

7

Experimentos Computacionais

Este capítulo objetiva validar a estratégia de paralelização hierárquica distribuída e o *middleware* de gerenciamento metaEasyGrid propostos nesta tese. Ambos foram integrados nas paralelizações desenvolvidas para o mTTP e para o problema da AGMD. Os experimentos computacionais realizados em um ambiente *grid* real são descritos e analisados nas Seções 7.2 e 7.3.

7.1

Considerações Iniciais

Os algoritmos testados e analisados neste capítulo foram desenvolvidos com a linguagem C++ e a versão 7.0.6 da implementação LAM [142] do padrão MPI [113, 114].

Os experimentos foram realizados em um ambiente *grid* real, conhecido por grid Sinergia [126]. Esse *grid* está localizado no Estado do Rio de Janeiro, e é composto por três sítios distribuídos em diferentes instituições e cidades, conforme ilustrado na Figura 7.1. O sítio pertencente à PUC é composto por dois *clusters*, localizados na cidade do Rio de Janeiro. O primeiro *cluster* é formado por 32 máquinas Pentium II 400 MHz e 280 Mbytes de RAM, enquanto o outro é composto por 32 máquinas Pentium IV 1.7 GHz e 256 Mbytes de RAM.

O segundo sítio está localizado na UFF e é formado por um *cluster* na cidade de Niterói, localizada a 40 quilômetros de distância do Rio de Janeiro, com 26 máquinas Pentium IV 2.6 GHz e três máquinas Pentium IV 3.2 GHz.

O terceiro sítio está instalado no LNCC, na cidade de Petrópolis, a 100 quilômetros do Rio de Janeiro, com sete máquinas Pentium IV 3.2 GHz. Todos os sítios são interconectados através de uma rede GigabitEthernet gerenciada pela RNP (Rede Nacional de Pesquisa) [88]. Nesse ambiente é usado o Globus Toolkit 2.4 [75] como *middleware* de serviço básico.

Durante a realização desta tese, foram realizados testes envolvendo até 100 máquinas pertencentes aos três sítios do grid Sinergia. Todavia, neste capítulo serão mostrados resultados envolvendo no máximo 60 máquinas, pois essa era a máxima quantidade de recursos disponíveis durante a maior parte

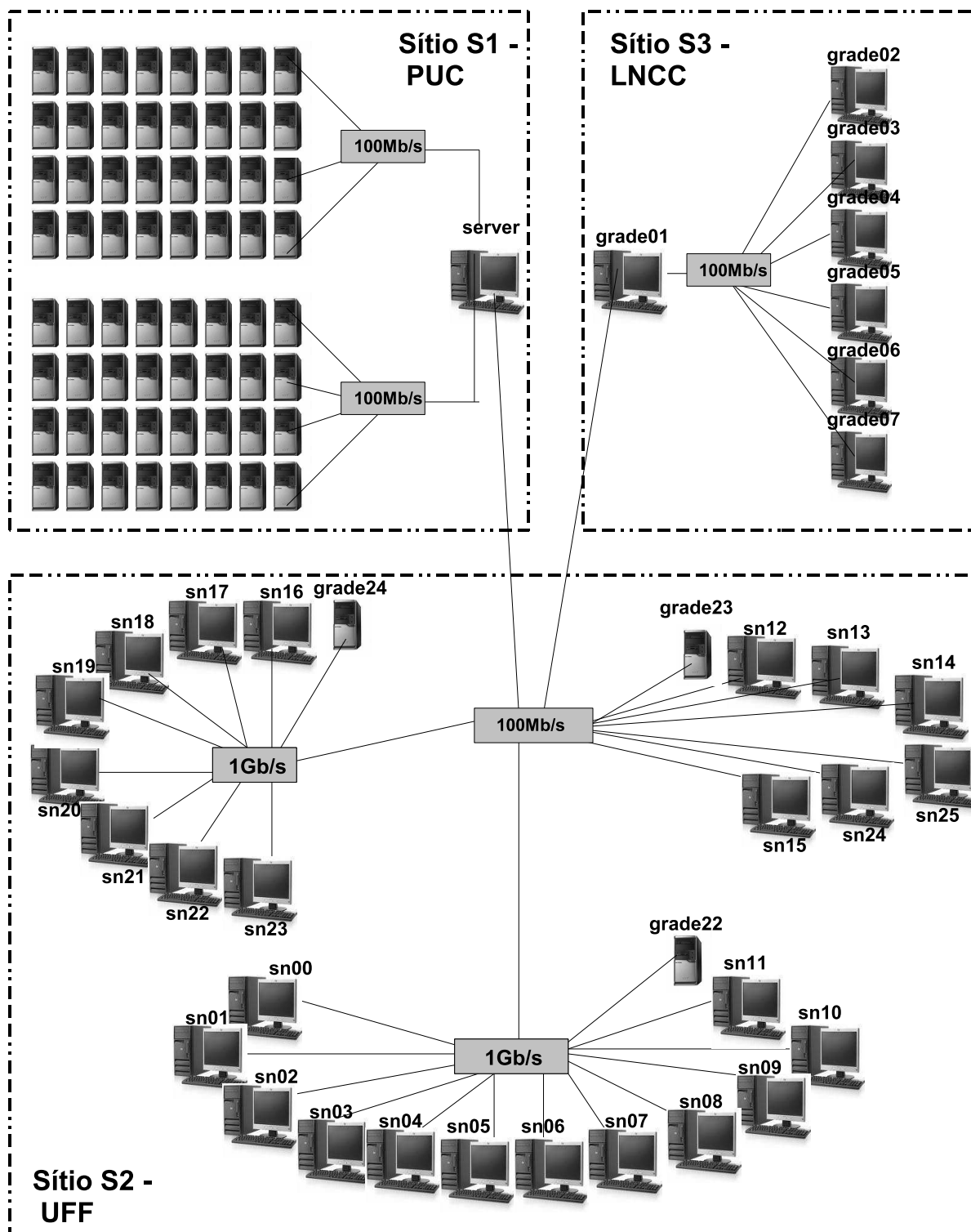


Figura 7.1: Configuração física do grid Sinergia.

do período de testes. A Tabela 7.1 apresenta os detalhes de configuração dos recursos usados.

Tabela 7.1: Recursos computacionais do grid Sinergia usados nos experimentos deste capítulo.

Local	Sítio	Máquinas	Processador	Cidade	Distância do S1
PUC	S1	32	PII 400 MHz	Rio de Janeiro	-
		32	PIV 1.7 GHz	Rio de Janeiro	-
UFF	S2	26	PIV 2.6 GHz	Niterói	40 Km
		03	PIV 3.2 GHz	Niterói	40 Km
LNCC	S3	07	PIV 3.2 GHz	Petrópolis	100 km

Por se tratar de um ambiente *grid* real, os experimentos não puderam ser sempre executados com recursos dedicados. Optou-se então por realizar os testes durante os finais de semana, nos feriados e à noite, quando o tráfego na rede é baixo e o uso dos recursos computacionais é mais facilmente gerenciado. Este capítulo foi dividido em duas seções. A Seção 7.2 traz os experimentos computacionais realizados com a implementação paralela AUDImTTP e a Seção 7.3 apresenta os resultados da implementação paralela AUDI-AGMD.

7.2

Experimentos Computacionais com AUDImTTP

Nesta seção, os resultados obtidos com a implementação paralela AUDImTTP são comparados com os alcançados pela versão PAR-MP, que apresentou melhor desempenho entre as versões paralelas analisadas no Capítulo 5.

Para realizar essa comparação os algoritmos PAR-MP e AUDImTTP foram programados para encerrarem suas execuções tão logo fosse encontrada uma solução com custo menor ou igual ao valor alvo estabelecido. Os tempos computacionais apresentados nas próximas seções foram medidos em segundos e representam a média sobre cinco execuções, com diferentes sementes para o gerador de números aleatórios.

7.2.1

Instâncias de Teste

A implementação AUDImTTP foi testada com três grupos diferentes de instâncias. A maioria dos testes realizados neste capítulo usaram as instâncias dos grupos *cir_n*, *nln* e a instância *br24*, apresentadas na Seção 5.6.1. Outro grupo de instâncias, denotado por *nfln*, foi criado a partir de dados reais,

usando para isso as distâncias entre as cidades das equipes da *National Football League* dos Estados Unidos.

Esse grupo contém as maiores instâncias do mTTP, formadas por $16 \leq n \leq 32$ equipes. Essas instâncias estão disponíveis em [145], onde encontram-se também os valores das melhores soluções conhecidas para cada uma delas.

7.2.2

Resultados Computacionais

Nesta seção são apresentados os experimentos computacionais que objetivam investigar a qualidade das soluções geradas a partir da implementação paralela AUDImTTP e a robustez desse algoritmo em um ambiente *grid* real.

Primeiro Experimento - Qualidade das Soluções

O objetivo deste experimento foi avaliar a qualidade das soluções alcançadas pelas implementações seqüencial, PAR-MP e AUDImTTP. Na Tabela 7.2 cada linha corresponde a uma instância testada. Para cada instância é mostrada a melhor solução encontrada pela implementação seqüencial, pela implementação paralela PAR-MP e pela implementação AUDImTTP. A última coluna dessa tabela mostra a melhoria percentual obtida com as soluções de AUDImTTP em relação ao algoritmo seqüencial.

Tabela 7.2: Qualidade das soluções encontradas pelas implementações seqüencial, PAR-MP e AUDImTTP com as instâncias do mTTP do primeiro e segundo grupos.

Instância	Seqüencial	PAR-MP	AUDImTTP	Melhoria (%)
circ8	140	140	140	-
circ10	276	272	272	1,45
circ12	456	456	456	-
circ14	714	714	696	2,52
circ16	1004	978	974	2,99
circ18	1364	1306	1306	4,25
circ20	1882	1882	1882	-
nl8	41928	41928	41928	-
nl10	63832	63832	63832	-
nl12	120655	120655	119608	0,87
nl14	208086	208086	199363	4,19
nl16	285614	279618	279077	2,29
br24	506433	503158	502971	0,68

As soluções apresentadas na Tabela 7.2 para a versão sequencial são as melhores alcançadas após cinco dias de processamento [147]. As soluções atingidas pelos algoritmos paralelos PAR-MP e AUDImTTP foram as melhores após 20 horas de processamento.

Os algoritmos paralelos PAR-MP e AUDImTTP foram executados em um ambiente distribuído com dez máquinas pertencentes a um sub-grupo de máquinas homogêneas do sítio S2, conforme ilustrado na Figura 7.2. A versão sequencial também executou em uma máquina pertencente ao sítio S2.

As listas de soluções de elite mantidas por PAR-MP e AUDImTTP foram definidas com tamanho igual a oito. A probabilidade Q de reiniciar a execução de um processo trabalhador (no caso da versão PAR-MP) ou provedor de solução (no caso da implementação AUDImTTP) a partir da fase de busca local ILS foi fixada em 10%.

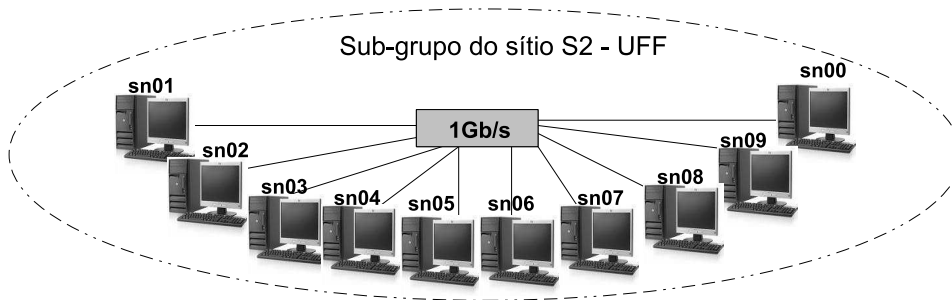


Figura 7.2: Configuração usada nos experimentos realizados com um sub-grupo de máquinas do sítio S2.

Destacam-se em **negrito** na Tabela 7.2 os custos das soluções que a implementação paralela AUDImTTP conseguiu melhorar. Como pode ser notado, a implementação AUDImTTP conseguiu resultados melhores do que as versões sequencial e PAR-MP para seis das 13 instâncias testadas. Para as instâncias *circ8*, *circ10*, *circ18*, *nl8*, *nl10* e *nl12* os valores encontrados foram iguais aos melhores valores conhecidos na literatura.

Esses resultados ilustram que, quanto à qualidade das soluções, o algoritmo paralelo com a hierarquia de listas é melhor que as demais implementações. Ele é capaz de melhorar em até 4,19% o custo da solução da instância *nl14*, cuja solução não havia sido melhorada por PAR-MP.

Verificada a superioridade do algoritmo paralelo AUDImTTP em relação à qualidade das soluções, decidiu-se investigar os fatores responsáveis por esse melhor desempenho. Na próxima seção serão analisadas as características que tornam promissor o uso da estratégia hierárquica distribuída.

Segundo Experimento - Estratégia

O segundo experimento objetiva comparar o uso da estratégia proposta para paralelizar metaheurísticas com a estratégia tradicional mestre-trabalhador, implementada na versão PAR-MP. Este teste foi realizado com as mesmas dez máquinas do sítio S2 usadas no experimento anterior. Em princípio, o ambiente de teste usado é favorável à estratégia de paralelização mestre-trabalhador, pois é homogêneo e interligado por rede LAN.

Para tornar a comparação mais equilibrada, o teste que gerou os resultados apresentados na Tabela 7.3 para a implementação AUDImTTP usou apenas uma lista de intensificação. Assim, PAR-MP executou com um processo mestre e nove processos trabalhadores (um em cada máquina). A implementação AUDImTTP empregou um processo gerenciador da lista de cooperação, um processo gerenciador da lista de intensificação e nove provedores de solução, sendo que um dos processos provedores executou concorrentemente com o gerenciador da lista de intensificação em um dos recursos. O tamanho da lista de soluções de elite usada nas duas execuções foi igual a oito. O parâmetro que representa a probabilidade Q de reiniciar a execução de um processo a partir da fase de busca ILS foi fixado em 10%.

