações possíveis para um grafo com 8 vértices e 12 arcos.

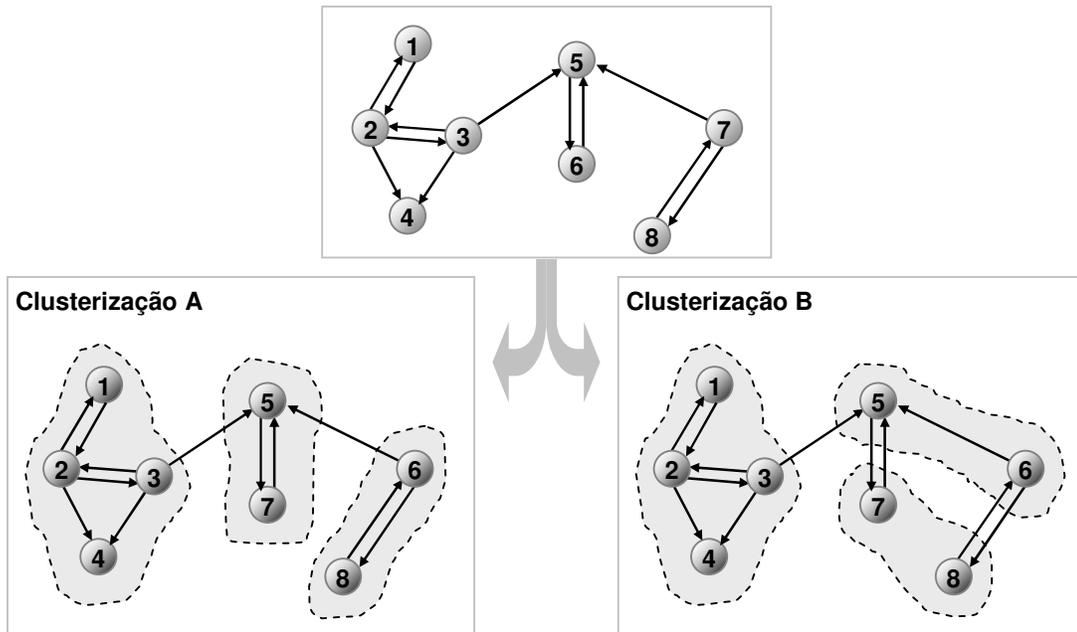


Figura 3.3 – Exemplos de clusterização de um grafo orientado com 8 vértices

Em cada uma das duas clusterizações indicadas na Figura 3.3 cada *cluster* é delimitado por uma linha tracejada. De forma intuitiva podemos considerar a clusterização A mais adequada do que a clusterização B, tendo em vista que nesta existe um *cluster* sem qualquer arco interno e existe também um número maior de arcos entre diferentes *clusters* do que em A.

3.2. Clusterização com Algoritmos Genéticos

Conforme apresentado no Capítulo 2, os algoritmos genéticos (AGs) são indicados para a solução de problemas de complexidade computacional elevada, fornecendo soluções sub-ótimas em intervalos de tempo adequados. Graças a estas características, os AGs vêm sendo muito utilizados na literatura para a solução de problemas de clusterização [9, 10], existindo diferentes versões propostas para atender a problemas de diferentes origens, seja para clusterização automática, como em [13, 40], ou não [12, 28, 29, 32].

Segundo Falkenauer [16], a utilização de AGs tradicionais para a solução de problemas de clusterização possui alguns problemas que merecem atenção:

1. elevada redundância no esquema de codificação dos indivíduos, devido à simetria na representação;
2. o operador de cruzamento pode gerar indivíduos filhos que não possuam características dos indivíduos originais;
3. outros operadores genéticos podem prejudicar fortemente os indivíduos referentes a soluções consideradas relativamente boas.

Dessa forma, no desenvolvimento de um AG para clusterização, devem ser avaliadas características particulares na representação dos indivíduos, na seleção e na aplicação dos operadores de cruzamento e mutação, que serão tratadas com maiores detalhes a seguir. Além disso, a função de aptidão deve ser modelada de acordo com o problema em questão, em que o objetivo do AG poderá ser a sua maximização ou minimização.

3.2.1. Representação dos Indivíduos

A importância da codificação dos indivíduos para a qualidade das soluções obtidas pelo AG foi apresentada no Capítulo 2. Devido a aspectos inerentes aos problemas de clusterização, este componente do AG deve ser tratado ainda com mais cuidado.

Segundo Cole [10], as diferentes maneiras de representar os indivíduos em AGs para clusterização são baseadas em dois esquemas principais. No primeiro esquema, cada elemento a ser agrupado recebe um valor, seja ele um inteiro ou uma seqüência de bits, que corresponde ao gene e indica em qual *cluster* o respectivo elemento se encontra. No segundo esquema, os valores dos genes correspondem aos elementos e as posições dos genes indicam como os elementos estão divididos nos *clusters*.

Cole [10] apresenta os seguintes tipos de representação dos indivíduos baseados no primeiro esquema:

- *group-number*: um vetor de n inteiros é utilizado para representar uma clusterização de n elementos, onde a i -ésima posição do vetor indica o número do *cluster* do i -ésimo elemento. Utiliza-se uma codificação *group-number*

binária quando a clusterização possuir apenas dois *clusters*, em que os valores 0 e 1 são os identificadores dos *clusters*;

- *matriz*: uma clusterização de n elementos em k *clusters* é representada por uma matriz $n \times k$, em que cada linha corresponde a um *cluster* e cada coluna corresponde a um elemento. Cada elemento (i, j) da matriz pode assumir o valor 0 ou 1, onde o valor 1 indicará que o elemento da coluna j está no *cluster* i . Cada elemento pode estar somente em um *cluster*, portanto, poderá haver apenas uma ocorrência do valor 1 em cada coluna.

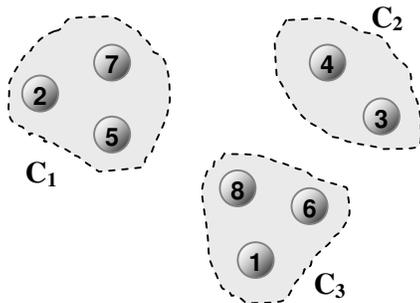
As principais representações dos indivíduos baseadas no segundo esquema são:

- *permutação com separadores*: uma cadeia de inteiros é utilizada e valores inteiros maiores do que o total de elementos, ou outros valores adequados, são utilizados para indicar os limites dos *clusters* da solução;
- *permutação gulosa*: é utilizado em uma k -clusterização, onde cada indivíduo é representado por uma cadeia de caracteres, sendo que os primeiros k elementos da cadeia representam as k sementes dos k *clusters*. A localização de cada um dos demais elementos da cadeia corresponde ao *cluster* da semente com a qual ele possua a maior similaridade.

A Figura 3.4 apresenta um exemplo de clusterização em que oito elementos de um conjunto, referentes a pontos no espaço bidimensional, são agrupados em *clusters*, indicados com delimitações por linhas tracejadas. Na Figura 3.4 são indicadas também as diferentes formas de representações dos indivíduos correspondentes à clusterização apresentada no exemplo.

A partir das representações de indivíduos apresentadas, é importante avaliar os aspectos indicados na Seção 2.1.1: completeza, validade e simetria. É de fácil constatação que as representações apresentadas são completas, tendo em vista que cada elemento pode possuir um valor em um domínio especificado de tal forma que seja possível a obtenção de todas as possibilidades de agrupamentos com outros elementos. Desta forma, todo o espaço de busca pode ser alcançado pelas representações.

Clusterização



(a) *Group-number*

[3 1 2 2 1 3 1 3]

(b) Matriz

$$\begin{bmatrix} 0 & 1 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \end{bmatrix}$$

(c) Permutação com separadores (separador = 9)

[2 5 7 9 3 4 9 1 6 8]

(d) Permutação gulosa

[2 3 1 5 7 6 8 4]

Figura 3.4 – Exemplos das diferentes formas de representação de indivíduos em AGs para problemas de clusterização.

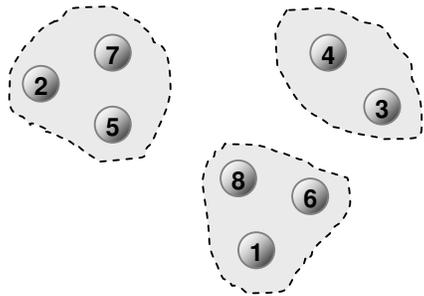
Sobre o segundo aspecto, a validade, as representações apresentadas podem gerar soluções inválidas para problemas de k -clusterização. Neste caso, após a aplicação de cada um dos operadores genéticos pode-se obter um indivíduo com um número de *clusters* diferente de k . Assim, deve-se adotar uma das seguintes abordagens: eliminação do indivíduo inválido, seguida da obtenção de novo indivíduo (aplicando novamente o operador genético), ou um ajuste induzido do indivíduo para torná-lo válido. É importante ressaltar que para a clusterização automática este cuidado não é necessário, tendo em vista que, não havendo alguma restrição específica, todos os indivíduos gerados são válidos (o número de *clusters* pode variar a cada aplicação dos operadores genéticos).

Com relação ao terceiro aspecto, a simetria, cada uma das formas de representação de indivíduos apresentadas acima possui um certo grau de simetria e, portanto, pode gerar indivíduos diferentes que representem uma mesma clusterização. Isto pode ser facilmente constatado alterando-se apenas os números identificadores dos *clusters*, sem alterar a distribuição dos elementos nos *clusters*. A Figura 3.5 apresenta diferentes indivíduos correspondentes a uma mesma clusterização, utilizando a representação *group-number*. De forma análoga, este exemplo pode ser estendido aos demais tipos de representações.

Nos tipos de representações de indivíduos apresentados, a ocorrência da simetria afeta fortemente a utilização do operador de cruzamento. Nestes casos os operadores de

cruzamento tradicionais (de um ponto, de múltiplos pontos e uniforme) não são sensíveis ao contexto das clusterizações representadas pelos indivíduos envolvidos, ou seja, eles atuam sobre os elementos dos indivíduos e não sobre os *clusters*.

Clusterização



$C = \{\{X_2, X_5, X_7\}, \{X_3, X_4\}, \{X_1, X_6, X_8\}\}$

Diferentes Indivíduos (*group-number*)

- A) [3 1 2 2 1 3 1 3]
- B) [1 2 3 3 2 1 2 1]
- C) [2 3 1 1 3 2 3 2]
- D) [6 2 5 5 2 6 2 6]
- E) [5 6 4 4 6 5 6 5]

(...)

Figura 3.5 – Exemplo de simetria na representação *group-number*

Como exemplo, aplicando-se o operador de cruzamento de um ponto sobre os indivíduos A e B da Figura 3.5, e considerando o ponto de cruzamento entre o terceiro e o quarto elementos, temos como indivíduos gerados os apresentados na Figura 3.6. A situação apresentada na Figura 3.6 é uma situação extrema do problema de simetria em que, partindo-se de dois indivíduos que representam uma mesma clusterização, são obtidos indivíduos que representam clusterizações totalmente diferentes daquela representada pelos pais. Pela concepção original dos AGs, estes dois indivíduos deveriam ter sido gerados idênticos aos pais, tendo em vista que as informações compartilhadas são as mesmas. Como dito anteriormente, a simetria afeta também as representações em matriz, permutação com separadores e permutação gulosa.

Indivíduos Pais

- A) [3 1 2 | 2 1 3 1 3]
- B) [1 2 3 | 3 2 1 2 1]

Indivíduos Filhos (cruzamento de um ponto)

- F1) [3 1 2 | 3 2 1 2 1]
- F2) [1 2 3 | 2 1 3 1 3]

$C_A = \{\{X_2, X_5, X_7\}, \{X_3, X_4\}, \{X_1, X_6, X_8\}\}$

$C_B = \{\{X_2, X_5, X_7\}, \{X_3, X_4\}, \{X_1, X_6, X_8\}\}$

$C_{F1} = \{\{X_2, X_6, X_8\}, \{X_3, X_5, X_7\}, \{X_1, X_4\}\}$

$C_{F2} = \{\{X_1, X_5, X_7\}, \{X_2, X_4\}, \{X_3, X_6, X_8\}\}$

Figura 3.6 – Exemplo de repercussão da simetria no operador de cruzamento

Com o intuito de resolver o problema de simetria nas representações dos indivíduos nos AGs voltados para a clusterização, uma nova representação, proposta por Falkenauer [16, 17], utiliza uma estrutura orientada a *clusters* em substituição à representação orientada aos elementos dos *clusters*. Nesta nova representação, cada indivíduo é dividido em duas partes: a primeira parte, chamada “parte dos itens”, é idêntica à representação *group-number*; e a segunda parte, chamada “parte dos *clusters*”, possui uma codificação baseada em “um gene (elemento do indivíduo) para cada *cluster*”. Os rótulos dos *clusters* são apresentados na segunda parte na mesma ordem em que eles aparecem na primeira parte. Os AGs que utilizam esta nova representação são denominados GGAs (*Grouping Genetic Algorithms* – algoritmos genéticos para agrupamento).

Para exemplificar a representação utilizada no GGA, a Figura 3.7 mostra um indivíduo, que corresponde à clusterização indicada na representação puramente *group-number*, e o indivíduo equivalente na representação do GGA.

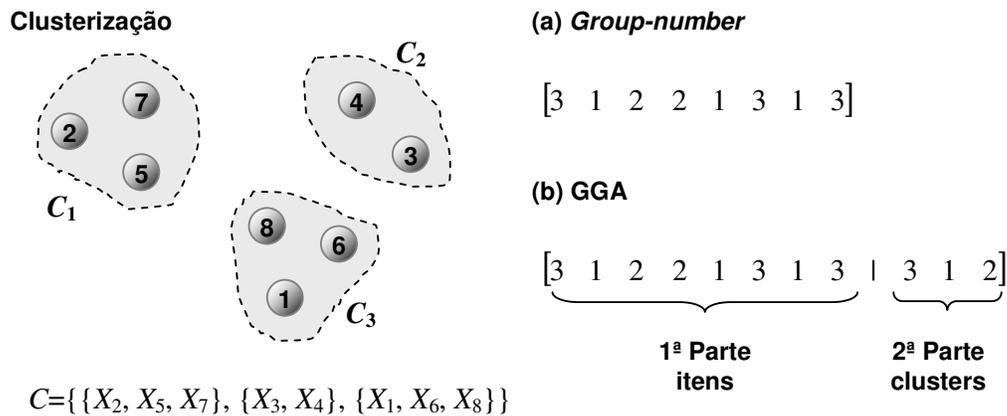


Figura 3.7 – Exemplo de representação de indivíduos no GGA

Na Figura 3.7, os três elementos da segunda parte do indivíduo com representação GGA (b), indicam que clusterização possui três *clusters* (3, 1 e 2) e a informação sobre quais elementos estão em cada *cluster* é obtida na parte dos itens. É importante salientar que, nesta representação, os indivíduos poderão ter tamanhos diferentes, de acordo com o número de *clusters* da clusterização que eles representem.

Para eliminar o problema da simetria, no GGA os operadores genéticos serão aplicados apenas à parte dos *clusters*. A utilização do operador de cruzamento com o enfoque GGA será tratada com mais detalhes na Seção 4.3.

3.2.2. AG para Clusterização Automática de Grafos Orientados

Apesar do problema de particionamento de grafos já ser bastante explorado, existem diferentes modelos para representar este problema. Um modelo aplicado à Engenharia de Software foi proposto por Doval, Mancoridis e Mitchell [13] e será utilizado como base para os algoritmos propostos neste trabalho, sendo, dessa forma, descrito com maiores detalhes a seguir.

Doval et al. [13] propuseram um algoritmo genético tradicional para obter, de forma automática, um bom particionamento (ou clusterização) de um grafo de dependências de módulos (MDG - *Module Dependency Graph*). Um MDG é uma maneira utilizada pelos projetistas de *software* para tornarem sistemas complexos mais compreensíveis e corresponde a um grafo orientado onde os módulos de um sistema são representados pelos vértices e as dependências estáticas entre os módulos são representadas pelos arcos do grafo. A Figura 3.8 apresenta um exemplo de MDG, referente aos módulos do sistema operacional Mini-Tunis, e uma possível clusterização associada, apresentados em [13].

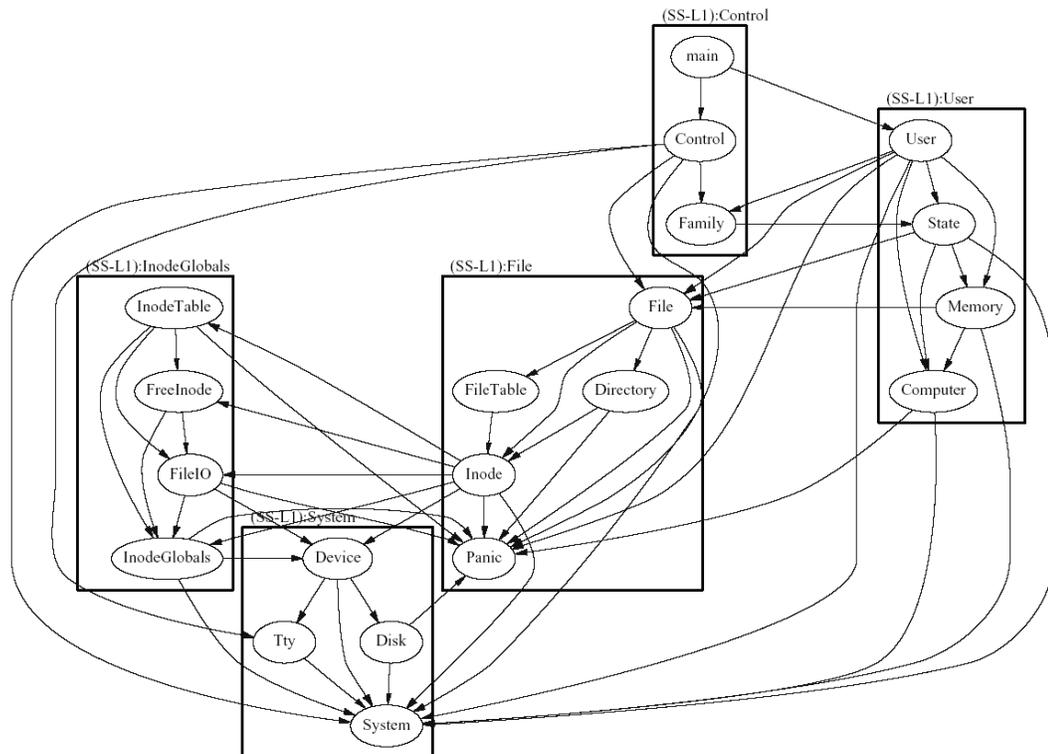


Figura 3.8 – Clusterização do MDG para o Sistema Operacional Mini-Tunis [13]

No algoritmo genético proposto em [13], e referenciado a partir daqui como AGT (Algoritmo Genético Tradicional), é realizada uma clusterização automática, não sendo necessário especificar previamente o número de *clusters* em que o MDG deve ser particionado. Durante a execução do AGT, cada indivíduo da população corresponde a uma solução válida, podendo cada solução possuir um número diferente de *clusters* das demais. Para avaliar a qualidade das clusterizações de um grafo orientado não ponderado, é apresentada uma função-objetivo que leva em consideração esta característica da clusterização automática.

Representação dos Indivíduos

Seja um grafo orientado $G = (V, E)$, onde V corresponde a um conjunto de n vértices $\{v_1, v_2, \dots, v_n\}$, e E representa o conjunto de arcos (v_i, v_j) tal que $v_i, v_j \in V$, onde cada arco incide de v_i e é incidente em v_j .

Dado um conjunto finito de *clusters* C , uma clusterização de G pode ser definida pela função $Q: V \rightarrow C$ que mapeia cada vértice de V em um *cluster* de C , sendo $P(C_i)$ o conjunto de vértices associado ao *cluster* C_i , isto é, $P(C_i) = \{v \in V : Q(v) = C_i\}$.

No AGT cada indivíduo da população deve representar um esquema de clusterização Q . Para isto, cada indivíduo corresponderá a um vetor de inteiros $\{e_1, e_2, \dots, e_n\}$ onde cada elemento e_i de índice i do vetor indica o identificador do *cluster* do vértice v_i , ou seja, $Q(v_i) = e_i$, onde $1 \leq i \leq n$. Conforme visto na Seção 3.2.1, esta representação é referenciada na literatura como codificação *group-number* [10].

A Figura 3.9 mostra um exemplo de indivíduo utilizando a codificação *group-number* para representar uma clusterização de um grafo orientado com 8 vértices.

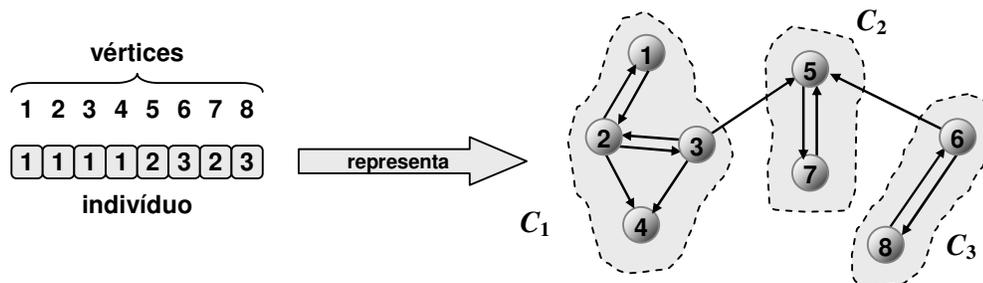


Figura 3.9 – Exemplo de codificação *group-number* para clusterização de um grafo orientado

População Inicial

A população inicial do AGT é gerada de forma aleatória tal que, para cada indivíduo j da população, a cada vértice v é associado um *cluster* $C_i \in C$ tal que $|C| \leq n$, sendo n o número de vértices do grafo. Doval et al. [13] propõem que o número de indivíduos da população inicial (tamanho da população) seja igual a $10 \times n$. Contudo, observamos que, utilizando esta proposta, o tamanho da população pode se tornar muito grande para grafos com número elevado de vértices (por exemplo da ordem de centenas de vértices). O tamanho da população é mantido fixo em cada geração ao longo do processo evolutivo do AGT.

Seleção e Reprodução

Partindo-se da população inicial, o processo evolutivo é iniciado de forma que diferentes populações de indivíduos façam parte das diferentes gerações. Cada geração corresponde a uma iteração do AGT, em que ocorrem os passos de seleção e reprodução dos indivíduos de uma população para gerar uma nova população sobre a qual serão aplicados os operadores genéticos.

Durante a seleção e reprodução de uma população, p indivíduos são escolhidos da população atual levando-se em consideração os valores das suas respectivas funções de aptidão. A seleção utilizada no AGT corresponde à estratégia da roleta, complementada com elitismo, sendo este último responsável por garantir que o indivíduo mais apto da população corrente seja selecionado para a próxima população. Os p indivíduos selecionados de uma geração formam a nova população, que será utilizada na próxima geração do AGT.

Função de Aptidão

Para medir a qualidade da configuração de uma clusterização (solução) para um grafo orientado, Doval et al. [13] apresentam uma função que leva em consideração as conexões entre os vértices do grafo. A função, denominada Qualidade de Modularização (*MQ – Modularization Quality*), é utilizada como função de aptidão do algoritmo genético. O objetivo do AGT será encontrar um bom (possivelmente ótimo) particionamento, através da maximização da função *MQ*.

Para calcular o valor da função MQ de uma clusterização, é necessária a definição de duas outras funções que levam em consideração o total de arcos que conectam vértices internos a um *cluster* e o total de arcos envolvendo vértices de diferentes *clusters*. São elas:

- Intraconectividade (A_i): a intraconectividade A_i de um *cluster* i é:

$$A_i = \frac{\mu_i}{N_i^2} \quad (3.8)$$

onde μ_i é total de arcos internos ao *cluster* i e N_i o total de vértices do *cluster* i .

- Interconectividade ($B_{i,j}$): considerando um par de *clusters* i e j , com ε_{ij} sendo total de arcos entre os *clusters*, N_i e N_j os totais de vértices dos *clusters* i e j , respectivamente, a medida da interconectividade $B_{i,j}$ entre o par de *clusters* é :

$$B_{i,j} = \begin{cases} 0 & \text{se } i = j \\ \frac{\varepsilon_{ij}}{2N_i N_j} & \text{se } i \neq j \end{cases} \quad (3.9)$$

A função de intraconectividade de um *cluster* considera o total de arcos internos a ele em relação ao número máximo de arcos possível. É considerada nesta função a possibilidade da existência de laços, ou seja, ligações que envolvam apenas um vértice.

Os valores da intraconectividade e interconectividade variam entre 0 e 1 e uma clusterização de boa qualidade é aquela que possui um valor grande para o somatório da intraconectividade de cada um dos *clusters* e um valor pequeno para o somatório da interconectividade de todos os pares de *clusters* possíveis no particionamento.

A função MQ , que considera a intraconectividade total e a interconectividade total de uma clusterização é então definida como sendo:

$$MQ = \begin{cases} \frac{\sum_{i=1}^k A_i}{k} - \frac{\sum_{i,j=1}^k B_{i,j}}{\frac{k(k-1)}{2}} & \forall k > 1 \\ A_i & k = 1 \end{cases} \quad (3.10)$$

onde A_i é a intradependência do *cluster* i , $B_{i,j}$ é a interdependência entre *clusters* i e j e k é o total de *clusters* da solução.

O valor de MQ , equação 3.10, corresponde à diferença entre a média de intraconectividade e a média de interconectividade, podendo variar entre -1 e 1 . Tendo em vista que a função MQ premia a obtenção de *clusters* coesos (com alto valor de intraconectividade) e penaliza particionamentos com muitas dependências entre seus *clusters* (alto valor de interconectividade), quanto maior for o valor de MQ , melhor a clusterização. A função MQ é então utilizada como função de aptidão do AG, tendo como objetivo a sua maximização.

Na Figura 3.10 são apresentadas duas clusterizações diferentes para um mesmo grafo e os respectivos valores para a função MQ . O valor de MQ para a clusterização A é superior ao valor referente à clusterização B e, dessa forma, esta pode ser considerada uma clusterização pior do que aquela, que por sua vez pode ainda não ser a melhor clusterização possível (solução ótima) para o grafo do exemplo. Para termos certeza que uma clusterização é a solução ótima devemos verificar, para todas as possíveis combinações dos vértices em *clusters*, qual delas possui o maior valor para a função MQ .

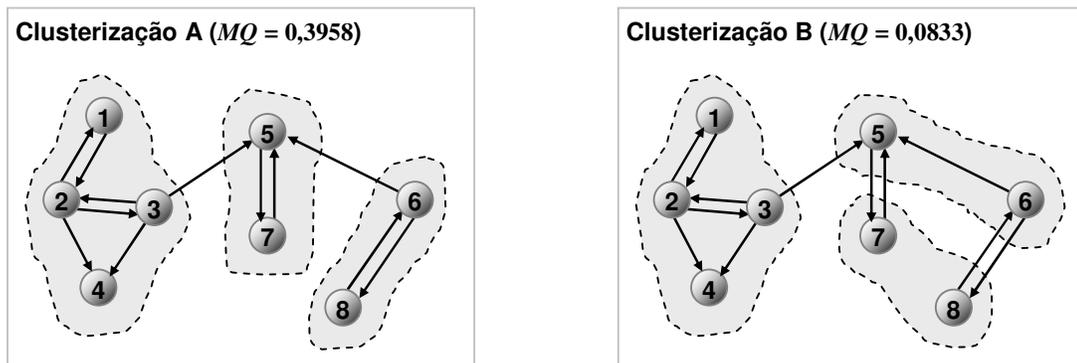


Figura 3.10 – Exemplos de valores para a função MQ associada a diferentes clusterizações

Operador de Cruzamento

O operador de cruzamento é aplicado imediatamente após a seleção dos indivíduos da atual população e reprodução na nova população, sendo usado para combinar pares de indivíduos (os indivíduos pais) com o objetivo de obter novos indivíduos (indivíduos filhos).

O operador de cruzamento utilizado é o cruzamento de um ponto, em que, um trecho de um indivíduo do par selecionado é trocado com mesmo trecho do outro indivíduo do par. O trecho inicia em uma posição i ($1 \leq i \leq n$) do indivíduo, obtida de forma aleatória, onde n é o número de vértices do grafo (e tamanho do indivíduo). Os dois indivíduos do par são então alterados pela troca dos valores dos elementos entre as posições $i + 1$ e n (inclusive). A Figura 3.11 mostra um exemplo de aplicação do operador de cruzamento, onde foi selecionada a quarta posição para ser o ponto de cruzamento.

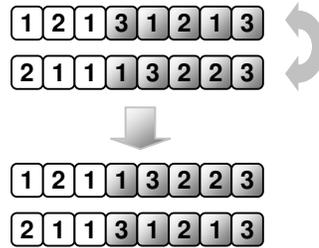


Figura 3.11 – Exemplo de aplicação do operador de cruzamento no AGT

Operador Mutação

O operador mutação é aplicado a cada indivíduo da população, após a aplicação do operador de cruzamento. No operador mutação definido por Doval et al. [13], o valor de cada i -ésimo elemento do indivíduo, correspondente ao identificador do *cluster* do vértice v_i com $1 \leq i \leq n$, possui uma probabilidade de ser trocado por um valor aleatório q , tal que $1 \leq q \leq n$. Assim, o número máximo de *clusters* será o número de elementos (vértices) do problema (grafo) associado.

A taxa de aplicação do operador mutação é definida no AGT como sendo igual a $0,004 \times \log_2(n)$. A Figura 3.12 mostra um exemplo de aplicação do operador mutação sobre um indivíduo no AGT.

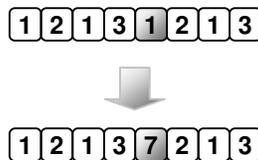


Figura 3.12 – Exemplo de aplicação do operador mutação no AGT

3.3. Resultados Experimentais

Esta seção apresenta os resultados obtidos a partir da implementação e execução do AGT proposto por Doval et al. [13] e apresentado na Seção 3.2.2. O AGT, assim como os demais algoritmos descritos neste trabalho, foram implementados em C padrão ANSI, compilados e montados pelo compilador “gcc” e executado em uma plataforma com um processador ATHLON de 2,4 GHz, 256 MB de memória RAM e sistema operacional GNU/Linux. Durante a realização de cada um dos experimentos deste trabalho nenhum outro processo de usuário concorria pelo processador com o processo do experimento.

