

BRUNO GARCIA DA SILVA



Otimização de Leiautes Realistas

Usando Sistemas Multiagente

Dissertação apresentada como requisito parcial à obtenção do grau de mestre em Ciências. Curso de pós-graduação em Informática Aplicada – PPGIA, Centro de Ciências Exatas e de Tecnologia – CCET, Pontifícia Universidade Católica do Paraná – PUCPR.

Área de Concentração:

Sistemas Inteligentes & Sistemas Produtivos

Orientador: Prof. Dr. Marcos Augusto Hochuli Shmeil

Curitiba

2002



Silva, Bruno Garcia da.
Otimização Prática Realista de Leiautes Usando Multiagente

Dissertação – Pontifícia Universidade Católica do Paraná. Programa de Pós-Graduação em Informática Aplicada.

1. Sistemas Multiagente 2. Inteligência Artificial 3. Leiautes 4. Sistemas Produtivos

I. Pontifícia Universidade Católica do Paraná. Centro de Ciências Exatas e de Tecnologia Programa de Pós-Graduação em Informática Aplicada II-t



ATA DA SESSÃO PÚBLICA DE DEFESA DE DISSERTAÇÃO DE MESTRADO
DO PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA APLICADA
DA PONTIFÍCIA UNIVERSIDADE CATÓLICA DO PARANÁ

DEFESA DE DISSERTAÇÃO Nº 066

Aos 31 dias do mês de outubro de 2002 realizou-se a sessão pública de defesa da dissertação “**Otimização de Leiaute Realista Usando Sistemas Multiagentes**”, apresentada por **Bruno Garcia da Silva**, como requisito parcial para a obtenção do título de **Mestre em Informática Aplicada**, perante uma Banca Examinadora composta pelos seguintes membros:

Prof. Dr. Marcos A. H. Shmeil
PUCPR (Presidente)

assinatura

parecer (aprov/reprov.)

Prof. Dr. Milton Pires Ramos
TECPAR

Prof. Dr. Hilton J. S. Azevedo
CEFET-PR

Prof. Dr. Edson E. Scalabrin
PUCPR

Conforme as normas regimentais do PPGIA e da PUCPR, o trabalho apresentado foi considerado aprovado (aprovado/reprovado), segundo avaliação da maioria dos membros desta Banca Examinadora. Este resultado está condicionado ao cumprimento integral das solicitações da Banca Examinadora, conforme registrado no Livro de Defesas do programa.

Prof. Dr. Carlos Maziero
Diretor do PPGIA PUCPR

04/12/02
Data e assinatura, após homologação da defesa pelo colegiado.



À minha família e amigos, em especial ao meu
pai que sempre acreditou em mim, mesmo
quando eu mesmo não acreditava.

Agradecimentos

Ao meu pai Roberto que sempre acreditou em mim e por isto deu seu mais completo e irrestrito apoio para todos os meus sonhos e loucuras. Ao professor e orientador Doutor Marcos Augusto Hochuli Shmeil pelas suas contribuições e principalmente pela sua profunda paciência comigo. Ao Professor Dr. Edson Emilio Scalabrin pelas inúmeras sugestões e correções durante o desenvolvimento deste trabalho. Ao meu amigo Valdair e Prof. Dr. Marco Antonio Candido que tanto contribuíram com sugestões e apoio para tornar este trabalho possível. A todas as pessoas que me apoiaram e me ajudaram na realização deste trabalho: Sonia Kazumi Iramina, Edson Pacheco, Orlando, Milton Pires Ramos e Hilton José Silva de Azevedo.

Sumário

Agradecimentos.....	vi
Sumário.....	vii
Lista de Figuras.....	ix
Lista de tabelas.....	xi
Lista de abreviações.....	xii
Lista de abreviações.....	xii
Resumo.....	xiii
Abstract.....	xiv
1. Introdução.....	1
1.1 Desafio.....	4
1.2 Motivação.....	4
1.3 Área de pesquisa.....	6
1.4 Objetivo geral.....	6
1.5 Objetivos específicos e metodologia.....	6
1.6 Restrições e variáveis de um leiaute.....	8
1.7 Contribuições.....	9
1.8 Organização da dissertação.....	9
2. Estado da arte.....	11
2.1 Inteligência Artificial.....	11
2.1.1 Inteligência Artificial Distribuída.....	12
2.1.2 Sistemas Multiagente.....	12
2.1.3 O que são agentes?.....	13
2.1.4 Taxionomia de um agente.....	14
2.2 Sistemas produtivos e arranjo de leiaute.....	15
2.2.1 Problema de atribuição quadrática.....	17
2.2.2 Algoritmos construtivos.....	18
2.2.3 Busca de uma solução ótima.....	22
2.2.4 Algoritmo de busca tabu.....	23
2.3 Considerações gerais.....	24
3. Sistema AVOLI.....	26
3.1 Definições de aspecto e área do Leiaute.....	27
3.2 Definições de aspecto e área dos departamentos.....	29
3.2.1 Departamentos móveis e suas informações.....	29
3.2.2 Departamentos fixos e suas informações.....	31
3.3 Áreas ocupadas.....	32
3.4 Restrições de localização.....	33
3.5 Análises de fluxos entre os departamentos.....	34
3.6 Considerações sobre as distâncias entre os departamentos.....	35
3.7 Considerações sobre a função objetiva.....	36

3.8	Algoritmo básico do AVOLI	39
3.8.1	Métodos de criação da lista de departamentos	39
3.8.2	Método de construção de leiaute do sistema AVOLI.....	43
3.8.3	Pré-processamento.....	45
3.9	Posições e informações sobre departamentos	46
4.	Sistema multiagente	48
4.1	Comunicação entre agentes	48
4.2	Heurísticas usadas pelos agentes.....	49
4.3	Ambiente	56
5.	Testes, Resultados e Comparações	60
5.1	Usando QAP	61
5.2	Usando Departamentos de Tamanhos Variados	64
5.3	Usando Departamentos de Tamanhos Variados com Restrições de adjacência e orientação	67
6.	Conclusões.....	74
6.1	Resultados obtidos	74
6.2	Os agentes.....	76
6.3	O mundo	77
6.4	A comunicação entre os agentes e o mundo	78
6.5	Trabalhos Futuros.....	79
7.	Referências	82
	ANEXO A - Usando o sistema AVOLI.....	85
	ANEXO B - Algoritmos e Fluxogramas.....	92
	ANEXO C – Vocabulário Utilizado pelos Agentes	100
	ANEXO D - Os Melhores Leiautes Obtidos Através de SMA	102

Lista de Figuras

Figura 2-1 Árvore de divisão e Leiaute gerado.....	21
Figura 3-1 Sugestões do sistema AVOLI para a área do leiaute.....	28
Figura 3-2 Definições iniciais da dimensão do leiaute, número de departamentos, área ocupadas, métodos, etc.....	28
Figura 3-3 Informações sobre os departamentos móveis.....	31
Figura 3-4 Informações sobre os departamentos fixos.....	31
Figura 3-5 Informações sobre as áreas ocupadas.....	33
Figura 3-6 Informações sobre restrições de adjacências.....	34
Figura 3-7 Informações sobre os fluxos entre os departamentos.....	35
Figura 3-8 Visualização dos resultados.....	39
Figura 3-9 Configurações gerais do sistema.....	40
Figura 3-10 Lista de construção.....	41
Figura 3-11 Departamentos organizados por fluxos para a construção da lista automaticamente.....	42
Figura 3-12 Visualização do Pré-Processamento.....	46
Figura 3-13 Informações diversas dos departamentos.....	47
Figura 3-14 Disposição dos departamentos no interior do leiaute.....	47
Figura 4-1 Representação de um agente em Blocos.....	50
Figura 4-2 Comportamentos, fluxos e memória de troca.....	52
Figura 4-3 Exemplo de lista sugerida do agente 1 para o mundo.....	53
Figura 4-4 Informações que o agente tem da vizinhança atual do mundo.....	53
Figura 4-5 Informações que os agentes tem sobre os fluxos e relacionamentos de adjacências.....	54
Figura 4-6 Exemplo de comunicação de um agente.....	54
Figura 4-7 Acompanhamento do Agente.....	55
Figura 4-8 Log de comunicação do agente.....	55
Figura 4-9 Representação em blocos do agente mundo.....	57
Figura 4-10 Histórico completo de todas as mudanças feitas no leiaute pelos agentes.....	57
Figura 4-11 Contribuições dos agentes e suas listas sugeridas.....	59
Figura 5-1 Tempo em segundos que levou cada teste da primeira bateria de testes.....	62
Figura 5-2 Contribuição dos agentes no teste 1 com 5,6,7,8 e 12 departamentos.....	63
Figura 5-3 Contribuição dos agentes no teste 1 com 15, 20, 30 e 40 departamentos.....	63
Figura 5-4 Melhorias efetivas na utilidade final do leiaute em 1000 ciclos.....	64
Figura 5-5 Tempo em segundos que levou cada teste da segunda bateria de testes.....	65
Figura 5-6 Contribuição dos agentes no teste 2 com 5, 6, 7 e 8 departamentos.....	65
Figura 5-7 Contribuição dos agentes no teste 2 com 12, 15, 20 e 30 departamentos.....	66
Figura 5-8 Melhorias efetivas na utilidade final do leiaute no teste 2 com 12, 15, 20 e 30 departamentos.....	66

Figura 5-9 Melhorias efetivas na utilidade final do leiaute no teste 2 com 5,6,7 e 8 departamentos.....	67
Figura 5-10 Tempo em segundos que levou cada teste da terceira bateria de testes.....	68
Figura 5-11 Contribuições dos agentes no teste 3 com 8 e 12 departamentos.....	70
Figura 5-12 Melhorias efetivas na utilidade final do leiaute no teste 3 com 8 e 12 departamentos.....	70
Figura 5-13 Contribuições dos agentes no teste 3 com 15, 20 e 40 departamentos.....	71
Figura 5-14 Melhorias efetivas na utilidade final do leiaute no teste 3 com 15, 20 e 40 departamentos.....	71
Figura 5-15 Contribuições dos agentes no teste 3 com 30 departamentos.....	72
Figura 5-16 Melhorias efetivas na utilidade final do leiaute no teste 3 com 30 departamentos.....	72
Figura 5-17 Contribuições dos agentes no teste 3 com 55 departamentos.....	73
Figura 5-18 Melhorias efetivas na utilidade final do leiaute no teste 3 com 30 departamentos.....	73
Figura 6-1 Demonstrativo do uso da CPU.....	75
Figura 6-2 Melhorias efetivas no leiaute.....	76
Figura 6-3 Variação da utilidade total durante 500 ciclos.....	77
Figura 6-4 Participação dos agentes na melhoria geral do leiaute em 500 ciclos.....	79
Figura 6-5 Depois de mais 1000 ciclos o departamento +3 nunca colaborou.....	80
Figura A.1 Tela da configuração do ambiente.....	86
Figura A.2 Informações sobre áreas ocupadas ou mortas.....	87
Figura A.3 Informações sobre os departamentos fixos.....	88
Figura A.4 Informações sobre os departamentos moveis.....	89
Figura A.5 Informações sobre os fluxos entre os departamentos.....	90
Figura A.6 Informações sobre relacionamento entre os departamentos.....	91
Figura B. 1 Fluxograma da solução inicial.....	93
Figura B. 2 Leiaute mostrado graficamente.....	96
Figura B. 3 –Fluxograma do algoritmo construtivo.....	99
Figura D.1 - Leiaute dos melhores resultados encontrados pelos agentes na primeira bateria de teste.....	102
Figura D.2 - Leiaute dos melhores resultados encontrados pelos agentes na segunda baterias de testes.....	105
Figura D. 3 Leiaute dos melhores resultados encontrados pelos agentes na terceira baterias de testes.....	108
Figura D. 4 Leiaute dos melhores resultados encontrados pelos agentes na terceira baterias de testes (cont.).....	109

Lista de tabelas

Tabela 2-1 Definição e variações de IA.....	11
Tabela 2-2 Algumas das dimensões para a classificação de agentes	14
Tabela 2-3 - Comparativo de restrições tratadas por vários métodos	25
Tabela 5-1 Testes comparativos – [Chi01], [Nug68] e [Mar02].....	62
Tabela 5-2 Departamentos de tamanhos variados.....	65
Tabela 5-3 Departamentos de tamanhos variados com restrições e áreas ocupadas.....	68
Tabela D.1 - Fluxos entre os departamentos da primeira bateria de testes.....	103
Tabela D.2 - Fluxos entre os departamentos da primeira bateria de testes (continuação).....	104
Tabela D.3 - Listas construtoras da primeira bateria de testes.....	104
Tabela D.4 - Fluxos entre os departamentos da segunda bateria de testes	106
Tabela D.5 - Listas construtoras da segunda bateria de testes	107
Tabela D.6 - Informações sobre departamentos móveis da segunda bateria de testes	107
Tabela D.7 - Listas construtoras da terceira bateria de testes	110
Tabela D.8 - Informações sobre departamentos fixos da terceira bateria de testes.....	110
Tabela D.9 – Informações sobre áreas ocupadas da terceira bateria de testes.....	110
Tabela D.10 – Informações sobre departamentos móveis da terceira bateria de testes.....	111
Tabela D.11 - Fluxos entre os departamentos da terceira bateria de testes	112
Tabela D.12 - Fluxos entre os departamentos da terceira bateria de testes (continuação)	113
Tabela D.13 - Fluxos entre os departamentos da terceira bateria de testes (continuação)	114

Lista de abreviações

ACR	Algoritmo de Construção Realista
AG	Agente
ALDEP	Automated Layout Design Program
AVOLI	Ambiente Visual para Otimização de Leiaute Industrial
CAD	Computer Aided Design
CORELAP	Computerized Relationship Layout Planning
CRAFT	Computerized Relative Allocation of Facilities Technique
Dep	Departamento
GA	Genetic Algorithm
MIP	Mixed Integer Program
PPGIA	Programa de Pós Graduação de Informática Aplicada
PUCPR	Pontificia Universidade Católica do Paraná
QAP	Quadratic Assignment Problem
SA	Simulated Annealing
SMA	Sistemas multiagente
TB	Tabu Search

Resumo

Este trabalho apresenta o problema do leiautes e às disposições de seus departamentos considerando seus fluxos produtivos. A problemática de otimização de leiautes se enquadra na categoria conhecida como problemas da classe NP-completo que são de difícil resolução em um tempo polinomial. Para tratar deste problema apresentamos um sistema completo computacional chamado de AVOLI¹ (Ambiente Visual para Otimização de Leiautes Industrial). Este sistema procura ser uma ferramenta útil e de fácil utilização que auxilie na construção de leiautes totalmente viáveis e realistas, pois trata de uma ampla gama de variáveis e restrições encontradas no mundo real. O sistema AVOLI trata a problemática do leiaute em duas etapas distintas: geração de um leiaute inicial totalmente viável e a otimização deste leiaute usando uma nova metodologia de sistema multiagente reativa não cooperativa.

A eficiência do sistema AVOLI na construção de leiautes realistas é validada através de testes e comparações realizadas com problemas reportados na literatura da engenharia da produção.

Palavras-Chave: sistema multiagente, otimização, layout, sistemas produtivos.

¹ Para obter uma cópia do sistema AVOLI escreva-me no e-mail: brunogarcia69@brturbo.com.br ou brunogarcia69@msn.com

Resumo

Este trabalho apresenta o problema do leiautes e às disposições de seus departamentos considerando seus fluxos produtivos. A problemática de otimização de leiautes se enquadra na categoria conhecida como problemas da classe NP-completo que são de difícil resolução em um tempo polinomial. Para tratar deste problema apresentamos um sistema completo computacional chamado de AVOLI¹ (Ambiente Visual para Otimização de Leiautes Industrial). Este sistema procura ser uma ferramenta útil e de fácil utilização que auxilie na construção de leiautes totalmente viáveis e realistas, pois trata de uma ampla gama de variáveis e restrições encontradas no mundo real. O sistema AVOLI trata a problemática do leiaute em duas etapas distintas: geração de um leiaute inicial totalmente viável e a otimização deste leiaute usando uma nova metodologia de sistema multiagente reativa não cooperativa.

A eficiência do sistema AVOLI na construção de leiautes realistas é validada através de testes e comparações realizadas com problemas reportados na literatura da engenharia da produção.

Palavras-Chave: sistema multiagente, otimização, layout, sistemas produtivos.