Com a adoção de uma única lista de intensificação é possível avaliar se há ganho de desempenho em se manter a hierarquia entre as listas de soluções de elite implementadas pelo algoritmo AUDImTTP. Na Tabela 7.3 são apresentados, para cada uma das 13 instâncias do primeiro e do segundo grupos, o valor alvo usado como critério de parada de ambos os algoritmos e os tempos computacionais médios em segundos sobre cinco execuções para a versão PAR-MP e para o algoritmo AUDImTTP.

Apesar da implementação AUDImTTP ter executado com apenas uma lista de intensificação, ela obteve melhor desempenho para todas as instâncias com tempo de execução superior a 200 segundos. Por exemplo, para a instância *circ16* a versão PAR-MP apresentou tempo de processamento quase sete vezes o do algoritmo AUDImTTP. Todavia, para as instâncias pequenas, o desempenho de PAR-MP foi melhor. Isso ocorre porque, para as instâncias pequenas, o tempo necessário para ativar todos os processos do *middleware* SGA metaEasyGrid e criar a hierarquia de listas é maior do que o tempo necessário para atingir o critério de parada.

A redução no tempo de processamento das instâncias *circ10*, *circ16*, *circ18*, *nl16* e *br24* foi considerável. Essa redução pode ser atribuída a dois fatores: uso da técnica de renovação na lista de intensificação e gerência do ambiente *grid* pelo *middleware* SGA metaEasyGrid.

Com o objetivo de investigar dentre esses fatores qual é o primordial

Tabela 7.3: Tempos médios em segundos das implementações PAR-MP e AUDImTTP com apenas uma lista de intensificação, executadas em dez máquinas em ambiente LAN com tamanho das listas de soluções de elite igual a oito.

Instância	Alvo	PAR-MP	AUDImTTP
circ8	140	0,03	1,28
circ10	276	629,43	297,54
circ12	456	0,18	2,57
circ14	714	0,47	2,53
circ16	982	3.744,72	575,66
circ18	1308	6.703,00	3.570,78
circ20	1882	26,64	27,49
nl8	41928	0,05	1,86
nl10	63832	43,18	43,19
nl12	120655	2,71	3,69
nl14	208086	2,43	4,75
nl16	280116	3.894,20	2.705,62
br24	504000	4.905,63	1.154,02

para o bom desempenho da implementação AUDImTTP, foram feitos testes com algumas variações nas implementações de PAR-MP e AUDImTTP. Esse experimento foi realizado apenas com as cinco instâncias (*circ10*, *circ16*, *circ18*, *nl16* e *br24*) em que a implementação AUDImTTP conseguiu redução no tempo de processamento no experimento anterior.

Para cada instância mostrada na Tabela 7.4, são apresentados o valor alvo usado como critério de parada, o tempo médio em segundos sobre cinco execuções para a versão PAR-MP implementada com variação de sementes, e para a implementação AUDImTTP com e sem ativação da operação de renovação.

A variação de sementes implementada no algoritmo PAR-MP significa que uma nova semente será enviada em cada estabilização que um processo trabalhador sofrer. Na variação da implementação do algoritmo AUDImTTP com uma única lista de intensificação, sem aplicação da técnica de renovação, significa que a lista de cooperação torna-se uma cópia fiel da única lista de intensificação. Nesse caso, a implementação AUDImTTP tem um comportamento similar ao da versão PAR-MP em relação à diversidade de soluções na lista.

Inicialmente, compararam-se as duas implementações do AUDImTTP. Nessa análise, notou-se que o uso da técnica de renovação traz melhor benefício para o algoritmo AUDImTTP, pois com a renovação das listas de intensificação aumenta a variedade das soluções a serem pesquisadas.

Tabela 7.4: Tempos médios em segundos sobre cinco execuções das implementações PAR-MP e AUDImTTP executadas em dez máquinas em ambiente LAN com tamanho da lista de soluções de elite igual a oito.

Instância	Alvo	PAR-MP com sementes diferentes(s)	AUDImTTP(s)	
			Sem renovação	Com renovação
circ10	276	304,47	299,22	297,54
circ16	982	1.628,32	613,27	575,66
circ18	1308	5.014,92	4.843,44	3.570,78
nl16	280116	3.290,95	3.128,01	2.705,62
br24	504000	2.193,14	1.236,55	1.154,02

A redução no tempo computacional com a técnica de renovação foi em média 10,63% em relação à execução sem renovação. Contrariamente, como em PAR-MP não há renovação, as execuções com instâncias maiores tendem a ficar com sua única lista de soluções de elite ultrapassada, pois sem renovação as soluções ficam armazenadas na lista por longos períodos de tempo.

A fim de analisar se o bom desempenho da implementação AUDImTTP é consequência da gerência feita pelo SGA metaEasyGrid, comparou-se a implementação de AUDImTTP com a versão PAR-MP. Para tornar essa comparação mais apropriada, analisou-se o comportamento da versão de PAR-MP com a implementação AUDImTTP sem a técnica de renovação. Nesse caso, ambas as implementações trabalham com múltiplas sementes durante a execução e, apesar da implementação AUDImTTP ter executado com duas listas (uma lista de cooperação e uma lista de intensificação), sua lista de cooperação é uma cópia da lista de intensificação, agindo como se houvesse uma única lista.

Como pode ser observado na terceira e quarta colunas da Tabela 7.4, embora a versão PAR-MP tenha obtido um bom desempenho, a implementação AUDImTTP teve um desempenho superior para as cinco instâncias analisadas. A redução no tempo de processamento alcançada foi de 23,21% em relação a PAR-MP. Esse desempenho é alcançado pelo fato do *middleware* SGA metaEasyGrid gerenciar a execução da implementação AUDImTTP. Com o *middleware* metaEasyGrid é feito o balanceamento de carga entre as máquinas envolvidas na execução e a criação das tarefas é realizada de acordo com a demanda do ambiente.

Com os experimentos apresentados, conclui-se que a superioridade do algoritmo AUDImTTP é resultante da integração entre a estratégia de paralelização hierárquica distribuída e o *middleware* SGA metaEasyGrid. A primeira possibilita trabalhar com a hierarquia das listas de soluções de elite

e com a técnica de renovação. Por outro lado, o *middleware* gerencia eficientemente o ambiente, adequando a execução da aplicação às mudanças ocorridas.

Terceiro Experimento - Multiplicidade das Listas

Neste experimento é avaliado o comportamento do algoritmo AUDImTTP com o uso de mais de uma lista de intensificação. A Tabela 7.5 apresenta, para cada instância testada, o alvo usado como critério de parada e os tempos médios em segundos sobre cinco execuções do algoritmo AUDImTTP com uma e duas listas de intensificação, respectivamente.

Tabela 7.5: Tempos médios em segundos sobre cinco execuções da implementação AUDImTTP com uma e duas listas de intensificação em um ambiente LAN formado por 10 máquinas.

Instância	Alvo	Uma lista (segundos)	Duas listas (segundos)
circ8	140	1,28	0,92
circ10	274	297,54	252,26
circ12	456	2,57	1,28
circ14	714	2,53	1,81
circ16	984	575,66	515,86
circ18	1310	3.570,78	3.482,06
circ20	1882	27,49	24,31
nl8	41928	1,86	0,98
nl10	63832	43,19	41,80
nl12	120655	3,69	3,36
nl14	208086	4,75	2,78
nl16	280117	2.705,62	2.685,08
br24	503158	1.154,02	1.129,62

Esses resultados mostram os benefícios de se trabalhar com a distribuição de listas de intensificação, pois com sua multiplicidade é possível dividir a carga dos processos que a gerenciam e aumentar a diversidade das soluções armazenadas. Essa última vantagem ocorre pelo fato de cada lista manter suas próprias soluções, independente das soluções armazenadas nas demais listas pelos outros gerenciadores.

Como este experimento foi desenvolvido em um ambiente LAN, surgiu a idéia de analisar os benefícios da multiplicidade de listas de intensificação em ambiente WAN. As instâncias testadas nesse novo experimento foram *circ10*, *circ16*, *circ18*, *nl16* e *br24*, por serem as instâncias mais pesadas e cujos tempos de processamento foram maiores que 200 segundos no experimento apresentado na Tabela 7.5.

Nos experimentos relatados na Tabela 7.6, os resultados em ambiente LAN referem-se à execução em um *cluster* local, formado pelas máquinas pertencentes ao sítio S2 do grid Sinergia. Os resultados em ambiente WAN dizem respeito à execução realizada em máquinas pertencentes a dois diferentes *clusters*, um localizado no sítio S1 e o outro no sítio S2, conforme ilustrado na Figura 7.3.

Tabela 7.6: Tempos médios em segundos sobre cinco execuções das implementações PAR-MP e AUDImTTP executadas em ambientes LAN e WAN com dez processadores, com a implementação AUDImTTP executando com duas listas de intensificação.

Inst.	PAR-MP (s)			AUDImTTP (s)			$\frac{\text{PAR-MP}}{\text{AUDImTTP}}$	
	LAN	WAN	(%)	LAN	WAN	(%)	LAN	WAN
circ10	629,43	747,52	18,76	252,26	257,95	2,26	2,50	2,90
circ16	3.744,72	4.537,74	21,18	515,86	521,49	1,09	7,26	8,70
circ18	6.703,00	7.532,03	12,37	3.482,06	3.516,47	0,99	1,93	2,14
nl16	3.894,20	4.500,76	15,58	2.685,08	2.723,68	1,44	1,45	1,65
br24	4.905,63	5.106,51	4,09	1.129,62	1.190,35	5,38	4,34	4,29
		Média	14,04			2,01	3,50	3,94

A primeira coluna da Tabela 7.6 informa as instâncias analisadas. A segunda e terceira colunas informam os tempos médios em segundos sobre cinco execuções da versão PAR-MP em ambientes LAN e WAN, respectivamente. A quarta coluna apresenta as diferenças percentuais nos tempos de execução em ambiente LAN em relação ao ambiente WAN para PAR-MP. A quinta e sexta colunas informam o tempo computacional médio em segundos sobre cinco execuções da implementação AUDImTTP em ambientes LAN e WAN, respectivamente. A sétima coluna apresenta as diferenças percentuais nos tempos de execução da implementação AUDImTTP quando executada sob os dois ambientes. As duas últimas colunas apresentam a redução no tempo de processamento de AUDImTTP e PAR-MP para os dois ambientes LAN e WAN, respectivamente. O critério de parada utilizado para ambos os algoritmos foi a obtenção de soluções com custo menor ou igual aos valores alvos mostrados na Tabela 7.4.

Os resultados na Tabela 7.6 destacam as vantagens da estratégia proposta, mesmo que o ambiente de execução LAN seja favorável ao modelo de paralelismo mestre-trabalhador por ser homogêneo e dedicado. Além disso, os experimentos em ambiente WAN mostram que o desempenho de PAR-MP degrada significativamente (em média 14,04%), enquanto que a implementação

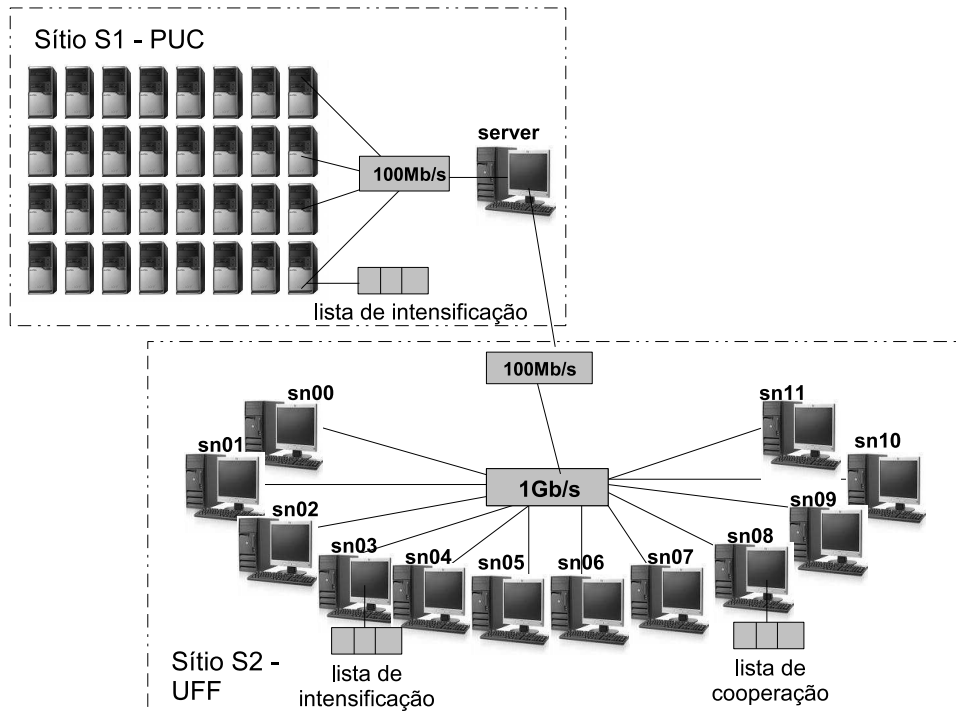


Figura 7.3: Configuração usada nos experimentos realizados sob ambiente WAN formado pelos sítios S1 e S2 do grid Sinergia.

AUDImTTP mantém quase o mesmo desempenho, com uma degradação média de apenas 2,01%.

A redução média no tempo de processamento de AUDImTTP em relação a PAR-MP na execução em ambiente WAN é 3,94, levemente maior do que em LAN (3,50), consolidando a vantagem de se trabalhar com a distribuição das listas de intensificação, principalmente em ambientes que há muita troca de mensagens entre diferentes redes LANs.

A Figura 7.4 mostra um gráfico do tipo `tttplot` [5] para a instância *circ10*, executada 100 vezes para cada uma das implementações PAR-MP e AUDImTTP em ambientes LAN e WAN. Esse gráfico mostra a distribuição de probabilidade do tempo de cada versão convergir para um dado valor alvo. O valor alvo usado como critério de parada para a instância *circ10* foi 274. O objetivo desse experimento é analisar com um maior número de execuções o comportamento dessas duas implementações em ambientes diferentes.