Os grafos utilizados nos experimentos deste trabalho foram gerados de forma artificial, principalmente devido à inexistência de instâncias de nosso conhecimento para problemas de particionamento de grafos orientados não ponderados, usando o modelo proposto em [13]. Para gerar os grafos, com características diferentes, como número vértices, número de arcos e topologia de formação, foi desenvolvido e implementado um programa gerador de grafos, descrito na Seção 3.3.1.

3.3.1. Programa Gerador de Grafos

Para a realização dos experimentos deste trabalho, foi desenvolvido e implementado um programa gerador de grafos (PGG). O objetivo do PGG é permitir a criação automática de grafos orientados não ponderados a partir de parâmetros fornecidos pelo usuário. A lista de adjacências do grafo criado é, então, armazenada em um arquivo a ser utilizado pelos AGs implementados neste trabalho.

Os parâmetros utilizados na criação de um grafo são:

- número de vértices do grafo (n);
- número de *clusters* desejado na solução ótima (c);
- intraconecividade média dos *clusters* (A);
- interconectividade média entre os pares de *clusters* (B);
- topologia de formação do grafo, “anel” ou “estrela” (T).

Com base nos parâmetros fornecidos, os passos executados pelo PGG são:

1. definir aleatoriamente a localização de cada um dos n vértices do grafo em um dos c clusters;
2. inserir arcos em cada um dos clusters de tal forma que o valor da sua intraconectividade chegue ao valor mais próximo possível de A ;
3. inserir arcos entre vértices de diferentes clusters de tal forma que o valor da interconectividade média entre cada par de clusters esteja o mais próximo possível de B .

No passo 2, dependendo da topologia de formação definida por T , a inserção dos arcos dentro de cada cluster será realizada de forma circular entre os vértices (topologia anel) ou de forma que sejam realizadas ligações de cada vértice a todos os demais (topologia estrela). As Figuras 3.13 e 3.14 exemplificam as formações de um cluster de quatro vértices para uma intraconectividade fornecida como sendo $A = 0,5$, considerando os tipos de topologia anel e estrela, respectivamente. No passo 3, as ligações entre os clusters vão sendo inseridas seguindo o tipo da topologia, através da escolha dos pares de clusters e da escolha dos vértices internos aos clusters de cada par. Após a geração do grafo com a distribuição dos vértices nos c clusters, a clusterização é avaliada e o valor de MQ obtido pode ser considerado um limite inferior para o valor de MQ da melhor clusterização possível para o grafo.

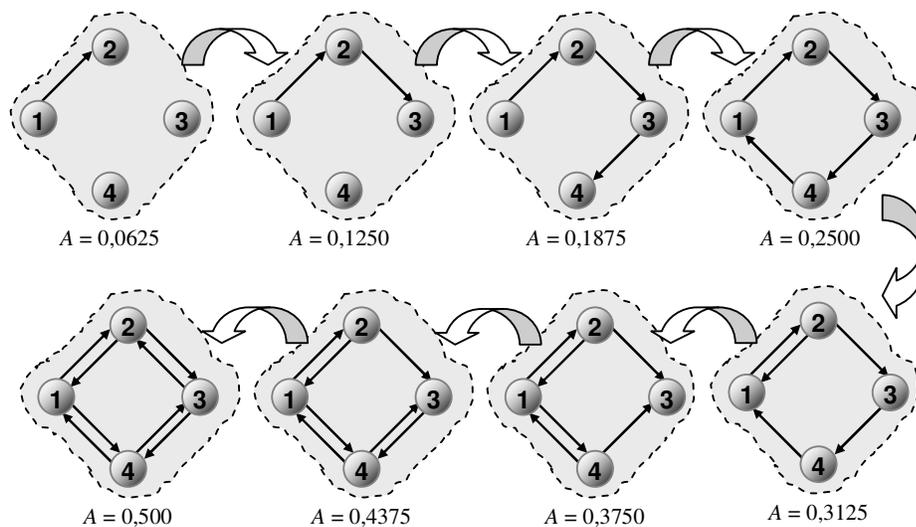


Figura 3.13 – Exemplo de formação de cluster no PGG, com $A = 0,5000$ e topologia anel

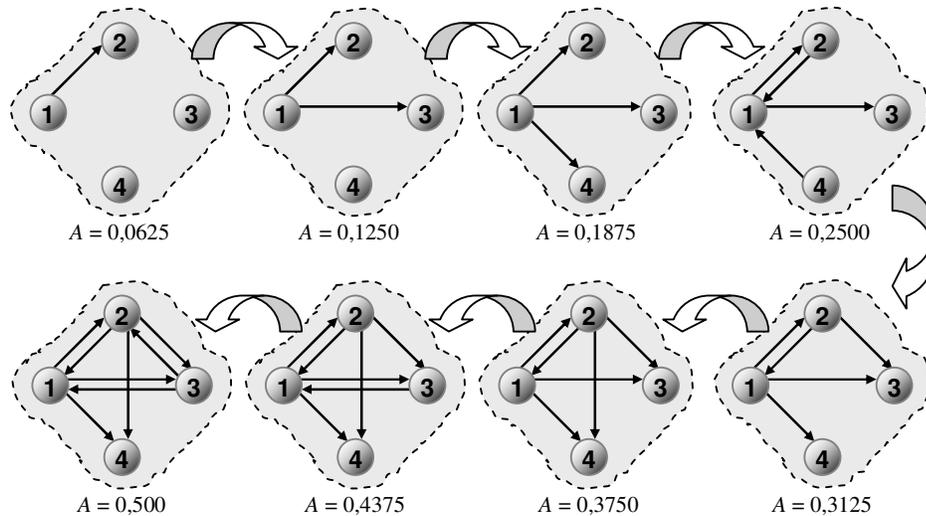


Figura 3.14 – Exemplo de formação de *cluster* no PGG, com $A = 0,5000$ e topologia estrela

É importante ressaltar que, na geração do grafo, apesar de ser fornecido um valor para o parâmetro “número de *clusters* desejado na solução ótima (c)”, não é possível garantir que o valor obtido para a função MQ após a geração corresponda à solução ótima para o grafo gerado. Com base no exposto e no fato de também não ser possível garantir que uma determinada solução obtida após a execução de um AG seja ótima, para fins de comparação entre as diferentes versões de AGs implementadas neste trabalho, será utilizada a expressão “solução *best*” para um determinado grafo como sendo a melhor solução para o problema de clusterização, considerando o valor de MQ obtido na geração do grafo e a melhor solução obtida após todos os experimentos realizados com os algoritmos que utilizam de forma individual as propostas apresentadas no Capítulo 4.

3.3.2. Resultados Obtidos com o Algoritmo Genético Tradicional

Nesta seção são apresentados os resultados dos experimentos realizados com o AGT. Os seguintes parâmetros do AGT foram considerados, conforme propostos originalmente por Doval et al. [13]:

- *tamanho da população*: o número de indivíduos das populações foi mantido fixo ao longo das gerações, com valor igual a $10 \times n$, onde n é o total de vértices do grafo;

- *total de iterações*: o número total de gerações na execução do AGT é de $200 \times n$, onde n é o número de vértices do grafo;
- *geração da população inicial*: a população é gerada de forma aleatória, podendo cada indivíduo corresponder a uma clusterização com número total de *clusters* diferente dos demais;
- *taxa de aplicação do operador de cruzamento*: para cada indivíduo este parâmetro é definido da seguinte forma: para populações de 100 indivíduos ou menos é igual a 80% ; para populações acima de 1.000 indivíduos corresponde a um valor igual a 100%; para populações entre 100 e 1.000 indivíduos varia linearmente entre 80% e 100%.
- *taxa de aplicação do operador mutação*: igual a $0,004 \times \log_2(n)$, onde n é o número de vértices do grafo.

Utilizando-se os valores propostos por Doval et al. [13], tanto o tamanho da população quanto o número total de iterações podem assumir valores muito elevados caso o grafo de entrada possua um número de vértices da ordem de centenas de vértices. Apesar desta constatação, nos experimentos desta seção são mantidos os valores propostos para que os resultados obtidos possam refletir de forma precisa a execução do AGT.

A Tabela 3.1 apresenta informações sobre cada grafo criado para utilização nos experimentos com o AGT, incluindo o total de vértices e de arcos do grafo, a topologia de formação artificial do grafo pelo PGG, bem como o valor da função MQ da solução *best* (chamado de MQ_{best}) e o número de *clusters* desta solução *best*.

Nome do Grafo	Número de Vértices	Número de Arcos	Topologia de Formação	MQ_{Best}	Número de Clusters
A10	10	30	Anel	0,7083	3
A20	20	110	Anel	0,7371	4
A40	40	233	Anel	0,7087	5
A60	60	1085	Anel	0,7402	3
A80	80	758	Anel	0,7780	10
A100	100	888	Anel	0,8092	7
E10	10	30	Estrela	0,7083	3
E20	20	110	Estrela	0,7089	4
E40	40	233	Estrela	0,6485	8
E60	60	1085	Estrela	0,7402	3
E80	80	758	Estrela	0,7075	10
E100	100	921	Estrela	0,6980	16

Tabela 3.1 – Grafos utilizados nos experimentos com o AGT

Devido à utilização de valores iguais para alguns parâmetros na geração de diferentes grafos, pode-se observar algumas coincidências nos valores nas colunas *número de vértices*, *número de arcos*, *MQ best* e *número de clusters* para diferentes grafos apresentados na Tabela 3.1. A Tabela 3.2 apresenta os parâmetros utilizados para a execução do AGT para cada grafo da Tabela 3.1.

Nome do Grafo	Tamanho da População	Número de Iterações	Taxa de Crossover	Taxa de Mutação
A10	100	2.000	0,8000	0,0160
A20	200	4.000	0,8222	0,0200
A40	400	8.000	0,8667	0,0240
A60	600	12.000	0,9111	0,0240
A80	800	16.000	0,9556	0,0280
A100	1.000	20.000	1,0000	0,0280
E10	100	2.000	0,8000	0,0160
E20	200	4.000	0,8222	0,0200
E40	400	8.000	0,8667	0,0240
E60	600	12.000	0,9111	0,0240
E80	800	16.000	0,9556	0,0280
E100	1.000	20.000	1,0000	0,0280

Tabela 3.2 – Parâmetros de execução dos experimentos com o AGT

Nos experimentos, o AGT foi executado dez vezes para cada grafo utilizando os parâmetros descritos na Tabela 3.2. A Tabela 3.3 mostra os resultados obtidos pelas execuções do AGT para cada grafo, incluindo o valor do melhor *MQ* obtido nas execuções, a média aritmética dos valores para o *MQ* encontrados das dez execuções do AGT (*MQ* médio), o percentual do valor médio encontrado para *MQ* em relação ao *MQ* da solução *best* para o grafo (considerando o intervalo $[-1, 1]$ para todos os valores referentes à função *MQ*) e o tempo médio de execução para cada grafo (em segundos).

Nome do Grafo	<i>MQ Best</i>	Melhor <i>MQ</i> das Execuções	<i>MQ</i> Médio das Execuções	<i>MQ</i> Médio/ <i>MQ Best</i>	Tempo Médio de Execução (seg)
A10	0,7083	0,7083	0,7083	100,0	0,68
A20	0,7371	0,7371	0,7289	100,0	8,55
A40	0,7087	0,5919	0,5651	93,2	104,39
A60	0,7402	0,5695	0,5537	90,2	658,88
A80	0,7780	0,5035	0,4921	84,6	1.483,25
A100	0,8092	0,4765	0,4552	81,6	3.321,63
E10	0,7083	0,7083	0,7083	100,0	0,67
E20	0,7089	0,5957	0,5734	93,4	8,85
E40	0,6485	0,4077	0,3716	85,4	104,18
E60	0,7402	0,2165	0,1962	69,9	656,20
E80	0,7075	0,2132	0,1940	71,1	1.479,86
E100	0,6980	0,1544	0,1416	68,0	3.361,53

Tabela 3.3 – Resultados obtidos para os experimentos com o AGT

Gráficos comparativos entre os valores de *MQ* obtidos nas execuções para cada grafo e o *MQ best*, referentes aos dados da Tabela 3.3, são apresentados nas Figuras 3.15 a 3.18.

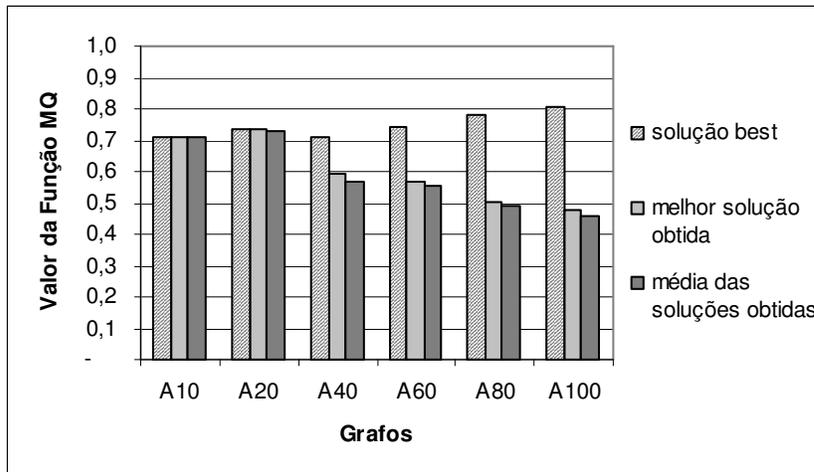


Figura 3.15 – Gráfico comparativo dos resultados do AGT para cada grafo com topologia anel

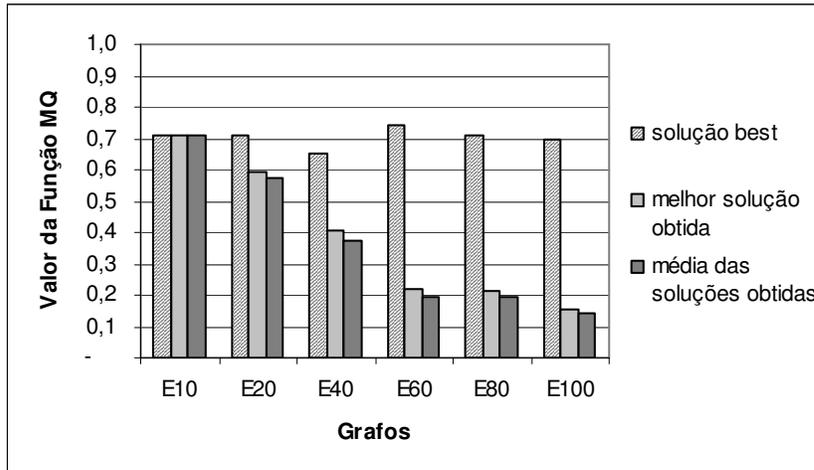


Figura 3.16 – Gráfico comparativo dos resultados do AGT para cada grafo com topologia estrela

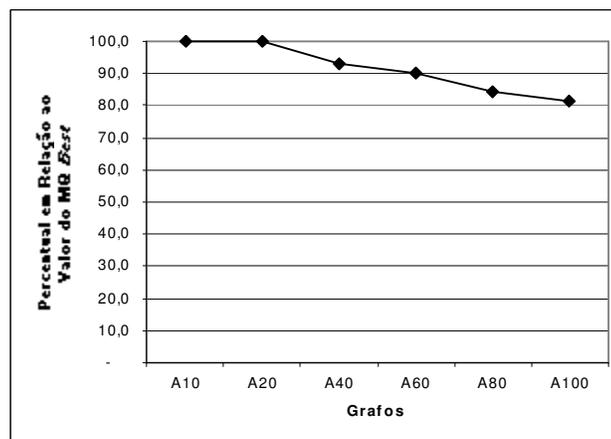


Figura 3.17 – Percentual do valor de MQ médio obtido nas execuções em comparação com o valor de MQ da solução *best* para cada grafo com topologia anel

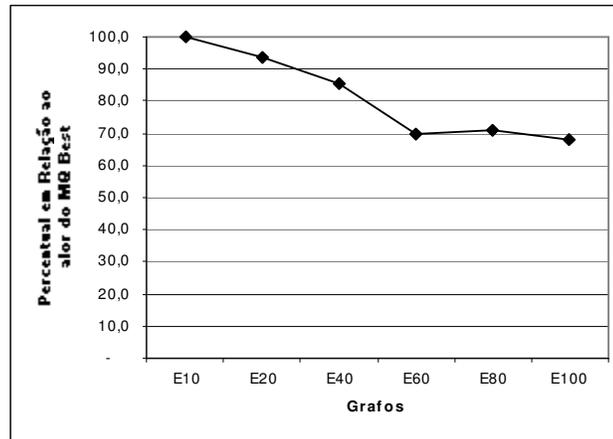


Figura 3.18 – Percentual do valor de MQ médio obtido nas execuções em comparação com o valor de MQ da solução *best* para cada grafo com topologia estrela

É possível observar nos resultados obtidos que, à medida que a cardinalidade do grafo de entrada é incrementada, a qualidade da melhor solução obtida pelo AGT vai se tornando pior em relação à solução *best* e, para grafos com mais de 20 vértices em nenhum experimento foi obtido um valor para MQ igual ao MQ *best*. Com base nos experimentos realizados podemos concluir que o AGT proposto por Doval et al. [13] não fornece resultados satisfatórios, mesmo para grafos com cardinalidade baixa – a partir de 20 vértices.

Além da qualidade das soluções obtidas, um aspecto que deve ser considerado é o tempo gasto para a execução do AGT. Os valores elevados utilizados para os parâmetros *tamanho da população* e *total de iterações* fizeram com que a curva do tempo de execução relativo ao tamanho dos grafos de entrada fosse muito acentuada, conforme os tempos médios de execução indicados nas Figuras 3.19 e 3.20. Como exemplo, uma execução do AGT para o grafo E100 finalizou, em média, após 56 minutos.

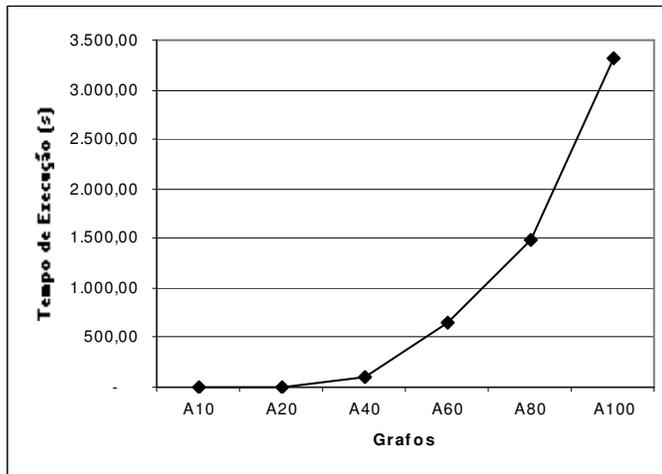


Figura 3.19 – Tempo médio de execução do AGT para cada grafo com topologia anel

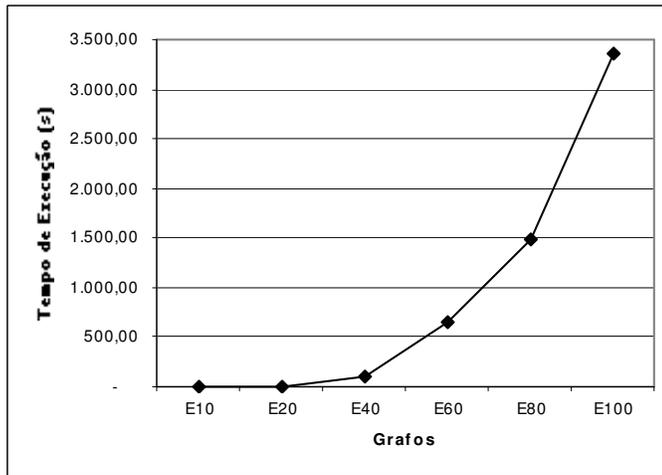


Figura 3.20 – Tempo médio de execução do AGT para cada grafo com topologia estrela

Para fins de comparação, os experimentos realizados com o AGT foram repetidos utilizando-se uma nova versão do AGT, chamada AGTa, em que foram reduzidos o número de indivíduos por população e o total de gerações, reduzindo por consequência o tempo total de execução para cada experimento. No AGTa os novos valores para estes parâmetros são:

- *tamanho da população*: para qualquer grafo o número de indivíduos das populações em cada uma das gerações será mantido fixo e igual a 100;
- *total de iterações*: o número total de gerações na execução do AGT será igual a $20 \times n$, onde n é o número de vértices do grafo;

Os demais parâmetros do AGTa foram mantidos idênticos aos do AGT.

A Figura 3.21 apresenta gráficos que comparam os tempos médios de execução do AGT e do AGTa para os experimentos realizados sobre os mesmos grafos de entrada e a Figura 3.22 mostra uma ampliação de cada um dos gráficos da Figura 3.21. Apesar da enorme redução no tempo de execução médio dos experimentos com o AGTa em relação ao AGT, a qualidade das soluções obtidas sofreu uma pequena redução para os grafos utilizados nos experimentos. Isto pode ser comprovado através das Figuras 3.23 e 3.24, que apresentam os percentuais dos valores de MQ das melhores soluções obtidas nas execuções do AGT e AGTa, em relação ao valor do MQ_{best} para cada grafo.

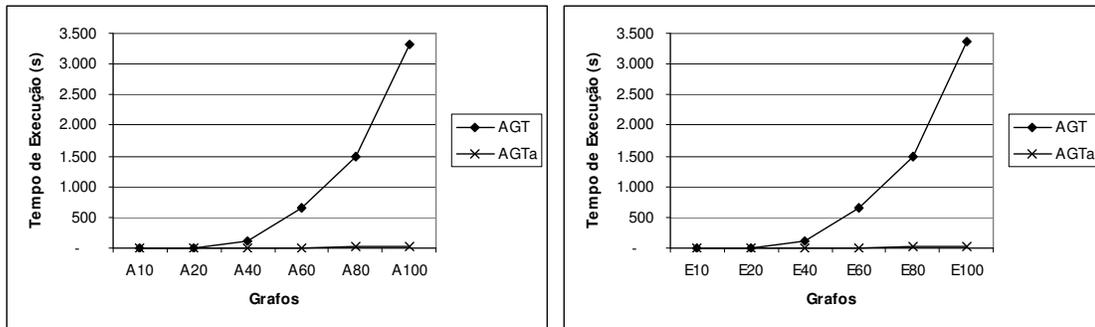


Figura 3.21 – Comparativo entre os tempos médios de execução do AGT e AGTa

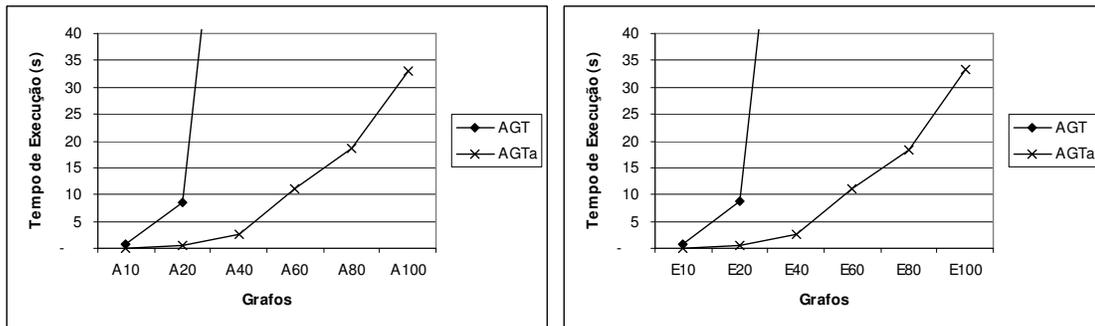


Figura 3.22 – Detalhes dos gráficos apresentados na Figura 3.21

Com base nos resultados apresentados nas Figuras 3.23 e 3.24 constata-se que os parâmetros *tamanho da população* e *total de iterações*, utilizados no AGT, são pouco relevantes para a qualidade das soluções, tendo em vista que as soluções obtidas mostraram-se pouco sensíveis ao elevado incremento no tempo de execução, em comparação com o AGTa.

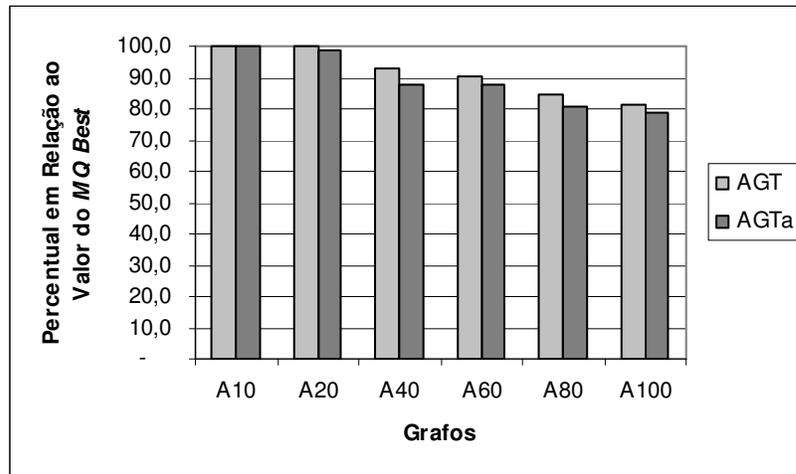


Figura 3.23 – Comparativo da qualidade das melhores soluções obtidas pelo AGT e AGTa em relação ao *MQ best* para os grafos com topologia anel

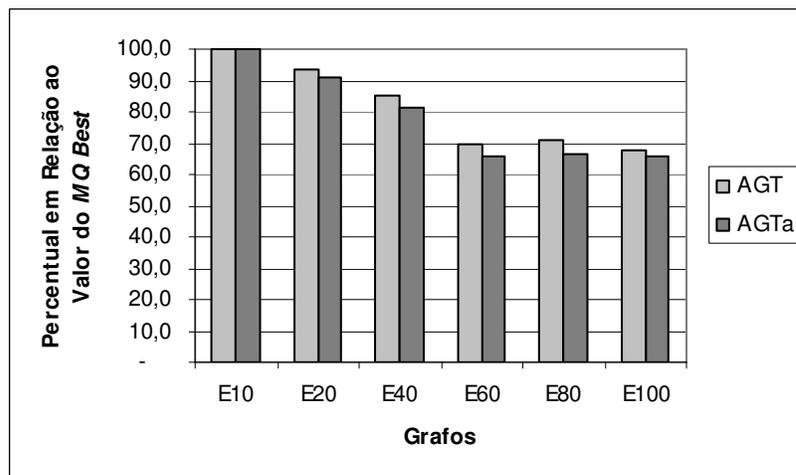


Figura 3.24 – Comparativo da qualidade das melhores soluções obtidas pelo AGT e AGTa em relação ao *MQ best* para os grafos com topologia estrela

A explicação para a constatação de que o AGT não é muito sensível aos parâmetros *tamanho da população* e *total de iterações* está relacionada ao seu caráter de não-convergência, o que pode ser confirmado através dos gráficos de acompanhamento da evolução do melhor indivíduo durante uma execução do AGT para os grafos A60 e A100, apresentados na Figuras 3.25 e 3.26, respectivamente. Nestes gráficos, são indicados o valor do *MQ best* para o grafo e o valor do *MQ* do melhor indivíduo em cada geração da execução do AGT. A não-convergência do AGT pode ser verificada pela evolução irregular das curvas ao longo do tempo, à medida que vão sendo processadas novas gerações do AGT.

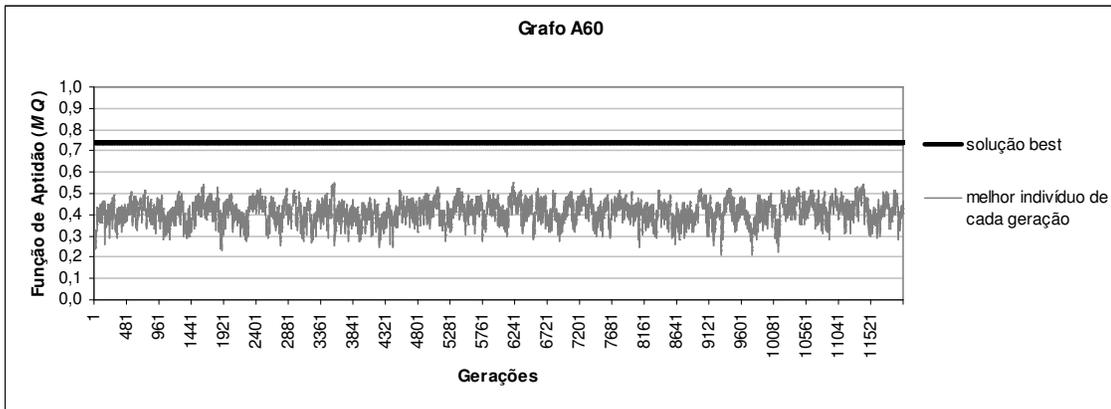


Figura 3.25 – Acompanhamento do melhor indivíduo obtido a cada geração do AGT para o grafo A60

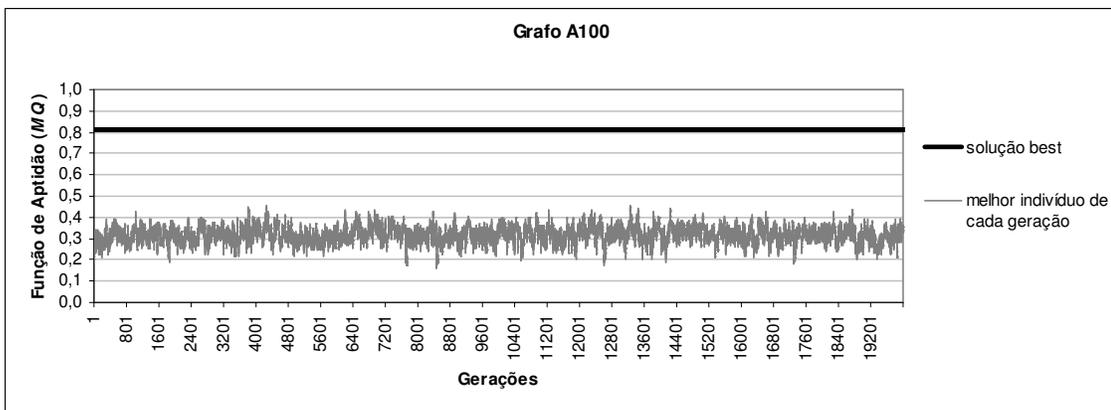


Figura 3.26 – Acompanhamento do melhor indivíduo obtido a cada geração do AGT para o grafo A100

A partir da análise do acompanhamento da execução do AGT foi verificado que devem ser incorporados melhoramentos ao AGT para que exista pelo menos um mínimo grau de convergência na população ao longo das gerações. A ocorrência da convergência no AGT seria o maior indicativo que as populações de determinadas gerações estariam se beneficiando das qualidades dos seus indivíduos ancestrais, permitindo a obtenção de melhores soluções com conseqüente incremento no desempenho global do AGT.