¹ Para obter uma cópia do sistema AVOLI escreva-me no e-mail: brunogarcia69@brturbo.com.br ou brunogarcia69@msn.com

Abstract

This work presents the problem of industrial layout and the disposition of departments considering its productive flows. The problem of layout improvement is including in a category called NP-completes class, which problems are of difficult resolution in a polynomial time. To treat this problem we presented a complete computational system called AVOLI ²(Visual Environment for the Improvement of Industrial Layout). This system tries to be a useful and easy tool to really aid the construction of viable and realistic layouts, because it is concerned to the wide range of variables and restrictions found in the real world. The AVOLI system treats the layout problem in two different stages: generation of an initial layout totally viable and the improvement of this layout using a new methodology based on multi-agent systems, with reactive and non-cooperative agents.

The efficiency of the AVOLI system in the construction of realistic layouts is validated through tests and comparisons accomplished on problems described in the technical literature of production engineering.

Keywords: multi-agent systems, layout improvements, productive systems.

² To obtain a copy of the AVOLI system writes me in the e-mail: brunogarcia69@brturbo.com.br or brunogarcia69@msn.com

Abstract

This work presents the problem of industrial layout and the disposition of departments considering its productive flows. The problem of layout improvement is including in a category called NP-completes class, which problems are of difficult resolution in a polynomial time. To treat this problem we presented a complete computational system called AVOLI ²(Visual Environment for the Improvement of Industrial Layout). This system tries to be a useful and easy tool to really aid the construction of viable and realistic layouts, because it is concerned to the wide range of variables and restrictions found in the real world. The AVOLI system treats the layout problem in two different stages: generation of an initial layout totally viable and the improvement of this layout using a new methodology based on multi-agent systems, with reactive and non-cooperative agents.

The efficiency of the AVOLI system in the construction of realistic layouts is validated through tests and comparisons accomplished on problems described in the technical literature of production engineering.

Keywords: multi-agent systems, layout improvements, productive systems.

² To obtain a copy of the AVOLI system writes me in the e-mail: brunogarcia69@brturbo.com.br or brunogarcia69@msn.com

1. Introdução

Este trabalho apresenta a concepção e implementação de um sistema multiagente para resolver o problema de geração e otimização do arranjado de um grande número de elementos ou objetos físicos (acima de 30) sobre um determinado espaço. Cada agente gerencia um objeto e tem um objetivo, que é encontrar a melhor posição possível, sobre o espaço ou ambiente em questão, para o objeto gerenciado respeitando-se um dado conjunto de restrições. A resolução do problema se dará no momento em que cada agente conseguirá satisfazer seu objetivo ou no final de um certo número de interações ou modificações no ambiente. As interações entre agentes são diretas quando elas se materializam por meio de trocas de mensagens do tipo oferta / contra-oferta ou indiretas quando elas se dão pela percepção das modificações no ambiente. Estas modificações ocorrem essencialmente quando dois agentes decidem permutar seus objetos de lugar. Os agentes tomam tal iniciativa como uma forma de cooperação.

O modelo agente se justifica quando o problema é complexo. Tal complexidade pode ser medida em função do grande volume de conhecimentos a manipular ou do elevado número de cálculos necessários para se obter cooperativamente a solução de um problema. A cooperação agente pode ser requerida para alcançar um tipo ou mais de objetivo(s), que são, por exemplo:

- acelerar a solução de um problema privilegiando o trabalho paralelo dos agentes (*um dos principais objetivos desta pesquisa*);
- obter várias soluções locais utilizando as capacidades dos outros agentes;
- melhorar a confiabilidade dos resultados utilizando o fato que os agentes podem verificar seus resultados;
- reduzir a possibilidade de duplicação de processamento atribuindo um subproblema a um número limitado de agentes, por exemplo, a um único; e

- reduzir o volume de comunicação trocando unicamente as informações necessárias.

O arranjo e distribuição de objetos em um determinado espaço serão chamados, doravante, simplesmente de leiaute. Este espaço poderá ser, por exemplo, uma superfície industrial, ou comercial. O termo leiaute, no dicionário Aurélio, possui três diferentes definições:

- esboço de anúncio, em que se apresentam ressaltados os seus diversos elementos (título, texto, ilustração, etc.);
- esboço, projeto, planejamento ou esquema de uma obra, apresentados graficamente; e
- distribuição física de elementos num determinado espaço.

Os elementos ou objetos aqui tratados podem ser departamentos, células, máquinas, grupo de pessoas ou outros componentes quaisquer que possuam dimensões conhecidas e precisam ser arranjadas num espaço conhecido do leiaute³. Estes elementos possuem peculiaridades próprias como: área, aspecto, orientação, relações de adjacências, etc., que são previamente determinadas de acordo com suas necessidades físicas e produtivas. Possuem também um conjunto de interações com os demais elementos do leiaute. Estas interações podem ocorrer na forma de recursos, materiais, informações ou qualquer outra coisa que afete a produção diretamente. Doravante usaremos a palavra departamento para generalizar todos e quaisquer elementos que serão dispostos no leiaute.

No projeto de um leiaute, o principal problema é determinar uma estrutura detalhada da distribuição física dos departamentos de modo a otimizar a produção diminuindo esforços. Esta diminuição pode ser obtida através da melhoria da movimentação e comunicação entre as pessoas envolvidas no processo produtivo, assim como a redução do tempo de transporte dos materiais otimizando seus fluxos. Outros fatores podem ser considerados, tais como: racionalização do espaço disponível, observância da segurança do trabalho e verificação das questões ergonômicas de um sistema produtivo.

O problema de leiaute é difícil de resolver e extremamente complexo devido à explosão combinatória levando em conta as inúmeras possibilidades de arranjos dos departamentos. O problema se torna ainda mais complexo se considerarmos o seguinte fato: os departamentos podem ter áreas e aspectos variáveis dentro de um limite mínimo e máximo conhecido. Desta forma, encontrar soluções ótimas para leiautes com inúmeros departamentos usando apenas métodos analíticos puramente matemáticos em um tempo computacional

³ O projeto de leiautes é um tema abordado em particular no campo da Engenharia de Produção.

aceitável é algo bastante difícil. Trata-se assim de um problema cuja solução é não-trivial e pertencente à classe dos problemas conhecidos como NP-Completo [Gar79].

O problema de leiaute tem sido amplamente pesquisado. Podem-se destacar trabalhos pioneiros nesta área como o de [Nug68] e trabalhos mais recentes como o de [Tam98] e [Mar02]. Pode-se encontrar nestes e em outros trabalhos novas técnicas e algoritmos que encontraram soluções aceitáveis, bem próximas do ótimo em um tempo computacional aceitável. Dentre estas técnicas e algoritmos pode-se citar como exemplo:

- Busca tabu [Glo89] e [Mar02];
- Algoritmo genético [Hol75]; e
- Recozimento simulado [Kir83].

Entretanto, são poucos os trabalhos que tratam de uma gama mais ampla de variáveis e restrições para a construção de leiautes verdadeiramente realistas. Neste trabalho será considerado não apenas os departamentos, mas também outras áreas que fazem parte do leiaute. Estas áreas, doravante chamadas de *áreas ocupadas*, podem ser, por exemplo, pilares de sustentação, escadas e elevadores, assim como outras áreas fixas de posição conhecida no interior do leiaute que podem afetar diretamente na disposição e produtividade dos departamentos. Considerar-se-á também a restrição de localização, como por exemplo, departamentos que necessariamente devam ocupar uma localização específica no interior do leiaute, assim como a restrição de adjacências dos departamentos. Estas adjacências são restrições que determinam quais departamentos devem necessariamente ficar colados ou separados. Estas e outras restrições serão explicadas detalhadamente mais adiante.

A abordagem utilizada para resolver o problema de leiaute é do tipo planejamento multiagente, onde os agentes são entidades simples e reativas. A solução do problema emerge da interação destes agentes. Estes agentes foram implementados em um ambiente de software, com interface gráfica para visualização / acompanhamento da evolução dos agentes. Este ambiente de software se chama AVOLI (Ambiente Visual para Otimização de Leiautes Industriais). E possui um conjunto de componentes semelhante aos sistemas multiagente situados definidos por J. Ferber [Fer95], que contém os seguintes elementos:

- um espaço ambiental E , que dispõe de uma topologia e uma noção de distância;
- um conjunto de objetos O . Estes objetos são situados, i.e. para todo objeto, é possível, num dado momento, associar uma posição em E . Estes objetos são passivos, i.e. eles podem ser percebidos, criados, destruídos e modificados pelos agentes.

- um subconjunto **A** de **O** que representa as entidades ativas do sistema, os agentes.
- um conjunto de relações / *acquaintances* **R** que unem os agentes entre si.
- um conjunto de operações **Op** que permitem aos agentes de **A** de perceber, produzir, consumir, transformar e manipular os objetos de **O**.

No contexto de trabalho, o ambiente será uma superfície qualquer com dimensões fixas, onde virtualmente haverá um conjunto de objetos físicos a serem arranjados e distribuídos para formar um certo leiaute. A manipulação destes objetos será efetuada por um conjunto de agentes através de um conjunto de operações. Estas operações serão articuladas em função das mensagens trocadas e das reações do ambiente na tentativa de modificá-lo. Os agentes e o ambiente AVOLI serão detalhados posteriormente.

1.1 Desafio

Dado um leiaute conhecido ou previamente construído através de um método qualquer, o desafio é aplicar técnicas de planejamento multiagente visando otimizá-lo. O processo de obtenção deste novo leiaute deve:

- realizar uma distribuição de departamentos de forma racional e produtiva;
- oferecer um ganho de produtividade final superior ao leiaute original;
- aceitar um grande número de departamentos (acima de 40);
- calcular a nova distribuição em um tempo aceitável computacionalmente;
- respeitar, dentre outras, as restrições dos departamentos tais como: área, razão de aspecto, adjacências, sem as violar.

1.2 Motivação

O esforço na determinação de bons leiautes apresenta dois aspectos motivadores, que são:

- aspecto econômico e
- aspecto científico.

Aspecto econômico

A redução de custos e a flexibilidade são itens essenciais para a sobrevivência de qualquer empresa com fins lucrativos ou não. A necessidade da flexibilidade se dá pela constante

reestruturação de departamentos que se faz necessária pela evolução tecnológica e necessidade do mercado consumidor. Isto significa dizer que alguns departamentos surgirão e outros desaparecerão em um tempo que pode ser acentuadamente curto, o que força o constante planejamento de um novo leiaute. No entanto, este leiaute deve sempre primar pela eficiência, que pode ser medida pela redução de custos observados em um bom posicionamento dos departamentos e um bom aproveitamento do espaço total. Este posicionamento dos departamentos no leiaute afetará diretamente os fluxos produtivos e os custos do produto final. Considerando que o tempo requerido para a geração de um novo e bom leiaute (sem deixar de levar em conta seus inúmeros arranjos possíveis) pode ser pequeno (de 1 hora a 1 dia), então o uso de ferramentas computacionais se tornam não só desejáveis como necessárias.

Aspecto científico

A otimização de um leiaute é um problema que pode gerar uma explosão combinatória e de difícil solução, i.e., trata-se de um problema do tipo NP-Completo [Gar79]. Tal complexidade pode ainda sofrer um acréscimo quando se procura construir e otimizar leiautes realistas. Eles podem incluir uma gama considerada de variáveis e restrições inflexíveis. Estas dificuldades têm motivado os pesquisadores da área em desenvolver novas técnicas e algoritmos para a otimização de leiautes. Neste sentido, propõe-se aqui a experimentação de técnicas da área de inteligência artificial, em particular, por meio de mecanismos de geração e otimização de planos. Estes últimos são obtidos a partir da interação de um conjunto de agentes simples.

Os principais motivos pelo qual escolheu-se o planejamento multiagente para a otimização de leiautes são:

- *paralelismo e distribuição*, onde os cálculos podem ser efetuados sobre diferentes agentes e estes estarem situados sobre diferentes máquinas;
- *sistema aberto*, onde pode-se inserir e retirar agentes do sistema perturbando minimamente o andamento da resolução de um problema;
- *múltiplas implementações*, onde pode-se chegar ao extremo de implementar cada competência de cada agente utilizando um algoritmo diferente; e
- *múltiplas competências*, onde pode-se ter agentes que resolvem diferentes partes de um problema.

1.3 Área de pesquisa

As pesquisas podem ser enquadradas basicamente em dois campos distintos, que são:

- planejamento multiagente; e
- sistema produtivo.

O sistema produtivo fornece o domínio de aplicação, definindo o problema, suas restrições, informações sobre leiautes, departamentos, fluxos entre departamentos, e bem como outras variáveis.

O paradigma multiagente fornece um modelo de coordenação distribuída entre entidades reativas autônomas independentes cuja interação é motivada por estímulos (trocas de mensagens, percepção do ambiente, modificação do ambiente). O resultado das interações deve ser um plano que corresponderá à solução do problema, i.e., a sugestão de um novo leiaute. Em outras palavras, a solução otimizada de um leiaute emergirá da interação destes agentes.

1.4 Objetivo geral

O objetivo geral é otimizar um leiaute envolvendo um grande número de departamentos (acima de 30) em tempo computacional aceitável (de 1 hora a 1 dia), e desenvolver um ambiente de simulação multiagente permitindo testar uma grande gama de algoritmos distribuídos e paralelos. A otimização de um leiaute significa reduzir os custos produtivos. Esta redução se dá por um melhor arranjo dos departamentos em uma superfície com dimensões definidas. Ela deverá também levar em conta todas as restrições impostas pelo problema e as mais reais possíveis.

1.5 Objetivos específicos e metodologia

Os objetivos específicos correspondem basicamente a um conjunto de atividades ligados à definição, implementação e teste de um conjunto de componentes do sistema AVOLI. Este último implementa uma arquitetura multiagente, onde os agentes são entidades reativas e especializadas na resolução de problemas de otimização de leiautes com um grande número de departamentos (acima de 30).

Para facilitar a compreensão e o cumprimento dos objetivos específicos, optou-se em executá-los em duas fases distintas, que são:

- criação do leiaute inicial; e
- otimização deste leiaute por meio da definição e implementação de um conjunto de agentes especializados.

Primeira fase

Esta fase compreende um conjunto de atividades ligado à criação do leiaute inicial, que são:

- levantar as informações e restrições essenciais para construir leiautes reais;
- estudar e implementar os algoritmos que realizarão tarefas básicas, tais como: geração e pré-processamento de leiautes, cálculo da função de avaliação da qualidade de um leiaute, detecção de vizinhanças;
- estudar e implementar métodos que permitam obter uma solução inicial, como por exemplo, a geração baseada em uma função aleatória;
- projetar e implementar uma interface gráfica para facilitar a entrada de valores para os parâmetros do sistema de otimização e permitir o acompanhamento das atividades dos agentes durante os cálculos. Incluem-se também formulários, planilhas, gráficos e desenho do leiaute;
- criar uma base de dados com diferentes leiautes para testar e validar o sistema. Os dados desta base serão retirados da literatura de Engenharia da Produção.

Segunda fase: otimização do leiaute criado

Esta segunda fase compreende um conjunto de atividades ligada à solução proposta, baseado no modelo agente, para a otimização de leiautes complexos, que são:

- montar uma arquitetura genérica de agente que contemple, em particular, módulos para tratar crenças, comportamentos múltiplos e outras informações;
- dotar cada agente de uma interface gráfica, privilegiando a exibição de elementos de informação que facilitem o acompanhamento visual do grau de sua satisfação e contribuição do agente na solução do problema;
- projetar e implementar um ambiente para a criação e execução dos agentes.
- definir e codificar os agentes do sistema AVOLI. Será implementado um agente quadro-negro que funcionará como o ambiente do sistema multiagente. Será também

definido um conjunto de critérios de avaliação que o ambiente usará para avaliar as contribuições dos agentes;

- dotar o ambiente de execução de procedimentos que facilitem a inclusão, acompanhamento e remoção de agentes do sistema de forma dinâmica;
- padronizar a comunicação entre os membros do sistema, incluindo linguagem de comunicação e vocabulário / ontologia; e
- verificar o desempenho e performance do método de resolução de problemas por coordenação — planejamento multiagente — confrontando-o a outros métodos da literatura de Engenharia da Produção.

É importante salientar que a primeira fase deste sistema foi desenvolvida em conjunto e fez parte também da dissertação de mestrado de V. C. Martins [Mar02]. Este último desenvolveu o método de otimização de leiautes em pseudotrocas tabu. Os trabalhos se diferenciam essencialmente na segunda fase. O método busca tabu com pseudotrocas para otimizar leiautes será usado posteriormente para inúmeras comparações de desempenho e está disponível no sistema AVOLI.