O benefício de usar listas distribuídas é confirmado pelos resultados apresentados na Figura 7.4. A degradação no desempenho do algoritmo AUDImTTP é muito menor do que em PAR-MP. Como observado no gráfico, as curvas referentes às execuções de AUDImTTP em ambientes LAN e WAN quase se sobrepõem. Esse melhor desempenho é causado pelo fato da maioria das mensagens serem trocadas localmente na versão AUDImTTP, dentro da mesma rede local. Isso é possível porque pode ser disparado um processo

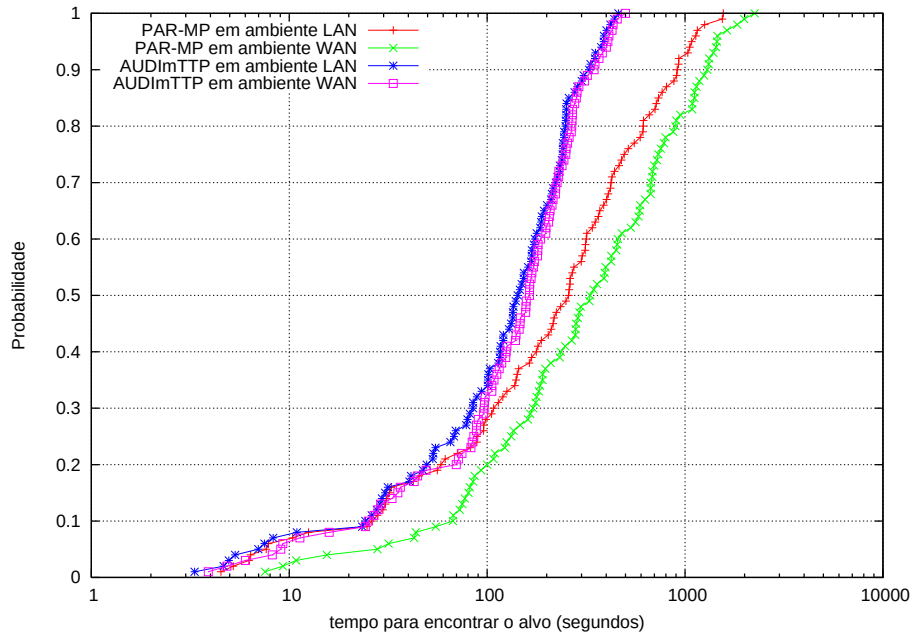


Figura 7.4: Distribuição de probabilidade empírica do tempo para atingir uma solução com custo menor ou igual ao valor alvo, considerando-se o alvo 274 para a instância *circ10*. As implementações PAR-MP e AUDImTTP foram executadas 100 vezes cada uma em ambientes LAN e WAN formados por dez máquinas pertencentes aos sítios S1 e S2.

gerenciador da lista de intensificação em cada sítio do *grid*.

Contrariamente, na versão PAR-MP todos os processos trabalhadores trocam mensagens com o processo mestre. Com isso, os processos trabalhadores que estão executando em uma rede diferente da que está localizado o mestre, tendem a ficar uma parte do tempo de processamento esperando por mensagens do mestre. Conseqüentemente, isso acarreta um considerável aumento no tempo total de execução.

No gráfico apresentado na Figura 7.4, também é possível notar que a diferença no tempo de execução entre as execuções da versão PAR-MP em ambientes LAN e WAN é visivelmente maior para as execuções que finalizam em até 100 segundos. Isso ocorre porque, nas execuções menores, a tendência é que a espera por mensagens faça maior diferença no tempo total de execução.

Quarto Experimento - Tamanho das Listas

Outro experimento realizado teve como objetivo avaliar se o bom desempenho da implementação AUDImTTP é devido ou não ao fato de ela ter uma maior capacidade de armazenamento de soluções de elite, se comparada com a versão PAR-MP.

Essa avaliação fez-se necessária porque a implementação AUDImTTP trabalha com pelo menos uma lista de soluções de elite a mais que a versão

PAR-MP. No caso mais simples, a implementação AUDImTTP adota uma lista de cooperação e uma de intensificação, contra apenas uma lista de soluções de elite mantida pela versão paralela PAR-MP.

O experimento baseou-se em executar PAR-MP com uma lista de soluções de elite de tamanho igual à soma do tamanho de todas as listas usadas em uma execução de AUDImTTP. Os valores apresentados na Tabela 7.7 são tempos computacionais médios em segundos sobre cinco execuções. Para cada instância analisada, são mostrados os resultados obtidos com as execuções de PAR-MP usando uma lista de 24 posições, de PAR-MP com uma lista de oito posições e AUDImTTP com três listas (uma de cooperação e duas de intensificação) de oito posições cada. Para cada uma das três configurações os testes foram realizados em ambientes LAN e WAN. O critério de parada adotado para todos os algoritmos foi a obtenção de uma solução cujo custo seja pelo menos igual a um dado valor alvo. Os valores alvos usados neste experimento foram os mesmos apresentados na Tabela 7.4.

Tabela 7.7: Tempos médios em segundos sobre cinco execuções das implementações PAR-MP e AUDImTTP usando listas de soluções de elite com 24 e 8 posições, no caso de PAR-MP, e três listas de soluções de elite com oito posições cada, no caso de AUDImTTP.

Instância	PAR-MP 1 lista de 24 posições		PAR-MP 1 lista de 8 posições		AUDImTTP 3 listas de 8 posições	
	LAN	WAN	LAN	WAN	LAN	WAN
circ10	663,70	782,60	629,43	747,52	252,26	257,95
circ16	3.946,11	4.613,17	3.744,72	4.537,74	515,86	521,49
circ18	6.928,72	7.593,24	6.703,00	7.532,03	3.482,06	3.516,47
nl16	3.945,38	4.605,95	3.894,20	4.500,76	2.685,08	2.723,68
br24	4.925,39	5.333,45	4.905,63	5.106,51	1.129,62	1.190,35

Os resultados confirmam que a quantidade total de soluções armazenadas na estrutura hierárquica e distribuída das listas de soluções de elite da implementação AUDImTTP não é a única responsável pelo bom desempenho dessa aplicação. A versão PAR-MP, mesmo tendo executado com uma lista que tem a mesma capacidade de armazenamento que a soma das três listas usadas na execução de AUDImTTP, não foi capaz de melhorar o seu desempenho. Ao contrário, o desempenho da execução de PAR-MP com uma lista de 24 posições foi pior do que com oito posições. Isso ocorre porque, com uma lista muito grande, soluções de baixa qualidade são armazenadas na lista de soluções de elite.

Com a adoção da hierarquia de listas na implementação AUDImTTP, as

soluções de elite são selecionadas em mais de um nível e apenas as melhores dentre elas são armazenadas na lista de cooperação. Dessa forma, nota-se claramente que o melhor desempenho da implementação AUDImTTP não está no fato de ela armazenar muitas soluções, mas sim de armazenar boas soluções que freqüentemente são renovadas.

Quinto Experimento - Escalabilidade

Nesta seção, avalia-se a escalabilidade da implementação AUDImTTP em relação à multiplicidade das listas. O experimento foi realizado com 60 máquinas do grid Sinergia.

Na Tabela 7.8 são mostrados os tempos médios em segundos sobre dez execuções para as implementações PAR-MP e AUDImTTP. Ambos algoritmos executaram até que o valor alvo indicado na Tabela 7.4 fosse alcançado.

Tabela 7.8: Tempos médios em segundos sobre dez execuções das implementações PAR-MP e AUDImTTP em 60 máquinas do grid Sinergia, sendo a última implementação testada em três, cinco e sete listas de intensificação.

Instância	PAR-MP	AUDImTTP			$\frac{\text{PAR-MP}}{\text{AUDImTTP}}$		
		3 listas	5 listas	7 listas	3 listas	5 listas	7 listas
circ10	72,11	36,84	31,02	32,57	1,96	2,32	2,21
circ16	980,66	345,03	319,22	321,54	2,84	3,07	3,05
circ18	5.647,42	1.822,45	1.783,69	1.801,76	3,10	3,17	3,13
nl16	2.759,23	1.975,99	1.678,51	1.708,97	1,40	1,64	1,61
br24	1.244,28	493,82	374,01	390,68	2,52	3,33	3,18
Média					2,36	2,71	2,64

A Tabela 7.8 apresenta na primeira coluna as instâncias testadas. A segunda coluna mostra o tempo computacional da versão PAR-MP para atingir o critério de parada. As próximas três colunas apresentam o tempo de execução do algoritmo AUDImTTP com três, cinco e sete listas de intensificação, respectivamente. Em seguida, é apresentada a redução no tempo computacional da implementação AUDImTTP com três, cinco e sete listas de intensificação em relação à versão PAR-MP.

Na execução da implementação AUDImTTP com três listas de intensificação, foi criada uma lista no sítio S1, uma no sítio S2 e a terceira no sítio S4, conforme mostrado na Figura 7.5. Na execução com cinco listas, criam-se duas no sítio S1, duas no sítio S2 e uma no sítio S3. Por último, definiram-se três listas no sítio S1, três no sítio S2 e uma no sítio S3.

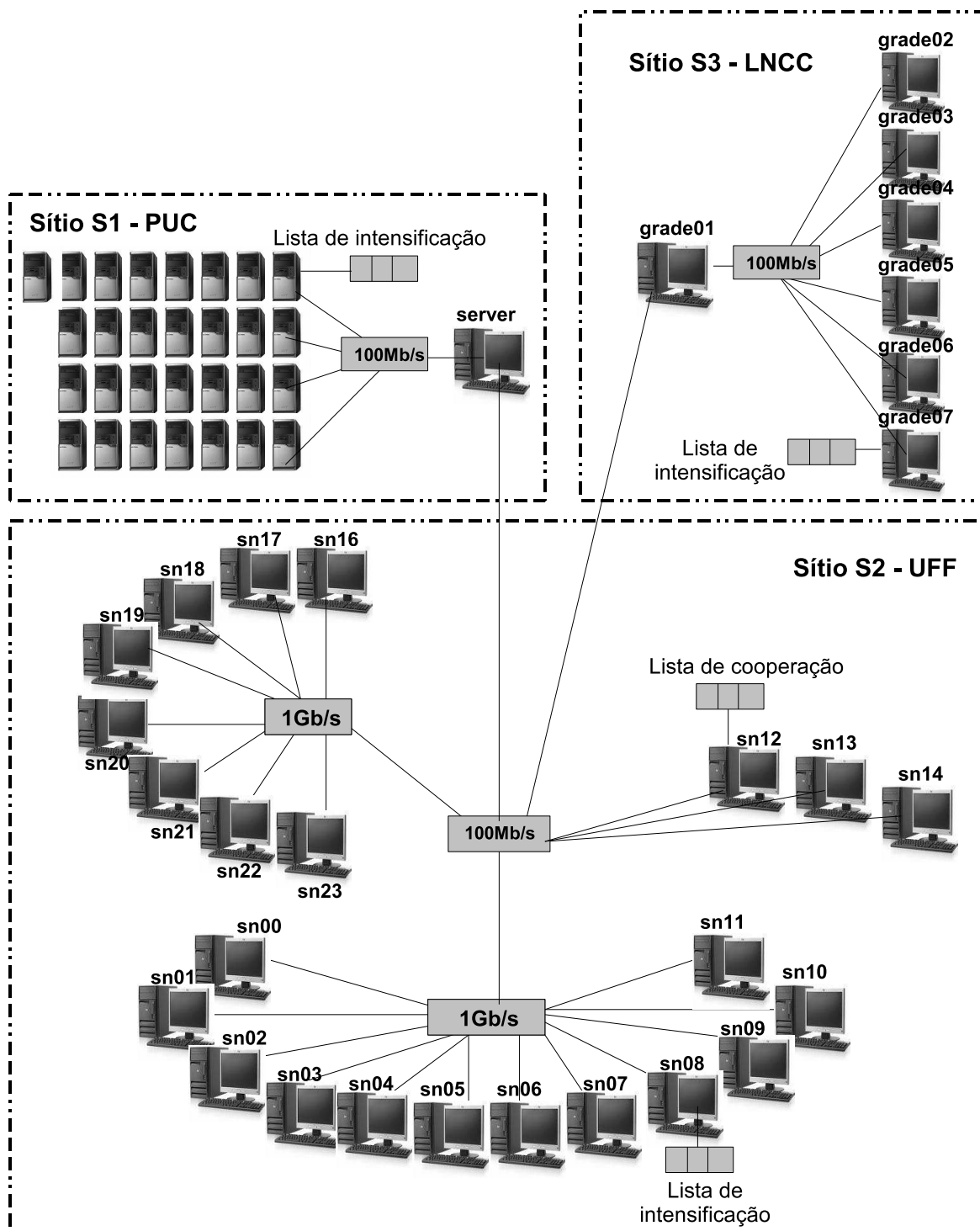


Figura 7.5: Configuração usada nos experimentos realizados com 60 máquinas através dos sítios S1, S2 e S3 do grid Sinergia.

A escolha da quantidade de listas de intensificação e da devida distribuição entre os sítios do *grid* foi feita de acordo com o número de máquinas em cada sítio e de sua localização física. Esse último fator justifica a criação, em todas as execuções, de uma lista de cooperação no sítio S3, visto que o mesmo foi formado por apenas sete máquinas.

Como pode ser observado na Tabela 7.8, todas as execuções da implementação AUDImTTP tiveram um tempo de processamento menor do que a execução de PAR-MP nas 60 máquinas. Dentre as três configurações de listas testadas com a implementação AUDImTTP, a execução com cinco listas de intensificação teve o melhor desempenho, alcançando uma redução média no tempo de processamento equivalente a 2,71. O tempo computacional médio da execução com sete máquinas foi levemente pior do que com cinco, apresentando uma redução média do tempo igual a 2,64 em relação a versão PAR-MP.

Contudo, os resultados mostraram que, para essa aplicação e com esse número de máquinas, a partir de sete listas o sistema torna-se saturado em relação ao número de gerenciadores das listas de intensificação.