3.4. Conclusão

Na literatura recente os algoritmos genéticos vêm sendo utilizados cada vez mais na solução de problemas de clusterização, tendo em vista o elevado custo computacional para a solução de problemas através de métodos que pesquisem exaustivamente o espaço de soluções. Neste capítulo foram discutidos o problema de clusterização e a utilização de

algoritmos genéticos para a solução desta classe de problemas. Foi apresentado também o AGT, que é um algoritmo genético tradicional para a solução do problema de clusterização aplicado a grafos orientados. Os experimentos realizados com o AGT mostraram que ele possui deficiências que devem ser tratadas.

Com base nos experimentos deste capítulo, é possível identificar pontos deficientes no projeto do AGT, como a incapacidade de obter a melhor solução mesmo para grafos com baixa cardinalidade, o elevado tempo de execução, a simetria na representação dos indivíduos e a falta de convergência da população. No próximo capítulo tais deficiências serão discutidas com maiores detalhes, bem como serão apresentadas propostas para minimizá-las.

Capítulo 4

Desenvolvimento e Análise Experimental de Novas Versões do AGT

Resultados preliminares, obtidos a partir da execução do AGT, mostraram-se insatisfatórios para grafos orientados contendo algumas dezenas de vértices, o que pode ser comprovado no Capítulo 3. Apesar destes resultados insatisfatórios, foi constatado que a função MQ do AGT permite uma boa avaliação da qualidade de uma clusterização, mesmo sem ter sido definida, *a priori*, a quantidade de *clusters* da solução.

A partir da análise do comportamento do AGT na bateria de experimentos do Capítulo 3 foi possível identificar seus principais pontos deficientes, com destaque para a falta de convergência da população ao longo das gerações e a simetria no conjunto de indivíduos gerados. Neste capítulo serão propostas adaptações ao AGT para buscar uma melhoria no seu desempenho em relação à qualidade das soluções obtidas, o que irá acarretar, inclusive, a manutenção de um bom desempenho para grafos da ordem de centenas de vértices. As propostas correspondem desde a otimização de parâmetros até o desenvolvimento de novos módulos para serem adicionados ou para substituírem módulos do AGT original.

As propostas de aperfeiçoamento do AGT foram implementadas como diferentes versões, com os seguintes destaques: alteração nos operadores de mutação e de cruzamento, a inserção de um procedimento de busca local e uma rotina de diversificação da população. Neste capítulo são descritas, de forma detalhada, cada uma das propostas para o AGT utilizado na clusterização de grafos orientados não ponderados, que correspondem às

principais contribuições deste trabalho. O objetivo principal deste capítulo é analisar separadamente cada versão que utiliza apenas uma adaptação proposta. As combinações das adaptações propostas são analisadas no Capítulo 5.

O AGT implementado para os experimentos deste capítulo, exceto nos casos em que alguma outra alteração seja indicada, irá trabalhar da mesma forma que o AGT original [13]. Tendo em vista o elevado custo computacional do AGT, e os experimentos realizados com a versão AGTa realizados na Seção 3.3, os seguintes parâmetros serão modificados, para viabilizar experimentos com grafos de dimensões elevadas:

- *tamanho da população*: o número de indivíduos das populações em cada uma das gerações será igual a $\text{máximo}[100, n]$, onde n corresponde ao número de vértices do grafo;
- *total de iterações*: o número total de gerações na execução do AGT será de $20 \times n$, onde n é o número de vértices do grafo;

Empiricamente foi constatado que estes novos valores para o tamanho da população e o total de iterações produz resultados próximos aos obtidos pelo AGT original, com um custo computacional bem inferior, conforme apresentado na Seção 3.3.

A Tabela 4.1 apresenta as características dos grafos utilizados neste capítulo, que inclui os grafos utilizados no Capítulo 3 e novos grafos com quantidades superiores de vértices, também criados artificialmente pelo programa gerador de grafos.

Na Tabela 4.1 cada linha corresponde a um grafo, sendo que a primeira coluna corresponde ao nome do grafo, seguida das colunas contendo o total de vértices e total de arcos do grafo. Na quarta coluna é indicada a topologia utilizada na inserção dos arcos para formação do grafo, explicada na Seção 3.3. Na coluna *MQ best* é apresentado o valor correspondente à solução *best* para o grafo, seguida pela coluna que indica o número de *clusters* desta solução.

Nome do Grafo	Número de Vértices	Número de Arcos	Topologia de Formação	<i>MQ Best</i>	Número de Clusters
A10	10	30	Anel	0,7083	3
A20	20	110	Anel	0,7371	4
A40	40	233	Anel	0,7087	5
A60	60	1085	Anel	0,7402	3
A80	80	758	Anel	0,7780	10
A100	100	888	Anel	0,8092	7
A160	160	3.208	Anel	0,9090	8
A200	200	1.975	Anel	0,8783	10
A500	500	22.594	Anel	0,8785	87
A1000	1000	38.430	Anel	0,9415	60
E10	10	30	Estrela	0,7083	3
E20	20	110	Estrela	0,7089	4
E40	40	233	Estrela	0,6485	8
E60	60	1085	Estrela	0,7402	3
E80	80	758	Estrela	0,7075	10
E100	100	921	Estrela	0,6980	16
E160	160	2.432	Estrela	0,7364	16
E200	200	1.959	Estrela	0,7662	20
E500	500	13.874	Estrela	0,8759	20
E1000	1000	37.878	Estrela	0,8865	30

Tabela 4.1 – Grafos utilizados nos experimentos com as adaptações propostas

4.1. Estudo do operador mutação

Nesta seção são apresentadas duas propostas para melhorar o desempenho do AGT a partir de modificações no seu operador mutação: alteração da taxa de aplicação do operador e a calibração do domínio de valores utilizados no operador, o que será explicado a seguir.

4.1.1. Alteração da Taxa de Aplicação do Operador Mutação

Dentre os diferentes aspectos do AGT analisados neste trabalho, inicialmente será tratada a avaliação da taxa de aplicação do operador mutação, referenciada neste trabalho simplesmente como “taxa de mutação”. Esta análise foi motivada pela verificação experimental da não convergência da população ao longo das gerações durante a execução do AGT, conforme mostrado na Seção 3.3.2 e Figuras 3.25 e 3.26. Como tratado no Capítulo 2, o parâmetro taxa de mutação é determinante para o grau de convergência da população durante a execução de um AG.

No AGT original [13], para um grafo com n vértices, a taxa de mutação utilizada corresponde a $0,004 \times \log_2(n)$. A justificativa para este valor, fornecida por Doval et al. [13], é que normalmente a taxa de mutação utilizada corresponde ao valor 0,004 quando a

representação dos indivíduos é realizada através de uma codificação binária. Dessa forma, Doval et al. consideram que, como na representação dos indivíduos no AGT é utilizado o sistema decimal no domínio dos valores inteiros, para obter o valor correto para a taxa de mutação deve-se multiplicar o 0,004 ao número de bits necessário para representar o maior valor inteiro que pode ser utilizado na codificação do indivíduo, o que é equivalente a $\log_2(n)$. Assim, para grafos com 10 vértices a taxa de mutação corresponde ao valor 0,0160, enquanto para grafos com 500 vértices este valor se torna 0,0360.

A Figura 4.1 apresenta um gráfico de distribuições percentuais cumulativas, gerado a partir de diferentes experimentos realizados com o AGT, que indica a probabilidade de uma execução do AGT obter uma solução em um determinado intervalo de tempo. Neste tipo de gráfico, que também será utilizado em outros experimentos deste capítulo, cada curva refere-se a uma bateria de execuções de cada versão do AGT para uma mesma instância. As características que diferem uma versão da outra nos diferentes experimentos são indicadas na legenda.

Nos experimentos realizados para a construção de um gráfico de distribuições percentuais cumulativas é utilizado um critério de parada definido como sendo um “valor alvo” a ser alcançado pelo valor da função de aptidão MQ (MQ -alvo) do melhor indivíduo gerado. Cada execução do AGT será, então, interrompida quando o valor de MQ da melhor solução obtida for igual ou superior ao valor de MQ -alvo. Para plotar o gráfico, para cada versão do AGT, os tempos obtidos para cada execução são inicialmente ordenados, obtendo t_1, t_2, \dots, t_m , com m sendo o total de execuções para a versão. Em seguida, é associada uma probabilidade $p_i = (i - 0,5)/m$ para cada i -ésimo tempo (t_i), e são plotados no gráfico os pontos (t_i, p_i) , para $i = 1, 2, \dots, m$.

Os resultados preliminares referentes às variações na taxa de mutação referem-se a experimentos com o grafo A60 e são apresentados na Figura 4.1. Nestes experimentos são utilizados diferentes valores para a taxa de mutação e, para cada experimento, foram realizadas 500 execuções distintas de cada versão do AGT e um valor alvo para a função MQ definido como 0,5. Enquanto no AGT original, para um grafo com 60 vértices, a taxa de mutação é definida como sendo 0,0240, nos experimentos da Figura 4.1, as taxas de mutação consideradas foram 0,0240, 0,0160, 0,0100, 0,0040 e 0,0002. Para indicar o valor da taxa de mutação utilizada no AGT, adotaremos a referência “+M(valor da taxa de mutação)”.

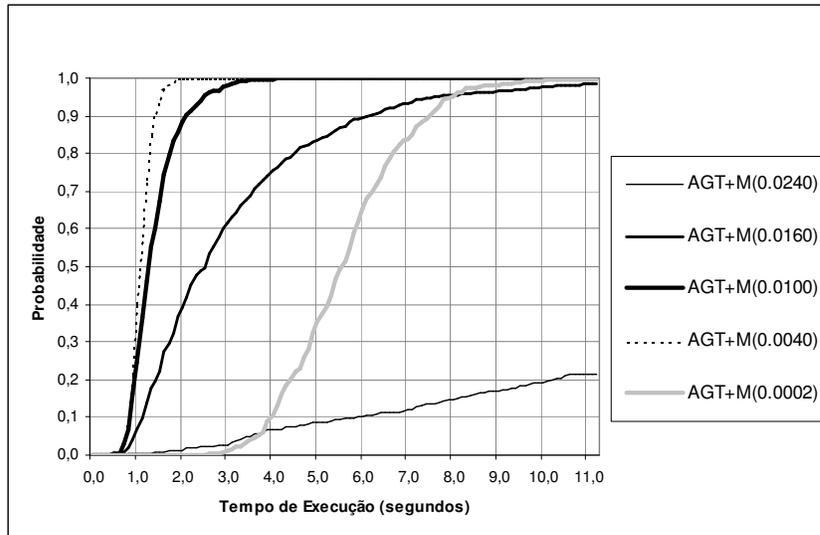


Figura 4.1 – AGT aplicado ao grafo A60, com diferentes valores para a taxa de mutação e MQ -alvo = 0,5

Através das curvas do gráfico da Figura 4.1 observa-se claramente que os melhores resultados foram obtidos para valores da taxa de mutação próximos a 0,004, que é bem inferior ao valor proposto originalmente no AGT. A questão agora é descobrir se existe uma relação entre as curvas do gráfico da Figura 4.1 e o grau de convergência do AGT para os diferentes valores da taxa de mutação. Para isto, foram gerados gráficos de acompanhamento do valor de MQ do melhor indivíduo e do valor médio de MQ considerando todos os indivíduos da população, a cada geração do AGT, para os diferentes valores da taxa de mutação. Cada uma das Figuras 4.2, 4.3, 4.4 e 4.5 exibe o resultado destes experimentos, referente a uma única execução do AGT para a clusterização do grafo A60 utilizando um valor para a taxa de mutação igual a 0,0240, 0,0160, 0,004 e 0,0002, respectivamente.

Nos resultados apresentados nos gráficos das Figuras 4.2 e 4.3, os valores definidos para a taxa de mutação (iguais a 0,0240 e 0,0160, respectivamente) fizeram com que o valor médio da função de aptidão dos indivíduos sofresse uma grande variação entre gerações próximas, e o mesmo ocorrendo para o valor de MQ do melhor indivíduo. Apesar do objetivo do operador mutação ser a investigação de novas áreas do espaço de busca, conforme visto no Capítulo 2, nos dois experimentos que utilizaram os maiores valores para taxa de mutação, as boas qualidades dos melhores indivíduos obtidos a cada geração não se perpetuam nas gerações seguintes. Isto ocorre porque, o operador mutação realiza a investigação do espaço de busca tão rapidamente que não permite a perpetuação dos bons genes de uma geração para outra – antes que isto possa ocorrer eles sofrem nova mutação.

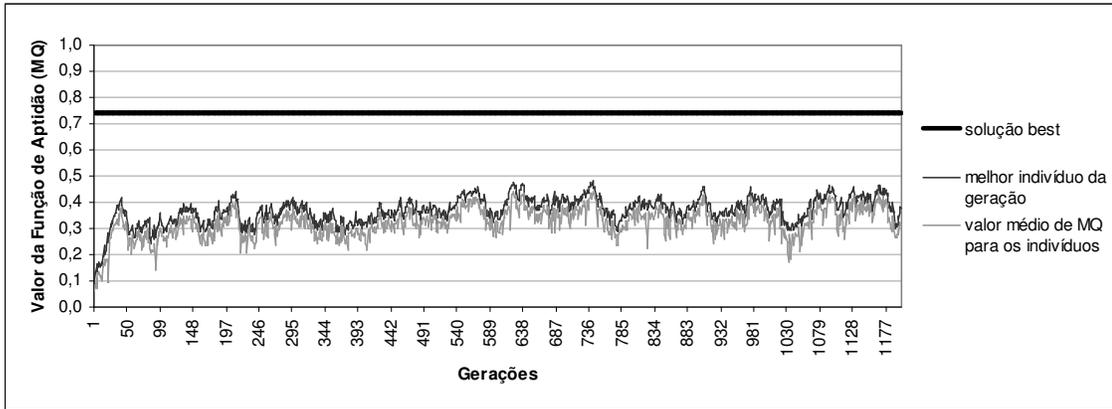


Figura 4.2 – Acompanhamento da evolução do melhor indivíduo e do *MQ* médio dos indivíduos ao longo das gerações, para o grafo A60 e valor para a taxa de mutação igual a 0,0240

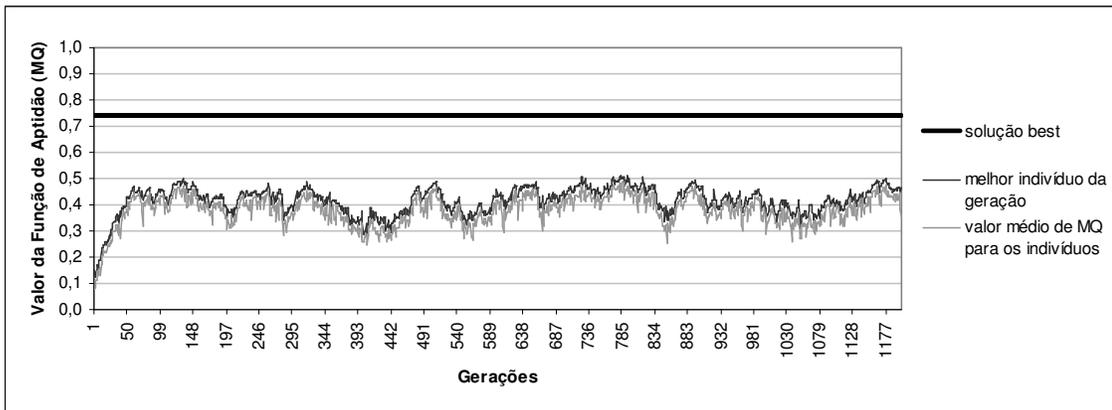


Figura 4.3 – Acompanhamento da evolução do melhor indivíduo e do *MQ* médio dos indivíduos ao longo das gerações, para o grafo A60 e valor para a taxa de mutação igual a 0,0160

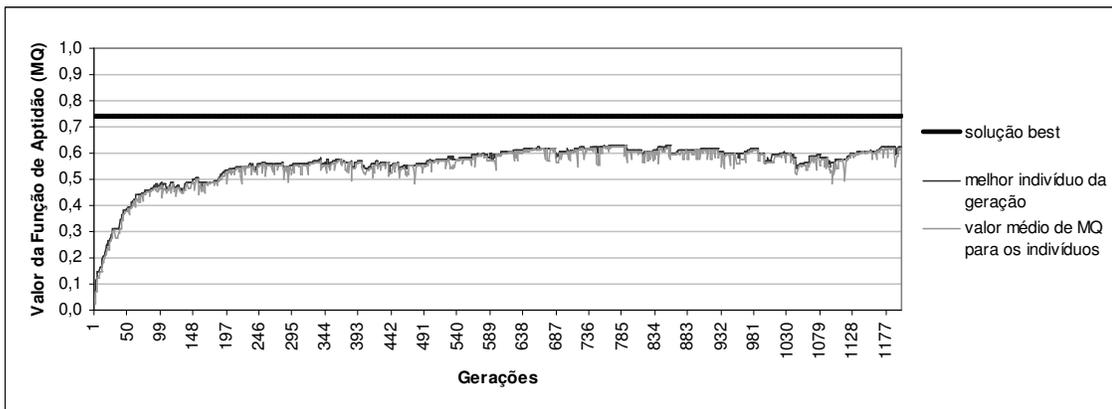


Figura 4.4 – Acompanhamento da evolução do melhor indivíduo e do *MQ* médio dos indivíduos ao longo das gerações, para o grafo A60 e valor para a taxa de mutação igual a 0,0040

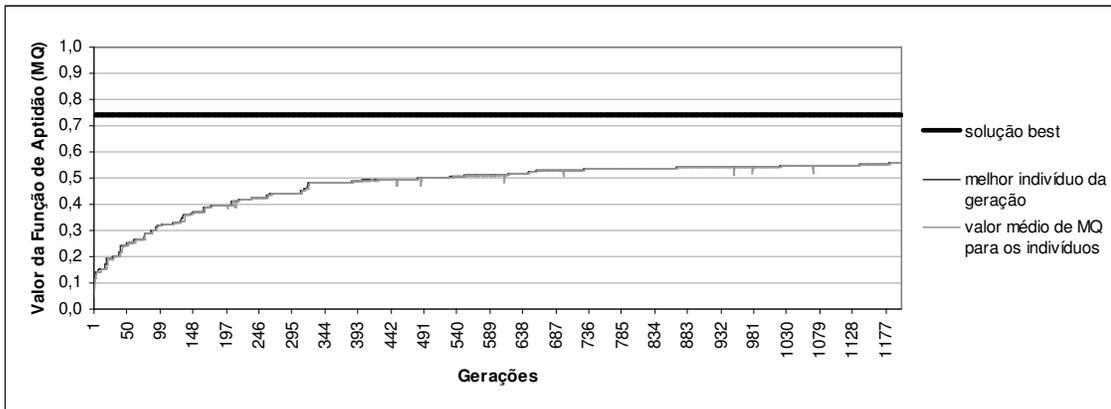


Figura 4.5 – Acompanhamento da evolução do melhor indivíduo e do MQ médio dos indivíduos ao longo das gerações, para o grafo A60 e valor para a taxa de mutação igual a 0,0002

Com a redução no valor da taxa de mutação, o algoritmo permite que os indivíduos com boas qualidades, e que tenham sido selecionados para reprodução, possam ter seus genes transmitidos para as gerações seguintes antes de sofrerem mutação. Isto pode ser verificado nos gráficos apresentados nas Figuras 4.4 e 4.5, em que os valores definidos para a taxa de mutação são iguais a 0,0040 e 0,0002, respectivamente. Nestes casos, os valores referentes ao MQ do melhor indivíduo e o valor médio de MQ a cada geração não são muito diferentes das gerações que se seguem, convergindo para um valor próximo à solução *best* sem, contudo, alcançá-la.

Outra constatação é que, quando são utilizados valores muito baixos para a taxa de mutação, como na Figura 4.5, passa a existir uma maior homogeneidade entre os indivíduos da população devido a uma redução na investigação do espaço de busca (o que pode ser verificado observando-se que a curva referente ao valor médio de MQ sobrepõe a curva referente ao valor de MQ do melhor indivíduo na quase totalidade das gerações). Neste caso, a falta da exploração no espaço de busca poderá retardar a obtenção da solução ótima ou mesmo impedir que ela seja alcançada em algum momento.

Portanto, nos experimentos apresentados na Figura 4.1, para o grafo A60 a melhor versão do AGT foi aquela que utilizou a taxa de mutação com valor igual a 0,004, por conseguir alcançar o alvo mais rapidamente. Entretanto, nenhuma das versões do AGT conseguiu obter a solução *best* para o referido grafo.

Com o intuito de verificar se a solução *best* para o grafo A60 pode ser alcançada com os melhores valores obtidos para a taxa de mutação, através de uma ampliação do total de gerações do AGT, foram realizados novos experimentos. Os resultados destes experimentos são apresentados nas Figuras 4.6 e 4.7, onde são utilizados os valores para a taxa de mutação iguais a 0,0040 e 0,0002, respectivamente, com um número total de gerações para cada versão igual a 20.000. Com base nas Figuras 4.6 e 4.7 e na Tabela 4.2 é possível observar que o aumento do número de gerações permitiu uma melhora na qualidade do melhor indivíduo obtido, em comparação com a versão com 1.200 gerações sem, contudo, alcançar a solução *best*. Além disso, o tempo computacional gasto no aumento das gerações foi muito elevado, o que tornou ruim o desempenho global do AGT. É importante observar que, apesar da versão com a taxa de mutação igual a 0,0002 obter a melhor solução de todos os experimentos apresentados, foi necessário um número elevado de gerações até que ela fosse alcançada (a melhor solução foi obtida na geração 14.976).

Taxa de Mutação	Número de Gerações	Melhor Solução Obtida	Tempo Total
0,0040	1.200	0,6300	10,70 segundos
	20.000	0,6557	174,94 segundos
0,0002	1.200	0,5580	10,83 segundos
	20.000	0,6944	166,62 segundos

Tabela 4.2 – Comparação entre as soluções obtidas para diferentes versões do AGT com o grafo A60 ($MQ_{best} = 0,7402$)

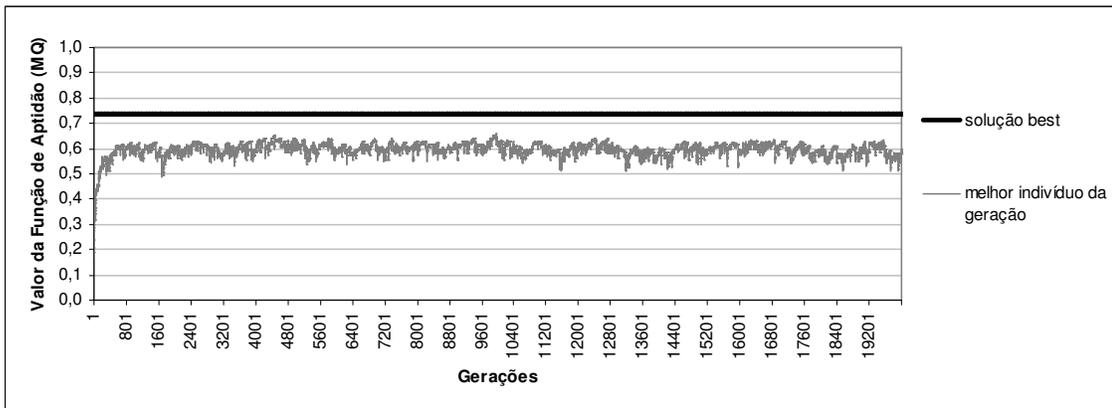


Figura 4.6 – Acompanhamento da evolução do melhor indivíduo ao longo de 20.000 gerações, para o grafo A60 e valor para a taxa de mutação igual a 0,0040

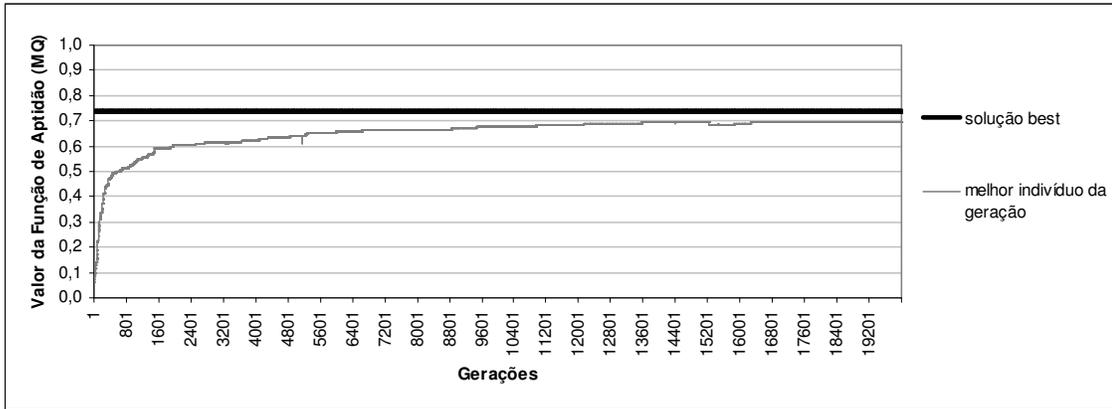


Figura 4.7 – Acompanhamento da evolução do melhor indivíduo ao longo de 20.000 gerações, para o grafo A60 e valor para a taxa de mutação igual a 0,0002

Foram realizados experimentos com outras instâncias de grafos para estender a análise da sensibilidade do AGT em relação ao valor da taxa de mutação e os resultados obtidos foram semelhantes aos resultados apresentados para o grafo A60. As Figuras 4.8 a 4.12 mostram tais resultados, em que o valor do MQ -alvo corresponde a 0,5, utilizando-se diferentes valores para a taxa de mutação, com um total de 200 execuções do AGT em cada experimento.

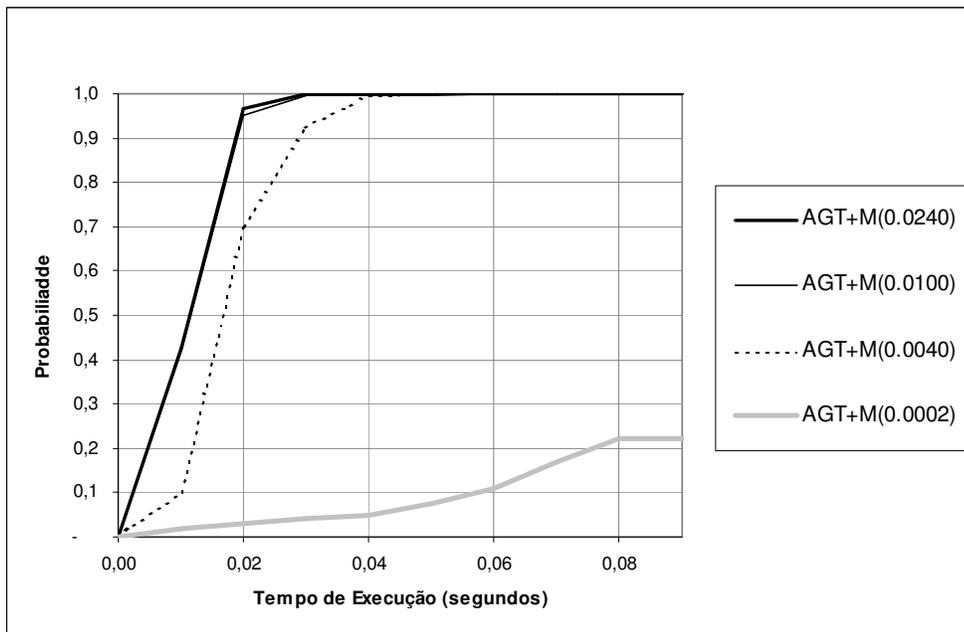


Figura 4.8 – AGT aplicado ao grafo A10, com diferentes valores para a taxa de mutação (MQ -alvo=0,5)

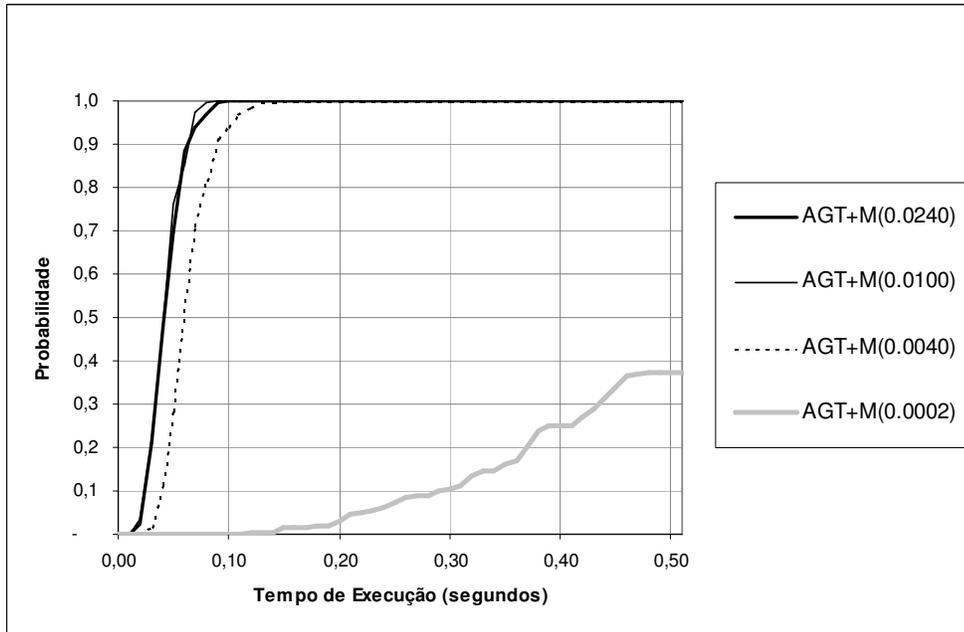


Figura 4.9 – AGT aplicado ao grafo A20, com diferentes valores para a taxa de mutação ($MQ\text{-alvo}=0,5$)