1.6 Restrições e variáveis de um leiaute

As restrições e variáveis que devem ser levadas em conta na otimização de um leiaute são as seguintes:

- *restrições de disponibilidade de área*, que são: (a) um departamento deverá ter uma área igual ou ligeiramente superior (devido a arredondamentos) aquela que foi inicialmente definida em seu mínimo de área necessária; e (b) o seu aspecto geométrico inicialmente definido nas suas razões de aspecto mínima e máxima, i.e., as proporções de um lado em relação ao outro lado do departamento deverão ser sempre preservadas, sem violações.
- *restrições de localização do departamento*, que são: (a) um departamento nunca poderá se sobrepor a um outro; (b) um departamento nunca poderá ser posicionado em um local dentro do leiaute onde haja um desperdício de sua área útil total por sobreposição de áreas ocupadas superior ao previamente definida; (c) um departamento nunca poderá ser posicionado fora do leiaute; (d) um departamento poderá ter sua posição definida a priori. Este departamento é conhecido como

departamento fixo; (e) um departamento poderá possuir restrições de adjacências predefinidas. I.e. o departamento deverá permanecer próximo ou longe de outros departamentos. Esta restrição será considerada mais relevante do que o fluxo entre os departamentos em questão.

O leiaute terá forma retangular e suas dimensões são conhecidas. Os departamentos que comporão o leiaute também têm formas retangulares e as proporções entre os seus lados são previamente especificadas dentro de um limite. Poderá haver ainda departamentos cujas suas posições são fixas e determinadas. As áreas ocupadas também são de aspectos retangulares e suas posições no interior do leiaute são predefinidas.

1.7 Contribuições

Uma contribuição é à otimização de leiautes realistas com um grande número de departamentos. Deve-se destacar também uso do modelo multiagente como arquitetura da solução proposta. O ambiente computacional construído para testar uma gama importante de algoritmos e fazê-los coabitar em um mesmo sistema.

Do ponto de vista industrial, a contribuição se apresenta por meio de uma ferramenta computacional que gera, em tempo aceitável, leiautes viáveis e aplicáveis ao mundo real. Espera-se, assim, ajudar as empresas a reduzirem seus custos de produção e aumentar seu nível de competitividade e produtividade, aspectos fundamentais para sua sobrevivência no mercado.

1.8 Organização da dissertação

Este capítulo apresentou algumas questões ligadas ao conceito de leiaute, assim como algumas definições básicas dos elementos que compõem um leiaute. Apresentou-se também a motivação, os objetivos e contribuições esperadas.

O capítulo II apresenta o estado da arte. Examinam-se algumas noções ligadas aos agentes e métodos de otimização de leiautes, bem como algumas considerações sobre as limitações destes métodos no tratamento de restrições do mundo real.

O capítulo III descreve em detalhe o escopo do problema e todos os dados necessários para o funcionamento do sistema AVOLI. Estes dados incluem: áreas, razões de aspectos, departamentos fixos e moveis, fluxos, restrições de adjacências, distâncias entre

departamentos, vizinhanças, algoritmo construtivo, pré-processamento, formulas básicas de avaliação de otimização do leiaute.

O capítulo IV apresenta o paradigma multiagente. Aqui são detalhados os agentes e o ambiente onde estes agentes se situam. Este detalhamento inclui: (a) os interesses dos agentes, suas percepções, comportamentos e modos de agir no ambiente; (b) as informações que os agentes precisam para manifestar suas intenções e reagir de acordo com suas crenças; (c) o comportamento do ambiente e seus mecanismos de controle e avaliação das propostas dos agentes, e, como decidir em prol da coletividade uma mudança no leiaute; (d) os mecanismos utilizados para operacionalizar a comunicação entre agentes e o mundo.

O capítulo V apresenta os testes e os resultados obtidos. Os testes foram subdivididos em três fases distintas e com graus crescentes de dificuldade. Estes resultados foram confrontados a resultados da literatura.

O capítulo VI relata as conclusões obtidas e os possíveis trabalhos futuros que poderão ser implementados para melhorar o sistema proposto. As conclusões concentram-se sobre a performance individual de cada agente e suas contribuições e o papel do ambiente, assim como o modo de comunicação empregado entre os membros do sistema.

Finalmente são apresentados alguns de anexos que trazem informações complementares sobre a operação passo a passo o sistema AVOLI, fluxogramas, mensagens aceitas pelos agentes e pelo ambiente, bem como o leiaute obtido nos testes e as tabelas de fluxos usadas.

2. Estado da arte

Este capítulo está dividido em duas partes, que são: inteligência artificial e sistemas produtivos. A primeira contempla um estudo sobre alguns aspectos da inteligência artificial utilizados para desenvolver o sistema AVOLI, notadamente os sistemas multiagente. A segunda parte centra-se sobre os sistemas produtivos, em particular, sobre questões ligadas ao problema de geração e otimização de leiautes.

2.1 Inteligência Artificial

A inteligência artificial busca conceber sistemas computacionais autônomos que tomam decisões e apresentam certas características encontradas apenas nos seres vivos, em especial, nos seres humanos. Procura-se assim modelar sistemas computacionais que tenham uma compreensão de um mundo específico e um comportamento próprio e autônomo baseado nesta compreensão. Agentes que, se possível, possam aprender e se desenvolver sozinhos, assim como executar suas tarefas de forma cada vez mais rápida e melhor, sem a constante intervenção humana. Vários autores como [Rus95] enquadram a IA a duas definições básicas e duas variações que dão quatro categorias como mostra a Tabela 2-1, combinando pensamento com ação e racionalidade com comportamento humano. No entanto não existe um consenso único entre todos os pesquisadores sobre uma definição de IA.

<i>Sistemas que pensam como os humanos</i>	<i>Sistemas que pensam racionalmente</i>
<i>Sistemas que agem como os humanos</i>	<i>Sistemas que agem racionalmente</i>

Tabela 2-1 Definição e variações de IA

2.1.1 Inteligência Artificial Distribuída

A inteligência artificial distribuída (IAD) é considerada um ramo da Inteligência Artificial [Bon88]. Este ramo está interessado em compreender e modelar ações e conhecimento de vários sistemas inteligentes ou não que interagem para o alcance de um objetivo comum. As execuções destes sistemas podem ser do tipo seqüencial, paralela ou mista. Entre suas principais características podemos destacar:

- possibilidade de paralelismo na resolução do problema;
- disposição e distribuição das atividades e/ou da inteligência;
- especialização das tarefas para um objetivo comum; e
- não comprometimento total do sistema multi-agente caso ocorram falhas de alguns agentes do mundo.

Vale aqui salientar que existem duas áreas fundamentais e distintas na IAD:

- *resolução distribuída de problemas* que está interessada em como um problema pode ser dividido em subproblemas e alocá-los a um conjunto de agentes ou nós. Estes nós geram resultados individuais, os quais são recolhidos para dar origem a uma solução global. Os nós possuem uma coordenação intermódulos centralizada.
- *sistemas multiagente* que estão interessados no comportamento individual e coletivo de unidades computacionais resolvedoras de problemas [Mou96]. Estas unidades, também conhecidas como agentes, têm capacidades de agir e reagir, mudando o ambiente que as comporta, se preciso; são caracterizadas pela ausência de uma coordenação centralizada e pelo alto grau de independência e autonomia dos seus membros.

2.1.2 Sistemas Multiagente

Qualquer conjunto de agentes que interagem entre si num determinado mundo pode ser considerado um sistema multiagente [Dur94]. Os agentes possuem ainda uma estrutura de organização de sociedade, quando o foco se situa no grande número de agentes, seus múltiplos papéis, atividades e evolução na comunidade como um todo [Shm99]. De uma forma geral, um sistema multiagente possui, no mínimo, os seguintes componentes:

- um conjunto de agentes homogêneos ou heterogêneos, que agem e reagem em um ambiente comum a todos, através de comportamentos distintos;

- um ambiente que os comporta, dotado de métricas e padrões conhecidos pelos agentes, o que possibilita aos agentes perceberem, localizarem e atuarem sobre os objetos e o ambiente; e
- um conjunto de objetos passivos, não agentes, que sofrem manipulações pelos elementos do conjunto de agentes.

2.1.3 O que são agentes?

Pode-se encontrar na literatura uma infinidade tanto de definições como de arquiteturas de agentes, cada qual com a finalidade de explicar o uso da palavra agente. Pode-se destacar algumas definições tais, como:

- em [Rus95] encontramos: “Um agente é uma entidade que pode perceber o seu ambiente através de sensores e agir sobre este ambiente através de atuadores”;
- em [Mae95] encontramos: “Agentes autônomos são sistemas computacionais, os quais inseridos num ambiente dinâmico e complexo, percebem e atuam automaticamente neste ambiente, e fazendo-o, compreendem um conjunto de objetivos ou tarefas para as quais foram projetados”;
- em [Smi94] encontramos a definição de agentes como uma entidade persistente de “Software” dedicada a uma finalidade específica. O termo persistente é empregado para distinguir agentes de rotinas; agentes possuem suas próprias idéias de como devem realizar suas tarefas e as suas próprias agendas;
- em [Hay95] encontramos: “Agentes inteligentes executam de forma contínua três funções: *perceber* as condições dinâmicas do ambiente, *agir* para afetar as condições do ambiente e *raciocinar* para interpretar as percepções, resolver problemas, inferir e determinar ações”; e
- em [Woo95] encontramos: “Agente é utilizado para denotar um sistema computacional que goza das seguintes propriedades: *autonomia*: agentes operam sem a intervenção direta dos seres humanos ou outras entidades, e exercem algum tipo de controle sobre as suas ações e estados internos; *habilidade social*: agentes interagem com outros agentes; *reatividade*: agentes percebem os seus ambientes e respondem rapidamente às trocas que neles ocorrem e, *pró-atividade*: agentes não agem apenas em respostas às alterações dos seus ambientes; eles são também capazes de exibir um comportamento orientado por objetivos através de iniciativas”.

2.1.4 Taxionomia de um agente

Definido o que seja um agente, ainda podemos classificá-los quanto à sua natureza e à classe a que ele pertence. As classes que definem um agente são baseadas em suas dimensões, sua concepção, constituição, granularidade, etc. Estas classes são úteis para o enquadramento dos agentes, permitindo assim decidir a classificação do sistema como heterogêneo ou homogêneo. Dentre vários esquemas possíveis de classificação (Tabela 2-2), encontramos em [Huh98] o que está representado a seguir:

Agente				Dimensão que caracterizam quanto à constituição
Natural		Artificial		origem
humano	Não-humano	Virtual	físico	existência
		Elementar	Super	granularidade
Dimensões que caracterizam quanto às capacidades e comportamentos				
.cognição / reaçãointeligênciaracionalidadeintencionalidade .autonomia .sociabilidade especialidade				

Tabela 2-2 Algumas das dimensões para a classificação de agentes

Conforme mostra Tabela 2-2, temos as dimensões que caracterizam os agentes quanto à constituição e quanto às capacidades e comportamentos. Quanto à constituição dos agentes pode-se ainda citar:

- Origem: *natural* quando o agente está contido no conjunto dos objetos ou fenômenos estudados pelas ciências naturais, ou *artificial* quando o agente é um artefato “man-made” [Sim68].
- Existência: *física* quando o agente pode alterar o mundo real ou *virtual* quando se trata de um componente de “software”.
- Granularidade: *elementar* quando os componentes do agente não são qualificados como agentes ou *super* quando o agente é constituído por partes menores as quais são agentes elementares ou super agentes.

Quanto às capacidades e comportamentos dos agentes pode-se destacar:

- *Cognitivos*: se dotados de observação, raciocínio e memória e capazes de criar e executar planos de ação para o alcance de seus objetivos. Ditos inteligentes podendo ser qualificados como racionais que buscam seus objetivos de forma certa sem evitá-los. E são intencionais na medida que são guiados pelas suas crenças, desejos e intenções (“BDI-architecture, Beliefs, Desires and Intentions”) [Rao95];
- *Reativos*: quando, na medida que percebem mudanças no seu ambiente, respondem rapidamente, com uma baixa atividade de raciocínio, a estas mudanças [Gol96];
- *Autonomia*: determina o quanto o agente possui de controle sobre suas ações [Dav96], podendo ser de completamente autônomo até totalmente não autônomo, quando o controle deste agente se encontra fora dele;
- *Sociabilidade*: classifica os agentes se possuem explicitamente, ou não, modelos dos outros agentes da comunidade e consideram este conhecimento para a tomada de decisões; e
- *Especialidade*: determina o domínio específico que cada agente possui para a resolução de determinado problema e como ele executa tal tarefa específica.

Em [Woo94] distingue-se ainda o uso do termo agente denotado por características fracas e fortes, que são:

- *características fracas*: os agentes possuem as seguintes propriedades: autonomia, sociabilidade, reatividade e pró-atividade; e
- *características fortes*: os agentes, além de possuírem todas as características fracas, possuem também outras aplicáveis usualmente aos seres humanos como: conhecimento, crenças, intenções, obrigações e emoções.

2.2 Sistemas produtivos e arranjo de leiaute

A sobrevivência de uma empresa em um mercado altamente competitivo depende de uma redução de custos, aumento da qualidade dos produtos e crescimento contínuo da produtividade. Desta forma, é imprescindível compreender como um leiaute pode influenciar positivamente nos custos de fabricação e nos índices de produtividade das empresas.

O leiaute é a maneira como homens, máquinas, equipamentos e dentre outros elementos ligados a um sistema produtivo estão dispostos em um determinado local. Pode-se

também definir leiaute como sendo a organização das instalações de uma companhia para promover o uso eficiente dos equipamentos, materiais, pessoas e energia [Mar02].

A análise de fluxo é essencial para a construção de um leiaute. Ela fornece informações importantes ao projeto do leiaute e contribui de forma decisiva para alcançar os seguintes objetivos:

- minimizar a distância viajada;
- minimizar regressos (contra-fluxos);
- minimizar fluxos cruzados;
- eliminar operações ou passos desnecessários no processo;
- combinar e encadear operações no processo; e
- minimizar custos de produção.

Existem várias técnicas e informações que se fazem necessárias para analisar um fluxo, tais como:

- roteiros de fabricação;
- operações;
- transportes;
- armazenamentos;
- inspeções;
- atrasos;
- diagrama de fluxo (fluxograma); e
- relação de operações.

Pode-se também destacar algumas perguntas chaves que ajudam analisar fluxos, tais como:

- pode-se eliminar este processo?
- pode-se automatizar este processo?
- pode-se combinar este processo com outro?
- pode-se mudar a rota para reduzir as distâncias viajadas dos matérias?
- pode-se posicionar os postos de trabalho mais intimamente?
- podem-se justificar apoios de produção para aumentar a eficiência?
- quanto custa produzir esta parte?
- vale a pena produzir esta parte?

- há fluxo cruzado?
- há regressos (contra-fluxos)?
- existem transportes excessivos?

2.2.1 Problema de atribuição quadrática

Vários pesquisadores trabalharam sobre as questões ligadas a disposição de leiautes e seus departamentos, cujos pioneiros são: Koopmans & Beckman [Koo57], Armour & Buffa [Arm63] e Nugent [Nug68]. Estes trabalhos concentram-se na otimização das localizações relativas dos departamentos sobre um único pavimento, o que caracteriza o problema leiaute de fábrica como sendo um problema de atribuição quadrática ou simplesmente QAP⁴ (do inglês "Quadratic Assignment Problem").

O QAP foi aplicado a um conjunto de situações que incluem: planejamento urbano, leiautes de painéis de controle e design de rede elétrica [Baz75]. A formulação QAP atribui toda máquina ou departamento a uma localização no interior de um leiaute e no máximo uma máquina ou departamento para cada localização. O custo de alocação para uma particular localização da construção é em função da localização dos departamentos que estão interagindo no sistema, i.e., para se conseguir um custo menor, é necessário dispor cada máquina ou departamento de maneira otimizada.

Em um QAP ignoram-se as restrições comumente encontradas em um sistema de manufatura real, por exemplo, a área a ser alocada a um departamento específico, sua razão de aspectos, localização obrigatória, adjacências, etc. Todos os departamentos possuem áreas quadradas iguais. Desta forma, os trabalhos produzidos nas últimas décadas não consideravam leiautes realistas, em particular, devido à complexidade inerente ao problema. É importante salientar que, no contexto desta dissertação, será feito um esforço singular para comparar os resultados obtidos pelos métodos QAP e AVOLI. Este último possui a capacidade de operar com um número de variáveis próprias de situações do mundo real.