Isso pode ser melhor visualizado no gráfico do tipo *tttplot* mostrado na Figura 7.6. Esse gráfico ilustra a execução da instância *circ10* pelas implementações PAR-MP, AUDImTTP com três, cinco, sete e nove listas de intensificação. Cada implementação foi executada 200 vezes e o alvo usado como critério de parada foi 274. As curvas relativas aos resultados obtidos com a implementação AUDImTTP indicam que seu desempenho é melhor do que o algoritmo PAR-MP.

Além disso, nota-se a saturação na multiplicidade das listas, pois com um número elevado de listas de intensificação há uma degradação no tempo computacional da implementação AUDImTTP. Por exemplo, a execução com nove listas de intensificação foi a que teve o pior desempenho dentre todas as configurações do AUDImTTP, sendo melhor apenas que a versão PAR-MP.

Esse gráfico deixa claro que há um ponto de saturação em relação ao número adequado de listas de intensificação. No ambiente formado por 60 máquinas esse ponto de saturação foi atingido pela instância *circ10* a partir de sete listas.

O uso de sete ou mais listas de intensificação nesse ambiente implica que mais de 10% dos recursos estão alocados exclusivamente para serem usados pelos processos gerenciadores das listas. Dessa forma, é subutilizada a capacidade de processamento que seria usada pelos processos provedores de solução. Uma alternativa para esse problema seria executar os processos gerenciadores das listas concorrentemente com os processos provedores de solução.

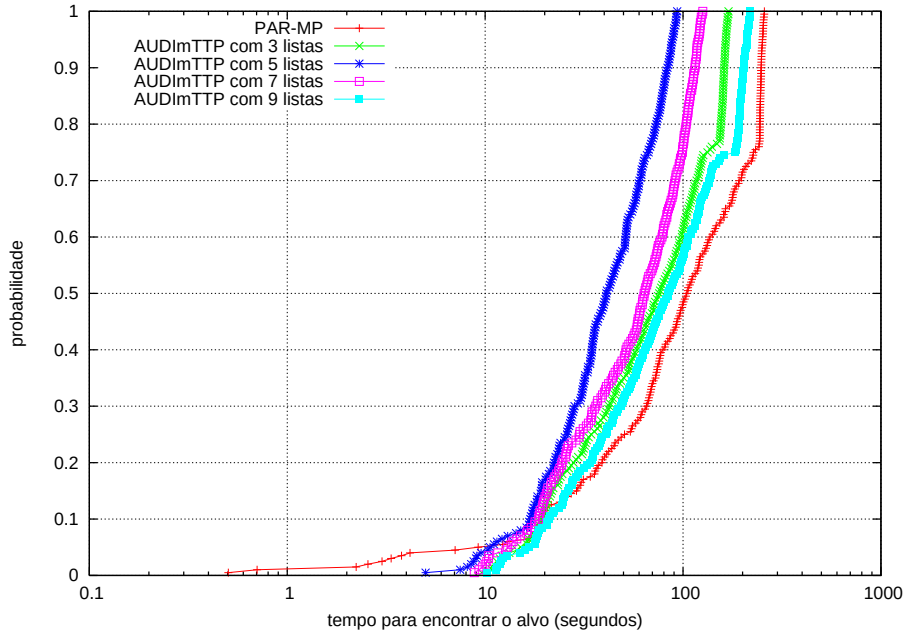


Figura 7.6: Distribuição de probabilidade empírica do tempo para atingir uma solução com custo menor ou igual ao valor alvo, considerando-se o alvo 274 para a instância *circ10*. As implementações PAR-MP, AUDImTTP com três, cinco, sete e nove listas de intensificação foram executadas 200 vezes cada uma em 60 máquinas do grid Sinergia.

Na Tabela 7.9 são apresentadas as melhorias percentuais nos tempos médios de processamento da implementação AUDImTTP em relação a PAR-MP. A primeira coluna da Tabela 7.9 nomeia as instâncias testadas. As demais colunas mostram o ganho percentual no tempo de execução das implementações AUDImTTP com três, cinco e sete listas em relação à versão PAR-MP. Esse ganho percentual foi dado através da fórmula:

$$Melhoria = \left(\frac{t(\text{PAR-MP}) - t(\text{AUDImTTP})}{t(\text{PAR-MP})} \right) \cdot 100,$$

onde $t(\text{PAR-MP})$ e $t(\text{AUDImTTP})$ representam, respectivamente, o tempo de processamento de PAR-MP e o tempo de processamento de AUDImTTP, ambos em 60 máquinas. O $t(\text{AUDImTTP})$ foi calculado para as execuções com três, cinco e sete listas de intensificação.

Os resultados mostrados na Tabela 7.9, ratificam a superioridade da implementação AUDImTTP em relação à versão PAR-MP. Dentre esses resultados, o maior percentual médio de melhoria equivale a 60,39%, atingido pelo algoritmo AUDImTTP quando executado com cinco listas de intensificação.

Tabela 7.9: Percentual de melhoria da implementação AUDImTTP com três, cinco e sete listas de intensificação em relação à versão PAR-MP.

Instância	Melhoria (%)		
	três listas	cinco listas	sete listas
circ10	48,91	56,98	54,83
circ16	64,82	67,45	67,21
circ18	67,73	68,42	68,10
nl16	28,39	39,17	38,06
br24	60,31	69,94	68,60
Média	54,03	60,39	59,36

Sexto Experimento - Qualidade das Soluções com o Grupo 3

Este experimento objetiva avaliar a qualidade das soluções obtidas pela implementação AUDImTTP para as instâncias do grupo *nfln*. Os resultados obtidos com este grupo de instância reforçam o bom desempenho demonstrado nos experimentos anteriores pelo algoritmo AUDImTTP.

Na Tabela 7.10, a primeira coluna nomeia todas as instâncias do grupo *nfl*. O valor da melhor solução conhecida na literatura para cada instância é apresentado na coluna indicada por S^* . Em seguida, são apresentadas as melhores soluções alcançadas pela heurística sequencial GRILS-mTTP e pela implementação paralela AUDImTTP, em um ambiente formado por dez máquinas do sítio S2.

Tabela 7.10: Custos das melhores soluções alcançadas pelo algoritmo GRILS-mTTP e pela implementação AUDImTTP para as instâncias do Grupo 3 do mTTP.

Instância	S^*	Sequencial	AUDImTTP
nfl16	248818	251289	249806
nfl18	299903	299903	299112
nfl20	359748	359748	359748
nfl22	418086	418086	418022
nfl24	467135	467135	465491
nfl26	554670	554670	548643
nfl28	618801	618801	609788
nfl30	740458	740458	739697
nfl32	924559	924559	914620

A implementação paralela AUDImTTP conseguiu melhorar sete das melhores soluções conhecidas para as nove instâncias do Grupo 3 e encontrou a melhor solução conhecida na literatura [145] para a instância *nfl20*. A instância *nfl16* foi a única em que AUDImTTP não atingiu a melhor solução conhecida.

Todavia, o custo da solução encontrada por AUDImTTP para essa instância ficou a 0,30% do valor S^* .

O maior tempo de execução foi 10,2 horas, consumido pela execução da instância *nfl24* enquanto o menor tempo foi igual a 3,01 horas, gastos na execução da instância *nfl18*.

7.3

Experimentos Computacionais com AUDI-AGMD

Os resultados computacionais obtidos pelo algoritmo paralelo AUDI-AGMD foram comparados com os alcançados pela heurística híbrida seqüencial, vista na Seção 2.4.2.

Os tempos computacionais apresentados nas próximas seções foram medidos em segundos e representam a média sobre cinco execuções, com diferentes sementes usadas pelos geradores de números aleatórios.

7.3.1

Instâncias de Teste

Os experimentos foram realizados sobre três grupos de instâncias de teste: grupos 1, 2 e 3. As instâncias de cada grupo possuem características específicas.

O Grupo 1 é formado por 43 instâncias usadas em [125], das quais 15 são grafos esparsos que contêm de 20 a 60 vértices, e 28 são grafos completos que contêm de 10 a 25 vértices. O diâmetro das instâncias desse grupo varia de 4 a 10. A solução ótima é conhecida para todas as instâncias desse grupo, e os custos das soluções ótimas apresentados nesta tese foram publicados a partir de [136].

O Grupo 2 contém 12 instâncias usadas em [80, 127]. Das 12 instâncias, seis possuem custos gerados aleatoriamente e as outras seis foram geradas no plano euclidiano. As instâncias desse grupo possuem valores de diâmetro iguais a 5, 7 e 9. Esse grupo contém somente instâncias esparsas, com diâmetro ímpar e várias arestas com custos iguais. Essa última característica aumenta a dificuldade de resolvê-las, pois como possuem muitas arestas com custos iguais, uma decisão por uma aresta dentre diversas de mesmo custo pode levar a soluções finais completamente diferentes. A solução ótima também é conhecida para todas as instâncias desse grupo, e os custos dessas soluções ótimas apresentados nesta tese foram extraídos a partir de [136].

O Grupo 3 é composto de 30 instâncias de grafos completos, provenientes da OR-Library [26] e usadas em [86, 95, 123, 136]. A biblioteca contém 15 instâncias para cada tamanho de grafo. Nesta tese, assim como em [86, 87, 136], são utilizadas apenas as cinco primeiras instâncias de cada grupo, contendo

50, 70, 100, 250, 500 e 1000 vértices. Esse é o grupo composto pelas instâncias mais difíceis. Sua dificuldade decorre do tamanho das instâncias e dos valores atribuídos aos custos, pois são valores próximos e fornecidos com oito casas decimais, exigindo mais tempo de processamento. A solução ótima não é conhecida para alguma instância desse grupo, e os custos apresentados nesta tese das melhores soluções conhecidas foram publicados a partir de [87, 136].

Todos os custos das soluções apresentados nesta seção são valores arredondados ou truncados na terceira casa decimal, seguindo o padrão adotado pelos demais autores [85, 86, 87, 136].

7.3.2

Resultados Computacionais

Os experimentos computacionais realizados com a implementação paralela AUDI-AGMD são apresentados e discutidos nesta seção. Os testes objetivam investigar a qualidade das soluções geradas e a robustez desse algoritmo diante de um ambiente *grid* real.

A probabilidade Q de reiniciar-se a execução de um processo provedor de solução a partir da fase de busca local ILS foi fixada em 10%. O tamanho das listas usadas em cada experimento depende do número de máquinas envolvidas em cada execução. Em todos os testes realizados nesta seção a implementação AUDI-AGMD adotou o mesmo tamanho para as listas de cooperação e de intensificação.

Primeiro Experimento - Comparação com a Heurística Sequencial

O objetivo dos experimentos realizados nesta seção é analisar a qualidade das soluções encontradas com a implementação AUDI-AGMD em relação à implementação sequencial. Para isso, o critério de parada utilizado por ambos algoritmos foi o tempo de execução. Esse tempo foi igual ao mesmo usado em [136], qual seja o tempo médio para executar 500 iterações da heurística sequencial para cada instância. Esses resultados foram obtidos em um sub-conjunto formado por dez máquinas pertencentes ao sítio S2, conforme ilustrado na Figura 7.2. A versão sequencial foi executada em uma máquina desse sítio. As listas de soluções de elite da implementação AUDI-AGMD foram definidas com tamanho igual a oito.

Os resultados desses experimentos são apresentados nas Tabelas 7.11, 7.12 e 7.13. Cada linha dessas tabelas corresponde a uma instância. As três primeiras colunas informam a quantidade $|V|$ de vértices, a quantidade $|E|$ de arestas e o valor $|D|$ do diâmetro da instância. Em seguida, os valores O^* das soluções ótimas são dados para as instâncias dos Grupos 1

e 2. Para as instâncias do Grupo 3 é fornecido o valor S^* da melhor solução conhecida antes deste trabalho. Para a versão seqüencial e para o algoritmo AUDI-AGMD, apresentam-se o custo da melhor solução encontrada sobre cinco execuções e o tempo médio em segundos dessas cinco execuções. Para as instâncias do Grupo 3, na Tabela 7.13 são apresentadas também as diferenças percentuais entre os custos das soluções encontradas pela versão seqüencial e pelo algoritmo AUDI-AGMD, em relação aos custos das melhores soluções conhecidas na literatura.

Como pode ser observado, para as instâncias do Grupo 1 o algoritmo paralelo conseguiu atingir todos os valores ótimos, superando a versão seqüencial que não foi capaz de encontrar a solução ótima para a instância de 60 vértices e 150 arestas desse grupo. O custo da solução encontrada pela versão seqüencial para essa instância está destacado em negrito na Tabela 7.11. Esse custo ficou a 1,6% do custo O^* da solução ótima alcançada pela implementação paralela AUDI-AGMD. A implementação paralela encontrou a solução ótima dessa instância em 35,96 segundos.

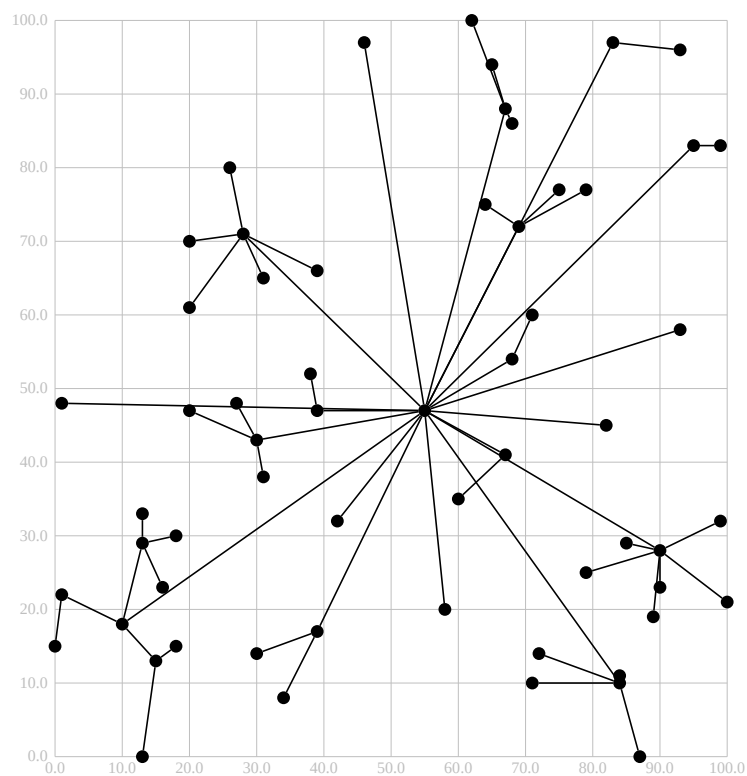
Em relação ao tempo de execução, a implementação paralela apresentou dois comportamentos diferentes. Para as instâncias menores, cujo tempo de processamento da versão seqüencial foi inferior a um segundo, a implementação paralela teve um desempenho pior. Isso ocorre porque o tempo necessário para que a implementação AUDI-AGMD ative todos os processos gerenciadores do SGA metaEasyGrid, crie as listas de cooperação e de intensificação, e inicialize os processos provedores em todas as máquinas envolvidas na execução é maior do que o tempo necessário para que a versão seqüencial finalize sua execução.