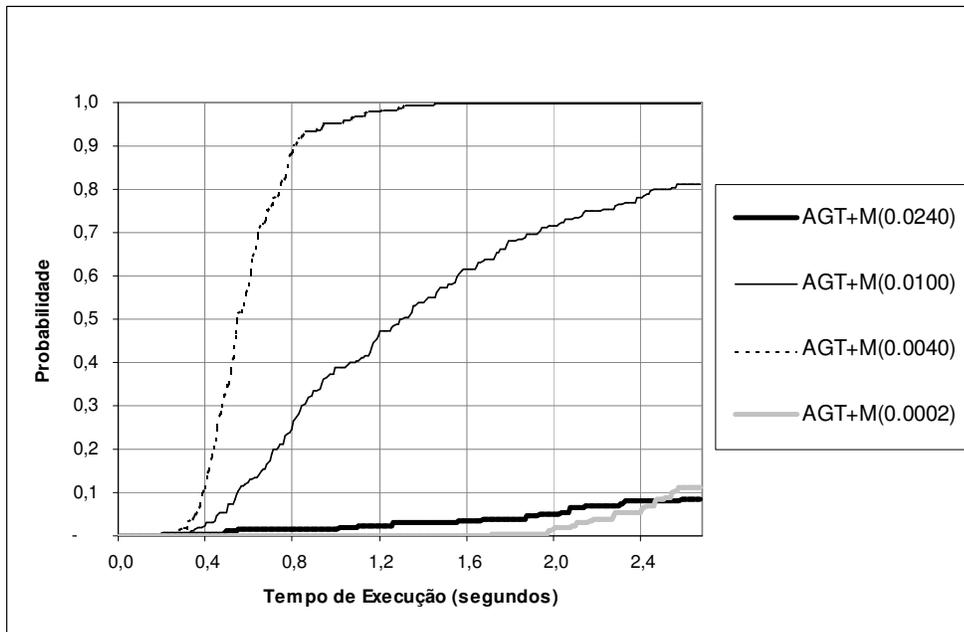


Figura 4.10 – AGT aplicado ao grafo A40, com diferentes valores para a taxa de mutação ($MQ\text{-alvo}=0,5$)

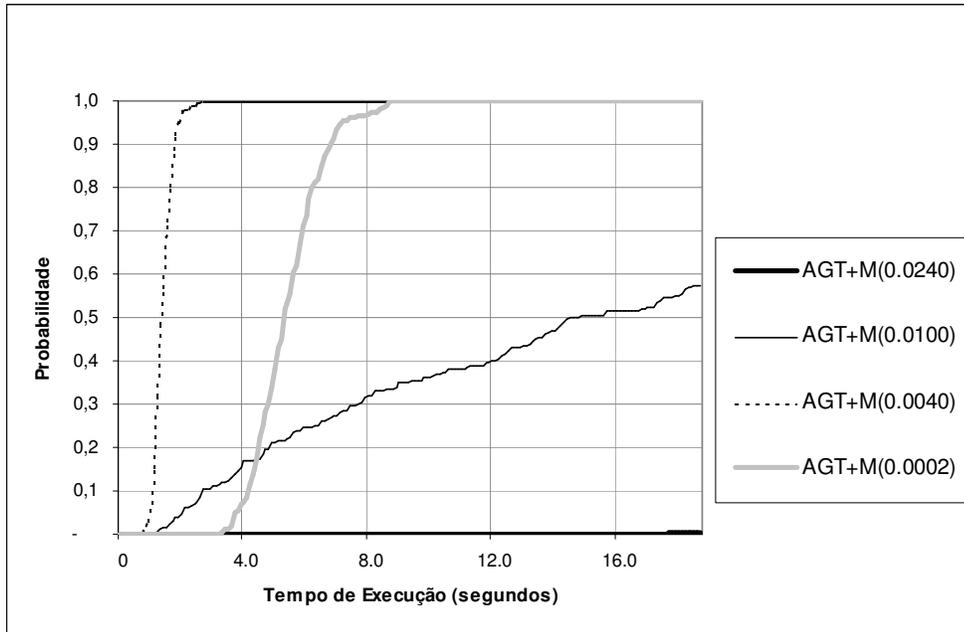


Figura 4.11 – AGT aplicado ao grafo A80, com diferentes valores para a taxa de mutação (MQ -alvo=0,5)

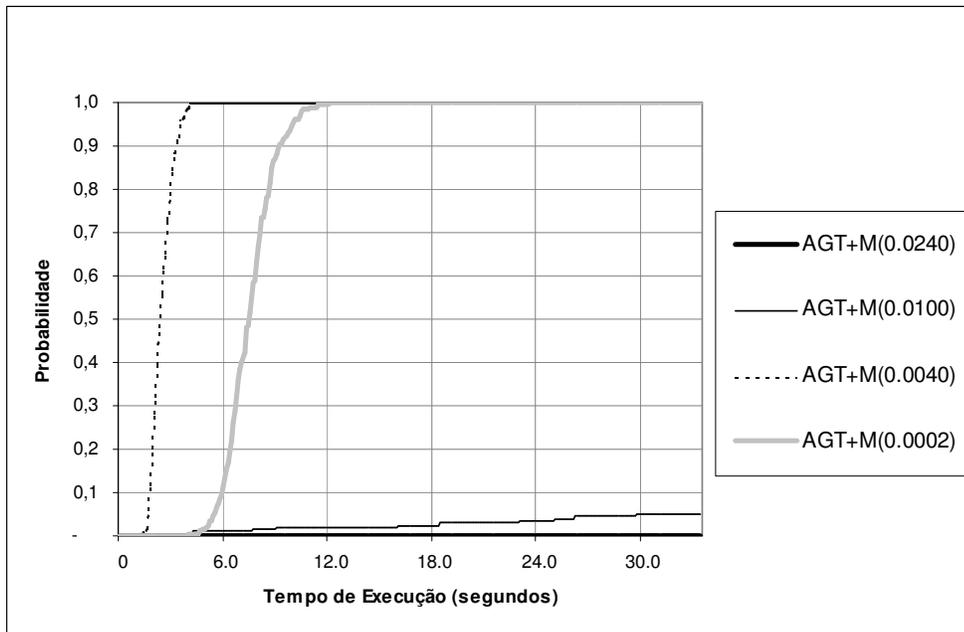


Figura 4.12 – AGT aplicado ao grafo A100, com diferentes valores para a taxa de mutação (MQ -alvo =0,5)

Nos experimentos das Figura 4.10 a 4.12, para os grafos com 40 ou mais vértices, a versão mais eficiente foi aquela que utilizou o valor para a taxa de mutação equivalente a

0,004 (da mesma forma que para o grafo A60, conforme Figura 4.1). Outra observação importante é que, para os grafos A80 e A100, a versão com a taxa de mutação igual a 0,024, apesar de utilizar um valor para a taxa de mutação inferior ao proposto no AGT original (para ambos os grafos a taxa de mutação deveria ser igual a 0,0280), não conseguiu alcançar o alvo. Isto deve-se à falta de uma convergência mínima necessária para se obter uma evolução da população ao longo das gerações.

Foram realizados novos experimentos utilizando-se o valor para a taxa de mutação equivalente a 0,004, com 10 execuções para cada versão aplicada a cada grafo, e os resultados são mostrados na Tabela 4.3 e Figuras 4.13 e 4.14. Nestes resultados pode-se observar, exceto para os grafos com 20 vértices, as melhores soluções obtidas foram superiores àquelas obtidas pelo AGT original. Entretanto, para os grafos com números maiores de vértices os resultados obtidos foram superiores sem, contudo, conseguirem obter a solução *best* para cada grafo. Os melhores resultados, obtidos pelo AGT ou AGT+M(0,004), são indicados em negrito na Tabela 4.3.

Grafo	Solução <i>Best</i>	Melhor Indivíduo AGT	Melhor Indivíduo AGT+M(0,004)	Média dos Melhores Indivíduos AGT+M(0,004)
A10	0,7083	0,7083	0,7083	0,6505
A20	0,7371	0,7371	0,7177	0,7070
A40	0,7087	0,5919	0,6508	0,6092
A60	0,7402	0,5695	0,6403	0,6213
A80	0,7780	0,5035	0,6778	0,6559
A100	0,8092	0,4765	0,6633	0,6445
E10	0,7083	0,7083	0,7083	0,7083
E20	0,7089	0,5957	0,4848	0,4609
E40	0,6485	0,4077	0,4787	0,4394
E60	0,7402	0,2165	0,3053	0,2721
E80	0,7075	0,2132	0,4002	0,3613
E100	0,6980	0,1544	0,3389	0,3172

Tabela 4.3 – Valores de *MQ* para soluções obtidas com as versões do AGT e AGT+M(0,004)

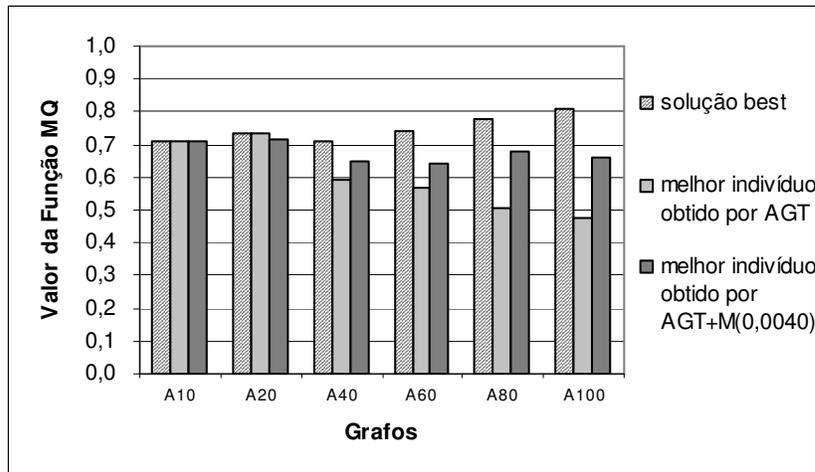


Figura 4.13 – Comparação de resultados obtidos pelo AGT e AGT+M(0,004) para grafos A10 a A100

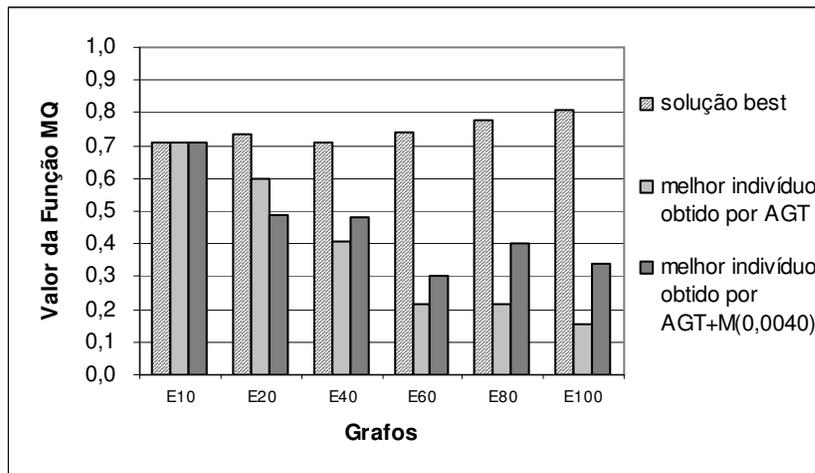


Figura 4.14 – Comparação de resultados obtidos pelo AGT e AGT+M(0,004) para grafos E10 a E100

A partir dos experimentos apresentados nesta seção, verificou-se que o desempenho do AGT é muito sensível ao valor definido para a taxa de mutação e, portanto, deve-se ter muito cuidado na definição deste parâmetro. Empiricamente foi verificado também que o valor para a taxa de mutação proposto no AGT original não é adequado, tendo em vista a redução na possibilidade dos genes dos melhores indivíduos serem transmitidos para gerações futuras. Além disso, para as instâncias utilizadas, mesmo ajustando-se o valor da taxa de mutação, o AGT não conseguiu obter a melhor solução para os grafos maiores. É possível concluir, então, que outros aspectos do AGT devem ser analisados para tentar obter melhores soluções na clusterização de grafos orientados.

4.1.2. Calibração do Operador Mutação

Conforme visto na Seção 3.2.2, a aplicação do operador mutação a um elemento de um indivíduo da população mantida pelo AGT corresponde à realização de uma alteração no seu valor. A consequência desta alteração é o transporte do elemento para um *cluster* c , obtido aleatoriamente no intervalo $1 \leq c \leq n$, onde n corresponde ao número de vértices do grafo. Este intervalo, definido no AGT proposto por Doval et al. [13], pode não ser adequado, tendo em vista que a possibilidade de criação de um novo *cluster*, durante a aplicação do operador mutação a um indivíduo, aumenta à medida que diminui o número total de *clusters* da solução representada pelo indivíduo.

Desta forma, a proposta de calibração do operador mutação consiste em, a aplicação do operador, considerando nc o número de *clusters* da melhor solução obtida até o momento, obter aleatoriamente o valor c no intervalo $1 \leq c \leq [nc + (10\% \text{ de } nc)]$ para que este seja o novo valor do elemento afetado. O objetivo desta alteração é permitir um ajuste progressivo no número máximo de *clusters* das soluções consideradas a cada geração, que deve convergir para um valor próximo ao número ótimo de *clusters*. A utilização da calibração no AGT será indicada acrescentando-se “+C” e os resultados dos experimentos que utilizaram esta proposta são apresentados na Tabela 4.4.

Grafo	Solução Best	Melhor Indivíduo AGT	Melhor Indivíduo AGT+C	Média dos Melhores Indivíduos AGT+C
A10	0,7083	0,7083	0,7083	0,6826
A20	0,7371	0,7371	0,7023	0,6929
A40	0,7087	0,5919	0,5260	0,4678
A60	0,7402	0,5695	0,5150	0,4874
A80	0,7780	0,5035	0,4369	0,4181
A100	0,8092	0,4765	0,4372	0,4080
E10	0,7083	0,7083	0,7083	0,7083
E20	0,7089	0,5957	0,5243	0,5061
E40	0,6485	0,4077	0,3228	0,2857
E60	0,7402	0,2165	0,1341	0,1093
E80	0,7075	0,2132	0,1435	0,1307
E100	0,6980	0,1544	0,1059	0,0972

Tabela 4.4 – Valores de *MQ* para soluções obtidas com as versões do AGT e AGT+C

Os gráficos referentes à comparação dos resultados do AGT e AGT+C mostrados na Tabela 4.4, são apresentados nas Figuras 4.15 e 4.16.

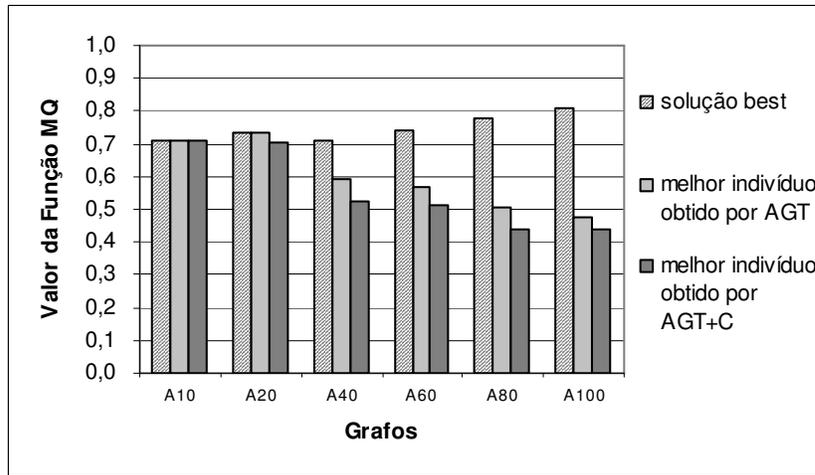


Figura 4.15 – Comparação de resultados obtidos pelo AGT e AGT+C para grafos A10 a A100

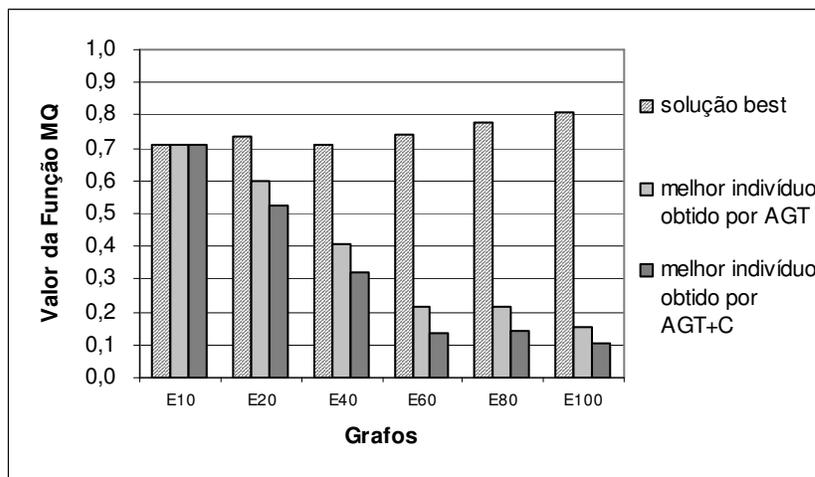


Figura 4.16 – Comparação de resultados obtidos pelo AGT e AGT+C para grafos E10 a E100

Com base nos resultados apresentados na Tabela 4.4 e Figuras 4.15 e 4.16, verifica-se que foram obtidos resultados piores do que o AGT original. Apesar disso, a calibração pode ser útil se combinada com outras propostas de alterações do AGT, o que será mostrado no Capítulo 5.

4.2. Novo Processo de Seleção de Indivíduos

Conforme apresentado no Capítulo 3, o processo de seleção de indivíduos de uma geração, utilizado pelo AGT original, para a constituição da população da geração seguinte corresponde ao método da roleta complementado com elitismo. Apesar de não existir um método de seleção que pode ser considerado o “melhor absoluto”, conforme visto no Capítulo 2, o objetivo desta seção é investigar de que forma a substituição do método de seleção do AGT original pelo método de torneio pode permitir ou não a obtenção de soluções melhores. No método de seleção por torneio utilizado nos experimentos desta seção, para a obtenção de cada indivíduo da população de uma geração, são selecionados de forma aleatória t indivíduos da população da geração anterior, com $t \geq 2$, e o indivíduo a ser reproduzido para a próxima geração será aquele que possuir o maior valor para a respectiva função de aptidão (MQ) dentre os t candidatos.

Nos experimentos realizados, cujos resultados são apresentados na Tabela 4.5 e Figuras 4.17 e 4.18, foram utilizados diferentes valores para o número de indivíduos que participam da seleção de cada indivíduo. A utilização do torneio na versão do AGT é indicada pelo acréscimo de “+T(t)”, onde t corresponde ao número de indivíduos candidatos que participam do torneio para a obtenção de cada indivíduo da população da próxima geração.

Grafo	Solução <i>Best</i>	Melhor Indivíduo AGT	Melhor Indivíduo AGT+T(2)	Melhor Indivíduo AGT+T(4)
A10	0,7083	0,7083	0,7083	0,7083
A20	0,7371	0,7371	0,7036	0,7371
A40	0,7087	0,5919	0,6782	0,6782
A60	0,7402	0,5695	0,6334	0,6313
A80	0,7780	0,5035	0,5019	0,4986
A100	0,8092	0,4765	0,4652	0,4728
E10	0,7083	0,7083	0,7083	0,7083
E20	0,7089	0,5957	0,5722	0,5722
E40	0,6485	0,4077	0,5196	0,5501
E60	0,7402	0,2165	0,2423	0,2366
E80	0,7075	0,2132	0,1926	0,2040
E100	0,6980	0,1544	0,1444	0,1456

Tabela 4.5 – Valores de MQ para soluções obtidas com as versões do AGT, AGT+T(2) e AGT+T(4)

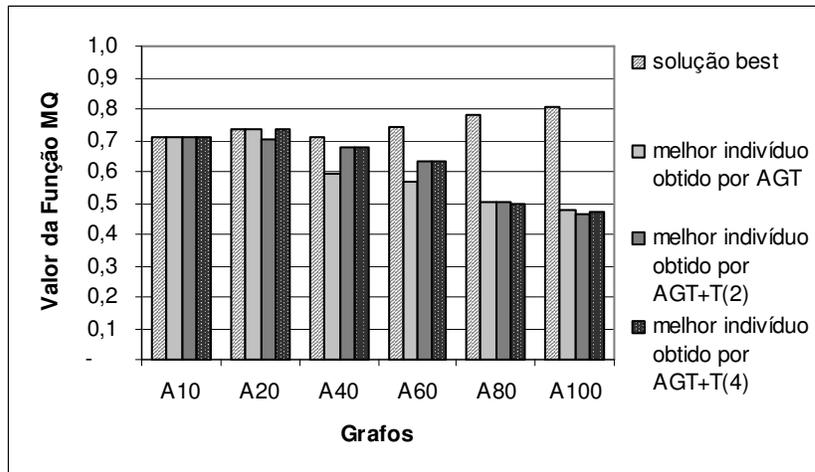


Figura 4.17 – Comparação de resultados obtidos pelo AGT, AGT+T(2) e AGT+T(4) para os grafos A10 a A100

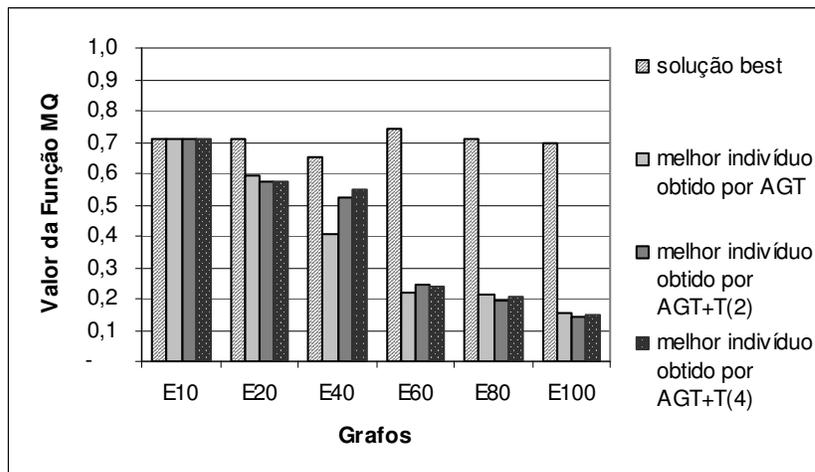


Figura 4.18 – Comparação de resultados obtidos pelo AGT, AGT+T(2) e AGT+T(4) para os grafos E10 a E100

Para cada experimento, utilizando-se os diferentes grafos, foram realizadas 10 execuções do algoritmo e é possível verificar, com base nos resultados mostrados na Tabela 4.5 e nas Figuras 4.17 e 4.18, que o melhor indivíduo obtido para cada versão com torneio foi superior, em alguns casos, em relação à melhor solução obtida pelo AGT que utiliza o método da roleta. Para avaliar mais precisamente, ainda que de forma empírica, os benefícios da substituição do método de seleção de roleta pelo método de torneio, a Figura 4.19 mostra o melhor indivíduo obtido a cada geração durante uma execução de cada uma das versões com torneio para um grafo com 100 vértices.

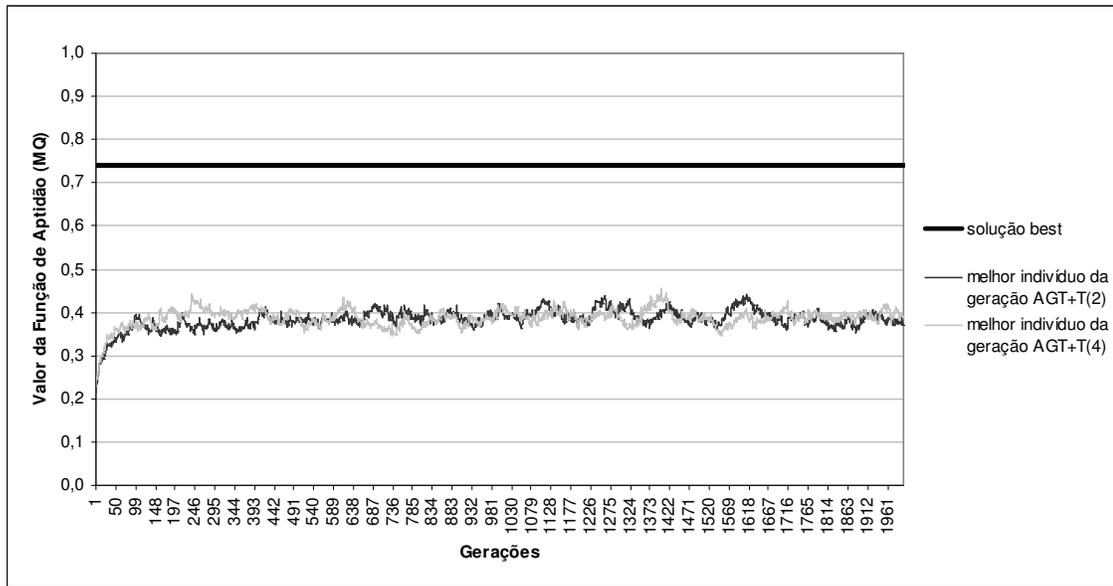


Figura 4.19 – Acompanhamento da evolução do melhor indivíduo ao longo de 2000 gerações do AGT+T(2) e AGT+T(4) utilizando o grafo A100

O gráfico da Figura 4.19 mostra uma menor dispersão do valor da função MQ para o melhor indivíduo obtido em cada geração do AGT+T(2) e AGT+T(4), em comparação com o experimento correspondente realizado com o AGT e apresentado na Figura 3.18 no Capítulo 3.

Dados estatísticos são mostrados na Tabela 4.6, apresentando a magnitude das flutuações dos valores da função MQ para os melhores indivíduos ao longo das gerações, através da indicação do valor médio de MQ nas gerações e do desvio padrão. Os valores para o AGT original correspondem aos resultados apresentados na Figura 3.26.

Versão	MQ do melhor indivíduo obtido em cada geração	
	Média Aritimética	Desvio Padrão
AGT	0,318176	0,038511
AGT+T(2)	0,383738	0,022144
AGT+T(4)	0,386689	0,020809

Tabela 4.6 – Valor médio de MQ ao longo das gerações e desvio padrão para as versões com e sem torneio utilizando o grafo A100

Através da Tabela 4.6 pode-se verificar que, para um mesmo grafo, apesar de existir uma considerável dispersão dos valores para MQ ao longo das gerações, ela é menor para as versões que utilizam torneio em comparação ao AGT original que utiliza o método da roleta. Assim, ao utilizar o método de torneio, existe uma convergência mais rápida para um

valor próximo ao valor médio do melhor indivíduo e uma maior manutenção das características do melhor indivíduo ao longo das gerações, em relação ao AGT original.

4.3. Estudo do Operador de Cruzamento

Outro aspecto que deve ser analisado no AGT é a representação utilizada para a codificação dos indivíduos. Conforme visto na Seção 3.2.2, o AGT utiliza a codificação *group-number*, em que cada indivíduo representa uma clusterização válida. Entretanto, conforme visto na Seção 3.2.1, esta codificação possui a propriedade de simetria, ou seja, diferentes indivíduos podem representar uma mesma solução de clusterização. Isto acarreta problemas na aplicação do operador de cruzamento como, por exemplo, a possibilidade de existirem dois pais diferentes que representam uma mesma clusterização e que, após a aplicação do operador de cruzamento, gerem dois indivíduos que representam duas clusterizações totalmente diferentes daquela que corresponde aos seus pais. Assim, as características dos dois indivíduos pais que foram selecionados para população podem não ser transmitidas para os filhos. Estendendo o problema, pode-se ter uma total descaracterização dos *clusters* dos indivíduos nos quais é aplicado o operador de cruzamento.

Com o objetivo de analisar as repercussões na qualidade das soluções obtidas com o AGT devido à simetria na representação dos indivíduos, foi desenvolvido um novo operador de cruzamento que elimina o problema de simetria nesta etapa do AGT. Neste novo operador de cruzamento é utilizada a idéia de representação dos indivíduos do algoritmo genético de agrupamento (GGA – *Grouping Genetic Algorithm*), proposto por Falkenauer [16, 17], e descrito na Seção 3.2.1. As referências à utilização do novo operador de cruzamento nos experimentos com o AGT, em substituição ao operador de cruzamento original, serão indicadas através do acréscimo de “+G”.

Apesar de utilizar a idéia do GGA, no AGT+G desenvolvido neste trabalho a representação dos indivíduos não foi modificada, ficando a eliminação da simetria restrita ao novo operador de cruzamento. A descrição do operador de cruzamento utilizado no AGT+G é apresentada na Figura 4.20.

Algoritmo para Cruzamento GGA entre dois Indivíduos, A e B

1. Obtém as partes dos *clusters* para os indivíduos A e B
2. Para cada um dos indivíduos A e B, seleciona um trecho da parte dos *clusters*, os segmentos de cruzamento de cada indivíduo (SCA e SCB, respectivamente), e identifica os *clusters* contidos nos segmentos, armazenando esta informação separadamente.
3. Os *clusters* do SCA são inseridos na parte de *clusters* do Indivíduo B e os valores dos *clusters* associados aos vértices Indivíduo B que estão nos *clusters* de SCA são alterados para refletirem os novos *clusters* adicionados. São utilizadas novas numerações para os *clusters* inseridos nos indivíduos, para que não ocorra uma união entre os *clusters* inseridos com os *clusters* já existentes no indivíduo.
4. O passo 3 é feito, inserindo-se o SCB no indivíduo A.

Figura 4.20 – Algoritmo do operador de cruzamento do AGT+G

Um exemplo de aplicação do operador de cruzamento utilizado no AGT+G é fornecido na Figura 4.21 em que, partindo-se de dois indivíduos pais, são gerados dois indivíduos filhos.

Conforme apresentado na Tabela 4.7 e Figuras 4.22 e 4.23, os resultados experimentais obtidos com as 10 execuções do AGT+G mostraram que a substituição do operador de cruzamento piorou a qualidade das soluções obtidas, em comparação às obtidas pelo AGT original. Entretanto, tais resultados não devem ser considerados definitivos no que diz respeito à não utilização do novo operador. Tendo em vista que outras características do AGT possuem forte influência nos resultados como, por exemplo, a taxa de mutação, e uma conclusão mais consistente só poderá ser obtida quando combinações de diferentes propostas de alterações do AGT forem consideradas concomitantemente com o novo operador de cruzamento, o que será realizado no Capítulo 5.