Um QAP possui uma forma particular de funcionamento. Por exemplo, têm-se m departamentos que precisam ser atribuídos a n localizações. Cada localização possui uma porção de espaço que será ocupada pelo departamento atribuído a ela. Um fluxo positivo é associado a cada par de departamentos como forma representação da interação entre os

⁴ QAP é um problema do tipo NP-Completo [Gar79].

mesmos. Desta forma, o problema consiste em atribuir cada departamento a uma localização específica de tal forma que a soma dos fluxos produtivos entre os departamentos pelo produto das distâncias entre todos os departamentos seja a menor possível. Pode-se inferir que cada localização será atribuída a um único departamento e conseqüentemente nunca haverá sobreposição de departamentos.

Um artifício para que os departamentos tenham diferentes tamanhos é dividi-los em vários departamentos. Criam-se então para estas frações elevados fluxos artificiais, o que os farão permanecer juntos. A abordagem QAP, no entanto, apresenta muitas limitações quando se opera sobre um conjunto de restrições do mundo real.

2.2.2 Algoritmos construtivos

Os algoritmos construtivos fazem com que os departamentos sejam posicionados no leiaute um de cada vez até que o mesmo seja completado [Tam98]. Assim, definido o problema e as dimensões do leiaute, próximo passo é automatizar o processo para dispor todos os departamentos do leiaute de uma maneira rápida e eficiente. Pode-se destacar, como instância destes algoritmos, os seguintes trabalhos:

- CORELAP – “**CO**mputerized **RE**lationship **L**eiaute **P**lanning” [Lee67];
- ALDEP – “**A**utomated **L**ayout **D**esign **P**rogram” [See67];
- “Hierarchical Approach” [Tam91]; e
- MIP – “**M**ixed **I**nteger **P**rogram” [Mon90].

CORELAP

O método constrói inicialmente um leiaute e calcula a proximidade média total de cada departamento em relação aos demais. Esta média é dada pela soma dos valores numéricos atribuídos sobre os relacionamentos de cada departamento com todos os outros departamentos que compõe o sistema. O departamento que tem a maior média total de proximidade é então atribuído ao centro do Leiaute.

Em caso de empate entre departamentos, escolhe-se o que tiver maior área. O próximo passo é percorrer a tabela de relacionamento e se algum departamento possui um relacionamento com o departamento inicialmente selecionado, este é posicionado no leiaute ao seu lado. Pode-se, no entanto, haver departamentos que não tenham relacionamentos com o

departamento selecionado. Neste caso, a tabela de relacionamento é novamente percorrida à procura de um departamento de média alto ainda não posicionado.

Caso dois ou mais departamentos sejam encontrados e que possuem o mesmo relacionamento com o departamento selecionado, então o departamento de maior média total de proximidade é selecionado. O terceiro departamento a compor o leiaute é determinado percorrendo-se novamente a tabela de relacionamento para verificar se existe um departamento ainda não atribuído e que possui relacionamento com o primeiro departamento selecionado. Se existir, este departamento é atribuído ao leiaute. Se houver empate, a média total de proximidade é usada conforme descrito anteriormente.

Se não existe nenhum departamento que mantém relacionamento com o segundo departamento, o procedimento é repetido considerando, em ordem, o relacionamento do terceiro, do quarto departamento, etc. O mesmo procedimento é repetido para todos os departamentos, exceto para os departamentos que já foram posicionados no leiaute. O procedimento continua até que todos os departamentos sejam posicionados no interior do leiaute.

ALDEP

Este algoritmo possui os mesmos objetivos e dados básicos de entrada do CORELAP. A diferença básica entre ALDEP e CORELAP é que CORELAP seleciona o primeiro departamento a ser posicionado no leiaute e resolve empates usando a média total de proximidade, enquanto ALDEP seleciona o primeiro departamento e resolve empates aleatoriamente. A diferença filosófica básica entre eles é que CORELAP procura produzir um único leiaute, enquanto ALDEP produz vários leiautes e deixa ao projetista a tarefa de escolher o que lhe mais agrada.

O primeiro departamento selecionado por ALDEP para compor o leiaute é selecionado aleatoriamente, conforme mencionado anteriormente. A tabela de relacionamento é então percorrida para determinar se existe um departamento que tem um relacionamento com o departamento selecionado aleatoriamente. Se existe um departamento, ele é selecionado para compor o leiaute. Se existir mais de um departamento, um deles é selecionado aleatoriamente. Se não existir nenhum departamento com um grau mínimo de relacionamento especificado pelo usuário, o segundo departamento a compor o leiaute será escolhido também

aleatoriamente. Uma vez que o segundo departamento já foi selecionado, o procedimento de seleção continuará até que todos os departamentos sejam selecionados para compor o leiaute.

O ALDEP inicia a seqüência de construção, atribuindo o primeiro departamento no lado esquerdo da região de particionamento. Cada novo departamento atribuído ao leiaute será disposto à direita do último departamento atribuído à região do leiaute. Após todos os departamentos já terem sido selecionados e atribuídos ao leiaute, o algoritmo ALDEP avalia a qualidade do leiaute atribuindo valores aos relacionamentos entre departamentos adjacentes. Se um departamento for adjacente a outro departamento com um relacionamento do tipo A, o valor 64 é atribuído à medida da qualidade. Um relacionamento do tipo E adiciona 16, um relacionamento do tipo I adiciona 4, um relacionamento do tipo O adiciona 1 à qualidade do leiaute, um relacionamento do tipo U não adiciona nada, e se dois departamentos estão adjacentes possuindo um relacionamento do tipo X, então o valor -1024 é adicionado à qualidade do leiaute. Toda vez que o ALPED é executado, ele pode gerar até vintes diferentes leiautes.

Hierarchical Approach

Este algoritmo é recente e foi apresentado por [Tam91] e melhorado por [Fur97] e [Aza00]. O algoritmo não usa a abordagem QAP, em particular, porque seus elementos não precisam ter necessariamente as mesmas medidas. Este método não considera as razões de aspectos mínimas e máximas definidas pelas necessidades dos departamentos, esta abordagem funciona de forma razoável em situações realistas. Esta abordagem utiliza para construir o leiaute uma estrutura do tipo árvore binária.

Inicialmente é produzida uma árvore binária baseada nos fluxos entre departamentos, de forma que os departamentos com forte acoplamento fiquem próximos na árvore. A partir desta organização em árvore é possível construir um leiaute dividindo-a várias vezes. A árvore será percorrida recursivamente a partir da raiz e as áreas correspondentes serão assinaladas na região de divisão.

Seja árvore de divisão da Figura 2-1, inicia-se o processo de divisão da região da seguinte maneira: começando-se pela raiz (nó -5), calcula-se o somatório das áreas dos departamentos que estão no ramo esquerdo. O valor obtido é usado como região de divisão, onde se varre a região até se obter a área correspondente a este somatório, mostrado na Figura 2.1. Faz-se então, a divisão, e em seguida continua-se a percorrer a árvore seguindo-se para o

nó -4 e novamente a área do ramo esquerdo (nó 1) é calculada e traduzida na região de divisão. Uma vez realizado esta divisão, a região dividida é então identificada (neste caso, o departamento 1). O processo de percorrer a árvore é recursivo e continua até que ela seja percorrida por completo, correspondendo ao final do processo de construção do leiaute por divisões.

Como se pode observar nesta abordagem, o aproveitamento da área total do leiaute é ótimo. No entanto ela não leva em consideração as razões de aspectos dos departamentos e em decorrência disto percebe-se que há violações de razões de aspectos dos mesmos. Além disto, não considera as áreas ocupadas, departamentos fixos e restrições de adjacências, o que poderia tornar a solução inviável no mundo real.

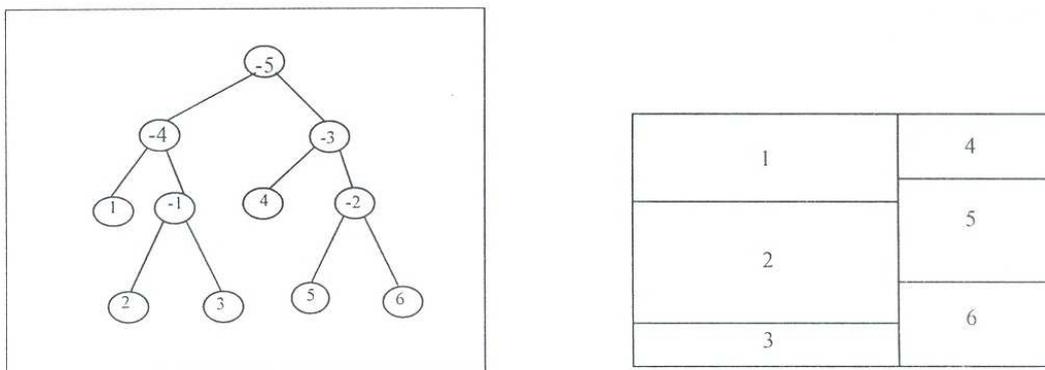


Figura 2-1 Árvore de divisão e Leiaute gerado

Teoria de Grafos

Foulds usa uma abordagem baseada em grafos. Nela é assumido que a localização desejável de cada par de departamentos adjacentes seja conhecida previamente. Inicialmente, o aspecto físico e a quantidade de área exigida pelo departamento são ignorados e cada departamento é então representado por um nó no grafo. As relações de adjacências serão representadas por arcos conectando dois departamentos adjacentes no grafo. O problema desta abordagem é que não trata nenhuma restrição, o que o torna inviável no mundo real.

MIP

MIP foi apresentado por [Mon90] e [Mon93]. O problema de leiaute também pode ser modelado como um problema de programação inteira mista (MIP). Um caso especial deste

modelo foi proposto por [Her91], onde são especificadas as larguras, comprimentos e as orientações dos departamentos. No MIP, a primeira solução obtida para o leiaute, geralmente é referida como um leiaute em “Bloco”, o qual mostra os contornos que satisfazem os requerimentos de área e aspecto de cada departamento. Um leiaute em bloco geralmente não especifica os corredores ou o arranjo de outros objetos como pilares e elevadores no interior de cada departamento. Em leiaute em bloco os departamentos que o compõem não podem ser divididos em duas ou mais partes.

2.2.3 Busca de uma solução ótima

Um algoritmo construtivo permite gerar um leiaute inicial, porém sem garantir que ele seja ótimo. O próximo passo é aplicar um algoritmo de refinamento que melhore a qualidade deste leiaute.

Vários algoritmos foram desenvolvidos na tentativa de se obter soluções ótimas para melhorar o leiaute inicial. Por exemplo, um algoritmo baseado na força bruta tenta todas as combinações possíveis de arranjos dos departamentos. Este algoritmo não permite obter soluções ótimas para problemas com mais de *nove* departamentos. O tempo de processamento é proibitivo além de requer uma quantidade gigantesca de memória. Desta forma, as pesquisas passaram a concentrar seus esforços na busca de soluções próximas da ótima através da utilização de heurísticas. Podem-se destacar algumas implementações visando minimizar os custos operacionais com leiaute como, por exemplo: CRAFT [Arm63], CORELAP [Lee67], ALDEP [See67], FLAC [Scr85].

CRAFT (Computerized Relative Allocation of Facilities Technique) [Arm63] foi o primeiro algoritmo a propor uma técnica computacional para resolver o problema de leiaute. A técnica busca minimizar o custo de transporte. Este último é dado pela distância que um produto percorrer em um processo de fabricação, levando em consideração o fluxo estimado (quantidade deste produto em um determinado período de tempo). O CRAFT começa determinando os centróides dos departamentos no leiaute inicial. Na sequência calcula-se a distância retangular entre os centróides dos departamentos. Estas distâncias são armazenadas em uma matriz, em seguida algoritmo avalia as permutações de departamentos com áreas iguais ou com uma borda em comum. Vários tipos de permutações são possíveis, a saber:

- permutação de pares de departamentos;

- permutação de três departamentos;
- permutação de um par, seguida pela permutação de três departamentos;
- permutação de três departamentos, seguida pela permutação de um par ou da melhor dentre a permutação de um par ou de três departamentos; e
- permutação de departamentos que oferecer a maior redução do custo de transporte.

O procedimento continua até não existir mais nenhuma permutação que reduza o custo de transporte. Uma desvantagem deste algoritmo é que o resultado final é inteiramente dependente da solução inicial, assim, é necessário executar o procedimento diversas vezes para a obtenção de resultados satisfatórios. Outro problema é que este algoritmo pode ficar num ótimo local, pois finaliza sempre que não há nenhuma melhora.

FACTORY (Cimtechnologies Corp. 1989) [Cor94] atua conjuntamente com o AUTOCAD (AutoDesk Inc.). Ele representa um avanço em relação às abordagens anteriores, principalmente, por explorar recursos de computação gráfica e a facilidade de interagir com o projetista. Existem também, no contexto de projetos gráficos, excelentes softwares como o MICROSTATION (Microsoft 1994) [Mic94] e 3D STUDIO. Porém não existem sistemas computacionais que combinem as características individuais destes softwares de forma amigável. Além disso, não existe uma metodologia apropriada que absorva o potencial destas ferramentas para o projeto de leiaute comercial.

2.2.4 Algoritmo de busca tabu

A busca tabu é uma metodologia heurística aplicado sobre problemas de otimização combinatória. Ele é um método de refinamento e pode ser empregado na busca de soluções próximas da ótima ([Glo89a] e [Glo97]). Em geral, este método é utilizado para resolver problemas ligados a área de pesquisa operacional. Ele segue os seguintes passos:

- a) escolher uma lista inicial que possua uma solução S .
- b) avaliar todas as listas candidatas ao movimento e suas soluções S' .
- c) selecionar a melhor lista admissível com solução S_{melhor} , onde: S_{melhor} é a melhor solução entre todas $S' \in N(S)$: S' não pertencendo a lista tabu e que satisfaça algum critério de aspiração que permite violar a lista tabu.
- d) atualizar a solução corrente $S \leftarrow S_{\text{melhor}}$, assim como a lista tabu.

- e) parar, se o número de iterações informado foi satisfeito, retornando o melhor resultado de todos, caso contrário voltar (b).

Este método precisa armazenar a melhor solução de todas as iterações, para retorná-la no final do processo e para testar os critérios de aspiração. Deve-se destacar que existe um tempo tabu especificado. Ele impede que uma mesma lista seja escolhida mesmo que ela seja a melhor das soluções na iteração corrente, exceto que esta lista apresente uma solução melhor do que todas as encontradas até então. Este tempo tabu é empírico. Ele impede que a solução se estabilize em um ótimo local. Estabelecer este tempo empírico é um grande inconveniente.

Uma variação da busca tabu foi sugerida para o refinamento da solução inicial [Mar02]. Ela baseia-se em uma pseudotroca e um tempo tabu dinâmico. Este método está disponível no sistema AVOLI (versão 2.0). Ele será usado essencialmente para servir como uma solução a ser confrontada a solução baseada em agentes, proposta deste trabalho.

O algoritmo construtivo usado será detalhado no Capítulo III.

2.3 Considerações gerais

Pode-se notar que a maioria dos algoritmos e softwares voltados para a solução de problemas de leiautes gera apenas diagramas de blocos, os quais representam soluções aproximadas não realistas, exceto o método busca tabu empregado em [Mar02]. O redesenho e as modificações por parte do projetista ou engenheiro de produção podem provocar um distanciamento significativo de uma solução ótima ou quase ótima encontrada. Estas ferramentas estão longe de serem práticas e eficientes. O trabalho proposto deve não apenas apresentar um novo método e favorecer um estudo comportamental de agentes para resolver a otimização de leiautes realistas, mas também contribuir com uma ferramenta computacional útil e de fácil manuseio. A Tabela 2-3 mostra um resumo das restrições tratadas pelo AVOLI e outros métodos relatados na literatura.

Os métodos propostos, baseados no modelo multiagente, parte também de uma solução inicial previamente construída por um outro método qualquer e tenta melhorá-la. O mesmo se aplica ao algoritmo da busca tabu e pseudotrocas também incorporado no sistema AVOLI. O sistema AVOLI pode também sugerir dimensões iniciais de um leiaute baseado nas dimensões de seus departamentos.