Por outro lado, para as instâncias maiores, a implementação paralela apresentou tempos de execução menores. Por exemplo, para as instâncias de 100 vértices a implementação AUDI-AGMD foi capaz de encontrar a solução ótima em média em 32,64% do tempo usado pela versão seqüencial.

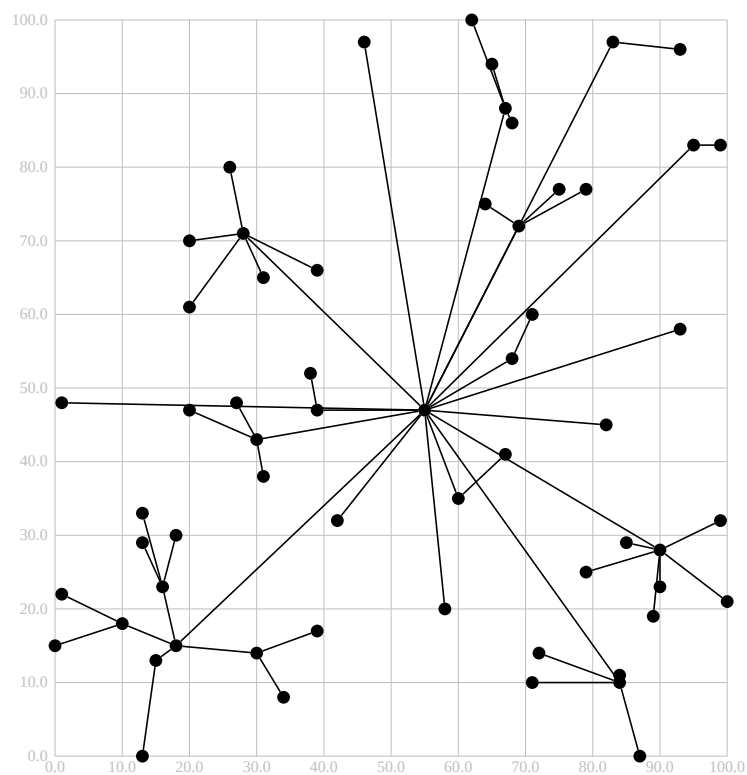
A fim de visualizar a diferença entre as soluções alcançadas pelos dois algoritmos, as Figuras 7.7 e 7.8 ilustram essas soluções para a instância de 60 vértices e 150 arestas. Essas figuras foram geradas a partir de um visualizador de grafos desenvolvido por Santos em [136]. As soluções apresentadas nas Figuras 7.7(a) e 7.7(b) estão apresentadas no plano, utilizando as coordenadas associadas a cada vértice do grafo. As soluções representadas nas Figuras 7.8(a) e 7.8(b) são as mesmas, mas desenhadas por níveis de profundidade a partir do vértice central (caso par) e das extremidades da aresta central (caso ímpar).

Tabela 7.11: Qualidade das soluções alcançadas pela versão sequencial e pelo algoritmo AUDI-AGMD com tempo de execução pré-determinado para as instâncias do Grupo 1.

$ V $	$ E $	D	O^*	Sequencial		AUDI-AGMD	
				Melhor	Tempo (s)	Melhor	Tempo (s)
10	45	4	252	252	0,10	252	2,20
		5	230	230	0,10	230	2,16
		6	221	221	0,10	221	2,30
		7	203	203	0,10	203	2,93
		8	232	232	0,10	232	2,87
		9	254	254	0,10	254	1,93
		10	254	254	0,10	254	1,78
15	105	4	346	346	0,40	346	2,36
		5	331	331	0,40	331	2,07
		6	314	314	0,40	314	2,41
		7	303	303	0,40	303	4,04
		8	240	240	0,40	240	5,40
		9	290	290	0,40	290	5,91
		10	286	286	0,40	286	2,39
20	190	4	349	349	0,50	349	2,24
		5	414	414	0,50	414	2,33
		6	298	298	0,50	298	2,42
		7	333	333	0,50	333	3,71
		8	331	331	0,50	331	4,01
		9	327	327	0,50	327	4,33
		10	324	324	0,50	324	2,32
25	300	4	500	500	1,00	500	2,31
		5	429	429	1,00	429	2,32
		6	378	378	1,00	378	1,89
		7	408	408	1,00	408	3,39
		8	369	369	1,00	369	5,65
		9	336	336	1,00	336	5,36
		10	379	379	1,00	379	2,03
20	50	4	442	442	5,00	442	2,14
		5	381	381	5,00	381	2,33
		6	329	329	5,00	329	2,17
		7	362	362	5,00	362	3,26
		8	366	366	5,00	366	3,01
		9	362	362	5,00	362	2,97
		10	359	359	5,00	359	2,07
40	100	4	755	755	10,00	755	3,38
		5	729	729	10,00	729	3,39
		6	599	599	10,00	599	3,41
		7	667	667	10,00	667	3,57
		8	580	580	10,00	580	3,62
		9	552	552	10,00	552	3,38
		10	574	574	10,00	574	2,10
60	150	5	968	983	50,00	968	35,96

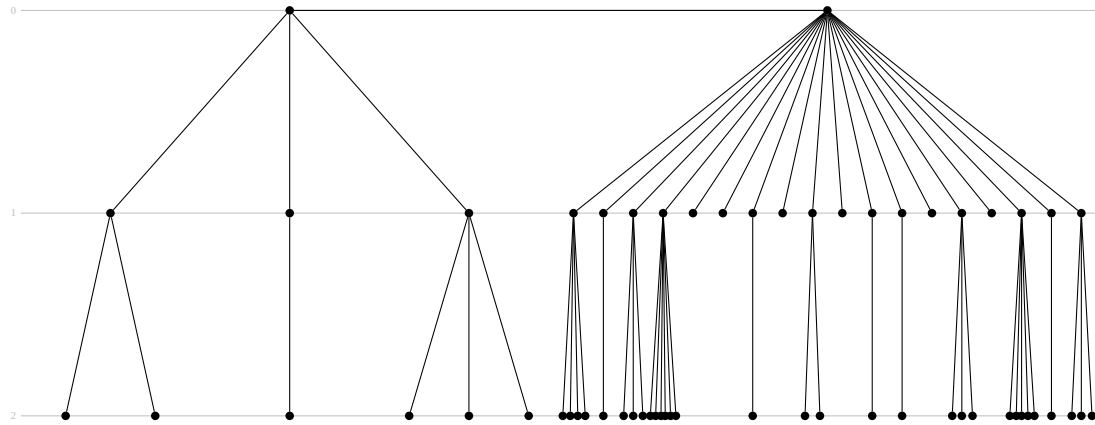


7.7(a): Solução obtida pela versão sequencial, com custo 983.

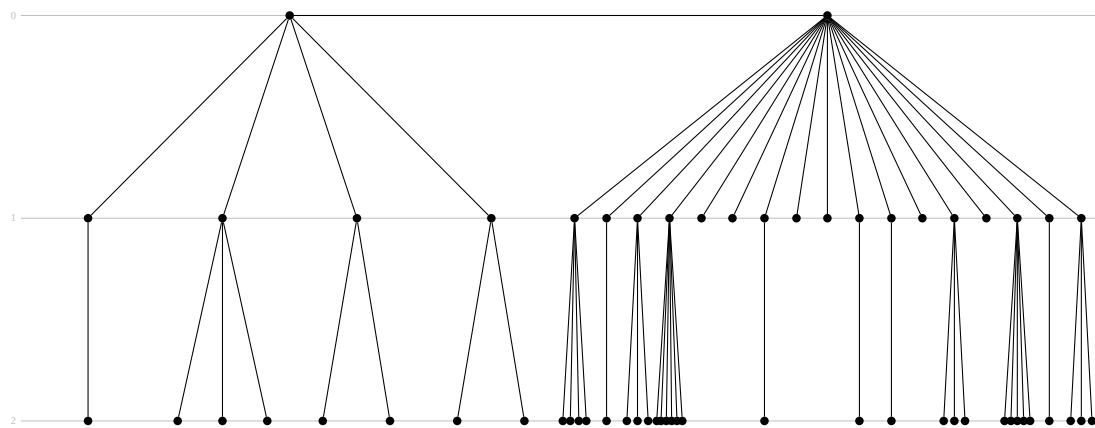


7.7(b): Solução obtida pela implementação paralela, com custo ótimo 968.

Figura 7.7: Soluções encontra para a instância do Grupo 1 com 60 vértices e 150 arestas.



7.8(a): Solução obtida pela versão sequencial, com custo 983.



7.8(b): Solução obtida pela implementação paralela, com custo ótimo 968.

Figura 7.8: Estrutura por níveis das soluções encontradas para a instância do Grupo 1 com 60 vértices e 150 arestas.

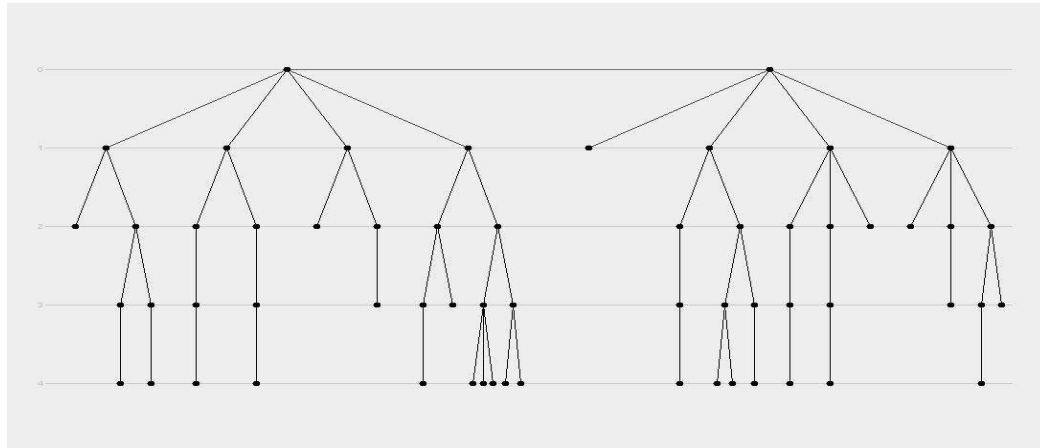
Na Tabela 7.12 é possível notar que o algoritmo seqüencial não atingiu a solução ótima para quatro das 12 instâncias pertencentes ao Grupo 2 (resultados apresentados em negrito). Para essas instâncias, o custo das melhores soluções alcançadas ficou no mínimo a 0,4% e no máximo a 1,3% do valor ótimo. A implementação paralela só não conseguiu encontrar as soluções ótimas para duas instâncias. Para essas, os custos ficaram no mínimo a 0,4% e no máximo a 0,5% do valor ótimo.

Quanto ao tempo, percebe-se que a implementação AUDI-AGMD tem um desempenho muito melhor que a versão seqüencial. Apesar de ter sido dado o mesmo tempo de execução que o algoritmo seqüencial, todas as soluções foram encontradas em um tempo significativamente menor. Por exemplo, no caso da quinta instância com 60 vértices e 600 arestas o algoritmo seqüencial executou por 50 segundos e não alcançou a melhor solução. O algoritmo paralelo encontrou a solução ótima em apenas 4,6 segundos.

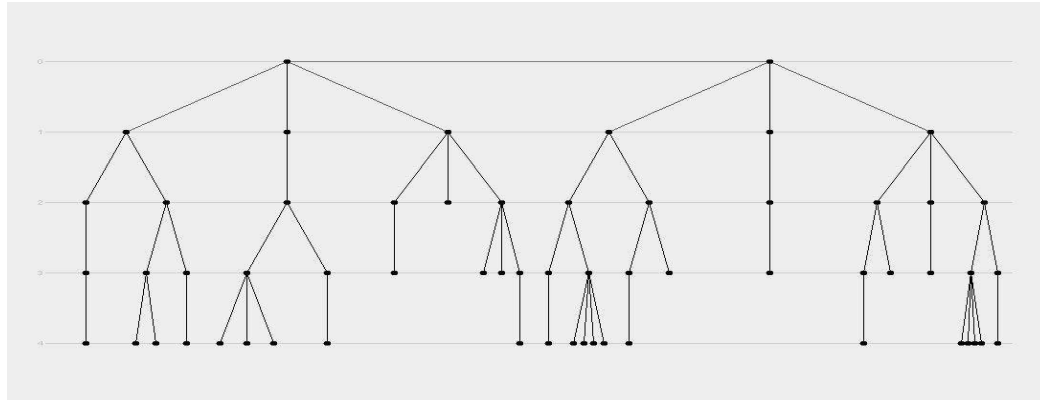
Para ilustrar a diferença de qualidade entre as soluções alcançadas pela versão seqüencial e pelo algoritmo paralelo, foi escolhida a instância do Grupo 2 com 60 vértices, 600 arestas e diâmetro 9, que é a nona instância da Tabela 7.12. Essa instância foi escolhida por ser uma das instâncias em que a solução ótima foi alcançada pela implementação paralela. A Figura 7.9(a) ilustra a solução encontrada pela heurística seqüencial, com custo igual a 741, e a Figura 7.9(b) a solução ótima alcançada pela implementação paralela, com custo 738.