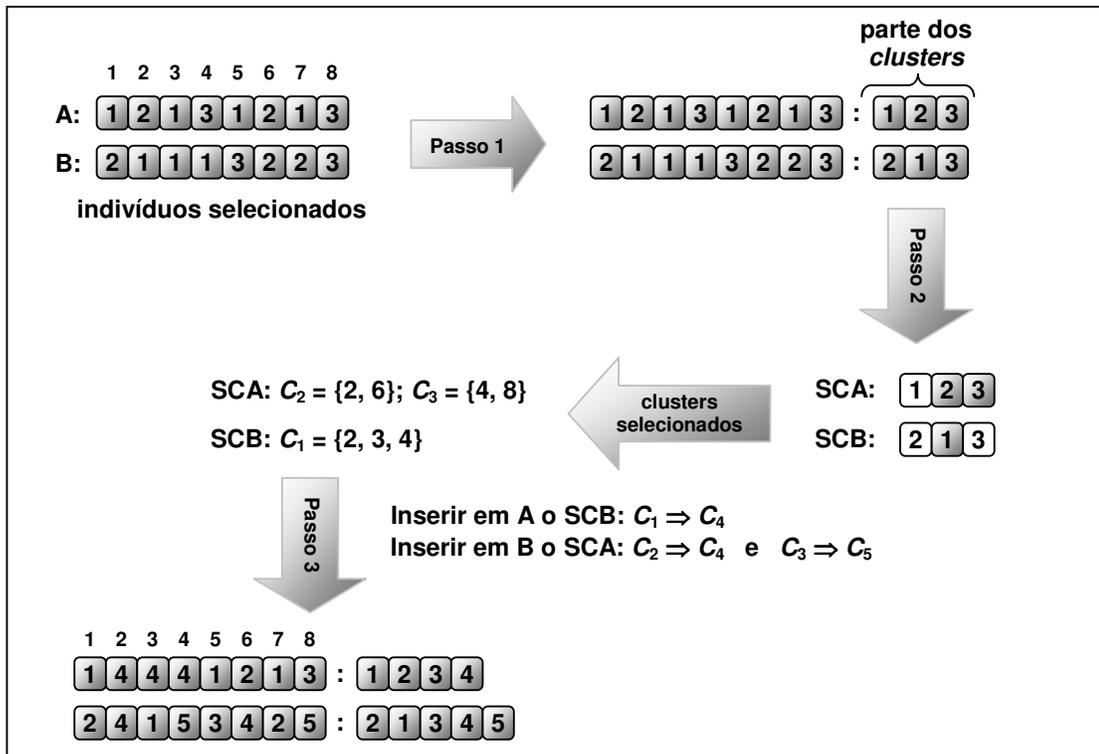


Figura 4.21 – Exemplo de utilização do operador de cruzamento do AGT+G

Grafo	Solução Best	Melhor Indivíduo AGT	Melhor Indivíduo AGT+G	Média dos Melhores Indivíduos AGT+G
A10	0,7083	0,7083	0,7083	0,7083
A20	0,7371	0,7371	0,6620	0,6456
A40	0,7087	0,5919	0,5283	0,4665
A60	0,7402	0,5695	0,4921	0,4852
A80	0,7780	0,5035	0,4476	0,4240
A100	0,8092	0,4765	0,4236	0,4046
E10	0,7083	0,7083	0,7083	0,7083
E20	0,7089	0,5957	0,5681	0,5317
E40	0,6485	0,4077	0,3111	0,2718
E60	0,7402	0,2165	0,1311	0,1106
E80	0,7075	0,2132	0,1622	0,1315
E100	0,6980	0,1544	0,1187	0,0985

Tabela 4.7 – Valores de MQ para soluções obtidas com as versões do AGT e AGT+G

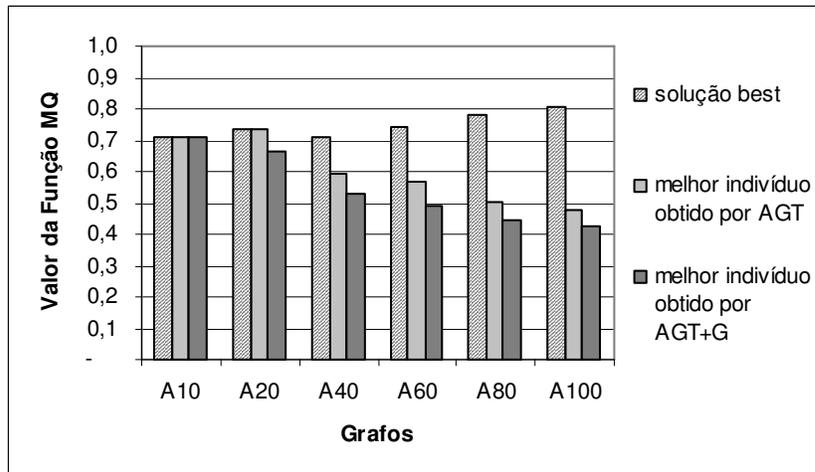


Figura 4.22 – Comparação de resultados obtidos pelo AGT e AGT+G para grafos A10 a A100

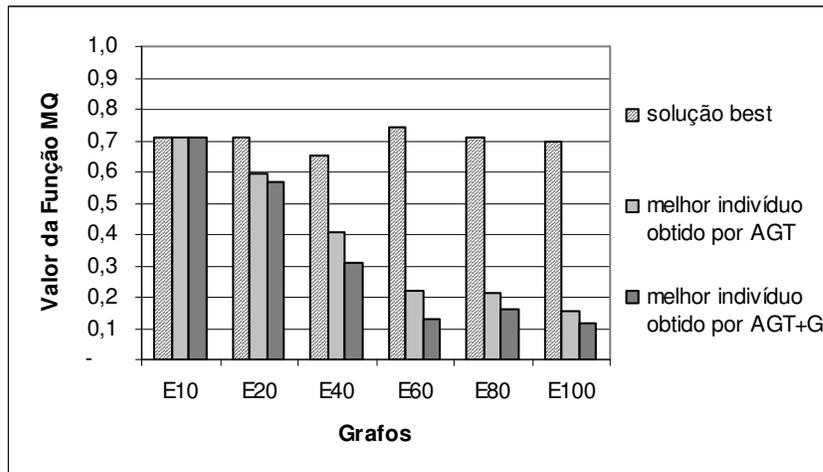


Figura 4.23 – Comparação de resultados obtidos pelo AGT e AGT+G para grafos E10 a E100

Uma observação importante sobre o novo operador de cruzamento diz respeito à convergência dos indivíduos da população ao longo das gerações. Com base na Figura 4.24, Tabela 4.8 e Figura 3.26 (no Capítulo 3) pode-se verificar que a dispersão da melhor solução obtida AGT+G foi reduzida em relação ao AGT original aplicado ao mesmo grafo A100, o que reforça a necessidade de aprofundar no estudo do AGT+G, o que será feito no Capítulo 5.

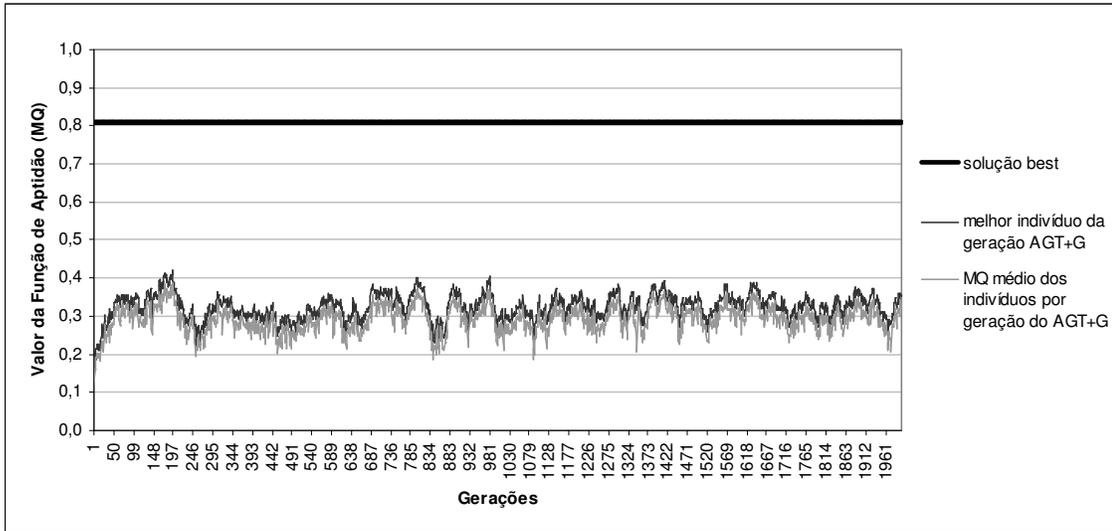


Figura 4.24 – Acompanhamento da evolução do melhor indivíduo ao longo de 2.000 gerações do AGT+G utilizando o grafo A100

Versão	MQ do melhor indivíduo obtido em cada geração	
	Média Aritimética	Desvio Padrão
AGT	0,318176	0,038511
AGT+G	0,324205	0,033051

Tabela 4.8 – Valor médio de *MQ* e desvio padrão para versões AGT e AGT+G para grafo A100

4.4. Procedimento de Busca Local

Existem muitos trabalhos que utilizam um módulo de busca local, também referenciado na literatura como *hill climbing*, para tentar melhorar a qualidade das soluções obtidas por um AG. Nesta abordagem, durante o processo evolutivo do AGT, enquanto o melhor indivíduo obtido a cada iteração não corresponder à solução ótima para a clusterização, pode-se tentar melhorá-lo através da aplicação de um procedimento de busca local. Basicamente, o procedimento de busca local consiste em uma investigação da vizinhança de uma solução, com o objetivo de obter um ótimo local, e normalmente é realizada através da substituição de valores dos genes de um indivíduo por valores que podem ser definidos aleatoriamente ou obtidos conforme alguma característica do problema e da instância.

No AGT com busca local proposto neste trabalho, e referenciado com a indicação “+B”, durante o procedimento de busca local parte-se de uma clusterização $C_0 \in C$, obtida

pelo AGT, e são analisadas várias clusterizações $C_1, C_2, C_3, \dots, C_n \in C$ da vizinhança de C_0 . A clusterização C_0 corresponde à melhor solução obtida em cada geração e cada uma das outras n clusterizações é obtida através de uma única modificação no identificador do *cluster* de algum vértice da clusterização obtida na iteração anterior. Os passos principais da busca local proposta neste trabalho são apresentados no algoritmo da Figura 4.25.

```

Algoritmo BuscaLocal (entrada: clusterização  $C_0$ )

1.  $MQ_C \leftarrow$  função de aptidão da clusterização  $C_0$ ;
2. Para  $i \leftarrow 1$  até  $n$  faça // para todos os  $n$  vértices do grafo
3.   Para cada cluster  $q \in Q$  computar  $nA_i(q)$  que correspondente à
      soma do total de arcos incidentes do vértice  $v_i$  nos
      vértices do cluster  $q$  e o total de arcos incidentes dos
      vértices de  $q$  no vértice  $v_i$ ;
4.    $qMax_i \leftarrow q$  tal que  $nA_i(q)$  seja o maior valor obtido em 3;
5.   Se  $qMax_i \neq cluster(v_i)$  então
6.      $NovoMQ_C \leftarrow$  valor da aptidão da clusterização  $C_i$ 
       considerando o vértice  $v_i$  contido no cluster  $qMax_i$ ;
7.   Fim se;
8.   Se  $NovoMQ_C > MQ_C$  então
9.     Substituir o cluster( $v_i$ ) na clusterização  $C_{i-1}$  por  $qMax_i$ ;
10.     $MQ_C \leftarrow$  função de aptidão da nova clusterização  $C_i$ ;
11.   Senão
12.     Manter  $v_i$  no cluster original da clusterização  $C_{i-1}$ ;
13.   Fim se;
14. Fim para;
15. Retornar a clusterização  $C_n$ .

```

Figura 4.25 – Algoritmo de busca local aplicado a uma clusterização C_0

Foi realizada uma bateria de experimentos para o AGT+B e as melhores soluções obtidas após 10 execuções para cada grafo são apresentadas na Tabela 4.9 e Figuras 4.26 e 4.27, comparando-as com as melhores soluções obtidas após 10 execuções do AGT. Nestes experimentos, a versão AGT+B conseguiu obter a solução *best* para todos os grafos

utilizados, de 10 a 100 vértices. Vale ressaltar que o valor médio para a função MQ dos melhores indivíduos obtidos nas 10 execuções do AGT+B foi muito superior ao melhor indivíduo obtido nas 10 execuções do AGT para os maiores grafos.

Grafo	Solução Best	Melhor Indivíduo AGT	Melhor Indivíduo AGT+B	Média dos Melhores Indivíduos AGT+B
A10	0,7083	0,7083	0,7083	0,7083
A20	0,7371	0,7371	0,7371	0,7371
A40	0,7087	0,5919	0,7087	0,6754
A60	0,7402	0,5695	0,7402	0,7339
A80	0,7780	0,5035	0,7780	0,7372
A100	0,8092	0,4765	0,8092	0,7827
E10	0,7083	0,7083	0,7083	0,6505
E20	0,7089	0,5957	0,7089	0,6974
E40	0,6485	0,4077	0,6485	0,6201
E60	0,7402	0,2165	0,7402	0,7402
E80	0,7075	0,2132	0,7075	0,6721
E100	0,6980	0,1544	0,6980	0,6564

Tabela 4.9 – Valores de MQ para soluções obtidas com as versões do AGT e AGT+B

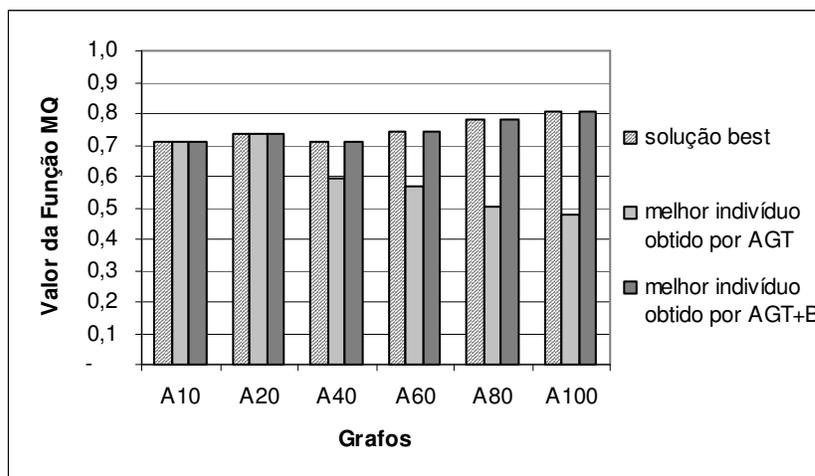


Figura 4.26 – Comparação de resultados obtidos pelo AGT e AGT+B para grafos A10 a A100

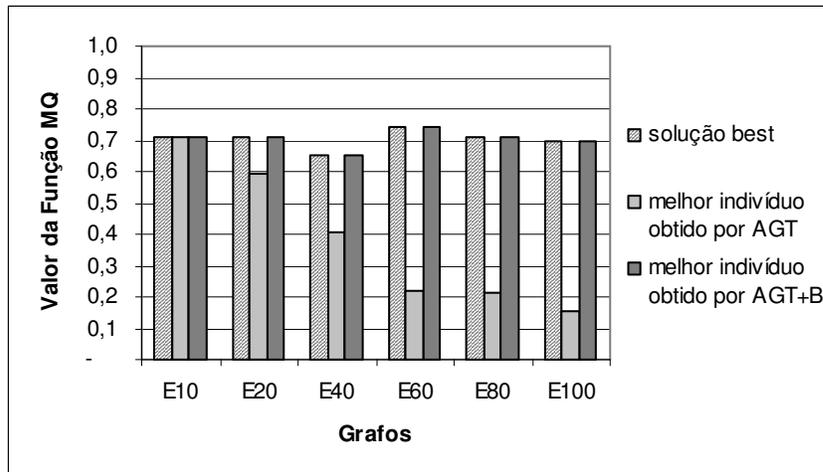


Figura 4.27 – Comparação de resultados obtidos pelo AGT e AGT+B para grafos E10 a E100

Com base na qualidade das soluções obtidas para os grafos de 10 a 100 vértices, estendemos os experimentos para grafos maiores, com 160, 200, 500 e 1000 vértices. Os resultados obtidos para os grafos maiores, com 10 execuções da versão original do AGT e da versão AGT+B para cada grafo, são apresentados na Tabela 4.10 e Figuras 4.28 e 4.29. Nestes experimentos observa-se novamente que a utilização da busca local permitiu a obtenção de resultados muito superiores do que a sua não utilização. Entretanto, para estas novas instâncias, a solução *best* foi obtida apenas para os grafos A160 (160 vértices) e E200 (200 vértices).

Grafo	Solução <i>Best</i>	Melhor Indivíduo AGT	Melhor Indivíduo AGT+B	Média dos Melhores Indivíduos AGT+B
A160	0,9090	0,5401	0,9090	0,8903
A200	0,8783	0,4198	0,8157	0,8066
A500	0,8785	0,3593	0,8314	0,8182
A1000	0,9415	0,4486	0,8708	0,8563
E160	0,7364	0,0597	0,7184	0,6922
E200	0,7662	0,0669	0,7662	0,7544
E500	0,8759	0,0079	0,7869	0,7666
E1000	0,8865	-0,0030	-0,0011	-0,0025

Tabela 4.10 – Valores de *MQ* para soluções obtidas com as versões do AGT original e AGT+B para instâncias grandes

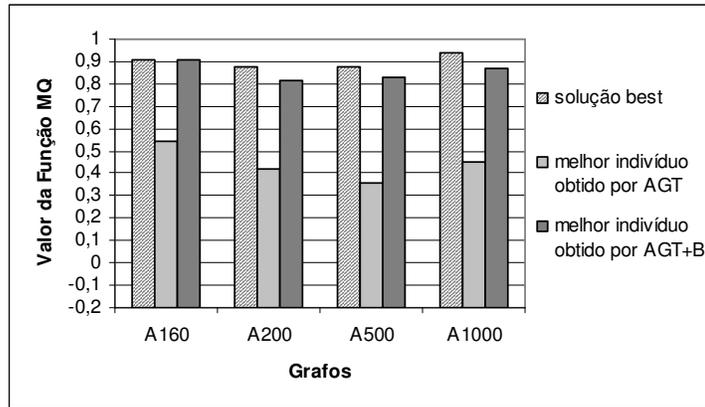


Figura 4.28 – Comparação de resultados obtidos pelo AGT e AGT+B para grafos A160 a A1000

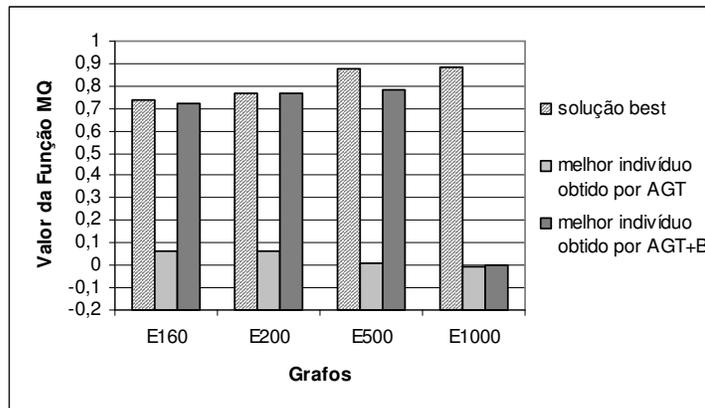


Figura 4.29 – Comparação de resultados obtidos pelo AGT e AGT+B para grafos E160 a E1000

Com relação aos experimentos realizados é importante observar que foram obtidos resultados muito ruins, com valores negativos, para a função MQ das melhores soluções obtidas com o grafo E1000 (1000 vértices), mesmo pelo AGT+B. Neste caso, a investigação do espaço de busca ao longo das gerações de cada uma das execuções do AGT+B não obteve, em momento algum, um indivíduo próximo à solução *best* que permitisse ao procedimento de busca local alcançá-la. Portanto conclui-se que, para se obter a eficácia no procedimento de busca local, o AGT deve ser capaz de gerar indivíduos que representem soluções contidas na região de busca próxima à da solução *best* e isto se torna mais difícil à medida que as instâncias utilizadas são maiores, tendo em vista que o espaço de busca cresce exponencialmente em relação ao tamanho do grafo. Como somente a utilização da busca local não é suficiente para gerar soluções de qualidade, uma sugestão é a sua utilização no AGT em conjunto com outras adaptações, o que será feito no Capítulo 5.

4.5. Diversificação da população

O procedimento de diversificação da população, proposto neste trabalho, corresponde à geração de uma nova população de indivíduos que substitui a população de uma geração sempre que o valor médio da função de aptidão (MQ) dos indivíduos da população não for incrementado durante um determinado número de gerações. Durante a execução do procedimento de diversificação, a geração da nova população não é realizada de forma aleatória, mas sim, aproveitando características do melhor indivíduo obtido pelo algoritmo até a geração corrente. A Figura 4.30 apresenta o algoritmo referente ao procedimento de diversificação.

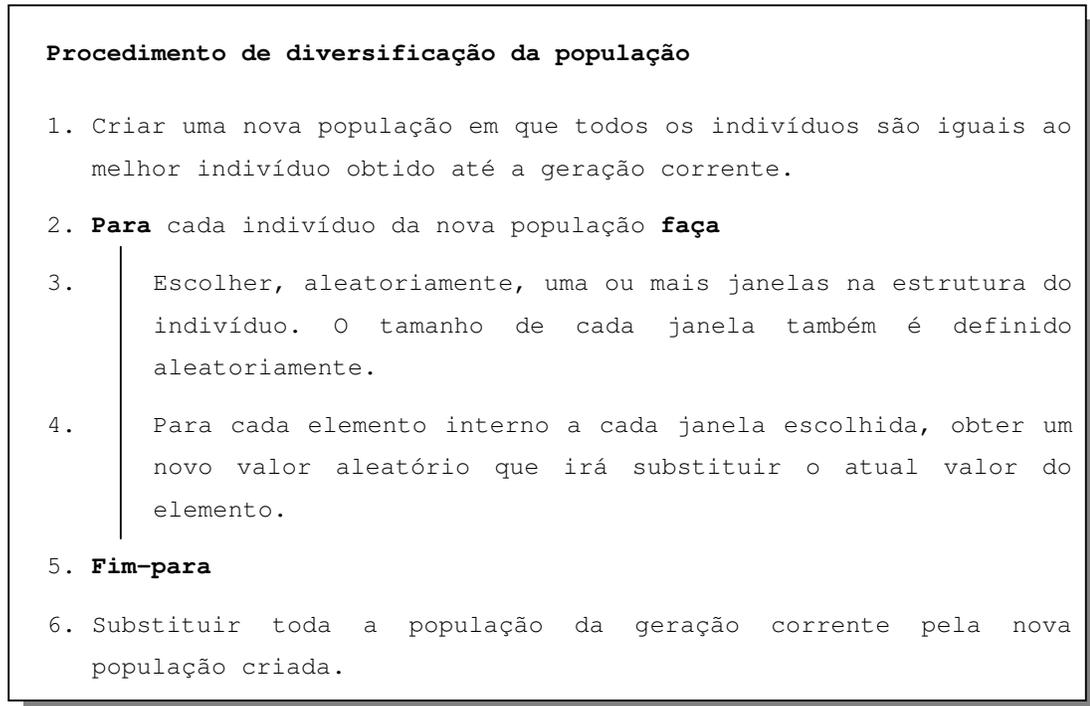


Figura 4.30 – Algoritmo de diversificação da população

A proposta da inserção do procedimento de diversificação surgiu a partir da constatação de que quando o valor médio da função de aptidão dos indivíduos não melhora durante um determinado número de gerações, a evolução da população pode ficar comprometida. Isto pode ocorrer devido a um dos seguintes fatores:

- *convergência prematura da população*: uma grande quantidade de indivíduos da população possui características semelhantes entre si e, portanto, correspondem a soluções de uma região específica do espaço de busca. Conforme visto na Seção 2.1, a falta de investigação de diferentes regiões do espaço de busca reduz a possibilidade de obtenção da solução ótima;
- *elevada dispersão dos indivíduos*: cada indivíduo da população sofre muitas alterações em um número reduzido de gerações, impedindo que as boas qualidades dos melhores indivíduos sejam transmitidas para as gerações seguintes e, dessa forma, a população não evolui.

A Figura 4.31 ilustra a aplicação do procedimento de diversificação para a clusterização de um grafo com 8 vértices e uma população com 6 indivíduos. O procedimento de diversificação foi considerado neste trabalho como uma geração do processo evolutivo e, após a sua aplicação, é realizado o passo de seleção dos indivíduos para constituírem a população da geração seguinte sobre a qual serão aplicados o operador mutação e o operador de cruzamento.

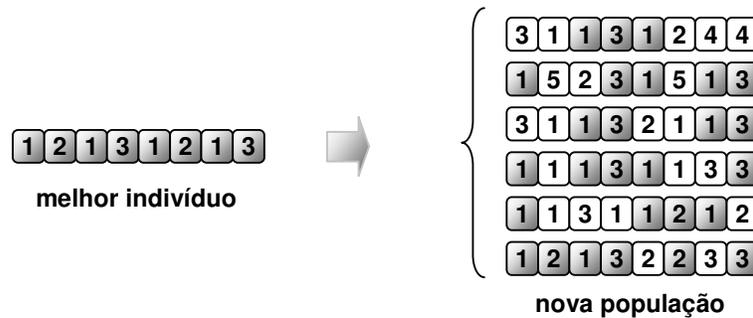


Figura 4.31 – Exemplo de aplicação do procedimento de diversificação da população

Um parâmetro a ser considerado nesta proposta é o número total de gerações sem que ocorra uma melhora no valor médio da função de aptidão, a partir do qual o procedimento de diversificação da população deverá ser ativado. Assim, a utilização deste procedimento no AGT será referenciada neste trabalho através da indicação “+D(*d*)” onde *d* é o número de iterações do algoritmo (gerações), sem a melhora no valor médio de *MQ*, para que o procedimento de diversificação seja ativado.

Os resultados dos experimentos preliminares realizados para avaliar a inclusão da proposta de diversificação são apresentados na Tabela 4.11 e Figuras 4.32 e 4.33. Nestes experimentos, cada versão foi executada 10 vezes para cada grafo, e são mostrados os melhores resultados obtidos para cada versão.

Grafo	Solução Best	Melhor Indivíduo AGT	Melhor Indivíduo AGT+D(10)	Melhor Indivíduo AGT+D(25)
A10	0,7083	0,7083	0,7083	0,7083
A20	0,7371	0,7371	0,7036	0,7124
A40	0,7087	0,5919	0,6766	0,6324
A60	0,7402	0,5695	0,6704	0,6276
A80	0,7780	0,5035	0,6955	0,6440
A100	0,8092	0,4765	0,6875	0,6204
E10	0,7083	0,7083	0,7083	0,7083
E20	0,7089	0,5957	0,6134	0,5722
E40	0,6485	0,4077	0,5145	0,4534
E60	0,7402	0,2165	0,3138	0,3182
E80	0,7075	0,2132	0,3994	0,3863
E100	0,6980	0,1544	0,3879	0,3213

Tabela 4.11 – Valores de MQ para soluções obtidas com as versões do AGT original, AGT+D(10) e AGT+D(25)

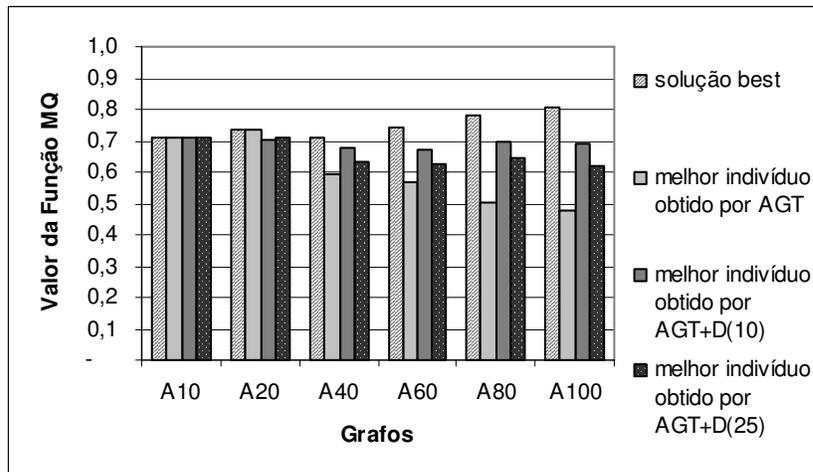


Figura 4.32 – Comparação de resultados obtidos pelo AGT, AGT+D(10) e AGT+D(25) para grafos A10 a A100

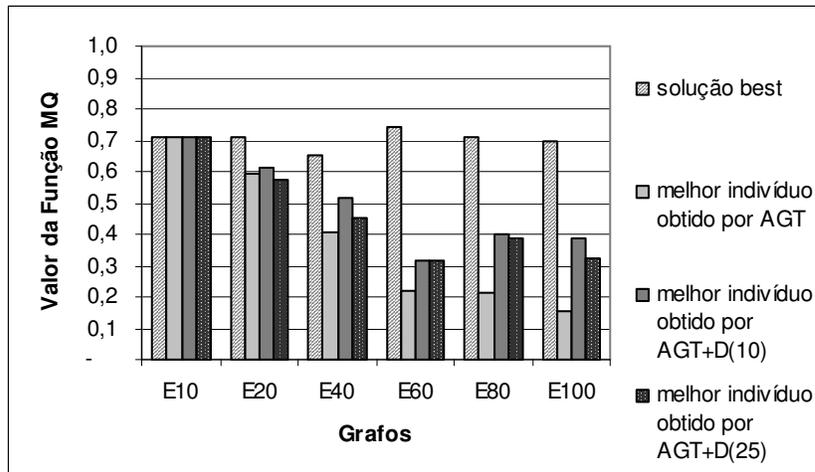


Figura 4.33 – Comparação de resultados obtidos pelo AGT, AGT+D(10) e AGT+D(25) para grafos E10 a E100

Com base nestes resultados, pode-se verificar que, para todas as instâncias com 40 ou mais vértices, as versões com a diversificação obtiveram resultados superiores ao AGT. Além disso, para os grafos maiores, a versão AGT+D(10) obteve melhores resultados do que AGT+D(25). Uma explicação para esta última constatação pode estar relacionada ao número de execuções do procedimento de diversificação da população: com um número menor de gerações sem evolução da população para que a sua ativação ocorra, o número de ativações da diversificação é muito maior no AGT+D(10) do que no AGT+(25). Os gráficos das Figuras 4.34 a 4.45 apresentam a evolução da população ao longo das gerações para a melhor execução do AGT com diversificação para os grafos A100 e E100, considerando a aplicação da diversificação após 10 gerações (Figuras 4.34 a 4.39) e 25 gerações (Figuras 4.40 a 4.45).