RESTRIÇÕES	[Tam91]	[Tam92]	[Fur97]	[Tam98]	[Chi01]	AVOLI
Quadro externo		X	X	X	X	X
Sobreposição	X	X	X	X	X	X
Adjacência						X
Não adjacência						X
Áreas desiguais	X	X	X	X	X	X
Razão de aspecto	X	X	X	X		X
Razão de área morta		X	X			X
Razão de redução						X
Orientação (V/H)	X					X
Departamentos Móveis	X	X	X	X	X	X
Departamentos Fixos	X					X
Áreas ocupadas		X	X			X
Tamanho do maior problema	30	30	30	20	40	55

Tabela 2-3 - Comparativo de restrições tratadas por vários métodos

3. Sistema AVOLI

O sistema AVOLI permite construir e otimizar leiautes. Ele implementa vários métodos de construção e otimização de leiautes, que são:

- planejamento multiagente;
- força bruta;
- busca tabu com pseudotrocas;
- construção automática;
- construção manual; e
- construção aleatória.

O sistema AVOLI deve possibilitar, facilmente, o acréscimo de outros métodos além dos supracitados. Ele deve também permitir o uso de vários métodos de forma seqüencial para aproveitar as melhores características de cada um na otimização de um leiaute específico. Deve-se salientar que uma característica importante do modelo multiagente é a naturalidade com que se podem expandir os comportamentos de um agente (um agente pode implementar vários métodos e ativar um ou outro em função de suas estratégias) e conseqüentemente do sistema.

Um requisito importante é que a entrada de informações sobre o ambiente do leiaute e seus departamentos seja mesma para todos os métodos implementados. Isto deve facilitar o uso da ferramenta AVOLI. Estas informações dizem respeito às definições iniciais para o funcionamento do sistema, tais como:

- áreas do leiaute e departamentos;
- razões de aspectos dos departamentos;
- reduções permitidas nos departamentos;
- localizações exatas de áreas ocupadas e departamentos fixos;
- tipo de solução inicial gerada;

- restrições de adjacências entre os departamentos;

Apresentar-se-á, na seqüência, uma breve descrição das informações iniciais requerida pelo sistema. Os detalhes de funcionamento e comando do sistema AVOLI são apresentados no anexo A.

3.1 Definições de aspecto e área do Leiaute

Ilustrativamente, um leiaute pode ser um barracão ou qualquer outro local físico disponível para a distribuição de departamentos no seu interior. Normalmente, as dimensões de um leiaute são conhecidas. Quando elas não o são, pode-se calculá-las a partir das dimensões do leiaute e das áreas e razões de aspecto de todos os departamentos (Figura 3-1). Estas medidas são dadas em termos de uma largura e um comprimento parametrizados (Figura 3-2). Em princípio, o leiaute é um bloco retangular e dimensões conhecidas. Apesar de ser possível, através de alguns artifícios, fazer com que a área do leiaute tenha também formatos irregulares. Para tanto bastaria posicionar alguns departamentos fixos sem nenhum fluxo ou relacionamentos nos limites do leiaute.

Considera-se o metro como a medida padrão para o leiaute, apesar de que a escala pode ser mudada dinamicamente. É importante salientar que o sistema, por incremento da complexidade e uma forte atenuação da performance, atua apenas com medidas baseadas nos números inteiros para as dimensões de largura e comprimento, tanto do leiaute, quanto dos departamentos fixos e móveis, bem como para as áreas ocupadas. Para uma maior precisão, se necessário, pode-se mudar a escala para decímetros ou centímetros ou ainda milímetros, apesar de que isto não é recomendado, pois acarretaria num acréscimo exponencial no processamento e no tempo final.

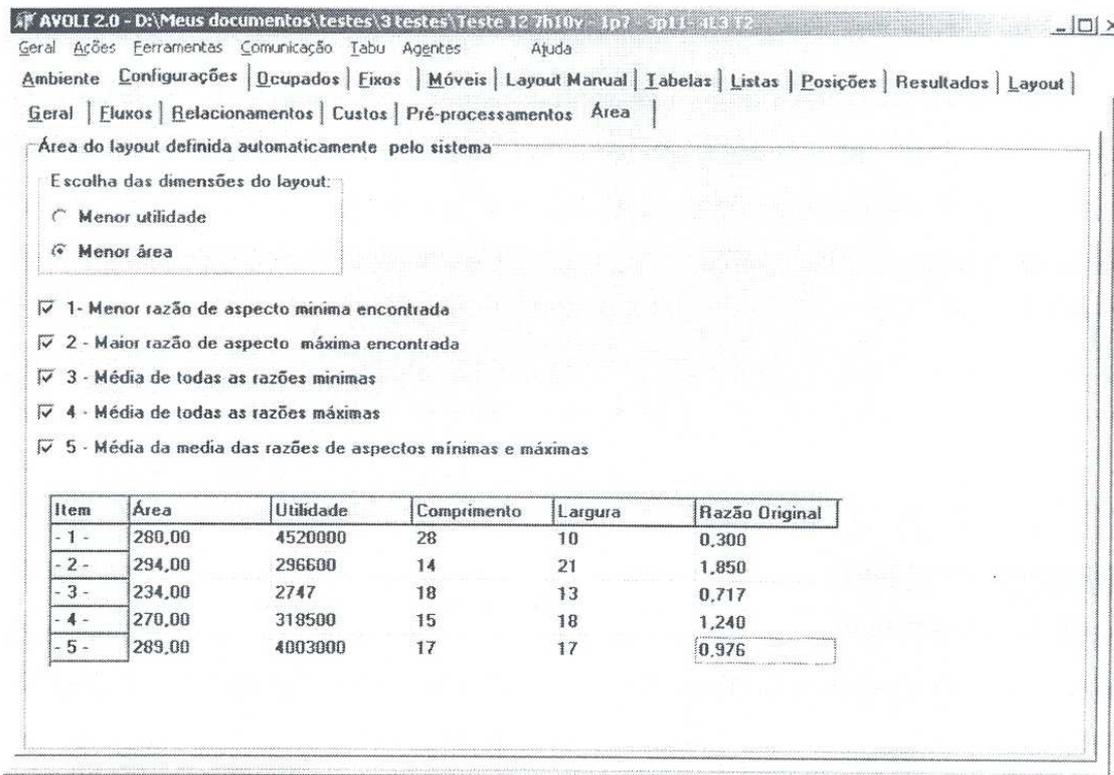


Figura 3-1 Sugestões do sistema AVOLI para a área do leiaute

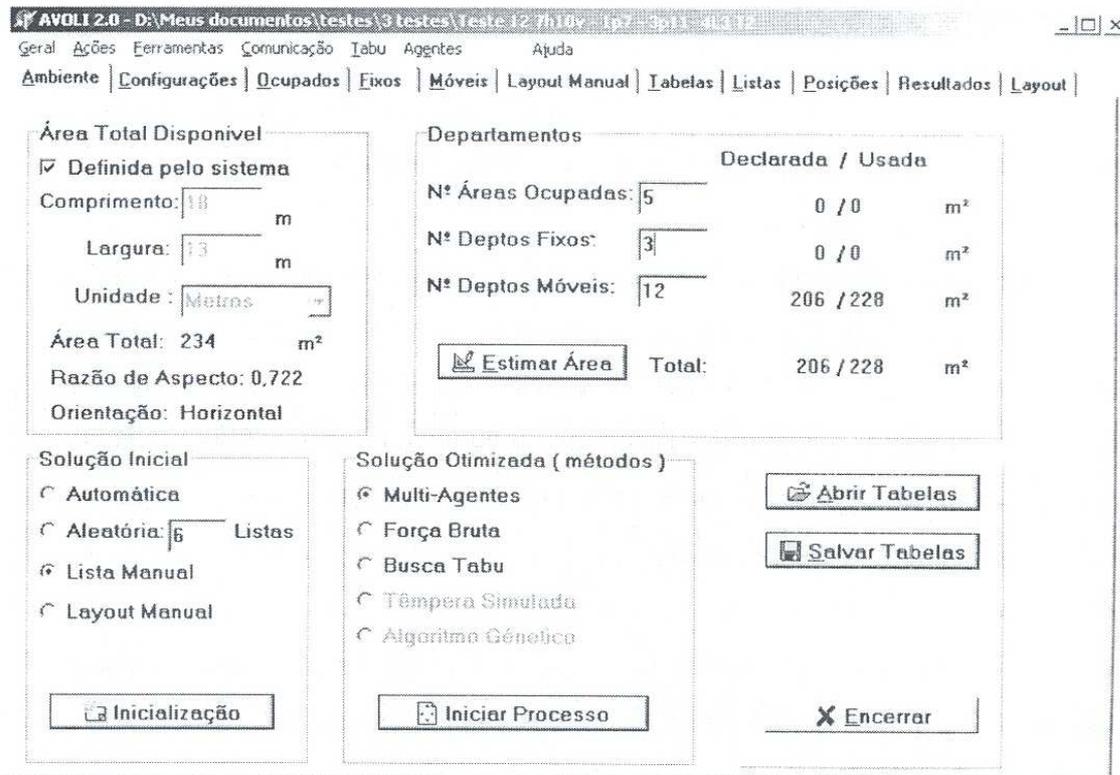


Figura 3-2 Definições iniciais da dimensão do leiaute, número de departamentos, área ocupadas, métodos, etc

3.2 Definições de aspecto e área dos departamentos

Os departamentos são blocos retangulares, que possuem um comprimento na horizontal e uma largura na vertical. A razão entre estas duas medidas chama-se de razão de aspecto. Por exemplo, um bloco quadrado é um caso especial de um retângulo de razão de aspecto 1 (um). Existem dois tipos básicos de departamentos que se pode definir no sistema AVOLI:

- departamentos de tipo móvel; e
- departamentos de tipo fixo.

3.2.1 Departamentos móveis e suas informações

Os departamentos móveis podem ser posicionados em qualquer parte no interior do leiaute, se e somente se, eles não se sobreponem a outros departamentos móveis ou fixos. Eles também não podem perder sua área útil por sobreposição de áreas ocupadas mais do que o definido. As perdas máximas por sobreposição de áreas ocupadas e outras informações sobre os departamentos móveis são definidas na sua grade no item 'Móveis' do sistema AVOLI (Figura 3-3). Estas informações são:

- área total do departamento móvel, dada pela fórmula da área do retângulo que é igual à largura (na vertical) pelo comprimento (na horizontal);
- sua orientação, que se definida, é uma restrição forte, ou seja, ela não pode ser violada. A orientação pode ser de três tipos e sempre estarão dentro dos limites estabelecidos na sua razão de aspecto: (a) *sem orientação ou orientação livre* (default): o campo na grade deve ser vazio; neste caso a impressão visual de que o departamento encontra-se na horizontal ou na vertical não será levada em consideração; (b) *orientação vertical*: marcada na grade do sistema com um 'V' ou 'vertical'. Informa ao sistema que necessariamente a largura do departamento deverá ser maior ou igual ao comprimento, dando assim a impressão que o departamento se encontra disposto sempre na vertical; e (c) *orientação horizontal*: marcada na grade do sistema com um 'H' ou 'horizontal'. Informa ao sistema que necessariamente o comprimento deverá ser maior ou igual à largura, dando assim a impressão de que o departamento se encontra na horizontal.
- quanto, em porcentagem, cada departamento pode perder de sua área útil sobreposta a áreas ocupadas. Estas especificações são do tipo restrição forte e nunca poderão ser

violadas no sistema. A menos que se marque nas configurações iniciais para não verificar e corrigir esta sobreposição de áreas.

- razão de aspecto mínima e máxima de cada departamento. Como os departamentos têm formatos retangulares, a razão de aspecto é definida como sendo a razão dada pelo comprimento do departamento dividido pela largura. Esta configuração permite uma maior flexibilidade em seu formato e uma possível melhora no arranjo do departamento no interior do leiaute. Estas especificações são do tipo restrições fortes, logo nunca serão violadas no sistema. O aspecto deve ser sempre preservado.
- definição do máximo de redução da área útil que o departamento pode sofrer. Este valor é dado em porcentagem e se nada for atribuído, será considerado como zero, ou seja, não será permitida nenhuma redução da área útil do departamento. O motivo para o qual existe esta informação é para permitir flexibilidade de posicionamento no sistema. Por exemplo, por mais que se deseje uma área de 100m^2 para um determinado departamento, só há disponível no leiaute 90m^2 e por isto este departamento não caberá no leiaute. Mas o departamento aceita uma redução máxima de até 20% de sua área útil total, previamente definida na redução máxima. No entanto deve-se salientar neste ponto que o sistema tentará medidas intermediárias em um número de passos definidos na sua configuração inicial, antes de chegar no mínimo de área possível. Então se a área de 100m^2 para o departamento não caber no leiaute, mas a de 90m^2 pode servir, então ele não usará os 80m^2 mínimos e sim a de 90m^2 . Por isto quanto maior o número de passos, mais refinado será a resposta, com um aproveitamento de área mais otimizado. O tempo de processamento será proporcional ao número de passos definidos.

No AVOLI, é importante notar que por questões de arredondamento a área total do departamento poderá diminuir um pouco. Mas se a área do departamento ficar abaixo do mínimo definido então o lado menor será arredondado para cima, ou seja, o próximo valor inteiro. O lado menor sempre será escolhido para minimizar os efeitos de aumento de área. Às vezes, devido a este arredondamento para cima, pode vir a acontecer que a área fique um pouco maior do que a definida originalmente. Quando isto acontece o valor de redução na grade de posições é marcada com um valor negativo, que indica o quanto à área foi acrescida de seu valor inicial por questões de arredondamento (Figura 3-8).

	Área	Orientação	Perda A. Morta (%)	R. Asp. Mín.	R. Asp. Máx.	Red. Máx. Área (%)	Área Mínima
+1	100		0	0,7	1	0	
+2	80		0	1	1	0	
+3	50		0	0,7	1,3	0	
+4	60	h	0	0,5	0,8	0	
+5	120		0	0,9	1	0	
+6	40		0	0,6	1	0	
+7	20		0	0,7	1,4	0	
+8	40	v	0	1	1	0	
+9	150	v	0	0,8	1,1	0	
+10	120	h	0	0,5	1,5	0	
+11	50		0	0,7	1,1	0	
+12	10		0	0,8	1,2	0	
+13	20		0	0,95	1,5	0	
+14	30		0	0,75	1,25	0	
+15	50		0	0,9	1,1	0	
+16	20		0	0,8	1,5	0	
+17	40	h	0	0,4	1,4	0	
+18	20		0	0,9	1,9	0	
+19	80	v	0	1	1	0	
+20	100		0	0,95	1,15	0	
+21	40		0	0,5	1,5	0	
+22	50		0	1	1,1	0	
+23	80		0	0,6	1	0	
+24	10		0	0,9	1	0	
+25	40		0	0,8	1,1	0	
+26	10		0	0,5	1,2	0	
+27	40		0	0,8	1	0	
+28	10		0	0,5	1,3	0	
+29	80	v	0	,7	1,05	0	

Figura 3-3 Informações sobre os departamentos móveis

3.2.2 Departamentos fixos e suas informações

Os departamentos fixos possuem uma localização pré-definida no leiaute e não podem ser movidos para outras posições. Eles são previstos para suprir algumas limitações do mundo real, como, por exemplo, se a localização do departamento é dada por alguma característica do ambiente e só lá este departamento pode ser posicionado. Eles serão sempre posicionados antes dos departamentos móveis, pois não possuem a mobilidade dos mesmos, e sua localização é de inteira responsabilidade do usuário.