Tabela 7.12: Qualidade das soluções alcançadas pela versão seqüencial e pelo algoritmo AUDI-AGMD com tempo de execução pré-determinado para as instâncias do Grupo 2.

V	E	D	O*	Seqüencial		AUDI-AGMD	
				Melhor	Tempo (s)	Melhor	Tempo (s)
40	400	5	612	612	10,00	612	3,58
		7	527	527	10,00	527	4,02
		9	495	495	10,00	495	4,39
		5	253	253	10,00	253	3,88
		7	171	171	10,00	171	6,04
		9	154	154	10,00	154	5,67
60	600	5	965	965	50,00	965	4,80
		7	789	796	50,00	793	17,22
		9	738	741	50,00	738	18,07
		5	256	257	50,00	257	5,13
		7	150	152	50,00	150	4,60
		9	124	124	50,00	124	8,94



7.9(a): Solução obtida pela versão sequencial, com custo 741.



7.9(b): Solução obtida pela implementação paralela, com custo ótimo 738.

Figura 7.9: Estrutura por níveis das soluções encontradas para a instância do Grupo 2 com 60 vértices, 600 arestas e diâmetro 9.

Na Tabela 7.13, além das informações mostradas nas Tabelas 7.11 e 7.12, são apresentadas também as diferenças percentuais entre os custos das soluções encontradas pela versão seqüencial e pelo algoritmo AUDI-AGMD, em relação aos custos das melhores soluções conhecidas na literatura.

Para a grande maioria das instâncias do Grupo 3, os resultados mostrados na Tabela 7.13 revelam que a implementação AUDI-AGMD foi superior à versão seqüencial, sendo capaz de encontrar soluções que o algoritmo seqüencial não encontrou.

A implementação AUDI-AGMD obteve soluções melhores que a implementação seqüencial para todas as instâncias desse grupo com 100 ou mais vértices.

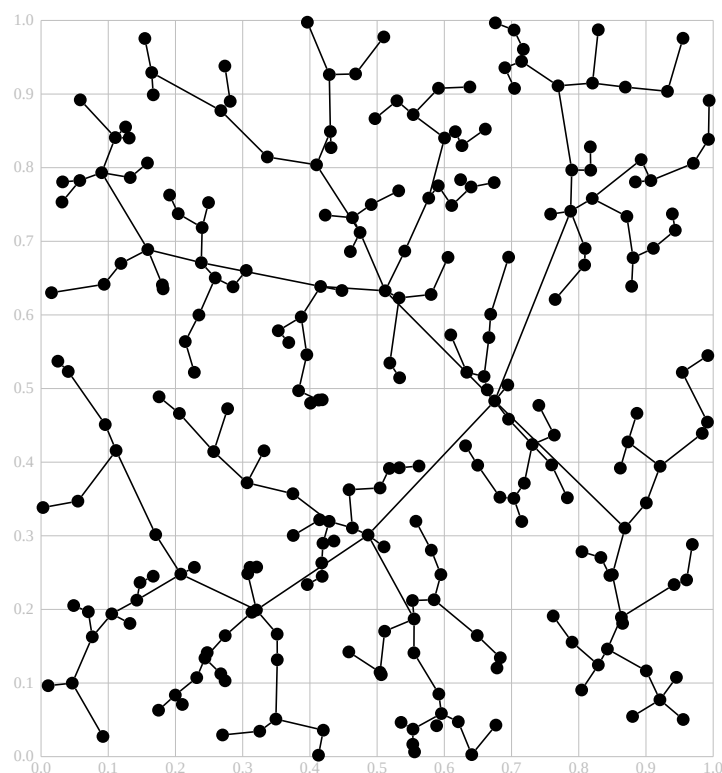
Mesmo tendo a implementação paralela executada com tempo pré-determinado, ela conseguiu encontrar uma solução superior à melhor solução conhecida na literatura para a quarta instância com 70 vértices e 2415 arestas, destacada em negrito na Tabela 7.13.

A superioridade da implementação AUDI-AGMD em relação à versão seqüencial torna-se mais clara para as instâncias maiores e mais difíceis. Por exemplo, a diferença percentual média entre os custos das soluções alcançadas para as instâncias de 500 vértices pelo algoritmo AUDI-AGMD e os custos das melhores soluções conhecidas foi de 1,40%, enquanto que a diferença percentual das soluções alcançadas pela versão seqüencial foi de 2,61%. Para as instâncias com 1.000 vértices as diferenças são 2,64% para os custos das soluções alcançadas por AUDI-AGMD e 3,53% para os custos das soluções da versão seqüencial.

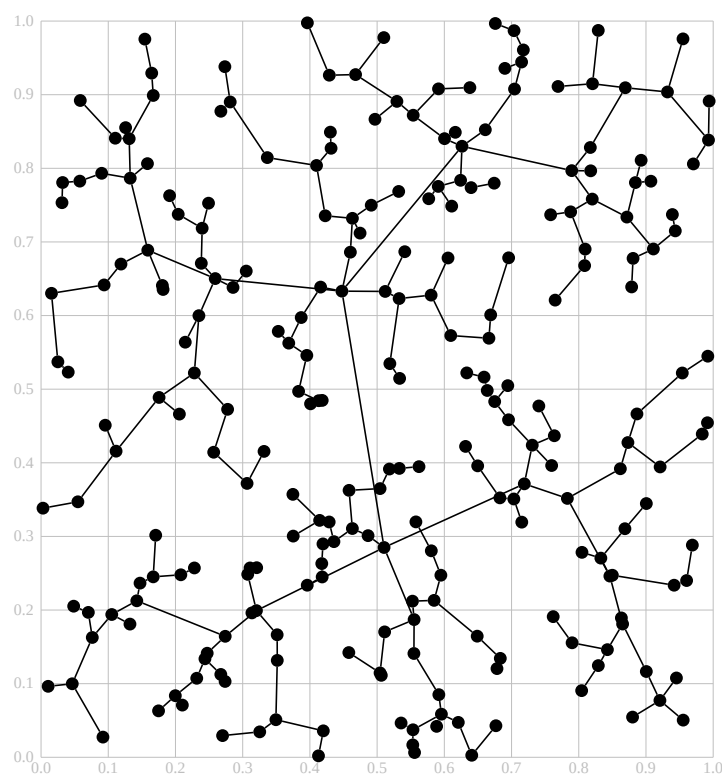
As Figuras 7.10 e 7.11 ilustram as duas visões das soluções alcançadas por cada implementação para a terceira instância com 250 vértices. Nesse caso, a solução alcançada pela versão seqüencial teve custo igual a 12,132 e a solução atingida pela implementação paralela teve custo 12,116, ficando a 0,93% do melhor valor conhecido na literatura.

Tabela 7.13: Qualidade das soluções alcançadas pela versão sequencial e pelo algoritmo AUDI-AGMD com tempo de execução pré-determinado para as instâncias do Grupo 3.

$ V $	$ E $	D	S^*	Sequencial			AUDI-AGMD		
				Melhor	dif. (%)	Tempo (s)	Melhor	dif. (%)	Tempo (s)
50	1225	5	7,599	7,599	0,00	10	7,599	0,00	2,10
			7,609	7,609	0,00	10	7,609	0,00	3,10
			7,240	7,240	0,00	10	7,240	0,00	1,72
			6,585	6,585	0,00	10	6,585	0,00	1,50
			7,248	7,248	0,00	10	7,248	0,00	1,31
70	2415	7	7,228	7,228	0,00	50	7,228	0,00	21,93
			7,080	7,080	0,00	50	7,080	0,00	16,27
			6,983	6,983	0,00	50	6,983	0,00	23,13
			7,499	7,499	0,00	50	7,486	-0,17	32,31
			7,245	7,245	0,00	50	7,245	0,00	18,04
100	4950	10	7,759	7,834	0,97	200	7,810	0,66	164,03
			7,849	7,943	1,20	200	7,881	0,41	173,08
			7,904	7,976	0,91	200	7,926	0,20	126,89
			7,977	8,044	0,84	200	8,032	0,69	157,84
			8,164	8,206	0,51	200	8,180	0,20	131,02
250	31125	15	12,231	12,451	1,80	1.500	12,412	1,48	973,34
			12,016	12,307	2,42	1.500	12,266	2,08	847,37
			12,004	12,132	1,07	1.500	12,116	0,93	1.036,99
			12,462	12,700	1,91	1.500	12,606	1,16	1.102,66
			12,233	12,444	1,72	1.500	12,438	1,68	1.087,94
500	124750	20	16,778	17,212	2,59	3.000	16,899	0,72	2.004,01
			16,626	17,063	2,63	3.000	16,714	0,53	2.131,29
			16,792	17,119	1,95	3.000	16,953	0,96	2.200,05
			16,796	17,250	2,70	3.000	17,15	2,11	1.987,32
			16,421	16,941	3,17	3.000	16,86	2,67	1.437,81
1000	499500	25	23,434	24,660	5,23	4.500	24,509	4,59	2.733,40
			23,464	24,460	4,24	4.500	24,286	3,50	3.021,23
			23,635	24,317	2,89	4.500	24,134	2,11	2.457,89
			23,787	24,609	3,46	4.500	24,270	2,03	3.000,43
			23,837	24,268	1,81	4.500	24,07	0,98	3.423,11

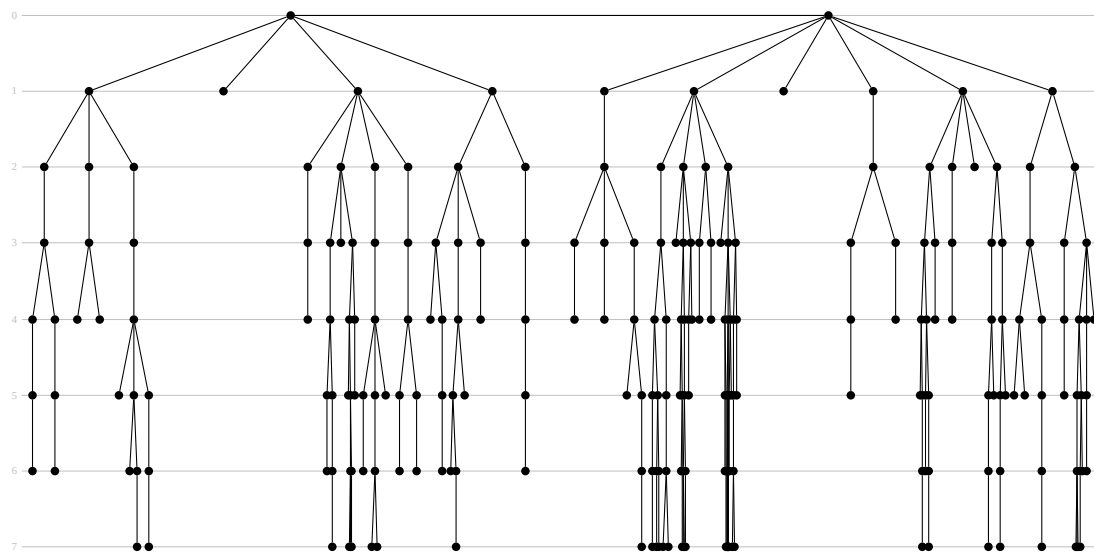


7.10(a): Solução obtida pela versão sequencial, com custo 12,132.

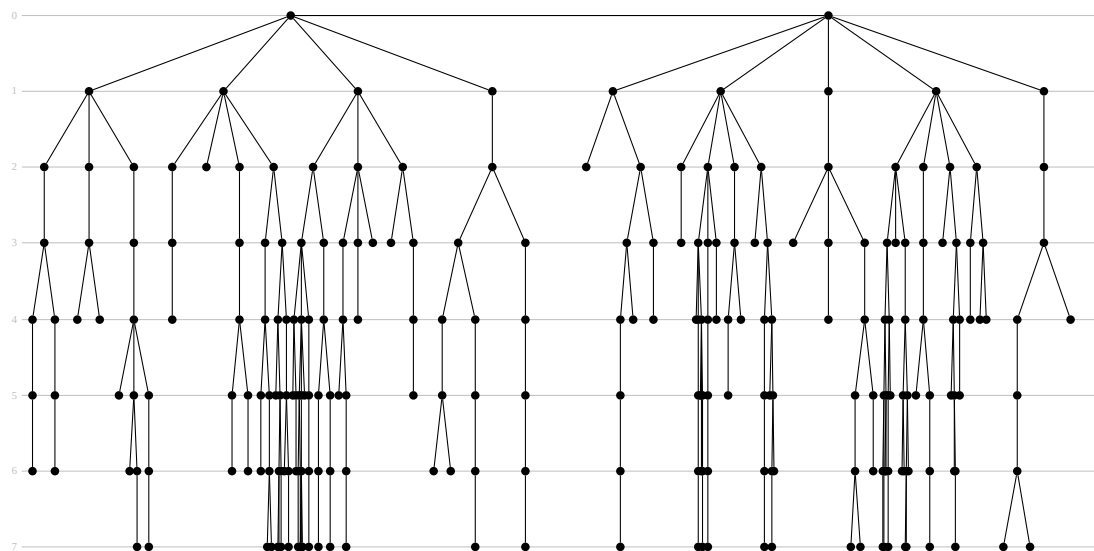


7.10(b): Solução obtida pela implementação paralela, com custo 12,116.

Figura 7.10: Soluções encontradas para a terceira instância do Grupo 3 com 250 vértices e 31.125 arestas.



7.11(a): Solução obtida pela versão seqüencial, com custo 12,132.



7.11(b): Solução obtida pela implementação paralela, com custo ótimo 12,116.

Figura 7.11: Estrutura por níveis das soluções encontradas para a terceira instância do Grupo 3 com 250 vértices e 31.125 arestas.

Segundo Experimento - Qualidade das Soluções do AUDI-AGMD

Os experimentos mostrados nesta seção objetivam analisar a qualidade das soluções que a implementação AUDI-AGMD é capaz de alcançar, independentemente de limitação de tempo. As instâncias usadas neste experimento foram as duas instâncias do Grupo 2 que não tiveram suas soluções ótimas atingidas no primeiro experimento e todas as instâncias do Grupo 3.