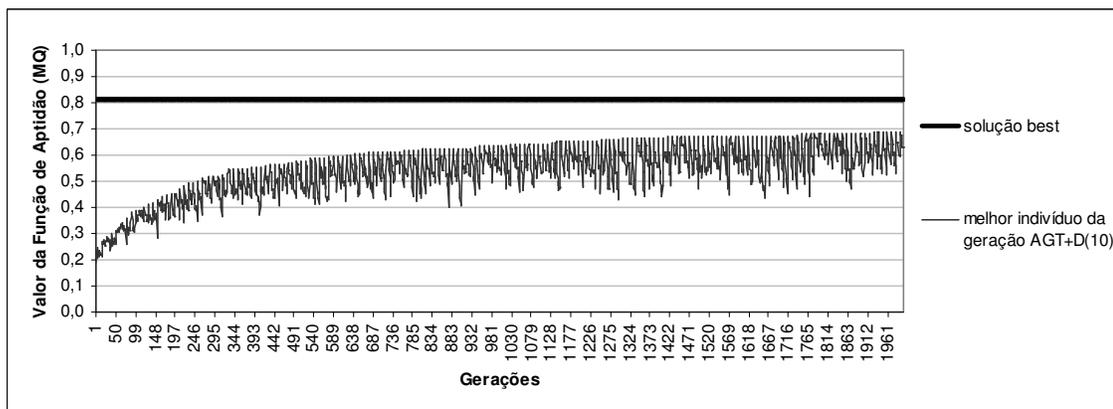


Figura 4.34 – Acompanhamento da evolução do melhor indivíduo ao longo de 2.000 gerações do AGT+D(10), para o grafo A100

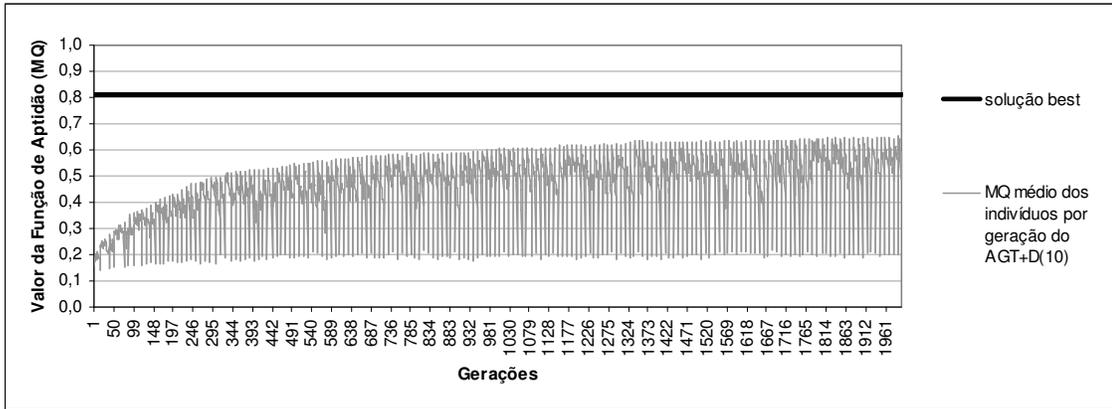


Figura 4.35 – Acompanhamento da evolução do valor médio para a função de aptidão dos indivíduos de cada uma das 2.000 gerações do AGT+D(10), para o grafo A100

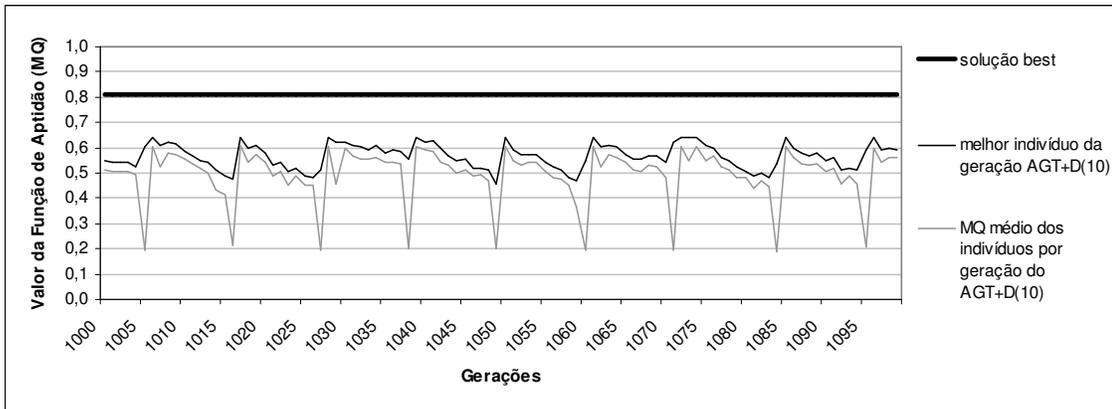


Figura 4.36 – Detalhes das Figuras 4.34 e 4.35 referentes às evoluções do melhor indivíduo e do valor médio para a função de aptidão dos indivíduos entre as gerações 1.000 e 1.100

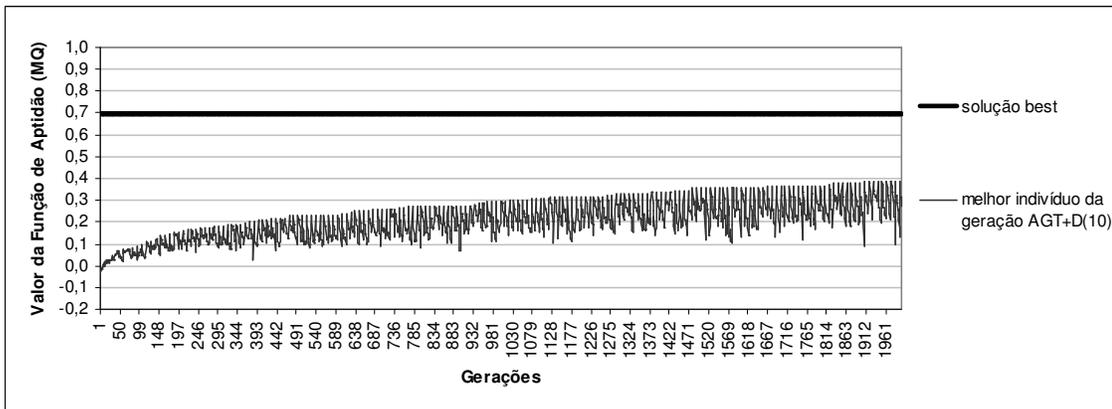


Figura 4.37 – Acompanhamento da evolução do melhor indivíduo ao longo de 2.000 gerações do AGT+D(10), para o grafo E100

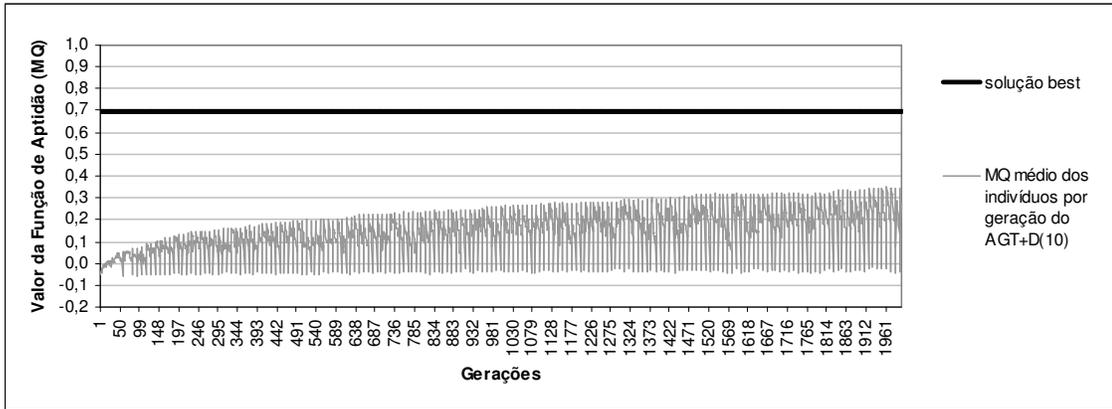


Figura 4.38 – Acompanhamento da evolução do valor médio para a função de aptidão dos indivíduos de cada uma das 2.000 gerações do AGT+D(10), para o grafo E100

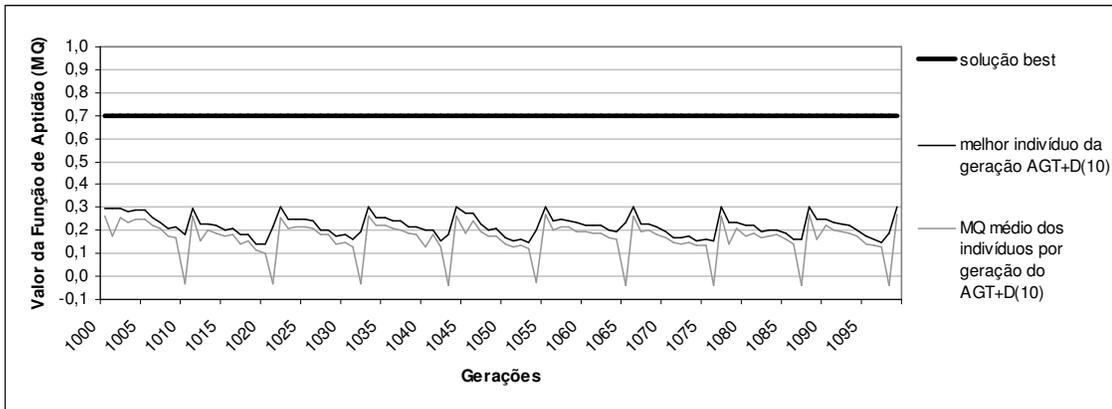


Figura 4.39 – Detalhes das Figuras 4.37 e 4.38 referentes às evoluções do melhor indivíduo e do valor médio para a função de aptidão dos indivíduos entre as gerações 1.000 e 1.100

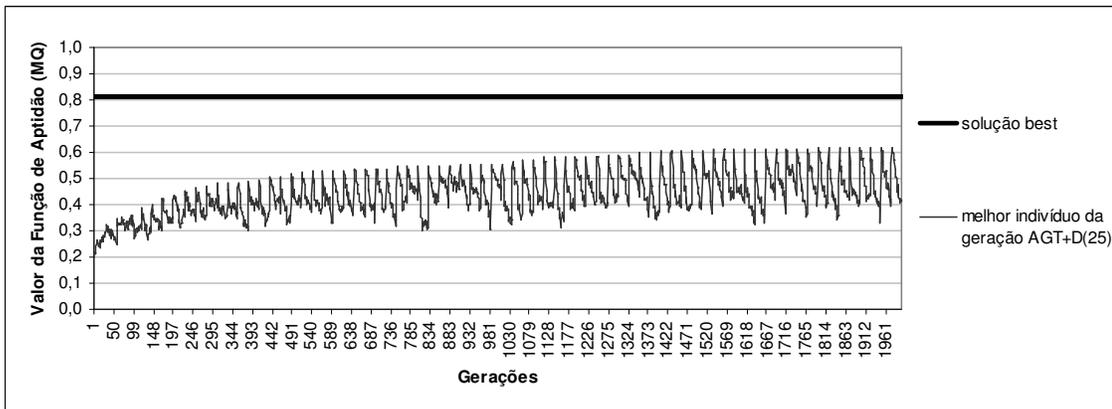


Figura 4.40 – Acompanhamento da evolução do melhor indivíduo ao longo de 2.000 gerações do AGT+D(25), para o grafo A100

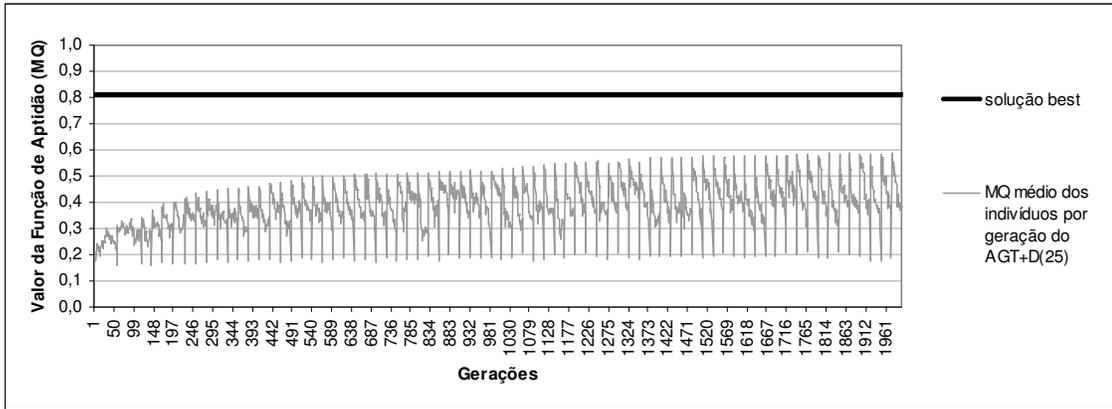


Figura 4.41 – Acompanhamento da evolução do valor médio para a função de aptidão dos indivíduos de cada uma das 2.000 gerações do AGT+D(25), para o grafo A100

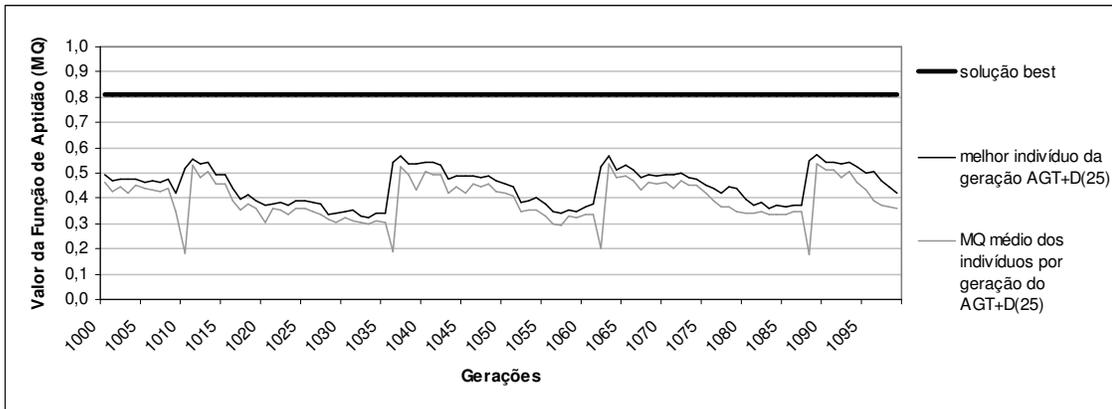


Figura 4.42 – Detalhes das Figuras 4.40 e 4.41 referentes às evoluções do melhor indivíduo e do valor médio para a função de aptidão dos indivíduos entre as gerações 1.000 e 1.100

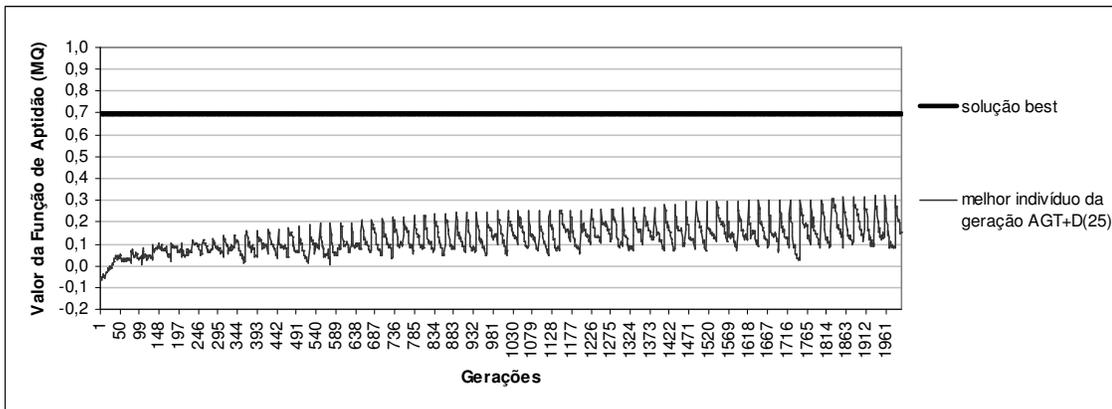


Figura 4.43 – Acompanhamento da evolução do melhor indivíduo ao longo de 2.000 gerações do AGT+D(25), para o grafo E100

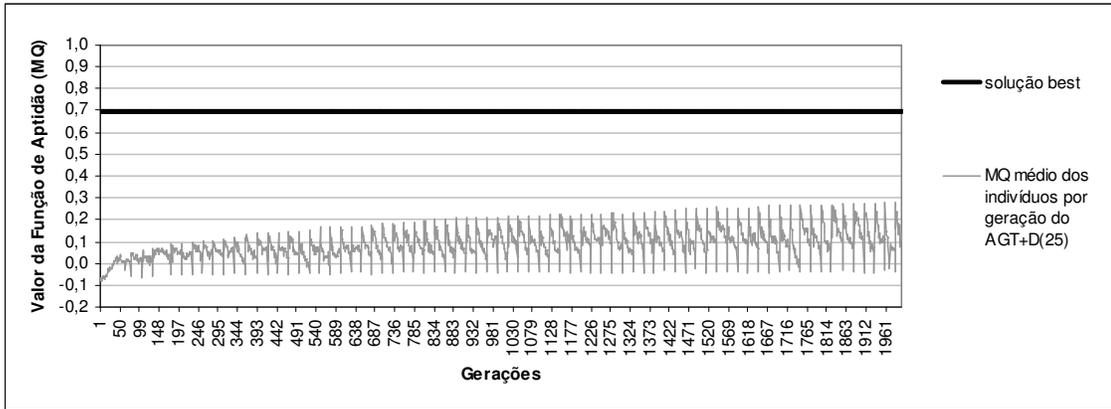


Figura 4.44 – Acompanhamento da evolução do valor médio para a função de aptidão dos indivíduos de cada uma das 2.000 gerações do AGT+D(25), para o grafo E100

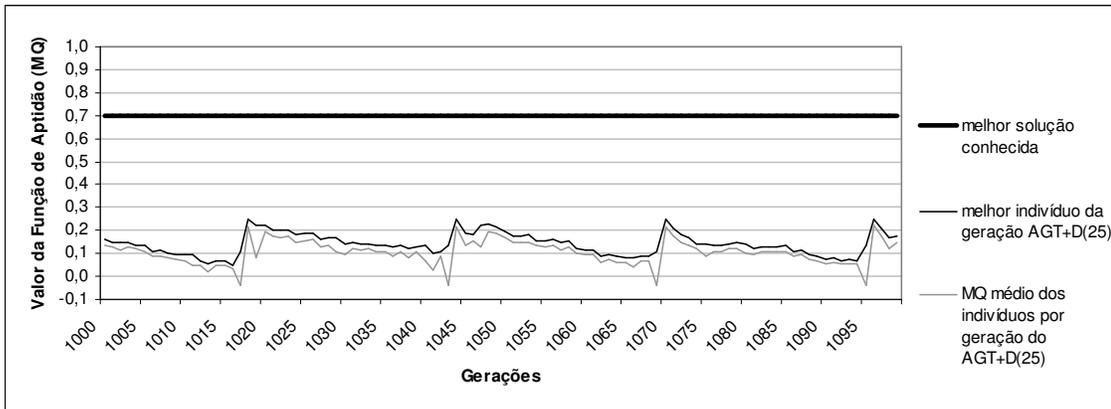


Figura 4.45 – Detalhes das Figuras 4.43 e 4.44 referentes às evoluções do melhor indivíduo e do valor médio para a função de aptidão dos indivíduos entre as gerações 1.000 e 1.100

A partir dos resultados apresentados nas Figuras 4.34 a 4.45 verifica-se que a inclusão do procedimento de diversificação da população ao AGT permitiu um aumento gradativo do valor do melhor indivíduo obtido a cada geração e do valor médio para a função MQ considerando todos indivíduos da população em cada geração. Isto permitiu que ocorresse uma evolução da população ao longo das gerações, em contraste com os gráficos de acompanhamento dos valores da função MQ para o AGT, apresentados na Seção 3.4.

Observando os detalhes dos gráficos, apresentados nas Figuras 4.36, 4.39, 4.42 e 4.45, verifica-se ocorrência de fortes crescimentos e quedas em alguns pontos das curvas dos valores de MQ para o melhor indivíduo e para o valor médio. As quedas correspondem aos valores referentes ao valor médio da função de aptidão dos indivíduos da população imediatamente após cada ativação do procedimento de diversificação, e são seguidos de fortes

crescimentos que correspondem à geração imediatamente seguinte, após o processo de seleção e a aplicação dos operadores genéticos. É importante observar também que, após o um crescimento da curva que segue o momento da diversificação, existe uma queda gradativa nos valores da função de aptidão do melhor indivíduo e no valor médio dos indivíduos. Este comportamento, que é recorrente após cada aplicação do procedimento de diversificação, deve-se ao caráter de dispersão do AGT, que possui todas as suas características originais entre cada duas aplicações consecutivas do procedimento de diversificação.

Apesar da verificação deste comportamento ascendente e descendente do valor de *MQ* ao longo das gerações, se forem considerados períodos maiores observa-se que ocorre um aumento gradativo na qualidade dos melhores indivíduos obtidos. Deve-se observar também que, para as 2.000 iterações das melhores execuções do AGT+D(10) para os grafos A100 e E100, os melhores indivíduos foram obtidos nas gerações 1960 e 1930, respectivamente. Considerando as 10 execuções do AGT+D(10) e AGT+D(25), as médias das iterações em que o melhor indivíduo foi obtido para cada grafo são apresentadas na Tabela 4.12.

Grafo	Solução Best	Total de Gerações	Versão AGT+D(10)		Versão AGT+D(25)	
			MQ Médio ¹	Média das Gerações ²	MQ Médio ¹	Média das Gerações ²
A10	0,7083	200	0,6955	25,3 (13%)	0,7083	50,4 (25%)
A20	0,7371	400	0,6963	229,3 (57%)	0,7010	328,1 (82%)
A40	0,7087	800	0,6491	721,4 (90%)	0,5934	724,1 (91%)
A60	0,7402	1.200	0,6545	1.179,6 (98%)	0,6057	1.160,0 (97%)
A80	0,7780	1.800	0,6642	1.566,1 (87%)	0,6168	1.549,6 (86%)
A100	0,8092	2.000	0,6621	1.970,5 (99%)	0,6024	1.944,2 (97%)
E10	0,7083	200	0,7083	45,0 (23%)	0,7083	70,0 (35%)
E20	0,7089	400	0,5613	312,5 (78%)	0,5275	280,6 (70%)
E40	0,6485	800	0,4764	716,0 (90%)	0,4065	725,1 (91%)
E60	0,7402	1.200	0,2898	1.093,9 (91%)	0,2678	1.159,8 (97%)
E80	0,7075	1.800	0,3807	1.563,3 (87%)	0,3416	1.582,9 (88%)
E100	0,6980	2.000	0,3648	1.968,3 (98%)	0,2992	1.955,7 (98%)

¹ Valor médio da função *MQ* para os melhores indivíduos obtidos nas 10 execuções

² Valor médio das gerações em que o melhor indivíduo foi obtido nas 10 execuções. Entre parênteses, o percentual do total de gerações.

Tabela 4.12 – *MQ* médio dos melhores indivíduos obtidos e valor médio das iterações em que o melhor indivíduo foi obtido nas 10 execuções das versões AGT+D(10) e AGT+D(25)

A partir dos dados apresentados na Tabela 4.12 é possível constatar que, para cada um dos grafos com 20 ou mais vértices, a geração em que o melhor indivíduo foi obtido é muito próxima da última geração da execução do AGT com diversificação. Com base nesta constatação foram realizados novos experimentos com estes grafos para verificar se o aumento no número de gerações irá permitir ao algoritmo melhorar as soluções obtidas. Nestes novos experimentos, foi realizada uma única execução da versão AGT+D(10) e da AGT+D(25) para cada um dos grafos de 20 a 100 vértices, considerando-se um total de 32.000 gerações. Os resultados obtidos são apresentados na Tabela 4.13 e Figuras 4.46 e 4.47.

Grafo	Solução Best	Versão AGT+D(10)		Versão AGT+D(25)	
		Melhor Indivíduo	Obtido na Geração	Melhor Indivíduo	Obtido na Geração
A20	0,7371	0,7371	10.569	0,7371	4.052
A40	0,7087	0,6681	888	0,6657	1.772
A60	0,7402	0,7136	14.315	0,7192	23.074
A80	0,7780	0,7657	20.608	0,7589	25.349
A100	0,8092	0,7817	27.702	0,7421	31.734
E20	0,7089	0,6134	27.871	0,6134	5.048
E40	0,6485	0,5614	12.695	0,5195	22.039
E60	0,7402	0,3488	24.826	0,3272	31.046
E80	0,7075	0,4652	31.975	0,4706	27.522
E100	0,6980	0,4985	29.341	0,4929	25.623

Tabela 4.13 – MQ dos melhores indivíduos obtidos e as gerações em que o melhor indivíduo foi obtido para uma execução das versões AGT+D(10) e AGT+D(25), com 32.000 gerações

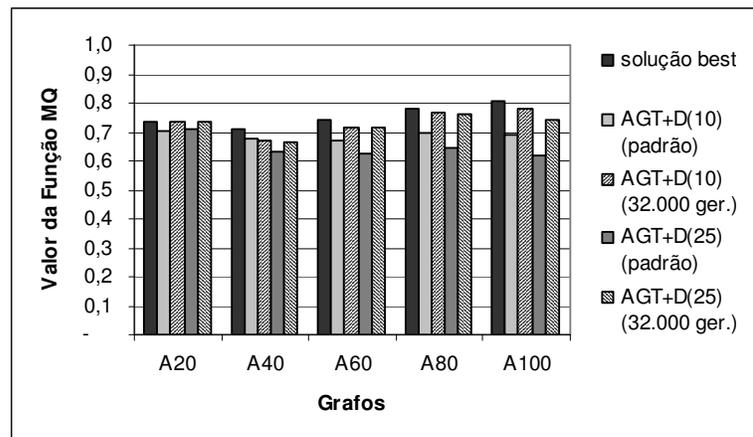


Figura 4.46 – Melhor indivíduo obtido pelas versões AGT+D(10) e AGT+D(25), com total de gerações padrão e com total de gerações igual a 32.000, para os grafos A20 a A100

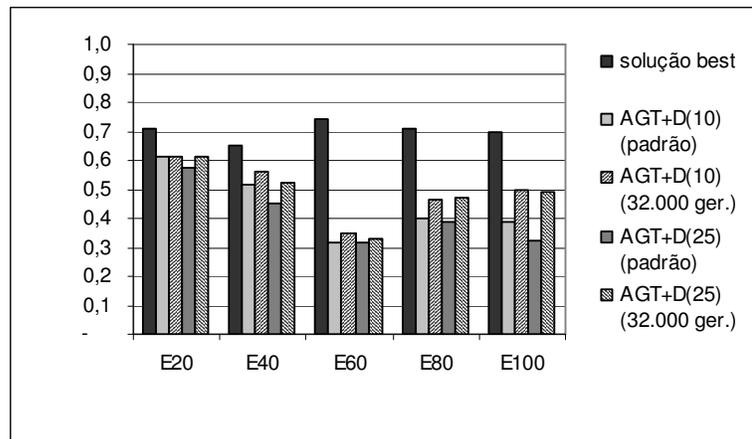


Figura 4.47 – Melhor indivíduo obtido pelas versões AGT+D(10) e AGT+D(25), com total de gerações padrão e com total de gerações igual a 32.000, para os grafos E20 a E100

Com o aumento do número de gerações para 32.000 foram obtidas soluções melhores do que utilizando o total de gerações igual a 2.000. Por outro lado, houve um aumento considerável no tempo total de execução de cada experimento, conforme apresentado na Tabela 4.14. O acompanhamento da evolução gradativa do melhor indivíduo da população ao longo de todas as gerações do AGT+D(10) e AGT+(25), utilizando-se os grafos A100 e E100, pode ser realizado a partir dos gráficos das Figuras 4.48 a 4.51.