	Área	Xinicio	Yinicio	Orientação	RA Morta	R. Asp. Mín.	R. Asp. Máx.
-1	60	28	1		0	,5	1
-2	450	69	36		0	1	1
-3	360	41	1		0	1	1
-4	200	1	47		0	1	1
-5	80	42	38	h	0	,6	1,2

Figura 3-4 Informações sobre os departamentos fixos

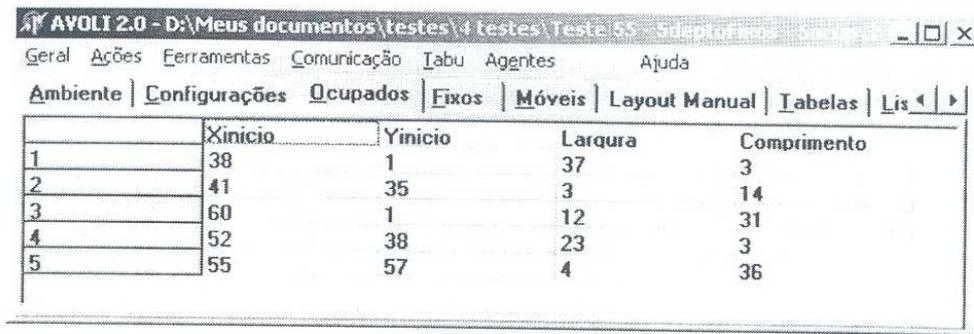
A Figura 3-4 mostra as informações que devem ser dadas ao sistema no item 'Fixos', que são:

- sua área total dada pela área do retângulo (igual aos departamentos móveis);
- coordenadas das posições X (na horizontal) e Y (na vertical) iniciais que correspondem ao canto superior esquerdo do departamento que indicará onde o departamento será posicionado no leiaute.
- sua orientação no leiaute. Os valores destes campos seguem o mesmo padrão dos departamentos móveis acima citados.
- razão da área morta em porcentagem que informa ao sistema o quanto o departamento pode perder com áreas sobrepostas as áreas ocupadas. No entanto, ao contrário dos departamentos móveis, não há nenhuma restrição forte neste caso (a localização deste tipo de departamento é responsabilidade do usuário). O sistema simplesmente se limitará a informar as violações ocorridas. Ele não tem autonomia para mudar a posição de um departamento fixo de lugar.
- razões de aspecto mínimo e máximo. Este item é igual ao dos departamentos móveis.

3.3 Áreas ocupadas

Também conhecidas como áreas mortas, são áreas inutilizadas no interior de um leiaute. Estas áreas são fixas, a saber: pilares, escadas, elevadores, ou outra área característica do local sem uma utilidade. Porém, não podem ser removidas. A existência destas áreas irá distorcer a forma retangular da área útil e podem comprometer as atividades de um departamento. Assim, nos departamentos móveis e fixos, para efeitos práticos e realistas, pode-se determinar o máximo de perdas de suas áreas úteis por sobreposição de áreas ocupadas.

As perdas máximas da área útil total de um departamento móvel são tratadas no sistema como uma restrição forte, i.e., elas nunca poderão ser violadas e conseqüentemente o departamento móvel nunca será posicionado em um local no interior do leiaute que o faça perder mais do que o permitido de sua área útil. Para os departamentos fixos tais perdas não serão consideradas como uma situação do tipo restrição forte. Desta forma, poder-se-á perder áreas superiores a área permitida previamente acordada, visto que os departamentos fixos são de inteira responsabilidade do usuário.



	Xinicio	Yinicio	Largura	Comprimento
1	38	1	37	3
2	41	35	3	14
3	60	1	12	31
4	52	38	23	3
5	55	57	4	36

Figura 3-5 Informações sobre as áreas ocupadas

As informações que devem ser fornecidas ao sistema no item 'Ambiente' e 'Ocupados' sobre as áreas ocupadas são:

- número total de áreas ocupadas na tela do sistema 'Ambiente';
- coordenadas no leiaute da posição X e Y iniciais da área ocupada que correspondem ao canto superior esquerdo da área ocupada para seu posicionamento no leiaute (Figura 3-5).
- largura e comprimento da área ocupada, que determinam sua área de ocupação, que aqui sempre será no formato retangular (Figura 3-5).

3.4 Restrições de localização

Existem situações suplementares, em ambiente realista, que precisam ser consideradas e tratadas como, por exemplo, restrições de adjacências, que, dependendo da situação, podem ser mais importantes que o próprio fluxo entre os departamentos. Pode haver casos de departamentos que, apesar de terem um alto fluxo entre departamentos, não podem ficar próximos. Por exemplo, em uma indústria que possui um departamento de solda e outro de pintura, o produto sairia direto da solda para a pintura, mas por motivo de segurança, o departamento de solda poderia acidentalmente soltar alguma faísca de solda e cair no departamento de pintura, acarretando um sério acidente. Tais departamentos devem ficar distantes.

Há situações que são exatamente o oposto da citada acima. Há dois departamentos que devem ser necessariamente vizinhos um do outro. Por motivos de segurança ou qualquer outro. Por exemplo, o ambulatório deve ficar perto de um departamento reconhecidamente problemático. Não existe, no entanto, um fluxo entre eles, ou pelo menos não se desejaria tê-lo, mas é altamente desejável que fiquem próximos.

	+1	+2	+3	+4	+5	+6
+1		1		1		
+2	1					
+3					-1	
+4	1					
+5			-1			
+6						

Figura 3-6 Informações sobre restrições de adjacências

O sistema AVOLI busca tratar todas estas variáveis. Estas restrições não são do tipo restrição forte. No entanto, uma severa penalidade na sua utilidade total final do leiaute poderá ocorrer se estas situações indesejáveis de adjacências estiverem presentes, diminuindo a qualidade do leiaute, com será explicado adiante. O motivo pelo qual isto não é tratado como uma restrição forte é que pode haver situações impossíveis de disposições dos departamentos ou por excesso de restrições de adjacências, ou por haver poucos departamentos em um pequeno leiaute, não permitindo que um fique afastado um do outro.

Os valores destes parâmetros são entrados pelo usuário através de uma tela de configurações (Figura 3-6). A codificação destes parâmetros é a seguinte:

- *1 (um)* para departamentos que necessariamente devem ficar juntos e serem vizinhos.
- *-1 (menos um)* para departamentos que necessariamente devem ficar longe um do outro, ou seja, não podem ser vizinhos.
- *0 (zero) ou campo vazio* para situações que não faz diferença se o departamento esta perto ou longe do outro. Neste caso só serão considerados os fluxos que existem entre eles para efeito dos posicionamentos.

3.5 Análises de fluxos entre os departamentos

A análise de fluxos é uma metodologia que avalia toda a produção e suas fases levando em conta vários fatores, tais como:

- transporte,
- inspeções,
- demoras,
- armazenamento,
- tipo e características do produto.

Esta análise ajuda a determinar quantos departamentos existirão, suas necessidades e dimensões desejadas, assim como os fluxos entre eles. Ela é feita por um especialista (Engenheiro de produção), que conheça a fundo todos os processos e métodos envolvidos na produção do produto.

Geral	Fluxos	Relacionamentos	Custos	Pré-processamentos	Área										
	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12	+13	+14	+15
+1	xxx	10	0	5	1	0	1	2	2	2	2	0	4	0	0
+2	10	xxx	1	3	2	2	2	3	2	0	2	0	10	5	0
+3	0	1	xxx	10	2	0	2	5	4	5	2	2	5	5	5
+4	5	3	10	xxx	1	1	5	0	0	2	1	0	2	5	0
+5	1	2	2	1	xxx	3	5	5	5	1	0	3	0	5	5
+6	0	2	0	1	3	xxx	2	2	1	5	0	0	2	5	10
+7	1	2	2	5	5	2	xxx	6	0	1	5	5	5	1	0
+8	2	3	5	0	5	2	6	xxx	5	2	10	0	5	0	0
+9	2	2	4	0	5	1	0	5	xxx	0	10	5	10	0	2
+10	2	0	5	2	1	5	1	2	0	xxx	0	4	0	0	5
+11	2	2	2	1	0	0	5	10	10	0	xxx	5	0	5	0
+12	0	0	2	0	3	0	5	0	5	4	5	xxx	3	3	0
+13	4	10	5	2	0	2	5	5	10	0	0	3	xxx	10	2
+14	0	5	5	5	5	1	0	0	0	5	3	10	xxx	4	4
+15	0	0	5	0	5	10	0	0	2	5	0	0	2	4	xxx

Figura 3-7 Informações sobre os fluxos entre os departamentos

Esta análise deverá anteceder os demais processos visto que ela determina todos os fluxos entre os departamentos (Figura 3-7). De posse destes dados, essenciais para a produção e confecção do leiaute, o próximo passo será o preenchimento da tabela de fluxos. Esta tabela será à base do processo de otimização de todas os métodos suportado pelo sistema. O anexo A detalha o seu preenchimento.

3.6 Considerações sobre as distâncias entre os departamentos

Em princípio, aproximar os departamentos que possuem os maiores fluxos entre si, melhorará a performance do processo produtivo. Isto diminuirá o esforço e o tempo do transporte de matérias necessário à produção. Sendo assim, a definição das distâncias entre todos os

departamentos é importante. As formas mais usuais para quantificar as distâncias entre dois pontos são ([Ask93] e [Tam91]):

- distância retangular; e
- distância euclidiana

Exemplo

Seja o ponto $P_a=(x_a,y_a)$ o centro geométrico do departamento a e $P_b=(x_b,y_b)$ o centro geométrico de um segundo departamento b . Pode-se obter a distância entre eles aplicando as seguintes equações:

Para o cálculo distância retangular:

$$d_{ab} = |x_a - x_b| + |y_a - y_b|$$

Equação 3-1 Distância retangular

Para o cálculo distância euclidiana:

$$d_{ab} = \sqrt{(x_a - x_b)^2 + (y_a - y_b)^2}$$

Equação 3-2 Distância euclidiana

Em ([Tam92], [Fur97], [Tam98] e [Chi01]), o cálculo da distância entre departamentos é sempre baseada no centro dos departamentos e aplicada ambas equações mesmo quando as entradas e saídas dos departamentos não são conhecidas. No sistema AVOLI optou-se pela equação que fornece a distância retangular entre os departamentos. Esta opção foi feita considerando o fato que a distância retangular é a mais usada, em particular, quando se busca fazer comparações com outros métodos da literatura.

3.7 Considerações sobre a função objetiva

A função objetiva fornece a utilidade total do leiaute (Figura 3-8). Esta função opera sobre os fluxos e as distâncias entre os departamentos. Deve-se destacar que a utilidade total que informa a qualidade do leiaute final, e dada pela seguinte formula:

$$F = \sum_i^n \sum_j^n f_{ij} d_{ij} * 2^{(P+A_p+A_L)}$$

Equação 3-3 Função objetiva

Onde

F é a função objetiva que fornece as utilidades totais do leiaute, que deve ser minimizada ao máximo;

f_{ij} é o fluxo conhecido entre o departamento i e o departamento j ;

d_{ij} é a distância retangular centro a centro entre os departamentos i e j ;

P é o número de departamentos não posicionados no interior do leiaute. Isto pode ocorrer quando por motivos de aspecto e/ou impossibilidade de redução de certo departamento, e conseqüentemente o sistema não consegue posicionar um ou mais departamentos no interior do leiaute. O resultado será fortemente penalizado.

A_p é o número de violações por adjacências, onde dois departamentos deveriam ficar distantes, porém estão pertos. Aqui a utilidade também é fortemente comprometida.

A_L é o número de violações por adjacências, onde dois departamentos deveriam ficar próximos, no entanto, estão distantes. A função objetiva é fortemente penalizada.

A otimização consiste tentar diminuir ao máximo o valor dado por esta função objetiva. Para tanto, vários posicionamentos são experimentados de várias formas no interior do leiaute sempre respeitando todas as restrições fortes, que são:

- a razão de aspecto do departamento deve necessariamente estar dentro do limite aceitável. I.e., deve estar contido no intervalo da razão mínima até a razão máxima definida previamente.

$$Aspecto_{\min.} \leq Aspecto_{dep.} \leq Aspecto_{\max.}$$

Onde

$Aspecto_{\min.}$ é a razão de aspecto mínimo previamente definido para o departamento;

$Aspecto_{dep.}$ é a razão efetivamente usada pelo departamento;

$Aspecto_{máx.}$ é a razão de aspecto máximo previamente definida para o departamento.

- o departamento jamais poderá ser posicionado em um local do leiaute que o faça perder uma parte de sua área útil por sobreposição de áreas ocupadas além daquela permitida e previamente definida.

$$Perdida_{dep.} \leq Perdida_{máx.}$$

Onde

$Perdida_{dep.}$ é quanto o departamento perdeu de sua área por causa de uma sobreposição de áreas ocupadas ou mortas;

$Perdida_{máx.}$ é quanto o departamento pode perder de sua área útil por causa de sobreposição de áreas ocupadas, pré-definido na tabelas das informações sobre os departamentos.

- o departamento pode ser reduzido, para um melhor posicionamento no interior do leiaute, dentro de um limite de reduções pré-estabelecido. Esta redução só será feita se necessário.

$$redução_{dep.} \leq redução_{máx.}$$

Onde

$redução_{dep.}$ é quanto o departamento teve sua área útil reduzida do inicialmente desejado;

$redução_{máx.}$ é o máximo de redução permitida para o departamento pré-definida na tabelas de características dos departamentos.

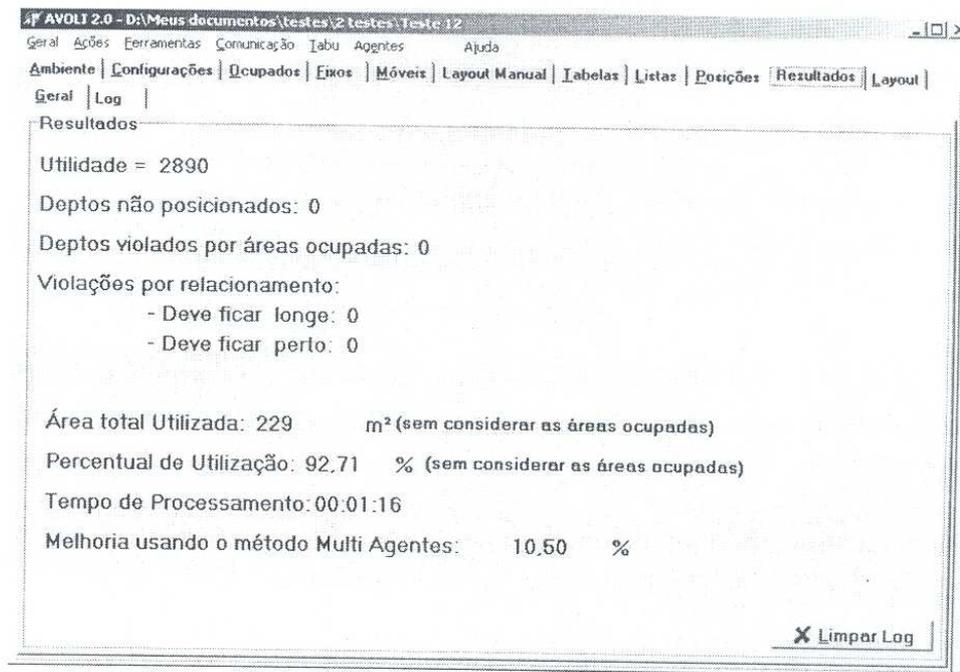


Figura 3-8 Visualização dos resultados

3.8 Algoritmo básico do AVOLI

A discussão que se segue explica como funcionam os algoritmos que serviram de base para a implementação do método de otimização de leiautes empregando um modelo multi-agente. Estes algoritmos, que são o coração do sistema, são comuns a todas os métodos construtivos e métodos de otimização de leiautes que o sistema AVOLI suporta ou poderá suportar para futuras expansões. Apresenta como funcionam as listas construtivas dos departamentos, diversos pré-processamento e o pesado algoritmo de construção realista.

3.8.1 Métodos de criação da lista de departamentos

A construção de leiautes começa de definição dos departamentos e seus fluxos. Estes departamentos podem ser móveis ou fixos. Estes últimos são representados, no sistema AVOLI, organizados em uma lista de números inteiros positivos e negativos especificando respectivamente os departamentos móveis e fixos. Esta lista possui também outras propriedades, tais como:

- tamanho, que corresponde a soma do número de departamentos móveis e fixos;
- histórico, reúne as n melhores listas obtidas com suas respectivas utilidades— n é configurável (Figura 3-9);

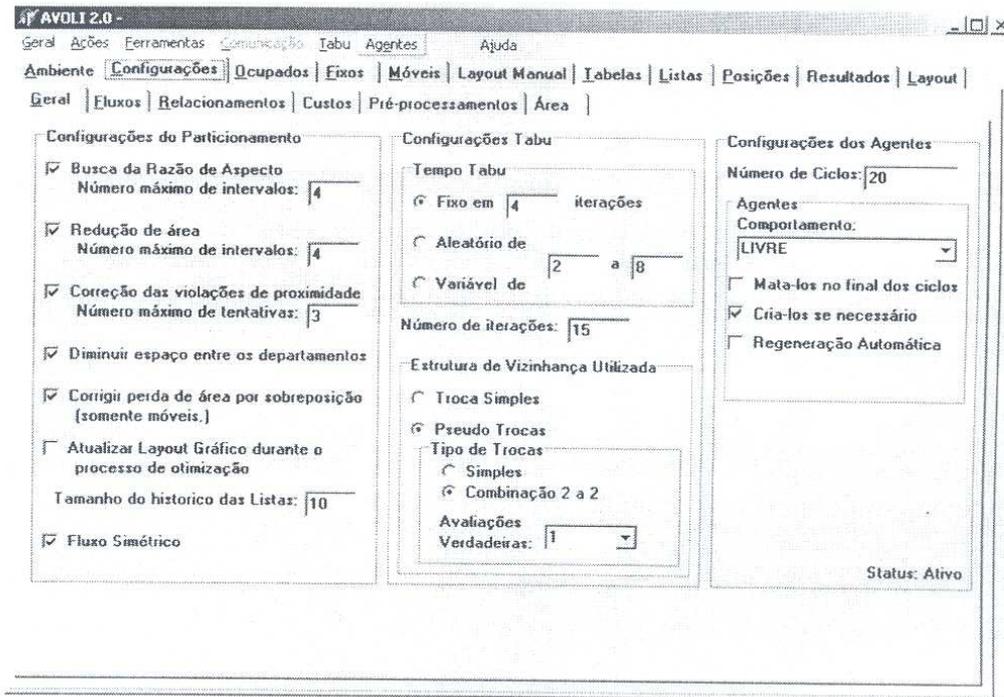


Figura 3-9 Configurações gerais do sistema

É importante frisar que todos os métodos de construção e otimização de leiautes operaram sobre esta lista de construção. Esta lista pode ser obtida inicialmente de três formas, que são:

- método manual;
- método aleatório; e
- método automático.