O critério de parada usado para as instâncias do Grupo 2 foi a obtenção de uma solução de custo igual ao valor ótimo conhecido. Para as instâncias do Grupo 3, adotaram-se os melhores valores conhecidos como critério de parada. Contudo, em ambos os casos, a execução era interrompida quando 100 estabilizações consecutivas dos processos gerenciadores das listas de intensificação ocorriam, mesmo que o valor alvo não tivesse sido atingido.

Os resultados desses experimentos são apresentados nas Tabelas 7.14 e 7.15. Cada linha dessas tabelas corresponde a uma instância. As três primeiras colunas informam a quantidade $|V|$ de vértices, a quantidade $|E|$ de arestas e o valor $|D|$ do diâmetro da instância. Em seguida, são apresentados os valores O^* das soluções ótimas para as instâncias dos Grupos 2. Para as instâncias do Grupo 3 é fornecido o valor S^* da melhor solução conhecida antes deste trabalho. Apresentam-se o custo da melhor solução encontrada sobre três execuções e o tempo médio em segundos dessas três execuções. Para as instâncias do Grupo 3, na Tabela 7.15 são apresentadas também as diferenças percentuais entre os custos das soluções encontradas pela implementação AUDI-AGMD, em relação aos custos das melhores soluções conhecidas na literatura.

Tabela 7.14: Qualidade das soluções alcançadas com a execução do algoritmo paralelo AUDI-AGMD para duas instâncias do Grupo 2, com dez processadores.

$ V $	$ E $	D	O^*	AUDI-AGMD	
				Melhor	Tempo (s)
60	600	7	789	789	65.556,34
		5	256	256	2.318,10

Com esse novo experimento, a implementação AUDI-AGMD foi capaz de encontrar as soluções ótimas para as duas instâncias que faltavam do Grupo 2. Dentre todas as instâncias desse grupo, a implementação paralela não havia conseguido atingir os valores ótimos para duas instâncias nos testes realizados com limitação de tempo, apresentados na Tabela 7.12. Para a instância com

60 vértices e diâmetro sete foram necessárias 18,21 horas de execução em dez máquinas para que o ótimo pudesse ser atingido.

Como o tempo necessário para resolver essa instância foi elevado, surgiu a idéia de executar a versão sequencial por longos períodos de tempo para averiguar se ela seria capaz e atingir as mesmas soluções que a implementação AUDI-AGMD. Esse experimento será relatado na próxima seção.

Os resultados dos experimentos com longos períodos de execução para as instâncias do Grupo 3 são apresentados na Tabela 7.15. Constata-se que, para as 30 instâncias desse grupo, a implementação AUDI-AGMD foi capaz de melhorar seis das melhores soluções conhecidas (destacadas em negrito) e atingir os melhores resultados em oito instâncias.

Para as demais instâncias, os custos ficaram no mínimo a 0,06% e no máximo a 1,14% da melhor solução conhecida na literatura. Em relação ao tempo de processamento, a execução mais longa demorou 6,65 horas para a terceira instância com 100 vértices e 4.950 arestas.

Tabela 7.15: Qualidade das soluções alcançadas com a execução do algoritmo paralelo AUDI-AGMD para as instâncias do Grupo 3, com dez processadores.

$ V $	$ E $	D	S^*	AUDI-AGMD		
				Melhor	Diferença (%)	Tempo (s)
50	1225	5	7,599	7,599	0,00	2,11
			7,609	7,609	0,00	3,11
			7,240	7,240	0,00	1,73
			6,585	6,585	0,00	1,50
			7,248	7,248	0,00	1,31
70	2415	7	7,228	7,228	0,00	21,93
			7,080	7,080	0,00	16,27
			6,983	6,981	-0,03	23,14
			7,499	7,486	-0,17	305,67
			7,245	7,238	-0,10	68,89
100	4950	10	7,759	7,757	-0,03	139,11
			7,849	7,849	0,00	23.965,71
			7,904	7,930	0,20	504,045
			7,977	7,973	-0,05	8.704,78
			8,164	8,176	0,15	336,92
250	31125	15	12,231	12,283	0,43	11.356,87
			12,016	12,123	0,89	8.978,35
			12,004	11,999	-0,04	6.006,41
			12,462	12,472	0,08	11.373,14
			12,233	12,272	0,32	9.261,90
500	124750	20	16,778	16,899	0,72	3.004,49
			16,626	16,810	1,11	4.30082
			16,792	16,907	0,68	11.572,04
			16,796	16,987	1,14	7.322,45
			16,421	16,555	0,82	12.431,17
1000	499500	25	23,434	23,488	0,23	6.103,87
			23,464	23,575	0,47	6.117,43
			23,635	23,663	0,12	5.782,18
			23,787	23,802	0,06	4.271,55
			23,837	23,887	0,21	5.003,86

Terceiro Experimento - Qualidade das Soluções do Seqüencial

O objetivo deste experimento é verificar a robustez do algoritmo seqüencial em relação à qualidade das soluções encontradas. Para isso, foram dados períodos longos de execução para que a versão seqüencial pudesse ter a chance de alcançar os mesmos resultados apresentados pela heurística paralela AUDI-AGMD nas Tabelas 7.14 e 7.15.

O critério de parada utilizado neste experimento foi o tempo de execução. O tempo de execução do algoritmo seqüencial foi fixado em $2T$ ou $10T$, sendo T o tempo usado pela versão AUDI-AGMD no experimento apresentado nas Tabelas 7.14 e 7.15. O tempo máximo de execução da versão seqüencial foi fixado em $2T$ para as instâncias em que o tempo de execução T da versão paralela foi maior do que 1.000 segundos, e $10T$ para as demais instâncias.

Nessa tabela são apresentados o grupo de cada instância, a quantidade $|V|$ de vértices, a quantidade $|E|$ de arestas, o valor $|D|$ do diâmetro e a identificação de cada instância do Grupo 3. Em seguida, a coluna nomeada por AUDI-AGMD repete as soluções encontradas pelo algoritmo paralelo no experimento anterior. Para finalizar, as duas últimas colunas apresentam os resultados alcançados com este experimento pela versão seqüencial e o tempo em segundos dado para cada instância executar.

Como pode ser observado na Tabela 7.16, a implementação seqüencial, mesmo tendo executado pelo menos o dobro do tempo usado pela implementação AUDI-AGMD, não conseguiu atingir os valores gerados por essa implementação para nenhuma das 27 instâncias.

O tempo de execução usado pelo algoritmo seqüencial contribui apenas para que o mesmo fosse capaz de melhorar três das suas próprias soluções anteriores, apresentadas da Tabela 7.12 e 7.13. Essas três instâncias estão destacadas em negrito na Tabela 7.16.

Percebe-se, com esse experimento, que a implementação paralela AUDI-AGMD é mais eficiente que a versão seqüencial, pois foi capaz de encontrar soluções que a versão seqüencial não encontrou, mesmo após executar por mais tempo. O algoritmo AUDI-AGMD, por exemplo, melhorou todos os resultados para as instâncias do Grupo 3 em relação à versão seqüencial. Além disso, para a instância com 60 vértices, 600 arestas e diâmetro 7 (primeira linha da Tabela 7.16) a implementação AUDI-AGMD encontrou o valor ótimo em 18,21 horas, enquanto que a versão seqüencial mesmo com 33 horas de processamento não o encontrou.

Tabela 7.16: Qualidade das soluções alcançadas pelo algoritmo seqüencial após executar por dez vezes o tempo usado pela implementação AUDI-AGMD para as instâncias com tempo de execução inferior a 1.000 segundos no experimento da Seção 7.3.2, e pelo dobro do tempo usado pela implementação AUDI-AGMD para as demais instâncias.

Grupo	$ V $	$ E $	id	AUDI-AGMD	Seqüencial	Tempo (s)
2	60	600	-	789	796	120.000,00
			-	738	740	2.300,00
			-	256	257	5.000,00
			-	150	152	400,00
3	70	2415	3	6,981	6,983	230,00
			4	7,486	7,499	3.000,00
			5	7,238	7,245	690,00
	100	4950	1	7,757	7,835	1.400,00
			2	7,849	7,943	48.000,00
			3	7,930	7,961	5.000,00
			4	7,973	8,048	17.500,00
			5	8,176	8,204	3.400,00
	250	31125	1	12,283	12,446	22.700,00
			2	12,123	12,307	18.000,00
			3	11,999	12,132	12.000,00
			4	12,472	12,700	22.800,00
			5	12,272	12,444	18.600,00
	500	124750	1	16,899	17,212	6.000,00
			2	16,810	17,063	8.600,00
			3	16,907	17,119	23.000,00
			4	16,987	17,250	14.600,00
			5	16,555	16,941	24.800,00
	1000	499500	1	23,488	24,660	12.200,00
			2	23,575	24,460	12.200,00
			3	23,663	24,317	11.500,00
			4	23,802	24,607	8.600,00
			5	23,887	24,268	10.000,00

Quarto Experimento - Escalabilidade

O objetivo deste experimento é analisar o comportamento da implementação paralela AUDI-AGMD em ambientes heterogêneos e não dedicados, com quantidades diferentes de recursos. O experimento investiga o desempenho de AUDI-AGMD em termos de escalabilidade.

Para esse teste foi escolhida uma instância de cada tipo do Grupo 3. Cada instância escolhida foi executada cinco vezes em 15, 30 e 60 máquinas. A quantidade de listas de intensificação para todas as configurações de máquinas foi fixada em três. Cada lista de intensificação do experimento foi criada em um sítio do grid Sinergia e a lista de cooperação foi sempre ativada no sítio S2.

Na Tabela 7.17 é mostrada, para cada instância selecionada do Grupo 3, a quantidade $|V|$ de vértices, a quantidade $|E|$ de arestas, o valor D do diâmetro e suas identificações. Na quarta coluna é apresentado o valor alvo usado como critério de parada para cada instância. As três últimas colunas apresentam o tempo médio em segundos sobre cinco execuções do AUDI-AGMD em 15, 30 e 60 máquinas.

Tabela 7.17: Tempos médios em segundos sobre cinco execuções da implementação AUDI-AGMD para algumas instâncias do Grupo 3 em 15, 30 e 60 máquinas pertencentes aos três sítios do grid Sinergia.

$ V $	$ E $	D	id	Alvo	AUDI-AGMD		
					15 CPUs	30 CPUs	60 CPUs
50	1225	5	1	7,599	1,96	3,72	10,65
70	2415	7	3	6,983	9,34	8,89	7,16
100	4950	10	3	7,981	49,23	35,11	24,08
250	31125	15	1	12,450	3.723,27	2.344,49	1.317,78
500	124750	20	2	17,063	2.627,14	2.496,45	2.006,08
1000	499500	25	4	24,609	3.837,79	3.401,71	2.916,25

Como pode ser visto na Tabela 7.17, a implementação paralela AUDI-AGMD é escalável. Os tempos de processamento diminuem com o aumento no número de máquinas disponíveis para execução. Isso significa que houver melhor aproveitamento da capacidade de processamento disponível através do aumento no número de máquinas. No entanto, para a instância com 50 vértices, 1.225 arestas e diâmetro 5 (primeira linha da Tabela 7.17) essa redução no tempo de processamento não ocorreu. Essa instância gastou apenas 1,31 segundos para atingir o valor alvo 7,248 em dez máquinas, conforme mostrado na Tabela 7.11. Conseqüentemente, o tempo necessário para criar os processos em um número maior de máquinas e gerenciar todos esses processos é maior

do que o tempo de execução nas dez máquinas. Isso significa que, nesse caso o aumento no poder computacional não se faz mais necessário, pois a execução da instância atingiu um ponto de saturação em relação a quantidade de recursos.

Para as instâncias mais difíceis, o aproveitamento de um maior número de recursos disponíveis no ambiente *grid* torna-se mais evidente.

Esse melhor aproveitamento é possível porque não há um único elemento central responsável por controlar todos os processos da aplicação, assim quanto maior poder de processamento, mais rápido será a execução da aplicação. Além disso, mesmo que um número maior de máquinas seja usado e, consequentemente, um número maior de processos seja criado, não há sobrecarga no ambiente porque o SGA metaEasyGrid cria os processos de acordo com a carga do ambiente e demanda da aplicação.

Outro fator que influencia em escalabilidade da estratégia proposta é a existência das listas de intensificação, pois sua distribuição no ambiente evita que muitas mensagens sejam transmitidas em toda a rede, concentrando a maior parte da troca de informações em redes locais.

Para ratificar os resultados do experimento acima relatado, foi escolhida uma instância do Grupo 3 para ser executada 100 vezes com as mesmas configurações do grid Sinergia, ou seja, em 15, 30 e 60 máquinas. A Figura 7.12 ilustra o gráfico do tipo *tttplot*, que mostra a distribuição empírica de probabilidades do tempo necessário para o algoritmo AUDI-AGMD encontrar para a quinta instância de 100 vértices do Grupo 3 uma solução com custo menor ou igual ao valor alvo 8,211.

A escalabilidade da implementação AUDI-AGMD é comprovada pelas curvas do gráfico. A curva mais à esquerda refere-se à execução em 60 máquinas, seguida pelas execuções em 30 e 15 máquinas. Além disso, nota-se que existem diferenças de tempo consideráveis entre a execução mais rápida e mais lenta para todas as três configurações.

Todavia, com a execução realizada em 60 máquinas essa diferença torna-se menor, demonstrando que essa execução é mais estável que as demais. Por exemplo, a implementação mais demorada em 60 máquinas levou pouco mais de 100 segundos, a implementação mais demorada em 30 máquinas levou quase 200 segundos e, a implementação mais demorada em 15 máquinas levou quase 400 segundos.