Grafo	Versão AGT+D(10)		Versão AGT+D(25)	
	Padrão	32.000 Gerações	Padrão	32.000 Gerações
A10	0,1 s	10,6 s	0,1 s	10,5 s
A20	0,4 s	30,8 s	0,4 s	31,6 s
A40	2,6 s	100,1 s	2,6 s	100,0 s
A60	10,5 s	260,4 s	10,7 s	264,5 s
A80	18,0 s	350,2 s	18,2 s	354,1 s
A100	32,4 s	507,1 s	32,6 s	514,2 s
E10	0,1 s	10,5 s	0,1 s	10,5 s
E20	0,4 s	32,1 s	0,4 s	32,7 s
E40	2,6 s	99,2 s	2,6 s	101,1 s
E60	10,6 s	276,2 s	10,7 s	280,7 s
E80	18,0 s	356,2 s	18,2 s	358,3 s
E100	32,8 s	519,7 s	33,1 s	522,3 s

Tabela 4.14 – Tempo de execução utilizado pelas versões AGT+D(10) e AGT+(25) com número de gerações padrão e com 32.000 gerações

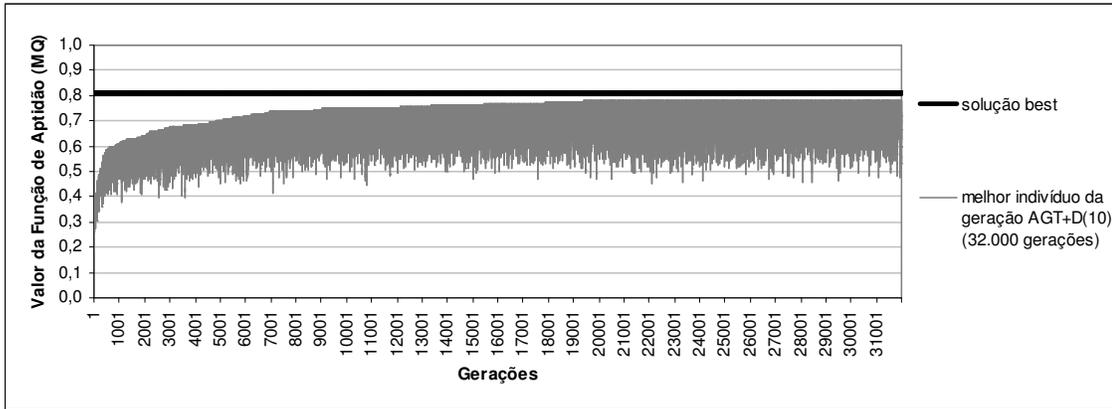


Figura 4.48 – Acompanhamento da evolução do melhor indivíduo ao longo de 32.000 gerações do AGT+D(10), para o grafo A100

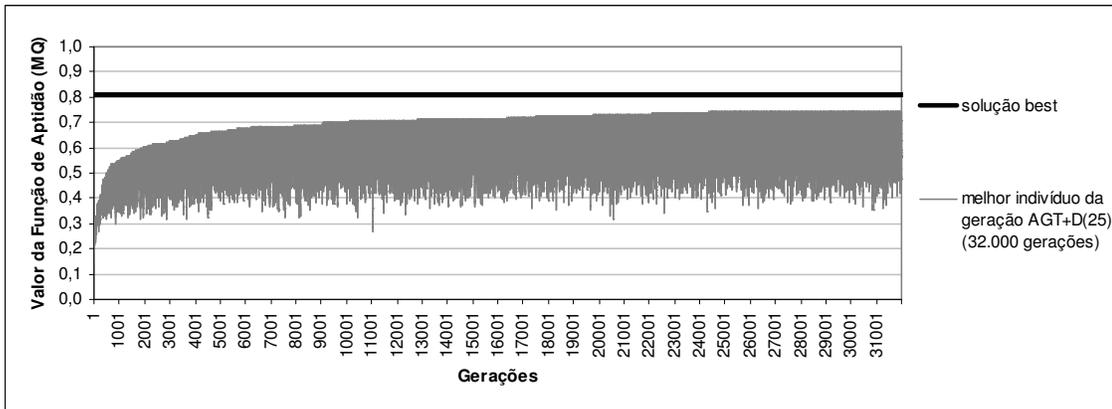


Figura 4.49 – Acompanhamento da evolução do melhor indivíduo ao longo de 32.000 gerações do AGT+D(25), para o grafo A100

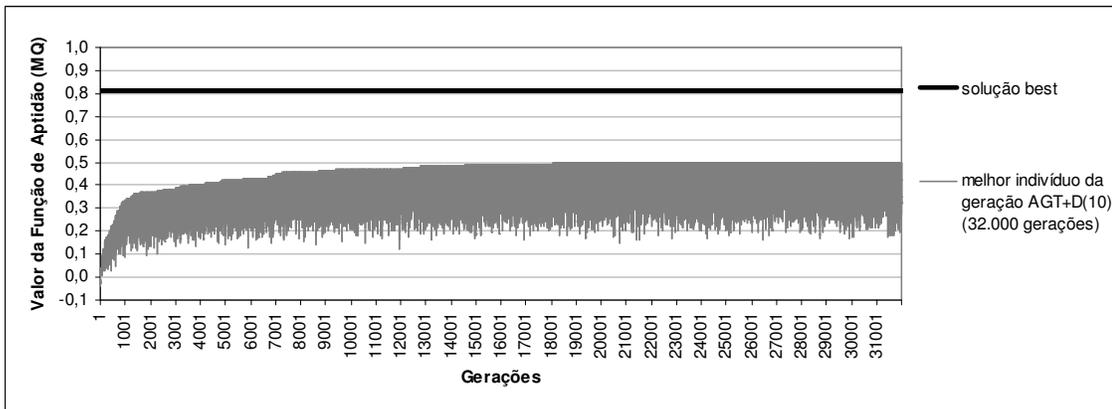


Figura 4.50 – Acompanhamento da evolução do melhor indivíduo ao longo de 32.000 gerações do AGT+D(10), para o grafo E100

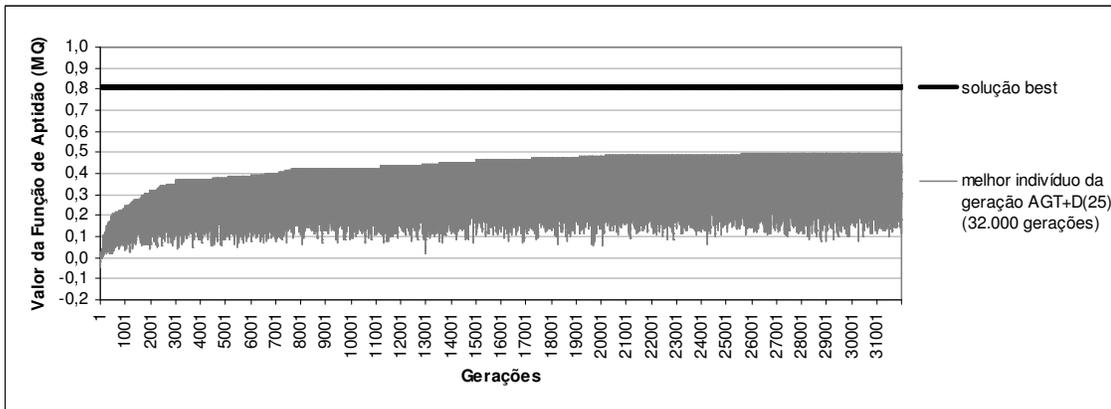


Figura 4.51 – Acompanhamento da evolução do melhor indivíduo ao longo de 32.000 gerações do AGT+D(25), para o grafo E100

A partir dos resultados obtidos com o acréscimo do procedimento de diversificação da população ao AGT é possível concluir que:

1. houve uma melhora significativa na qualidade das soluções obtidas em comparação com o AGT original;
2. aumentando-se o total de gerações do AGT com o procedimento de diversificação pode-se obter melhores resultados, apesar do aumento no tempo de execução utilizado;
3. a definição do intervalo de aplicação do procedimento de diversificação também interferiu nos resultados, em que, para um intervalo igual a 10 gerações sem a evolução da população, os resultados foram melhores do que quando foi utilizado um intervalo maior, igual a 25 gerações.

4.6. Comparação entre as Diferentes Propostas

Nas seções anteriores todas as propostas de adaptações do AGT foram comparadas, uma a uma, com a versão original do AGT. As Tabelas 4.15 e 4.16 realizam consolidações mostrando, de forma comparativa, os resultados obtidos por todas as propostas sugeridas para melhorar o desempenho do AGT, utilizando os grafos de 10 a 100 vértices.

A Tabela 4.15 apresenta o valor para a função MQ do melhor indivíduo obtido por cada versão do AGT.

Grafo	Solução Best	AGT	AGT +M(0,0040)	AGT +C	AGT +T(2)	AGT +T(4)	AGT +G	AGT +B	AGT +D(10)	AGT +D(25)
A10	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083
A20	0,7371	0,7371	0,7177	0,7023	0,7036	0,7371	0,6620	0,7371	0,7036	0,7124
A40	0,7087	0,5919	0,6508	0,5260	0,6782	0,6782	0,5283	0,7087	0,6766	0,6324
A60	0,7402	0,5695	0,6403	0,5150	0,6334	0,6313	0,4921	0,7402	0,6704	0,6276
A80	0,7780	0,5035	0,6778	0,4369	0,5019	0,4986	0,4476	0,7780	0,6955	0,6440
A100	0,8092	0,4765	0,6633	0,4372	0,4652	0,4728	0,4236	0,8092	0,6875	0,6204
E10	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083
E20	0,7089	0,5957	0,4848	0,5243	0,5722	0,5722	0,5681	0,7089	0,6134	0,5722
E40	0,6485	0,4077	0,4787	0,3228	0,5196	0,5501	0,3111	0,6485	0,5145	0,4534
E60	0,7402	0,2165	0,3053	0,1341	0,2423	0,2366	0,1311	0,7402	0,3138	0,3182
E80	0,7075	0,2132	0,4002	0,1435	0,1926	0,2040	0,1622	0,7075	0,3994	0,3863
E100	0,6980	0,1544	0,3389	0,1059	0,1444	0,1456	0,1187	0,6980	0,3879	0,3213

Tabela 4.15 – Valores de *MQ* para os melhores indivíduos obtidos com o AGT original, e com cada uma das alterações propostas

Os dados descritos na Tabela 4.15 são apresentados graficamente nas Figuras 4.52 e 4.53.

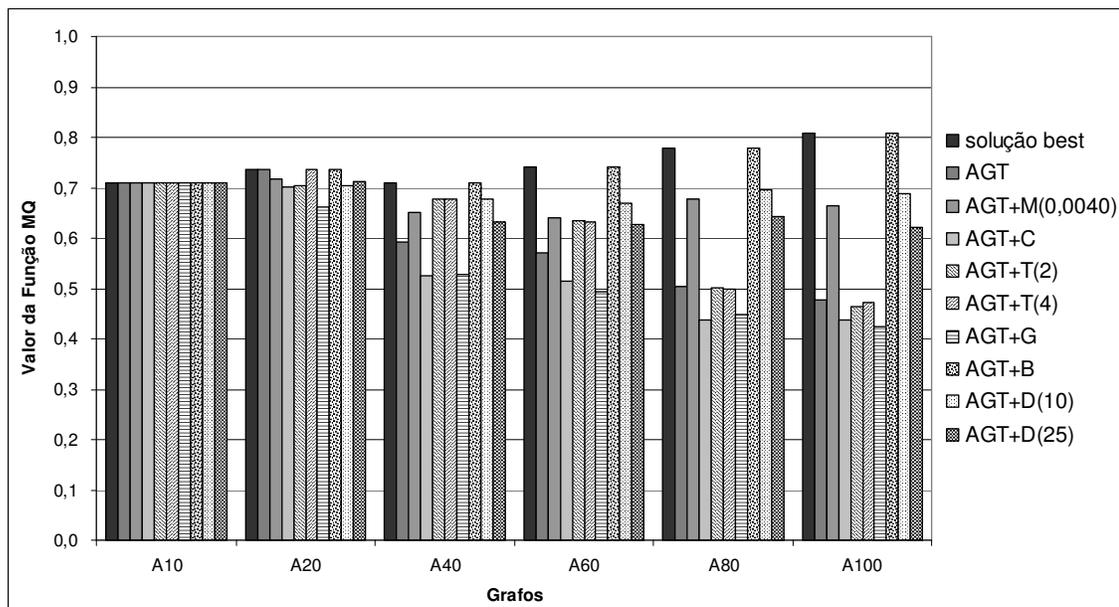


Figura 4.52 – Comparação dos melhores indivíduos obtidos pelo AGT e pelas versões propostas, para os grafos A10 a A100

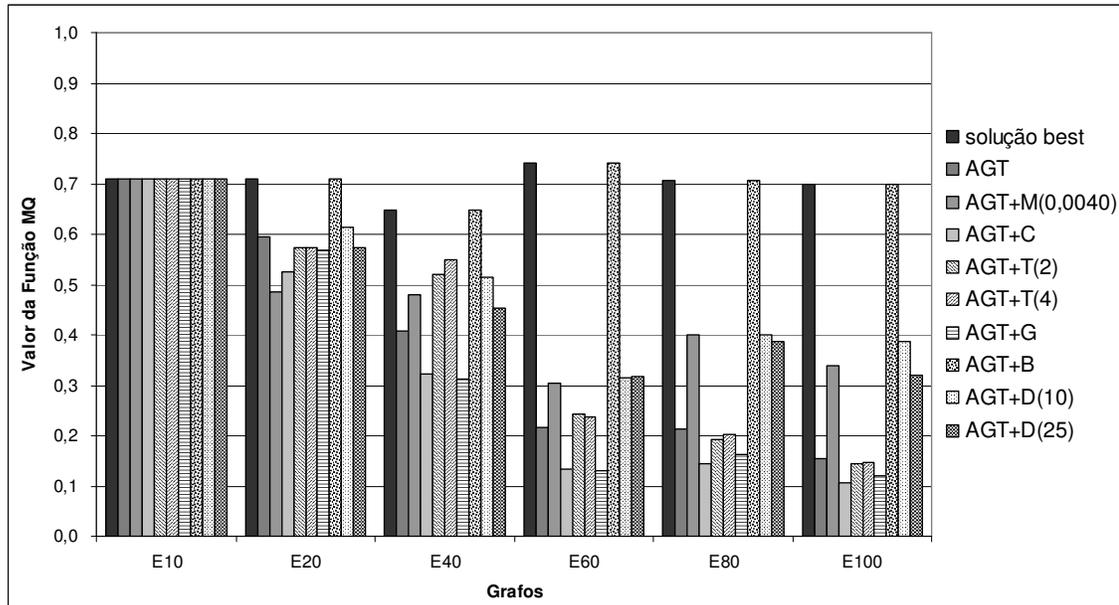


Figura 4.53 – Comparação dos melhores indivíduos obtidos pelo AGT e pelas versões propostas, para os grafos E10 a E100

Analisando os resultados dos melhores indivíduos obtidos para cada uma das versões indicadas na Tabela 4.15, é possível concluir que:

1. os melhores resultados foram obtidos pela versão com a inserção da busca local, que obteve a solução *best* para todos o grafos e um resultado médio muito próximo da solução *best*;
2. para os grafos maiores (com 60 ou mais vértices), após a versão com busca local, as versões que obtiveram os melhores resultados foram as versões com redução no valor da taxa de mutação e com a inserção do procedimento de diversificação;
3. as propostas de calibração (AGT+C) e de cruzamento utilizando a representação GGA (AGT+G) obtiveram os piores resultados, de uma forma geral, mesmo se comparados com o AGT original. Para resultados mais contundentes é necessário analisar o comportamento destas versões em conjunto com as demais, o que será feito no Capítulo 5;
4. os resultados obtidos pelas versões com seleção por torneio também obtiveram resultados não satisfatórios para grafos maiores.

A Tabela 4.16 apresenta, de forma comparativa, o valor médio de MQ nas 10 execuções de cada versão, para cada grafo. As Figuras 4.54 e 4.55 mostram tais resultados graficamente.

Grafo	Solução Best	AGT	AGT +M(0,0040)	AGT +C	AGT +T(2)	AGT +T(4)	AGT +G	AGT +B	AGT +D(10)	AGT +D(25)
A10	0,7083	0,7083	0,6505	0,6826	0,7083	0,7083	0,7083	0,7083	0,6955	0,7083
A20	0,7371	0,7289	0,7070	0,6929	0,6784	0,7075	0,6456	0,7371	0,6963	0,7010
A40	0,7087	0,5651	0,6092	0,4678	0,6630	0,6626	0,4665	0,6754	0,6491	0,5934
A60	0,7402	0,5537	0,6213	0,4874	0,6236	0,6173	0,4852	0,7339	0,6545	0,6057
A80	0,7780	0,4921	0,6559	0,4181	0,4879	0,4889	0,4240	0,7372	0,6642	0,6168
A100	0,8092	0,4552	0,6445	0,4080	0,4504	0,4466	0,4046	0,7827	0,6621	0,6024
E10	0,7083	0,7083	0,7083	0,7083	0,7083	0,5100	0,7083	0,6505	0,7083	0,7083
E20	0,7089	0,5734	0,4609	0,5061	0,4914	0,5090	0,5317	0,6974	0,5613	0,5275
E40	0,6485	0,3716	0,4394	0,2857	0,4780	0,4864	0,2718	0,6201	0,4764	0,4065
E60	0,7402	0,1962	0,2721	0,1093	0,2080	0,2032	0,1106	0,7402	0,2898	0,2678
E80	0,7075	0,1940	0,3613	0,1307	0,1820	0,1849	0,1315	0,6721	0,3807	0,3416
E100	0,6980	0,1416	0,3172	0,0972	0,1362	0,1355	0,0985	0,6564	0,3648	0,2992

Tabela 4.16 – Valores de MQ médios obtidos com o AGT original, e com cada uma das alterações propostas

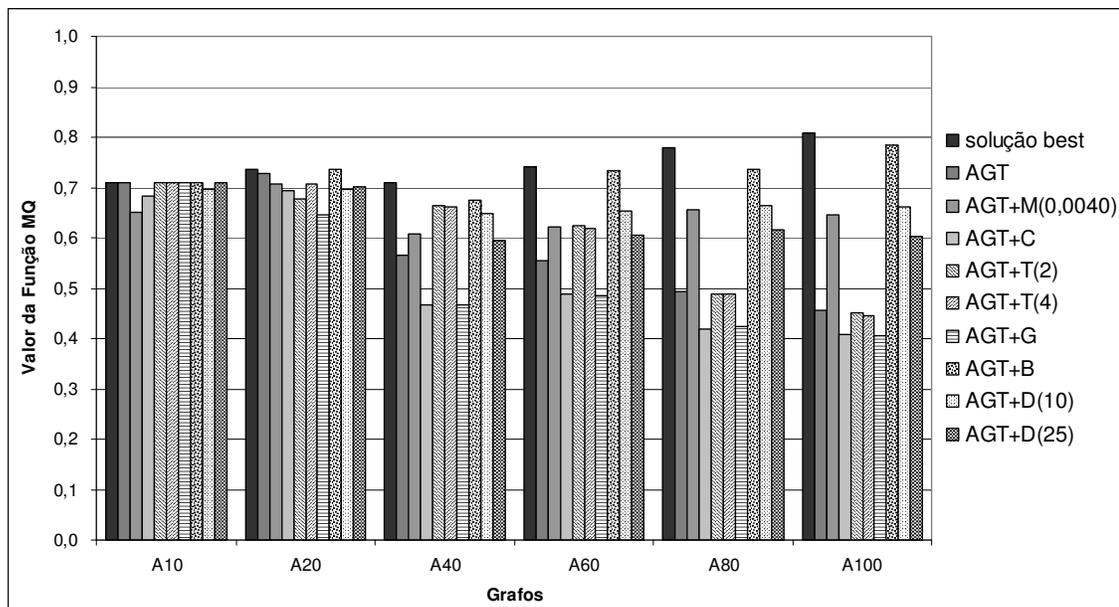


Figura 4.54 – Comparação dos valores de MQ médio referentes às 10 execuções do AGT e de cada versão proposta, para os grafos A10 a A100

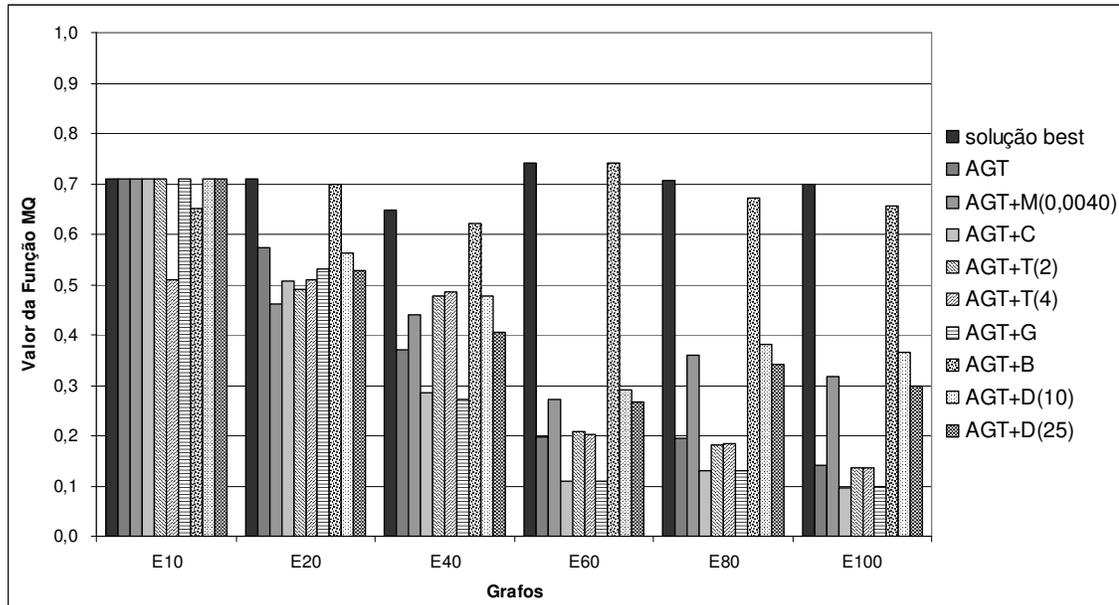


Figura 4.55 – Comparação dos valores de *MQ* médio referentes às 10 execuções do AGT e de cada versão proposta, para os grafos E10 a E100

Para todos os experimentos realizados, o comportamento médio das diferentes versões corresponde ao comportamento da execução que obteve o melhor indivíduo para cada versão e para cada grafo. Um aspecto importante a ser analisado é o tempo de execução utilizado para cada versão, conforme apresentado na Tabela 4.17.

Grafo	AGT	AGT +M(0,0040)	AGT +C	AGT +T(2)	AGT +T(4)	AGT +G	AGT +B	AGT +D(10)	AGT +D(25)
A10	0,682	0,071	0,070	0,083	0,082	0,070	0,067	0,067	0,067
A20	8,548	0,427	0,433	0,465	0,446	0,446	0,374	0,426	0,427
A40	104,391	2,568	2,649	2,690	2,682	2,656	2,238	2,558	2,589
A60	58,884	10,696	11,068	11,107	11,102	11,087	7,958	10,507	10,727
A80	1.483,246	18,008	18,482	19,322	19,298	18,575	15,367	18,033	18,194
A100	3.321,629	32,463	32,982	34,417	34,354	32,960	27,849	32,383	32,619
E10	0,673	0,065	0,071	0,082	0,091	0,072	0,065	0,067	0,068
E20	8,845	0,445	0,447	0,483	0,488	0,445	0,367	0,429	0,435
E40	104,177	2,556	2,631	2,728	2,716	2,636	2,260	2,566	2,596
E60	656,204	10,570	10,991	11,092	11,109	10,982	8,146	10,589	10,650
E80	1.479,860	17,844	18,413	19,191	19,187	18,425	15,547	18,031	18,180
E100	3.361,533	32,528	33,373	34,620	34,763	33,533	28,490	32,843	33,084

Tabela 4.17 – Tempo médio de execução (em segundos) utilizado para cada versão aplicada a cada grafo

Com base nos dados indicados na Tabela 4.17 verifica-se que as versões que obtiveram os melhores resultados foram também aquelas que tiveram os menores valores para

o tempo médio de execução. A explicação para estes resultados é que as melhores versões trabalham com indivíduos referentes a soluções com número de *clusters* menores do que as piores versões – uma tendência identificada nos experimentos é que as versões iniciam com soluções com uma grande quantidade de *clusters* e este valor vai sendo reduzido à medida que a qualidade dos indivíduos melhora. Dessa forma, existe um ganho na computação da função de aptidão de indivíduos com número de *clusters* menores, o que resulta em um tempo de execução inferior, se comparado com indivíduos com quantidades maiores de *clusters*.

Vale observar ainda que o valor elevado para o tempo de execução do AGT em relação às demais versões deve-se aos valores originais utilizados para os parâmetros tamanho da população e número de gerações, que são bem maiores no AGT do que nas versões propostas.

4.7. Conclusão

Neste capítulo foram apresentadas as propostas para aperfeiçoamento do AGT para a solução do problema de clusterização automática de grafos orientados não ponderados [13]. Para cada proposta foram realizados vários experimentos e os resultados foram comparados com a versão original. Vale ressaltar que as análises realizadas neste capítulo referem-se à utilização separada de cada proposta de adaptação para o AGT. A utilização conjunta de diferentes propostas em um mesmo algoritmo será realizada no Capítulo 5.

Algumas das propostas apresentadas forneceram resultados superiores aos resultados obtidos pelo AGT original, mesmo com tempo médio de execução muito inferior a este. Dentre as propostas que se destacaram estão a inserção do procedimento de busca local, a inserção do procedimento de diversificação e a alteração no parâmetro taxa de mutação. A inserção do procedimento de busca local permitiu a obtenção da solução *best* para todos os grafos de 10 a 100 vértices utilizados nos experimentos, não conseguindo, entretanto, manter este mesmo desempenho para grafos maiores. As demais propostas, apesar de não conseguirem a obtenção de bons resultados para os experimentos considerados, possuem conceitos que devem ser investigados. Esta nova investigação será realizada no próximo capítulo, através de experimentos realizados a partir da combinação de todas as adaptações propostas.

Capítulo 5

Versões Híbridas do AGT

Neste capítulo serão apresentados os resultados computacionais de experimentos envolvendo novas versões do AGT. Estas novas versões recebem a denominação de algoritmos evolutivos híbridos (AEH), por utilizarem a combinação de mais de uma das propostas de aperfeiçoamento do AGT apresentadas no Capítulo 4. O objetivo deste capítulo é identificar quais combinações das propostas podem obter resultados melhores em relação à utilização das propostas separadamente. Para isso, foram realizados experimentos com cada uma das possíveis combinações dos oito itens descritos a seguir, totalizando cento e trinta e cinco versões diferentes:

- taxa de aplicação do operador mutação igual a 0,0040;
- calibração no operador mutação;
- seleção por torneio com 2 indivíduos participantes, ou;
- seleção por torneio com 4 indivíduos participantes;
- operador de cruzamento utilizando proposta GGA;
- diversificação após 10 gerações sem evolução média da população, ou;
- diversificação após 25 gerações sem evolução média da população;
- inserção do procedimento de busca local.

5.1. Resultados Computacionais

Para cada versão híbrida, foram realizados 10 experimentos preliminares para cada grafo com 10, 20, 40, 60, 80 e 100 vértices (com topologias de formação anel e estrela), totalizando 16.200 execuções. Os resultados médios são mostrados na Tabela 5.1.

Ord.	Versão	Média	Ord.	Versão	Média	Ord.	Versão	Média
1	AEH13578	99,87	46	AEH358	98,98	91	AEH1357	91,50
2	AEH14568	99,86	47	AEH78	98,98	92	AEH146	91,42
3	AEH124568	99,86	48	AEH138	98,93	93	AEH14	91,35
4	AEH12468	99,86	49	AEH38	98,92	94	AEH135	91,15
5	AEH1368	99,86	50	AEH578	98,90	95	AEH16	91,10
6	AEH123568	99,86	51	AEH458	98,85	96	AEH1256	91,01
7	AEH12378	99,86	52	AEH68	98,83	97	AEH1235	90,97
8	AEH1468	99,86	53	AEH1358	98,83	98	AEH148	90,90
9	AEH12478	99,86	54	AEH568	98,83	99	AEH157	90,83
10	AEH2378	99,86	55	AEH1248	98,76	100	AEH17	90,63
11	AEH13568	99,86	56	AEH1238	98,73	101	AEH56	90,36
12	AEH124578	99,85	57	AEH12578	98,63	102	AEH126	90,26
13	AEH14578	99,85	58	AEH1268	98,56	103	AEH256	89,89
14	AEH1378	99,84	59	AEH12568	98,55	104	AEH26	89,77
15	AEH24578	99,84	60	AEH1278	98,54	105	AEH1257	89,56
16	AEH2468	99,84	61	AEH12358	98,54	106	AEH127	88,86
17	AEH23568	99,83	62	AEH1578	98,42	107	AEH57	88,54
18	AEH23578	99,83	63	AEH58	98,41	108	AEH257	88,19
19	AEH1478	99,83	64	AEH128	98,34	109	AEH15	87,80
20	AEH2368	99,79	65	AEH1258	98,24	110	AEH27	87,60
21	AEH24568	99,79	66	AEH158	98,13	111	AEH247	87,23
22	AEH123578	99,77	67	AEH168	97,79	112	AEH237	87,21
23	AEH2478	99,73	68	AEH1568	97,78	113	AEH2356	87,20
24	AEH12368	99,69	69	AEH178	97,69	114	AEH236	87,09
25	AEH3568	99,46	70	AEH18	97,44	115	AEH2456	87,07
26	AEH4568	99,46	71	AEH12457	92,58	116	AEH246	86,98
27	AEH468	99,42	72	AEH1457	92,55	117	AEH2457	86,97
28	AEH378	99,42	73	AEH137	92,44	118	AEH23	86,93
29	AEH368	99,40	74	AEH1247	92,42	119	AEH2357	86,82
30	AEH4578	99,40	75	AEH145	92,42	120	AEH24	86,74
31	AEH478	99,40	76	AEH147	92,39	121	AEH357	86,63
32	AEH238	99,37	77	AEH1236	92,28	122	AEH457	86,57
33	AEH1458	99,36	78	AEH1245	92,19	123	AEH37	86,49
34	AEH3578	99,34	79	AEH124	92,17	124	AEH46	86,46
35	AEH28	99,22	80	AEH1246	92,10	125	AEH245	86,33
36	AEH2458	99,22	81	AEH123	92,05	126	AEH356	86,32
37	AEH248	99,18	82	AEH1356	92,04	127	AEH125	86,15
38	AEH12458	99,13	83	AEH13	92,00	128	AEH456	86,08
39	AEH2358	99,10	84	AEH12357	91,99	129	AEH35	85,97
40	AEH258	99,07	85	AEH136	91,92	130	AEH47	85,92
41	AEH268	99,05	86	AEH156	91,89	131	AEH36	85,90
42	AEH2568	99,05	87	AEH1456	91,79	132	AEH12	85,78
43	AEH278	99,05	88	AEH12356	91,67	133	AEH45	85,48
44	AEH2578	99,03	89	AEH1237	91,61	134	AEH235	85,38
45	AEH48	99,02	90	AEH12456	91,53	135	AEH25	79,37

Tabela 5.1 – Resultados médios dos experimentos com as versões híbridas, para grafos de 10 a 100 vértices

Na Tabela 5.1 são apresentados os valores referentes à média aritmética dos percentuais da melhor solução, obtida em cada execução, em relação à solução *best* para cada grafo utilizado. Dessa forma, para cada versão híbrida, o valor médio engloba as 10 execuções para cada um dos 12 grafos utilizados (totalizando 120 experimentos). A partir dos resultados obtidos, as versões híbridas foram ordenadas na Tabela 5.1 de acordo com o valor médio indicado, fornecendo um *ranking* da melhor versão para a pior.

Para identificação das características de cada versão híbrida indicada na Tabela 5.1, devem ser considerados os seguintes indicadores no nome da versão:

- 1 = utilização da taxa de aplicação do operador mutação igual a 0,0040;
- 2 = utilização da calibração no número de *clusters* considerados no operador mutação;
- 3 = utilização do tipo de seleção por torneio, em substituição ao método da roleta, com dois indivíduos por torneio;
- 4 = utilização do tipo de seleção por torneio, em substituição ao método da roleta, com quatro indivíduos por torneio;
- 5 = substituição do operador de cruzamento de um ponto, pelo operador de cruzamento proposto no GGA de Falkenauer [16];
- 6 = ativação do procedimento de diversificação da população após 10 gerações sem uma melhora no valor do *MQ* médio dos indivíduos da população;
- 7 = ativação do procedimento de diversificação da população após 25 gerações sem uma melhora no valor do *MQ* médio dos indivíduos da população;
- 8 = aplicação do procedimento de busca local ao melhor indivíduo de cada geração.

Desta forma, a versão AEH123568, por exemplo, corresponde ao AGT com a utilização das opções 1, 2, 3, 5, 6 e 8.