O algoritmo de construção a partir da lista de departamentos (Figura 3-10), monta o leiaute. Ele começa posicionando todos os departamentos fixos no leiaute e na seqüência aplica um processo análogo para posicionar os móveis. Este segundo momento leva em conta imperativamente todas as restrições e penalidades definidas para cada departamento no momento de sua criação. Este algoritmo será detalhado posteriormente.

3.8.1.1 Método manual

A inserção da lista de construção é feita manualmente no sistema. Nenhum método baseado em recursos computacionais é empregado. O usuário constrói o leiaute a partir de suas

convicções ou conhecimentos. Aplica-se, na seqüência, um algoritmo para determinar e posicionar efetivamente todos os departamentos no leiaute. Calcula-se então sua utilidade total usando a fórmula da função objetiva do item 3.7.

Utilidade	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
2596	+11	+9	+8	+1	+2	+10	+6	+14	+3	+13	+7	+12	+15	+5	+4
2596	+11	+9	+8	+1	+2	+10	+6	+14	+3	+13	+7	+12	+15	+5	+4
2598	+12	+11	+9	+5	+14	+15	+10	+8	+7	+13	+3	+2	+1	+6	+4
2606	+8	+11	+9	+12	+5	+6	+3	+1	+4	+13	+14	+2	+7	+15	+10
2618	+11	+8	+2	+1	+4	+10	+6	+15	+14	+5	+9	+12	+13	+7	+3
2618	+11	+8	+9	+12	+5	+6	+3	+1	+4	+13	+14	+2	+7	+15	+10
2623	+8	+11	+9	+5	+12	+10	+3	+1	+4	+13	+14	+2	+7	+6	+15
2635	+11	+8	+9	+5	+12	+10	+3	+1	+4	+13	+14	+2	+7	+6	+15
2641	+10	+12	+9	+11	+14	+2	+3	+1	+13	+6	+15	+7	+5	+8	+4
2645	+8	+11	+9	+12	+5	+6	+3	+15	+2	+7	+14	+1	+10	+4	+13
2653	+11	+8	+9	+12	+5	+6	+3	+1	+4	+13	+14	+2	+7	+10	+15

Figura 3-10 Lista de construção

3.8.1.2 Método aleatório

Inicialmente, obtém-se um valor α , gerado a partir de uma função de geração de números aleatório. Gera-se α listas de modo que todos os departamentos fixos e móveis sejam inseridos nas mesmas. Aplica-se, sobre cada lista, o algoritmo de construção de leiaute e calcula-se sua utilidade total. Escolhe-se então a lista portadora da melhor utilidade total e marca-a como a solução inicial.

3.8.1.3 Método Automático

A construção da lista se dá pela ordenação dos departamentos e seus fluxos. Este método de construção segue um conjunto de passos (Figura 3-11), que são:

- identificar o maior fluxo e os departamentos associados ao mesmo;
- inserir tais departamentos no início da lista;

- identificar o próximo maior fluxo e está ligado ao último departamento já inserido na lista e o departamento associado ao mesmo;
- inserir tal departamento na próxima entrada da lista;
- repetir o terceiro e quarto passo até todos os departamentos sejam inseridos na lista.

Com a lista criada, aplica-se o algoritmo de construção e calcula-se a função objetiva.

Por departamento				Por Fluxo			
Num.	Dep.	Dep.	Fluxo	Num.	Dep.	Dep.	Fluxo
1	+1	+2	5	1	+5	+6	10
2	+1	+3	2	2	+4	+5	5
3	+1	+4	4	3	+1	+2	5
4	+1	+5	1	4	+1	+4	4
5	+1	+6	0	5	+2	+3	3
6	+2	+3	3	6	+2	+6	2
7	+2	+4	0	7	+1	+3	2
8	+2	+5	2	8	+4	+6	2
9	+2	+6	2	9	+2	+5	2
10	+3	+4	0	10	+1	+5	1
11	+3	+5	0	11	+3	+5	0
12	+3	+6	0	12	+3	+6	0
13	+4	+5	5	13	+1	+6	0
14	+4	+6	2	14	+2	+4	0
15	+5	+6	10	15	+3	+4	0

Figura 3-11 Departamentos organizados por fluxos para a construção da lista automaticamente

É importante destacar que tal ordenação dos departamentos na lista pode influenciar diretamente o posicionamento dos mesmos no leiaute. I.e. pode haver situações onde os departamentos serão posicionados fora do leiaute considerando apenas a execução dos passos supracitados. Tem-se notado, no entanto, que para tais casos, se os departamentos com as maiores dimensões forem posicionados em primeiro, então os departamentos menores possuem mais chances de serem posicionados nas áreas desocupadas.

Se o algoritmo construtivo não conseguiu posicionar todos os departamentos no interior do leiaute, aplica-se então a seguinte heurística:

- seleccionar o departamento com a maior área e o insere no início da lista— empurrando os demais departamentos para uma posição a direita da lista até a posição em que se encontrava o departamento que se encontra no início da lista.
- tentar re-posicionar todos os departamentos;

- caso não tenha sido possível posicionar todos os departamentos no interior leiaute, selecionar o segundo maior departamento em área e inseri-lo na segunda posição da lista;
- repetir o processo até que todos os departamentos sejam posicionados na lista e respeite os limites do leiaute ou não que exista mais departamentos a serem deslocados para o começo da lista (para a próxima posição da lista)
- sugerir e modificar as dimensões iniciais do leiaute, se necessário e permitido. O leiaute terá assim um de seus lados aumentado alternadamente de uma unidade e todo o processo será relançado desde a ordenação por fluxo.
- indicar na lista construtiva, marcando com um 'X' na frente da indicação numérica do departamento os que não foram possíveis de posicionar dentro do leiaute. Na seqüência, calcular-se-á a utilidade total, atribuindo uma forte penalidade para cada departamento não posicionado e violações de adjacências. Isto resultará em uma solução extremamente ruim devido a não inclusão de todos os departamentos no leiaute ou violação de adjacências.

As modificações nas dimensões do leiaute são sugeridas, alternadamente, iniciando pelo comprimento e depois pela largura. Este processo se repete até que seja encontrada uma solução inicial viável em que todos os departamentos estejam posicionados no interior do leiaute.

3.8.2 Método de construção de leiaute do sistema AVOLI

Este é método efetua, com relação àqueles apresentado anteriormente um número maior de atribuições e conseqüentemente exigindo um esforço computacional adicional. Ele parte da lista construída previamente por qualquer método construtivo ou método de otimização. É importante salientar que todos os métodos construtivos e métodos de otimização (SMA e TB) operam diretamente sobre a lista construtiva e nunca sobre o leiaute diretamente.

O algoritmo de construção⁵ implementa as seguintes tarefas:

- posicionar no interior do leiaute todos os departamentos fixos e áreas ocupadas;
- encontrar uma posição válida no interior do leiaute para cada departamento móvel de acordo com a lista de construção, sempre tentando fazer com que o departamento

⁵ O anexo B apresenta de forma detalhada um fluxograma deste algoritmo.

móvel que está à direita na lista construtiva seja vizinho do departamento móvel ou fixo que está à esquerda da mesma lista. Ser vizinho significa que um departamento esteja encostado diretamente ao outro em qualquer ponto de seus limites ou bordas. Para que isto aconteça, se necessário, o algoritmo de construção pode: (a) fazer variar a razão de aspecto do departamento dentro dos seus limites mínimo e máximo; (b) efetuar diversas reduções de áreas totais dos departamentos até o limite máximo. Estas reduções de áreas são realizadas inicialmente de um modo individual para cada departamento. Não encontrado o posicionamento desejado para o departamento, faz-se então uma redução global levando em conta todos os departamentos;

- diminuir as áreas perdidas entre os departamentos (efeito dominó) empurrando todos os departamentos para cima e para a esquerda;
- determinar as vizinhanças para todos os departamentos; e
- verificar se não há nenhuma violação por adjacências. Se houver uma violação, tentar corrigi a violação modificando a lista construtiva. O número de tentativas é parametrizável.

Este algoritmo construtivo apresentou excelentes resultados, principalmente depois da implementação do pré-processamento que o tornou surpreendentemente mais rápido. Por exemplo, para posicionar trinta departamentos levou-se um pouco mais de quatro segundos para concluir o processo sobre um Pentium III a 600MHz. Estes trinta departamentos tinham baixa flexibilidade, i.e., não se permitia muitas variações de áreas e razões de aspectos.

Verificou-se que:

- o tempo é diretamente proporcional ao número de tentativas que o algoritmo faz para posicionar um departamento móvel. I.e., se a área do leiaute for de tamanho adequado e as razões de aspecto e reduções do departamento móvel forem bem amplas, existe a possibilidade de que o algoritmo não experimente um número elevado de combinações.
- o número de passos possíveis para mudar a razão de aspecto e reduzir a área de um departamento influencia diretamente no tempo de resposta. No entanto, quanto maior o número de passos, maior será a chance de se obter uma solução final refinada e precisa, i.e., com um bom aproveitamento da área total do leiaute. Os melhores resultados foram obtidos com um número médio de quatro passos, tanto para a razão de aspecto como para a redução de área.

- a definição da área total do leiaute depende de um conjunto de variáveis, que inclui as reduções possíveis e as razões de aspecto. Deve-se frisar que raramente ocorre um aproveitamento de cem por cento da área total do leiaute. Recomenda-se, a partir de algumas experimentações, que a soma total das áreas dos departamentos móveis e fixos não devem ultrapassar noventa e cinco por cento da área total do leiaute.
- os departamentos fixos, caso não sejam bem posicionados, provocarão uma perda importante da área de leiaute. Por exemplo, as áreas livres entre departamentos fixos e bordas em um leiaute dificilmente são aproveitadas.
- às áreas ocupadas e os departamentos móveis podem perder parte suas áreas úteis com sobreposições. É importante notar que o método assegura que não haverá perdas de área superior as perdas com sobreposições previamente acordadas. Todavia, tal observação vale apenas para os departamentos móveis, sabendo que o posicionamento dos departamentos fixos é de inteira responsabilidade do usuário. O sistema se limitará a informar o quanto o departamento fixo perderá de sua área total por sobreposição de áreas ocupadas.

3.8.3 Pré-processamento

As primeiras versões do AVOLI não havia pré-processamento. Os cálculos de todas as variações eram feitos no interior do próprio algoritmo de construção. Esta estratégia não era boa. Ela fazia com que o mesmo cálculo fosse feito centenas de vezes, onerando o processamento. Os cálculos de redução de área dos departamentos e as suas razões de aspecto com todas as suas variações eram as que mais contribuíam para a redução da performance do sistema. Criou-se então uma rotina de pré-processamento separa do algoritmo de construção (Figura 3-12). O tempo de execução de tal pré-processamento é menor que um segundo.

O pré-processamento cria matrizes dinâmicas com todas as combinações possíveis de razão de aspecto e redução para que o algoritmo de construção possa usar. O próprio algoritmo de construção ficou sensivelmente menor e mais eficiente por causa deste pré-processamento.

AVOLI 2.0 - D:\Meus documentos\testes\4 testes\Teste 55 - 5deptofixos - Socup - 10h1 (h - AVOLI 2.0) - 10/10/2004 10:00:00

Geral Ações Ferramentas Configuração Tabu Agentes Ajuda

Ambiente Configurações Ocupados Fixos Móveis Layout Manual Tabelas Listas Posições Resultados Layout

Geral Fluxos Relacionamentos Custos Pré-processamentos Área

Departamentos Móveis Departamentos Fixos Fluxos X Custos

Comprimento Largura

	P. 1	P. 2	P. 3	P. 4
+1	12	11	11	10
+2	9	9	9	9
+3	8	8	8	7
+4	11	10	9	9
+5	12	11	11	11
+6	8	7	7	6
+7	5	5	4	4
+8	6	6	6	6
+9	11	12	12	12
+10	15	12	12	13
+11	8	8	8	8
+12	4	4	4	4
+13	5	4	4	4
+14	6	6	5	5
+15	8	8	8	8
+16	5	4	4	4
+17	10	7	7	8
+18	5	4	4	3
+19	9	9	9	9
+20	10	10	10	10
+21	9	7	6	5

Calcular

Redução Nível: 1 << >>

Figura 3-12 Visualização do Pré-Processamento

3.9 Posições e informações sobre departamentos

A Figura 3-13 mostra diversas informações, tais como:

- posição final dos departamentos;
- quanto de redução ou aumento cada departamento teve sobre sua área (devido ao arredondamento dos lados de sua área),
- quanto de área útil cada departamento perdeu devido à sobreposição de áreas ocupadas, etc.

	Xinicio / Xfim	Yinicio / Yfim	Larg.	Comp.	Xcentro	Ycentro	Utilidade	RA Usada	Redução (%)	Ocupado (%)
-1	28,0 / 37,0	1,0 / 6,0	6,00	10,00	32,50	3,50	3905	0,6	0,000	0,000
-2	69,0 / 90,0	36,0 / 56,0	21,00	22,00	79,50	46,00	5400,5	1,0	-2,667	0,000
-3	41,0 / 59,0	1,0 / 19,0	19,00	19,00	50,00	10,00	4175	1,0	-0,278	0,000
-4	1,0 / 15,0	47,0 / 60,0	14,00	15,00	8,00	53,50	4626,5	0,9	-5,000	0,000
-5	42,0 / 51,0	38,0 / 45,0	8,00	10,00	46,50	41,50	2531	0,8	0,000	0,000
+1	17,0 / 28,0	15,0 / 23,0	9,00	12,00	22,50	21,00	3866,5	0,8	-8,000	0,000
+2	29,0 / 37,0	16,0 / 24,0	9,00	9,00	33,00	21,00	4958	1,0	-1,250	0,000
+3	66,0 / 73,0	22,0 / 28,0	7,00	8,00	69,50	30,00	5585,5	0,9	-12,000	0,000
+4	48,0 / 58,0	27,0 / 32,0	6,00	11,00	53,00	29,50	4049	0,5	-10,000	0,000
+5	5,0 / 16,0	35,0 / 44,0	10,00	12,00	10,50	40,50	5514	0,8	0,000	0,000
+6	56,0 / 63,0	22,0 / 26,0	5,00	8,00	59,50	24,00	4822	0,6	0,000	0,000
+7	1,0 / 4,0	34,0 / 39,0	5,00	4,00	2,50	36,00	4607	1,3	0,000	0,000
+8	32,0 / 37,0	25,0 / 31,0	7,00	6,00	34,50	29,00	3391	1,2	-5,000	0,000
+9	31,0 / 41,0	45,0 / 58,0	14,00	11,00	36,00	53,50	4759	1,3	-2,667	0,000
+10	17,0 / 31,0	33,0 / 40,0	8,00	15,00	24,00	42,50	2741	0,5	0,000	0,000
+11	32,0 / 39,0	38,0 / 44,0	7,00	8,00	35,50	41,00	3738	0,9	-12,000	0,000
+12	9,0 / 12,0	21,0 / 23,0	3,00	4,00	10,50	22,00	6996,5	0,8	-20,000	0,000
+13	17,0 / 21,0	41,0 / 44,0	4,00	5,00	19,00	42,50	4710	0,8	0,000	0,000
+14	74,0 / 79,0	22,0 / 26,0	5,00	6,00	83,50	29,00	5246,5	0,8	0,000	0,000
+15	81,0 / 88,0	22,0 / 28,0	7,00	8,00	84,50	25,00	4862	0,9	-12,000	0,000
+16	22,0 / 26,0	1,0 / 4,0	4,00	5,00	25,00	4,50	3118,5	0,8	0,000	0,000
+17	16,0 / 25,0	55,0 / 58,0	4,00	10,00	20,50	58,50	6105	0,4	0,000	0,000
+18	85,0 / 89,0	29,0 / 32,0	4,00	5,00	87,00	30,50	5524	0,8	0,000	0,000
+19	13,0 / 21,0	1,0 / 9,0	9,00	9,00	17,00	5,00	5399,5	1,0	-1,250	0,000
+20	7,0 / 16,0	25,0 / 34,0	10,00	10,00	11,50	29,50	4966	1,0	0,000	0,000

Figura 3-13 Informações diversas dos departamentos

A Figura 3-14 mostra graficamente como ficarão dispostos os departamentos no interior do leiaute.