Quinto Experimento - Ponto de Saturação

Os experimentos apresentados nesta seção objetivaram avaliar o comportamento da implementação AUDI-AGMD em relação à variação do número de listas de intensificação usadas na execução da aplicação. Para realizar este ex-

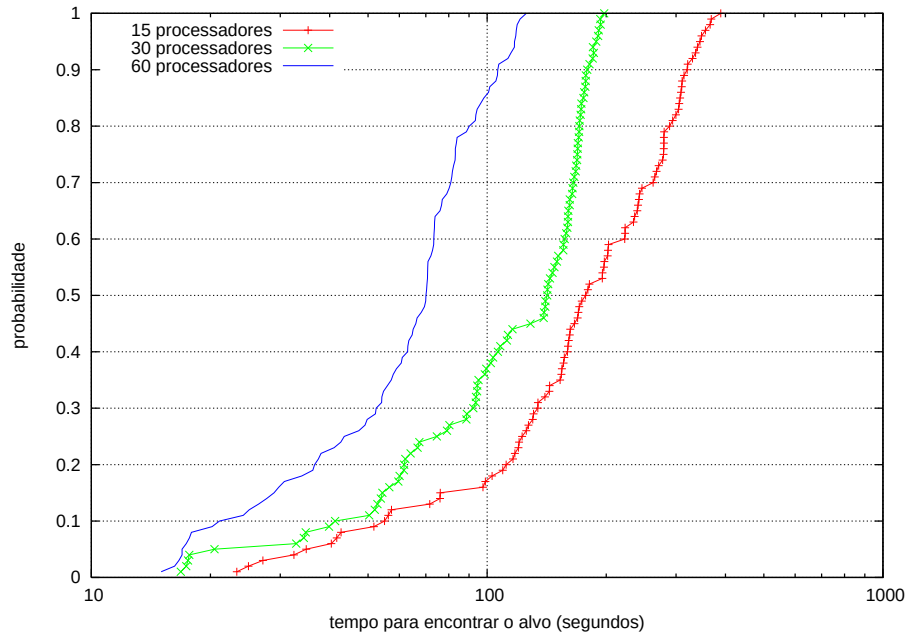


Figura 7.12: Distribuição de probabilidade empírica do tempo para atingir uma solução com custo menor ou igual ao valor alvo, considerando-se o valor 8,211 para a quinta instância de 100 vértices do Grupo 3. A implementação AUDI-AGMD foi executada 100 vezes em ambiente *grid* com 15, 30 e 60 máquinas.

perimento, foi escolhida uma instância de cada grupo. Essas instâncias foram executadas 100 vezes cada, utilizando-se sementes diferentes para o gerador de números aleatórios. As execuções foram programadas para encerrarem tão logo fosse encontrada uma solução com custo menor do que o valor alvo estabelecido.

Todas as execuções ocorreram em 60 máquinas do grid Sinergia, com três, cinco e sete listas de intensificação. Em cada configuração, todas as listas de intensificação tinham a mesma capacidade de armazenamento.

As Figuras 7.13, 7.14 e 7.15 ilustram gráficos do tipo *tttplot* para a instância de 60 vértices e 150 arestas do Grupo 1, para a instância com 60 vértices e 600 arestas do Grupo 2 e, para a quinta instância com 100 vértices do Grupo 3. Esses gráficos mostram a distribuição de probabilidade do tempo necessário para que a implementação paralela AUDI-AGMD encontre, para cada uma das instâncias testadas, soluções com custos menores ou iguais aos valores alvos 150, 738 e 7,787, respectivamente. Como pode ser observado, as três figuras mostram um comportamento semelhante, em que a execução em 60 máquinas com cinco listas de intensificação apresentou o melhor desempenho.

Na Figura 7.13 nota-se que existem diferenças de tempos consideráveis entre a execução mais rápida e mais lenta para as três configurações. Todavia, a execução com cinco listas foi a que obteve a menor variação, pois sua execução mais rápida demorou pouco mais de um segundo e a mais lenta

aproximadamente 24 segundos. Por outro lado, a execução com apenas três listas teve uma variação entre aproximadamente cinco e 60 segundos.

Com a Figura 7.14, percebe-se que a execução realizada com cinco listas teve quase 70% das suas execuções concluídas em menos de 100 segundos, enquanto que na configuração com sete listas essa porcentagem diminui para 60%, e com três listas essa porcentagem é pouco mais de 50%. Isso significa que a execução com cinco listas é capaz de convergir mais rapidamente para as boas soluções.

A Figura 7.15 ilustra a execução da instância pertencente ao grupo mais difícil, o Grupo 3. Essa maior dificuldade é notada pelo fato das três configurações terem tido tempos de execução que variaram entre oito segundos e até quase 600 segundos. Além disso, as curvas representantes das execuções com três e sete listas ilustram que apenas 20% dessas execuções encerraram com no máximo 100 segundos. Nas execuções com cinco listas esse percentual cresce para um pouco mais de 30%.

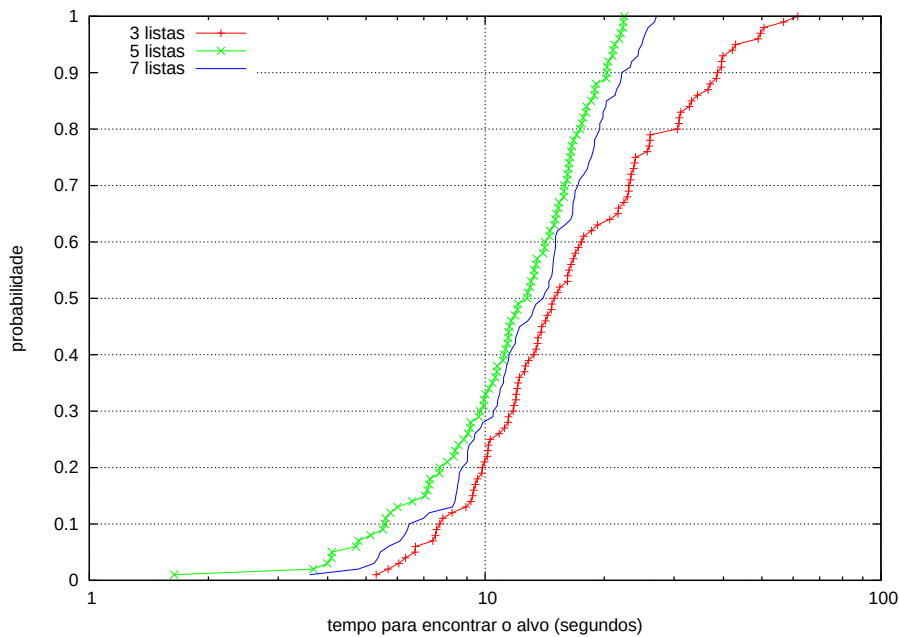


Figura 7.13: Distribuição de probabilidade empírica do tempo para atingir uma solução com custo menor ou igual ao valor alvo, considerando-se o alvo 968 para a instância de 60 vértices e 150 arestas do Grupo 1. A implementação AUDI-AGMD foi executada 100 vezes em 60 máquinas pertencentes ao grid Sinergia com três, cinco e sete listas de intensificação.

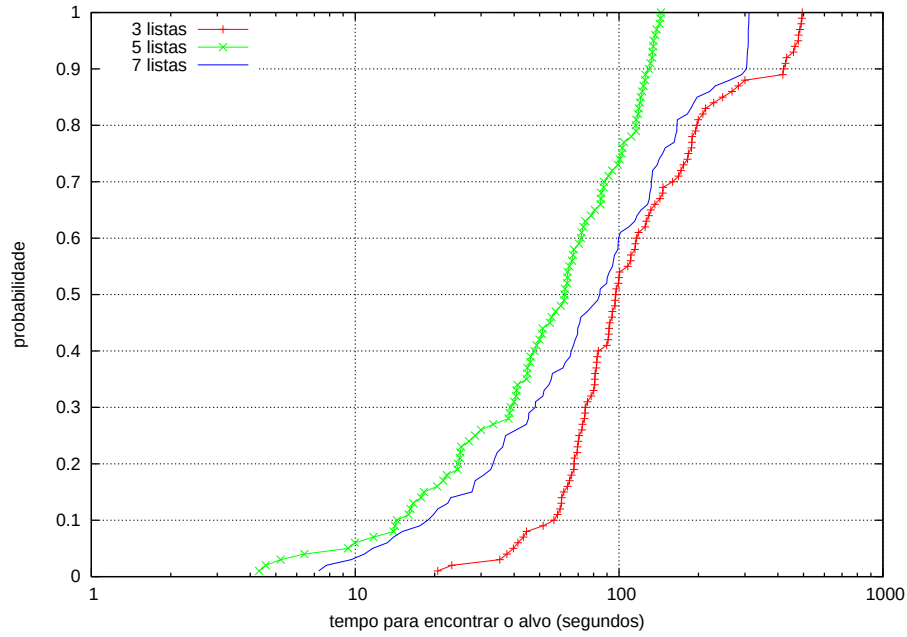


Figura 7.14: Distribuição de probabilidade empírica do tempo para atingir uma solução com custo menor ou igual ao valor alvo, considerando-se o alvo 738 para a instância de 60 vértices, 600 arestas e diâmetro 9 do Grupo 2. A implementação AUDI-AGMD foi executada 100 vezes em 60 máquinas pertencentes ao grid Sinergia com três, cinco e sete listas de intensificação.

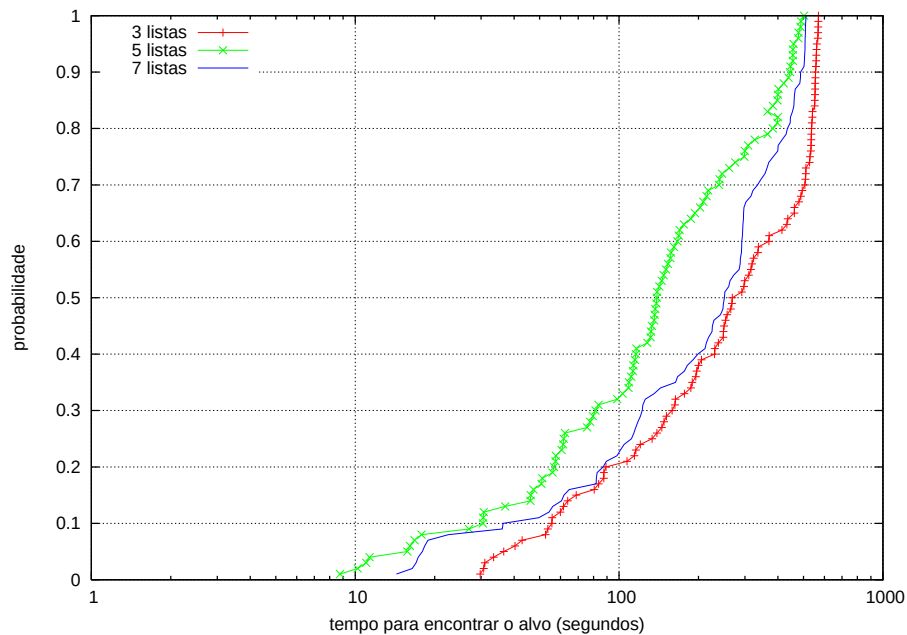


Figura 7.15: Distribuição de probabilidade empírica do tempo para atingir uma solução com custo menor ou igual ao valor alvo, considerando-se o alvo 7,787 para a primeira instância de 100 vértices do Grupo 3. A implementação AUDI-AGMD foi executada 100 vezes em 60 máquinas pertencentes ao grid Sinergia com três, cinco e sete listas de intensificação.

O fato da execução com cinco listas de intensificação ter tido um melhor desempenho que as demais configurações é devido ao equilíbrio alcançado entre a diversidade promovida pelas listas e o número de processos provedores associados a cada uma dessas listas.

Na execução com três listas, há muitos processos enviando soluções para as mesmas listas, podendo haver uma sobrecarga dos processos gerenciadores dessas listas de intensificação. Outro fator relacionado à grande quantidade de processo provedor vinculado a mesma lista é o preenchimento dessas com soluções parecidas, comprometendo assim a diversidade.

No outro extremo, a execução com sete listas inibe uma melhor seleção das soluções geradas, pois há poucos provedores associados em cada lista. Além disso, com a execução em 60 máquinas e sete listas, mais de 10% dos recursos estão alocados para os gerenciadores. Dessa forma, dependendo do número de recursos usado e da configuração física dos mesmos, há um ponto de saturação na quantidade de listas de intensificação.

7.4

Considerações Finais

Neste capítulo foram realizados diversos experimentos com a estratégia de paralelização desenvolvida durante esta tese, aplicada a dois problemas diferentes. Os experimentos objetivaram validar as idéias propostas no Capítulo 6, que foram a estratégia de paralelização hierárquica distribuída e o sistema de gerenciamento metaEasyGrid.

Os resultados apresentados mostraram que o desenvolvimento das heurísticas paralelas com a estrutura de listas proposta é uma estratégia eficiente para paralelizar metaheurísticas em ambientes *grid*, pois garante a cooperação intensiva entre os processos provedores, sem provocar um congestionamento de mensagens na rede.

Os experimentos também revelaram que outra característica promissora da estratégia hierárquica distribuída é a operação de renovação, implantada entre as listas de soluções. Essa operação proporciona uma constante e natural renovação das soluções usadas, garantindo diversidade nas buscas realizadas.

A estratégia de paralelização hierárquica distribuída proposta nesta tese pode ser usada por qualquer heurística que se utilize de listas de soluções de elite.

Como já mencionado, os *grids* computacionais apresentam características distintas dos ambientes de alto desempenho tradicionais e, por isso, são mais complexos para que as aplicações paralelas utilizem adequadamente os seus recursos. Todavia, com o *middleware* SGA metaEasyGrid a gerência do

ambiente fica transparente e a aplicação torna-se autônoma, capaz de se auto-adaptar às variações desse ambiente.

O SGA metaEasyGrid garante escalabilidade das aplicações autônomas porque trabalha com um modelo de execução no qual o número de processos MPI é proporcional ao grau de paralelismo da aplicação. O paralelismo não se restringe ao número de recursos do ambiente, como acontece no modelo tradicional usado pelo MPI. Além disso, a criação é proporcional a carga do ambiente e demanda da aplicação.

A integração da estratégia hierárquica distribuída com o *middleware* SGA metaEasyGrid transforma metaheurísticas paralelas em aplicações robustas, capazes de aproveitar a capacidade de processamento fornecida pelos ambientes *grids*. Essa integração demonstrou ser promissora na aceleração da convergência para boas soluções.

Como consequência de parte do trabalho apresentado neste capítulo foram publicados os artigos [14, 15].