Considerando todos estes experimentos apresentados na Tabela 5.1, as nove versões híbridas, que obtiveram os melhores resultados médios (e resultados similares) são indicadas na Tabela 5.2 (não sendo considerada uma ordem específica das versões).

Nome da Versão
AEH123568
AEH12378
AEH124568
AEH12368
AEH12478
AEH13578
AEH1368
AEH14568
AEH1468

Tabela 5.2 – Versões híbridas mais eficientes

Na Tabela 5.3 são apresentados os valores médios para a função MQ dos melhores indivíduos obtidos após a conclusão das 10 execuções de cada uma das melhores versões híbridas (indicadas na Tabela 5.2), para cada um dos grafos considerados.

Grafo	Sol. Best	AEH 123568	AEH 12378	AEH 124568	AEH 12468	AEH 12478	AEH 13578	AEH 1368	AEH 14568	AEH 1468
A10	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083
A20	0,7371	0,7371	0,7371	0,7371	0,7371	0,7371	0,7371	0,7371	0,7371	0,7371
A40	0,7087	0,6806	0,6806	0,6806	0,6806	0,6806	0,6818	0,6806	0,6806	0,6806
A60	0,7402	0,7402	0,7402	0,7402	0,7402	0,7402	0,7402	0,7402	0,7402	0,7402
A80	0,7780	0,7780	0,7780	0,7780	0,7780	0,7780	0,7780	0,7780	0,7780	0,7780
A100	0,8092	0,8092	0,8092	0,8092	0,8092	0,8092	0,8092	0,8092	0,8092	0,8092
E10	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083	0,7083
E20	0,7089	0,7089	0,7089	0,7089	0,7089	0,7089	0,7089	0,7089	0,7089	0,7089
E40	0,6485	0,6485	0,6485	0,6485	0,6485	0,6485	0,6485	0,6485	0,6485	0,6485
E60	0,7402	0,7402	0,7402	0,7402	0,7402	0,7402	0,7402	0,7402	0,7402	0,7402
E80	0,7075	0,7075	0,7075	0,7075	0,7075	0,7075	0,7075	0,7075	0,7075	0,7075
E100	0,6980	0,6980	0,6980	0,6980	0,6980	0,6980	0,6980	0,6980	0,6980	0,6980

Tabela 5.3 – Valores médios para a função MQ dos melhores indivíduos obtidos após a conclusão das 10 execuções de cada uma das melhores versões híbridas

Para cada uma das nove melhores versões híbridas (AEH), todas as execuções utilizando os grafos considerados obtiveram a solução *best*, exceto as que utilizaram o grafo A40, conforme mostrado na Tabela 5.3. Isto significa que cada linha da Tabela 5.3 contém os mesmos valores, ou seja, os valores médios são todos iguais ao *best* para todas as versões híbridas, excetuando-se aqueles referentes ao grafo A40.

Por outro lado, conforme mostrado na Seção 4.4 do Capítulo 4, apenas para os grafos A10, A20 e A60 a versão que utilizava exclusivamente o procedimento de busca local (AGT+B) conseguiu obter a solução *best* para todas as execuções realizadas. Dessa forma, podemos concluir que as nove melhores versões híbridas obtiveram resultados médios superiores à melhor proposta individual para melhoria do AGT, que corresponde à utilização do procedimento de busca local. Em outras palavras, observa-se uma maior robustez nas versões híbridas do que nas versões que utilizam apenas uma das propostas de aperfeiçoamento do AGT.

Com relação aos resultados obtidos para o grafo A40, de todos os experimentos realizados neste trabalho, apenas uma única execução do AGT+B, apresentada na Seção 4.4, conseguiu obter a solução *best*. Todos os demais experimentos realizados com o AGT+B e com as nove melhores versões híbridas obtiveram soluções com os valores para a função MQ iguais a, no máximo, 0,6926 (o melhor valor médio obtido para as nove melhores versões híbridas foi igual a 0,6818, conforme apresentado na Tabela 5.3).

Apesar das versões AEH2378 e AEH13568 terem obtido resultado médio igual ao resultado médio obtido por algumas das melhores versões (apresentadas na Tabela 5.2), a primeira não conseguiu obter a solução *best* para todas as execuções com o grafo A80 e a segunda não conseguiu obter a solução *best* para todas as execuções com o grafo E100. Além disto, estas duas versões também não conseguiram obter a solução *best* para o grafo A40. Portanto, estas versões não foram consideradas no grupo das nove melhores versões híbridas.

Um aspecto importante a ser observado com relação às nove melhores versões híbridas da Tabela 5.2 é que todas elas possuem no mínimo quatro propostas combinadas: alteração na taxa de aplicação do operador mutação, seleção por torneio, inclusão do procedimento de diversificação e inclusão do procedimento de busca local.

De todas as cento e trinta e cinco versões híbridas utilizadas, o pior resultado foi obtido pelos experimentos realizados com a versão que utiliza a combinação da proposta de calibração do operador mutação e do cruzamento GGA, ou seja, a versão AGT25.

A Tabela 5.4 apresenta o número médio de iterações em que o indivíduo equivalente à solução *best* foi alcançado em cada uma das nove melhores versões híbridas. Estes resultados variam muito de grafo para grafo, sendo que a última linha da Tabela 5.4 mostra o número médio de iterações referente a todas as execuções de cada versão híbrida, para todos os grafos considerados.

Com base nestes valores, as três versões que apresentaram convergência mais lenta, demorando, em média, mais iterações para alcançar a solução *best*, foram aquelas que utilizaram a proposta 7, ou seja, o procedimento de diversificação sendo ativado após 25 iterações sem melhora no valor do *MQ* médio. Neste caso, estas versões demoram mais a ativar o procedimento do que as demais, que utilizam a proposta 6. Ainda com relação ao número de iterações em que a solução *best* foi obtida, desconsiderando as três versões com a proposta 7, as outras duas versões que apresentaram convergência mais lenta foram aquelas que incluíram a proposta 3, ou seja, o torneio com 2 indivíduos. Neste último caso, a convergência da população sofre menos pressão do que quando é utilizado o torneio com 4 indivíduos.

Grafo	AEH 123568	AEH 12378	AEH 124568	AEH 12468	AEH 12478	AEH 13578	AEH 1368	AEH 14568	AEH 1468
A10	2	2	2	2	2	2	7	13	4
A20	109	85	122	56	66	82	93	156	86
A40	280	216	214	240	364	354	242	246	399
A60	10	14	6	6	6	7	5	3	5
A80	798	887	841	813	693	852	1027	772	582
A100	1616	1487	1480	1199	1497	1564	1604	1466	1652
E10	9	5	2	2	2	4	19	6	11
E20	30	31	16	20	34	30	19	15	19
E40	305	276	232	214	399	154	56	55	39
E60	31	31	31	29	30	27	28	26	28
E80	56	126	84	48	185	89	51	206	58
E100	60	219	55	180	151	282	73	65	56
Média	275,42	281,54	257,13	233,98	285,74	287,23	268,66	252,46	244,81

Tabela 5.4 – Número médio das iterações em que a solução *best* foi obtida para cada versão híbrida

Uma observação importante nesta análise é que quando dizemos que um algoritmo A exigiu, em média, mais iterações do que um outro algoritmo B, não necessariamente significa que o tempo de CPU médio de A tenha sido maior que o de B, já que uma iteração de A pode ser mais demorada do que uma iteração de B. A Tabela 5.5 mostra o tempo médio de execução para cada uma das melhores versões híbridas, utilizando cada grafo, considerando as 10 execuções.

Grafo	AEH 123568	AEH 12378	AEH 124568	AEH 12468	AEH 12478	AEH 13578	AEH 1368	AEH 14568	AEH 1468	Média	Desvio Padrão
A10	0,094	0,089	0,091	0,093	0,091	0,086	0,086	0,090	0,090	0,090	0,002
A20	0,464	0,448	0,464	0,463	0,446	0,430	0,440	0,439	0,443	0,449	0,011
A40	2,639	2,620	2,654	2,649	2,625	2,543	2,564	2,574	2,569	2,604	0,040
A60	8,925	8,504	8,916	8,908	8,472	8,336	8,763	8,762	8,784	8,708	0,202
A80	17,255	17,225	17,259	17,187	17,246	16,985	17,020	17,000	16,980	17,129	0,119
A100	31,222	31,283	31,345	31,154	31,416	31,101	30,679	30,902	30,946	31,116	0,234
E10	0,092	0,092	0,090	0,090	0,089	0,083	0,089	0,089	0,089	0,089	0,002
E20	0,466	0,448	0,464	0,464	0,446	0,423	0,445	0,445	0,447	0,450	0,012
E40	2,643	2,552	2,645	2,638	2,554	2,469	2,558	2,573	2,572	2,578	0,052
E60	9,065	8,626	9,073	9,055	8,596	8,456	8,893	8,866	8,908	8,838	0,211
E80	17,254	16,704	17,249	17,244	16,687	16,406	16,984	16,968	16,972	16,941	0,271
E100	32,221	31,360	32,189	32,166	31,406	30,932	31,804	31,765	31,797	31,738	0,399

Tabela 5.5 – Tempo médio para a execução completa de cada versão híbrida (em segundos)

Para verificar se são significativas as diferenças entre os tempos médios de execução de todas as versões utilizando cada grafo, as duas últimas colunas da Tabela 5.5 apresentam os valores referentes ao tempo médio de execução para cada grafo e o desvio padrão considerando todas as versões. Com bases nos valores indicados na Tabela 5.5 pode-se concluir que não existe uma relação direta entre o número de módulos utilizados em uma versão e o seu tempo médio de execução, tendo em vista que, mesmo comparando duas versões, estas podem apresentar comportamentos diferentes (em relação ao tempo de execução) para grafos diferentes. Pode-se concluir também que não existe uma diferença significativa entre os tempos médios de execução das diferentes versões híbridas.

Para verificar se as melhores versões híbridas para os grafos de 10 a 100 vértices mantêm seus desempenhos para grafos maiores, foram realizados também experimentos para os grafos com 160, 200, 500 e 1000 vértices. Para cada grafo foram realizados 2 experimentos para cada uma das nove melhores versões híbridas. Com o objetivo de reduzir o tempo de

execução das versões híbridas para os grafos com 500 e 1000 vértices, foram considerados o tamanho da população fixo e igual a 200 indivíduos e um total de 4000 gerações. Os valores médios para a função MQ , obtidos por cada versão aplicada a cada grafo, são mostradas na Tabela 5.6. São indicados em negrito os resultados que igualaram ou superaram a solução *best*.

Grafo	Sol. Best	AEH 123568	AEH 12378	AEH 124568	AEH 12468	AEH 12478	AEH 13578	AEH 1368	AEH 14568	AEH 1468
A160	0,9090	0,9090								
A200	0,8783	0,8882	0,8882	0,8885	0,8883	0,8878	0,8888	0,8885	0,8888	0,8892
A500	0,8785	0,8853	0,8852	0,8837	0,8841	0,8850	0,8711	0,8673	0,8668	0,8682
A1000	0,9415	0,9507	0,9499	0,9513	0,9467	0,9512	0,9257	0,9300	0,9303	0,9297
E160	0,7364	0,7364								
E200	0,7662	0,7662								
E500	0,8759	0,8759								
E1000	0,8865	0,8865								

Tabela 5.6 – Valores médios de MQ para as soluções obtidas com as versões híbridas do AGT para grafos com 160 e 200 vértices

Nos experimentos utilizando os grafos com dimensões elevadas, as versões híbridas AEH123568, AEH12378, AEH124568, AEH12469 e AEH12478 *conseguiram alcançar a solução best em todas as suas execuções. Vale observar que, para os grafo A200, A500 e A1000, estas versões híbridas conseguiram obter soluções melhores do que a solução best* (recordando, o valor *best* é o melhor valor entre o valor de MQ das soluções obtidas no Capítulo 4 e o valor de MQ obtido na geração do grafo associado a cada instância).

Outra observação importante, em relação aos experimentos indicados na Tabela 5.6, refere-se aos excelentes resultados obtidos para o grafo E1000. Na Seção 4.4 foi mostrado que a versão AGT+B, que inclui o procedimento de busca local (e nenhuma outra proposta de aperfeiçoamento do AGT), apesar de ter obtido um resultado superior em relação à versão AGT para o grafo E1000, teve um desempenho muito ruim, conseguindo a melhor solução com o valor para a função MQ igual a apenas -0,0011 (valor muito inferior à solução *best*, obtida por todas as nove versões híbridas, conforme apresentado na Tabela 5.6).

A partir dos resultados apresentados na Tabela 5.6, é possível verificar que as versões híbridas que não utilizam a calibração do operador mutação (AEH13578, AEH1368, AEH14568 e AEH1468) não conseguiram obter a solução *best* para os grafos A500 e A1000,

o que leva à conclusão que a inclusão desta característica, em conjunto com as outras quatro propostas presentes em todas as nove melhores versões híbridas, foi determinante para a qualidade das soluções obtidas para grafos com grandes dimensões.

Outro aspecto que deve ser considerado é o tempo de execução das versões híbridas para estes grafos com números elevados de vértices. A Tabela 5.7 apresenta o tempo de execução médio utilizado para a execução de cada versão híbrida, para cada grafo considerado, e o tempo de execução médio utilizado pela versão AGT+B, que foi a versão com menor tempo médio de execução entre as versões que utilizam apenas uma das propostas de aperfeiçoamento do AGT, conforme apresentado na Tabela 4.17 do Capítulo 4.

Grafo	AGT +B	AEH 123568	AEH 12378	AEH 124568	AEH 12468	AEH 12478	AEH 13578	AEH 1368	AEH 14568	AEH 1468
A160	00:03:54	00:03:18	00:03:10	00:03:17	00:03:16	00:03:10	00:03:08	00:03:15	00:03:15	00:03:15
A200	00:06:42	00:07:13	00:07:14	00:07:12	00:07:13	00:07:14	00:07:10	00:07:09	00:07:09	00:07:09
A500	01:19:46	01:22:16	01:22:17	01:22:03	01:22:07	01:22:13	01:22:58	01:22:52	01:22:59	01:22:34
A1000	04:08:14	04:18:49	04:22:44	04:20:40	04:19:01	04:16:21	04:16:20	04:28:00	04:28:13	04:32:03
E160	00:03:40	00:03:24	00:03:18	00:03:24	00:03:23	00:03:18	00:03:17	00:03:22	00:03:22	00:03:22
E200	00:06:43	00:07:12	00:07:02	00:07:12	00:07:13	00:07:01	00:07:56	00:07:09	00:07:09	00:07:09
E500	01:09:02	01:14:31	01:10:45	01:14:20	01:14:27	01:10:41	01:10:35	01:14:21	01:14:51	01:14:23
E1000	05:21:43	04:38:12	04:19:22	04:34:27	04:33:09	04:21:14	04:17:18	04:31:08	04:30:40	04:31:10

Tabela 5.7 – Tempo médio de execução completa para cada versão aplicada a cada grafo com 160, 200, 500 e 1000 vértices (no formato horas:minutos:segundos)

A partir da comparação dos tempos médios apresentados na Tabela 5.7, pode-se verificar que, para a quase totalidade dos experimentos, os tempos médios de execução das versões híbridas são similares entre si e similares aos tempos médios de execução do AGT+B. Em muitos experimentos as versões híbridas conseguiram completar suas execuções em quantidade de tempo menor do que a versão AGT+B e, para o grafo E1000, houve uma diferença significativa entre estes tempos de execução considerados.

5.2. Conclusão

Os resultados apresentados neste capítulo mostraram que a combinação das propostas apresentadas no Capítulo 4, através de versões híbridas do AGT, permite a obtenção de soluções ainda melhores do que se forem utilizadas separadamente. Além do excelente

desempenho obtido para os grafos de 10 a 100 vértices, algumas versões híbridas conseguiram manter este desempenho para grafos ainda maiores, com 160, 200, 500 e 1000 vértices.

A conclusão mais evidente, quando comparamos os resultados dos Capítulos 4 e 5, é que as versões híbridas (Capítulo 5) são mais robustas do que as versões apresentadas no Capítulo 4, isto é, aquelas apresentam um desempenho mais regular nos diversos experimentos realizados do que estas. A robustez de um algoritmo é importante na prática, pois mostra a confiabilidade das heurísticas aqui propostas.

As propostas de aperfeiçoamento do AGT que apresentaram resultados mais significativos, quando combinadas, foram as propostas de alteração na taxa de aplicação do operador mutação para o valor 0,0040, a seleção por torneio, a utilização do procedimento de diversificação e a utilização do procedimento de busca local. Vale mencionar que, para os grafos de grandes dimensões, as versões híbridas que utilizam a calibração no operador mutação se sobressaíram em relação àquelas que não a utilizam.

Capítulo 6

Conclusão e Trabalhos Futuros

O objetivo principal deste trabalho foi desenvolver algoritmos evolutivos (AEs) eficientes e realizar uma investigação experimental de diferentes versões de AEs para a clusterização de vértices em grafos orientados não ponderados, utilizando como ponto de partida um AG tradicional, proposto na literatura.

No Capítulo 2 os principais conceitos de algoritmos evolutivos (AEs) e algoritmos genéticos (AGs) foram apresentados através de uma revisão bibliográfica, com enfoque para a descrição de um AG simples e algumas propostas encontradas na literatura para o seu aperfeiçoamento.

No Capítulo 3 foi tratada a utilização dos AGs para a solução de problemas de clusterização. Um AG para clusterização de vértices em grafos orientados não ponderados foi apresentado, bem como os resultados de sua utilização para grafos com dimensões variadas. Os resultados dos experimentos mostraram que o desempenho do AG tradicional da literatura (AGT) não obtém bons resultados para grafos de médio e grande porte.

No Capítulo 4 foram apresentadas as propostas de aperfeiçoamento do AGT, e a realização de vários experimentos permitiu a comparação do desempenho das propostas com a versão original. Algumas versões apresentadas, como o AGT+B, mostraram um desempenho muito superior ao AGT.

No Capítulo 5 foram realizadas combinações das propostas apresentadas no Capítulo 4, e os resultados mostraram que as propostas combinadas (versões híbridas - AEH) podem melhorar ainda mais o desempenho de um AG sem prejuízo para o seu tempo total de execução. Os resultados também foram muito satisfatórios para grafos com grandes quantidades de vértices.

Contribuições

Apesar dos AEs serem muito utilizados para problemas de clusterização, é escasso o número de trabalhos que tratam o problema de clusterização automática de grafos orientados não ponderados. Dessa forma, este trabalho fornece várias contribuições relacionadas ao aperfeiçoamento de um AG tradicional desenvolvido para tratar este problema. São elas:

1. análise experimental de variações do parâmetro taxa de mutação e conseqüente verificação do alto grau de sensibilidade de um AG em relação a ele;
2. verificação das conseqüências da simetria na representação dos indivíduos do AG;
3. desenvolvimento de um procedimento de busca local eficiente a ser utilizado para a realização de uma exploração intensiva em uma região específica do espaço de busca, a cada geração do processo evolutivo do AG;
4. análise da sensibilidade do AG em relação ao método de seleção utilizado;
5. desenvolvimento de um procedimento de diversificação que reduz a possibilidade da convergência da população para um ótimo local;
6. análise experimental da combinação das diferentes versões híbridas e identificação dos casos em que a combinação de duas ou mais propostas de aperfeiçoamento é sinérgica e os casos em que a combinação se torna prejudicial.

É importante observar que as contribuições deste trabalho, algumas mais do que outras, podem também ser estendidas para os AGs com aplicações diversas, não ficando restritas ao problema de clusterização tratado.

Trabalhos futuros

Durante a elaboração deste trabalho surgiram várias possibilidades de pesquisa que podem agregar muita qualidade ao trabalho concluído. Entretanto, devido à impossibilidade de tratá-las em prazo adequado, colocamos estas possibilidades como trabalhos futuros. Dentre as possibilidades de continuidade deste trabalho, consideramos:

- *o desenvolvimento de versão paralela*: devido a características de paralelismo inerentes aos algoritmos genéticos, a utilização de uma versão para processamento paralelo de populações de indivíduos pode trazer benefícios no tempo total de execução e qualidade das soluções obtidas;
- *o desenvolvimento de técnicas cooperativas e adaptativas*: com a paralelização do AG, pode-se pensar em trabalhar com populações de indivíduos com comportamentos diferentes, executando em processadores diferentes, que cooperam entre si, como em [1] e [2];
- *a utilização das técnicas de mineração de dados*: identificação de padrões na estrutura de soluções elite que podem ser inseridos em indivíduos de uma população, através de novos operadores genéticos;
- *a utilização da técnica de religamento de caminho (path relinking)*: para tentar obter soluções de qualidade entre duas soluções elite da população.

Consideramos todas as possibilidades indicadas acima com grandes potencialidades para o tratamento de instâncias ainda maiores do que as tratadas aqui, com desempenho igual ou superior às versões propostas neste trabalho.

Publicações Associadas

Título: **Efficient Evolutionary Algorithms for the Clustering Problem in Directed Graphs**

Autores: Carlos Rodrigo Dias e Luiz Satoru Ochi;

Local: Apresentação e publicação nos Proceedings of the 2003 IEEE Congress on Evolutionary Computation (CEC2003), Canberra, Austrália;

Editora: IEEE-Press, ISBN: 0-7803-7804-0, vol 2, pp:983-990. ISBN (CD-ROM): 0-7803-7805-9 (softbound edition), 2003;

Editores: Ruhul Sarker, Robert Reynolds, Hussein Abbass, Kay Chen Tan, Bob McKay, Daryl Essam e Tom Gedeon.

Título: **Algoritmos Evolutivos Eficientes para o Problema de Clusterização em Grafos Orientados**

Autores: Carlos Rodrigo Dias e Luiz Satoru Ochi;

Local: Apresentação e publicação nos Anais do XXXV Simpósio Brasileiro de Pesquisa Operacional (XXXV SBPO), Natal, RN;

Editora: SOBRAPO, ISSN: 1518-1731, em CD-ROM, pp:1220-1232, 2003.

Título: **A Solução de Problemas de Particionamento em Grafos via Metaheurísticas Evolutivas**

Autores: Carlos Rodrigo Dias e Luiz Satoru Ochi;

Local: Encontro Regional de Matemática Aplicada e Computacional 2004 (ERMAC - 2004), Rio de Janeiro, RJ;

Editora: SBMAC, pp:26-27 (resumo), 2004;

Editores: Samuel Jurkiewicz, Rubens Sampaio, Nair Maria Maia de Abreu e Dani Gamerman.

Título: **Desenvolvimento e Análise Experimental de Algoritmos Evolutivos para o Problema de Clusterização Automática em Grafos Orientados**

Autores: Carlos Rodrigo Dias e Luiz Satoru Ochi;

Local: Aceito para apresentação e publicação nos anais do Brazilian Workshop on Evolutionary Computation, São Luiz, MA;

Editora: SBRN, 2004.

Referências Bibliográficas

- [1] Adamidis, P. *Co-operating populations: Improving the performance of Parallel Genetic Algorithms with Co-operating populations with different evolution behaviours*. Ph.D. thesis, 1997.
- [2] Adamidis, P. e Petridis, V. Co-operation populations with different evolution behaviors. In *1996 IEEE International Conference on Evolutionary Computation (ICEC '96)*, pp. 188-191, 1996.
- [3] Bäck, T. Optimization by Means of Genetic Algorithms. In *36th International Scientific Colloquium*, Technical University of Ilmenau, ed. E. Köhler, pp. 163-169, 1991.
- [4] Battiti, R. e Bertossi, A. Greedy, Prohibition, and Reactive Heuristics for Graph Partitioning. *IEEE Transactions on Computers*, 1999.
- [5] Battiti, R., Bertossi, A. e Cappelletti, A. Multilevel Reactive Tabu Search for Graph Partitioning. *Preprint UTM 554, Dip. Mat.*, Univ. Trento, Itália, 1999.
- [6] Beasley, D., Bull, D. R. e Martin R. R. An Overview of Genetic Algorithms: Part I, Fundamentals. *University Computing*, vol. 15, no. 2, pp. 58-69, 1993.
- [7] Berkhin, P. *Survey of Clustering Data Mining Techniques*. Accrue Software, 2002.
- [8] Ben-Dor, A., Shamir, R. e Yakhini, Z. Clustering gene expression patterns. *Journal of Computational Biology*, vol. 6, pp. 281-297, 1998.

- [9] Chiun, Y. e Lan, L. W. Genetic Clustering Algorithms. *European Journal of Operational Research* (135) 2, pp. 413-427, 2001.
- [10] Cole, R. M. Clustering with Genetic Algorithms. Master's thesis, Department of Computer Science, University of Western Australia, 1998.
- [11] Davidor, Y. Analogous Crossover. In *3rd International Conference on Genetic Algorithms*, pp. 98-103, San Mateo, California, 1989.
- [12] Demiriz, A., Bennett, K. P. e Embrechtz, M. J. Semi-Supervised Clustering Using Genetic Algorithms. R.P.I. Math Report No. 9901, Rensselaer Polytechnic Institute, Troy, New York, 1999.
- [13] Doval, D., Mancoridis, S. e Mitchell, B. S. Automatic Clustering of Software Systems using a Genetic Algorithm. In *1999 International Conference on Software Tools and Engineering Practice (STEP '99)*, 1999.
- [14] Drummond, L. M. A., Ochi, L. S., e Figueiredo, R. M. V. Design and Implementation of a Parallel Genetic Algorithm for the Travelling Purchaser Problem. In *Applied Computing/ACM*, pp. 257-263, 1997.
- [15] Ester, M., Kriegel, H. P., Sander, J., Wimmer, M. e Xu, X. Incremental Clustering for Mining in Data Warehousing Environment. In *Proceedings of the 24th VLDB Conference*, 1998.
- [16] Falkenauer, E. The Grouping Genetic Algorithms - Widening the Scope of the Gas. In *JORBEL - Belgian Journal of Operations Research, Statistics and Computer Science*, vol. 33 (1, 2), pp.79-102, 1993.
- [17] Falkenauer, E. A Hybrid Grouping Genetic Algorithm for Bin Packing. *Journal of Heuristics*, 2, pp. 530, 1996.
- [18] Fasulo, D. An Analysis of Recent Work on Clustering Algorithms. Technical Report, Dept. of Computer Science and Engineering, Univ. of Washington, 1999.

- [19] Fogel, L. J., Owens, A. J. e Walsh, M. J. *Artificial Intelligence Through Simulated Adaptation*. Wiley, New York, 1966.
- [20] Glover, F. e Laguna, M. Fundamentals of Scatter Search and Path Relinking. *Control and Cybernetics*, vol 29, no. 3, pp. 653-684, 2000.
- [21] Glover, F., Laguna, M. e Martí, R. *Scatter Search*. Springer-Verlag New York, Inc. New York, NY, USA, 2003.
- [22] Glover, F. Scatter Search and Star-Paths: Beyond the Genetic Metaphor. *OR Spektrum*, vol. 17, no. 2-3, pp. 125-138, 1995.
- [23] Goldberg, D. E. *Genetic Algorithms in search, optimization and machine learning*. Addison-Wesley, 1989.
- [24] Holland, J. H. *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor, 1975.
- [25] Hartuv, E., Schmitt, A., Lang, J. et al. An Algorithm for Clustering for Gene Expression Analysis. In *Proceedings of Third Annual International Conference on Computational Molecular Biology (RECOMB '99)*, 1999.
- [26] Koza, J. R. Hierarchical Genetic Algorithms Operating on Populations of Computer Programs. In *11th International Joint Conference on Artificial Inteligence*, ed. N. S. Sridharan, pp. 768-774, 1989.
- [27] Laguna, M. Scatter Search. In *Handbook of Applied Optimization*, eds. P. M. Pardalos e M. G. C. Resende, Oxford University Press, pp. 183-193, 2002.
- [28] Lorena, L. A. N. e Furtado, J. C. Constructive Genetic Algorithm for Clustering Problems. *Evolutionary Computation*, vol. 9, no. 3, pp. 309-327, 2001.
- [29] Lit, P. D., Falkenauer, E. e Delchambre, A. Grouping Genetic Algorithms: An Efficient Method to Solve the Cell Formation Problem. *Mathematics and Computers in Simulation*, 51, pp. 257-271, 2000.

- [30] Michalewicz, Z. *Genetic Algorithms + Data Structures = Evolution Programs*. Springer-Verlag, 1992.
- [31] Martí, R., Laguna, M. e Campos, V. Scatter Search vs Genetic Algorithms. Technical Report, University of Colorado at Boulder, (2002).
- [32] Maini, H. S., Mehrotra, K. G., Mohan, C. K. e Ranka, S. Genetic Algorithms for Graph Partitioning and Incremental Graph Partitioning. In *Proceedings of the 1994 Conference on Supercomputing*, pp. 449-457, 1994.
- [33] Moscato, P. On Evolution, Search, Optimization, Genetic Algorithms and Martial Arts: Towards Memetic Algorithms. Technical Report, Caltech Concurrent Computation Program, California Institute of Technology, 1989.
- [34] Mühlenbein, H. Genetic Algorithms. In *Local Search in Combinatorial Optimization*, John Wiley & Sons, Chichester, pp. 137-172, 1997.
- [35] Michalewicz, Z., Xiao, J. e Trojanowski, K. Evolutionary Computation: One Project, Many Directions. In *Proceedings of the 9th International Symposium (ISMIS '96)*, pp.189-201, 1996.
- [36] Oja, M., Nikkila, J., Toronen, P., Wong, G., Castren, E. e Kaski, S. Exploratory clustering of gene expression profiles of mutated yeast strains. In *Computational And Statistical Approaches To Genomics*, Kluwer Academic Publishers, pp. 64-78, 2002.
- [37] Ochi, L. S., Vianna, D. S. e Drummond, L. M. A. An asynchronous parallel metaheuristic for the period vehicle routing problem. In *Future Generations Computer Systems*, Elsevier, vol. 17, pp. 379-386, 2001
- [38] Radcliffe, N. J. e Surry, P. D. Formal Memetic Algorithms. In *Evolutionary Computing: AISB Workshop, Lecture Notes in Computer Science*, vol 865, pp. 1-16, Springer-Verlag, 1994.
- [39] Schwefel, H. P. *Numerical Optimization of Computer Models*. John Wiley & Sons, New York, 1981.

- [40] Tseng, L. Y. e Yang, S. B. A Genetic Clustering Algorithm for Data With Non-Spherical-Shape Clusters. *Pattern Recognition*, 33, pp. 1251-1259, 2000.
- [41] Whitley, D. A Genetic Algorithm Tutorial. *Statistics and Computing*, vol. 4, pp. 65-85, 1994.