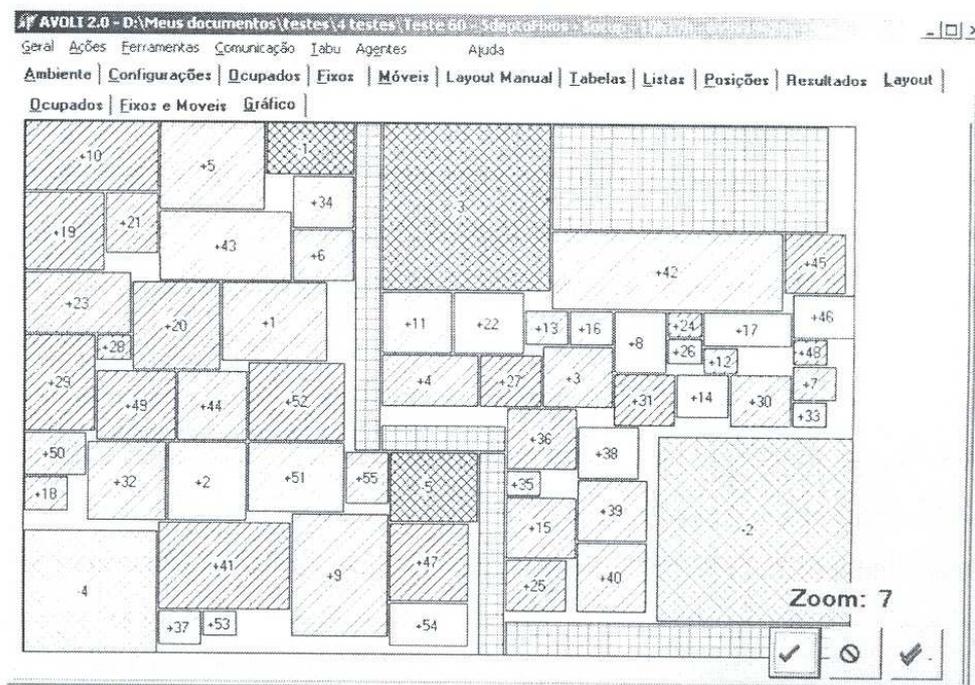


Figura 3-14 Disposição dos departamentos no interior do leiaute

4. Sistema multiagente

Este capítulo apresenta a abordagem empregada para construir os agentes, incluindo como eles percebem o seu mundo, suas crenças e reações. Apresentam-se também os mecanismos de comunicação, como funciona o comportamento dos agentes e quais são suas intenções. Descreve o mecanismo de memória dos agentes e como eles constroem suas listas de sugestões para o mundo. E finalmente explica como o mundo reage às ações dos agentes e suas possíveis contribuições para a melhoria do leiaute.

4.1 Comunicação entre agentes

Um fator essencial numa comunidade de agentes autônomos é as suas capacidades de comunicação e percepção do mundo e seus componentes. Esta comunicação deve ser de fácil compreensão e conhecida por todos os membros. Há a necessidade de uma padronização e uma semântica comum. Esta comunicação deve ser eficiente e rápida para que não haja atrasos nos processos e demasiadas falhas nos dados transmitidos.

Dentre as linguagens de comunicação pode-se destacar KQML (**K**nowledge **Q**uery and **M**anipulation **L**anguage) que possui protocolos para suporte em programação em rede para sistemas baseados em conhecimento ou agentes inteligentes.

O modo de comunicação no AVOLI (na opção multiagente) é do tipo “broadcast”, i.e., as informações são transmitidas através de eventos e lançada simultaneamente a todos os membros do mundo do sistema. Quando este evento é acionado, ele é monitorado por todos os membros, os quais decidem se esta informação lhes interessa ou será simplesmente ignorada.

Formato padrão das mensagens no AVOLI:

```
destinatário;remetente;ação;informaçãol;...;informaçãon
```

onde:

- *destinatário*: corresponde ao receptor da mensagem. O valor WORLD faz com que a mensagem seja enviada para o mundo. O nome de cada agente começa por AG seguido de um número. Este número será positivo para os agentes móveis e negativos para os agentes fixos (e.g., AG7 móvel, AG-3 fixo). O valor ALL faz com que a mensagem seja enviada para todos os agentes do sistema e inclusive para o mundo.
- *remetente*: identificador do emissor da mensagem; segue a mesma sintaxe da campo destinatário. O valor ALL não é permitido.
- *ação*: corresponde a ação ou informação que o remetente deseja transmitir. Cada ação/informação é composta de palavras e um vocabulário bem definido e de rápida interpretação e reconhecimento. As possíveis ações estão descritas no anexo C.
- *informação1;...;informaçãoN*: argumentos da ação.

4.2 Heurísticas usadas pelos agentes

Não existe unanimidade sobre o que é exatamente um agente. Aqui os agentes são sistemas autômatos que tem interesses, crenças, satisfação e comportamentos independentes (Figura 4-1). Os agentes interagem e observam o seu mundo e, que neste caso é o próprio leiaute, tentando promover mudanças sobre o mesmo. Cada departamento é representado por um agente. Cada agente pode ter mais de um comportamento ou estratégia. Ele pode mudá-la em função da situação.

Essencialmente os agentes agem e reagem baseados apenas em seus interesses individuais e sempre estão insatisfeitos com sua situação presente. Comportamento inspirado no mundo real, cujas pessoas, de um modo geral, sempre estão atrás de uma melhor colocação na sociedade. As pessoas de um modo geral estão sempre a procura de locais que consideram melhores. Estes locais normalmente oferecem melhores condições e qualidade de vida.

Baseando-se neste comportamento, os agentes AVOLI, de forma geral nunca estão satisfeitos e cobiçam seus vizinhos, acreditando que eles têm uma melhor colocação do que a sua no mundo – neste caso é o próprio leiaute. Um agente sempre tentará trocar de lugar com outro agente do mundo que tenha, ou pelo menos acredite, uma vizinhança melhor do que a sua. A melhor vizinhança é a que proporciona um maior fluxo com seus vizinhos diretos. I.e., é a soma dos fluxos de todos os departamentos que são diretamente adjacentes ao departamento desejado. Ser vizinho direto significa que existe pelo menos um ponto qualquer da borda do departamento em contato com outro departamento avaliado.

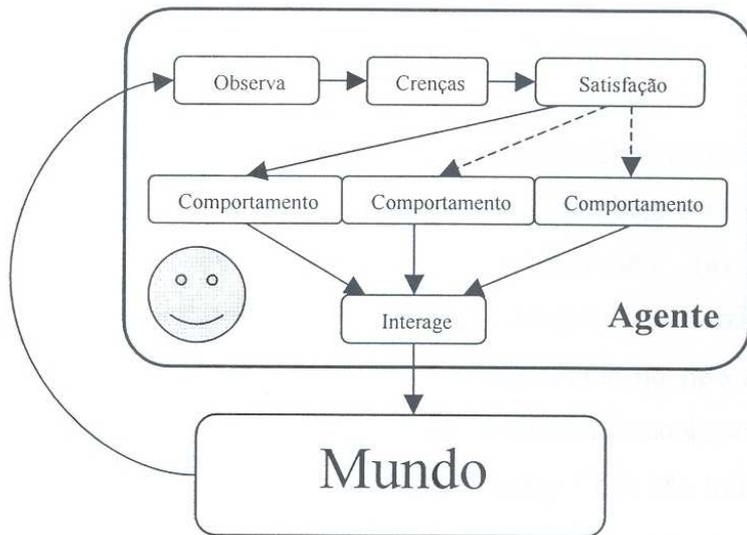


Figura 4-1 Representação de um agente em Blocos

O agente nunca está satisfeito com sua própria posição no mundo mesmo que esta seja melhor do que as outras localidades avaliadas. Desta forma, os agentes sempre tentando mudar, mesmo quando esta mudança piora sua situação presente. Tal comportamento evita que a solução fique em um ótimo local. Deve-se destacar também que na avaliação da vizinhança são considerados os relacionamentos de adjacências desejados e indesejados. Estas restrições de adjacências alteram consideravelmente o resultado final quanto ao desejo de permuta dos agentes. Se Baseado, neste fato, desenvolveu-se um algoritmo que avalia o desejo de troca do agente sobre um determinado local ocupado por outro departamento (ou agente) e cujo maior valor é sempre o mais preferível. O algoritmo é o seguinte:

$$A = \sum_a^n \sum_j^n F_{aj} \cdot V_{dj} + \sum_a^n \sum_j^n R_{aj} \cdot V_{dj} \cdot 1000 - T_d$$

Equação 4-1 Algoritmo que expressa o desejo de troca sobre uma localidade

onde:

- A é a função objetiva que expressa o desejo de troca do agente sobre uma vizinhança desejada d ;

- F_{aj} é o fluxo entre o departamento a , que corresponde ao departamento do agente que procura uma nova vizinhança, e o departamento j , que são todos os outros departamentos do mundo em questão;
- V_{dj} informa se o departamento j é vizinho do departamento desejado d . Estes valores de vizinhanças são dados através de uma tabela fornecida pelo mundo e cujo agente sempre terá uma cópia. Os valores possíveis são: (a) 1 (um) para indicar que j é vizinho de d ; e (b) 0 (zero) para indicar que j não é vizinho de d .
- R_{aj} indica um relacionamento de adjacência do tipo *deve ficar perto* ou *deve ficar longe* entre o departamento a , que corresponde ao departamento do agente que procura uma nova vizinhança, e o departamento j que são todos os outros departamentos do mundo em questão. Este relacionamento de adjacência pode assumir os seguintes valores: (a) 0 (zero) indicando indiferente. Não há nenhuma restrição entre os departamentos quanto se deve ou não ficar perto ou longe; (b) 1 (um) indica que os departamentos em questão devem ser vizinhos e assim ficarem próximos uns dos outros; e (c) -1 (menos um) indica que os departamentos em questão não podem ser vizinhos e devem necessariamente ficar longe uns dos outros.
- T_d é a variável que indica quantas vezes o departamento desejado d foi escolhido com o maior desejo de troca em ciclos anteriores.

A escolha da vizinhança mais desejada é decidida pelo primeiro valor encontrado mais alto na lista de fluxos (Figura 4-2). E para não haver uma repetição constante da mesma troca, foi acrescentado mais uma variável T_d que impede a ocorrência de tal fato. Este valor de T_d reduz o desejo de troca do agente sobre um departamento que já foi alvo de um desejo maior de permuta em ciclos anteriores. Toda vez que um departamento é escolhido como o melhor para permuta, este valor é incrementado em um. Esta variável funciona como uma memória de troca que impede que a mesma troca seja sempre repetida.

A avaliação do desejo de troca sempre terá um valor igual a zero quando ela for feita para o departamento do agente em questão. Isto não impede, no entanto, que a sua própria vizinhança seja escolhida como a mais desejada. Isto acontece principalmente por causa da variável T_d , que pode tornar os valores de desejo negativos para as demais trocas. Um valor negativo na avaliação do desejo pode ser também fruto do relacionamento do tipo *deve ficar longe*, que acarreta fortes perdas de desejos da permuta.

Dp E	Td	Dp1	Dp2	Fluxo
+1	4	+5	+1	-2
+2	3	+5	+2	9
+3	3	+5	+3	3
+4	0	+5	+4	0
+5	0	+5	+5	0
+6	0	+5	+6	2
+7	1	+5	+7	4

Figura 4-2 Comportamentos, fluxos e memória de troca

O agente escolhe, baseado no primeiro maior valor de seu desejo, o departamento que gostaria de trocar de lugar. Escolhido o departamento, o agente troca, na sua cópia da lista de construção, este departamento com o qual ele representa (Figura 4-3). Esta nova lista é enviada para avaliação no mundo, que decidirá se troca ou não de acordo com o interesse geral do mundo.

Após o envio da sua lista desejada para o mundo, o agente passa a observar constantemente se há mudanças no mundo. Quando observada, ele requisitará a nova lista construtiva usada pelo mundo naquele momento e a tabela de vizinhanças (Figura 4-4) que tal lista gerou. De posse destas novas informações, seus desejos de troca são novamente reavaliados. Novas listas de desejos são criadas e enviadas para o mundo e o processo de repete indefinidamente até que ocorra a eliminação do agente.

A eliminação de um agente pode se dar por varias formas. A mais comum é quando o mundo termina o número de ciclos previamente determinado e apresenta os resultados obtidos. O mundo pode eliminar, se assim for configurado, todo o agente em operação. Se assim for e houver novos ciclos, então o agente pode aproveitar sua memória e seus dados, partindo de onde parou. Todavia, se o agente for encerrado de alguma forma, seus dados e conhecimentos serão irremediavelmente perdidos, i.e, não há mecanismo para persisti-lo.

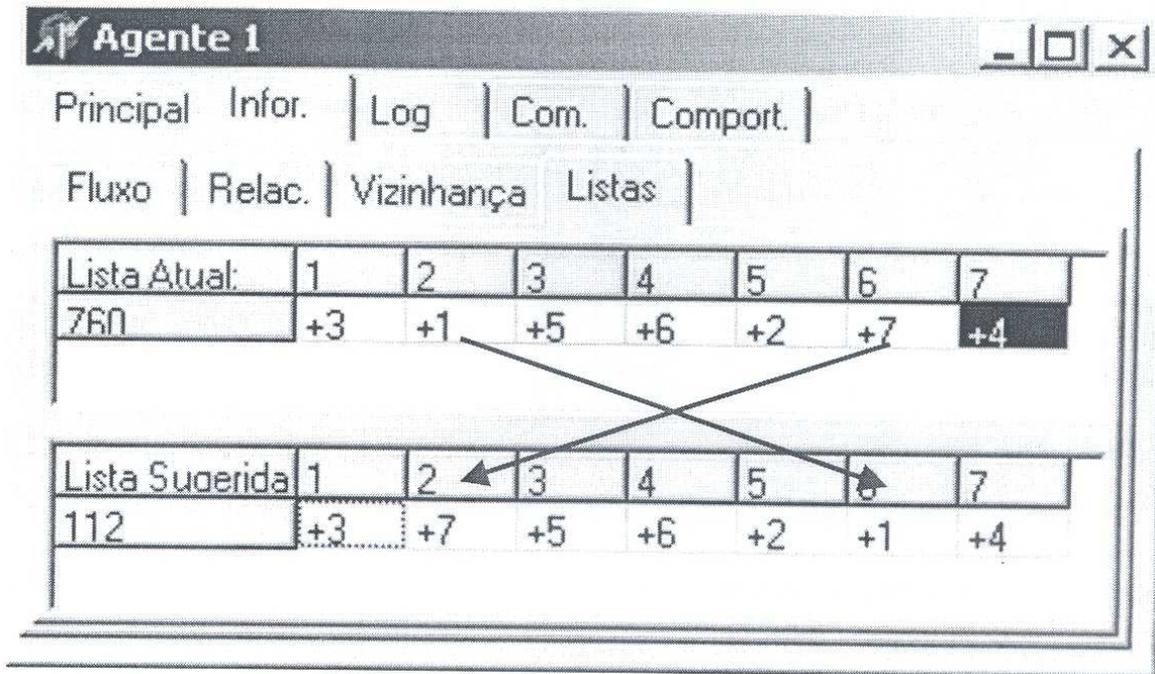


Figura 4-3 Exemplo de lista sugerida do agente 1 para o mundo

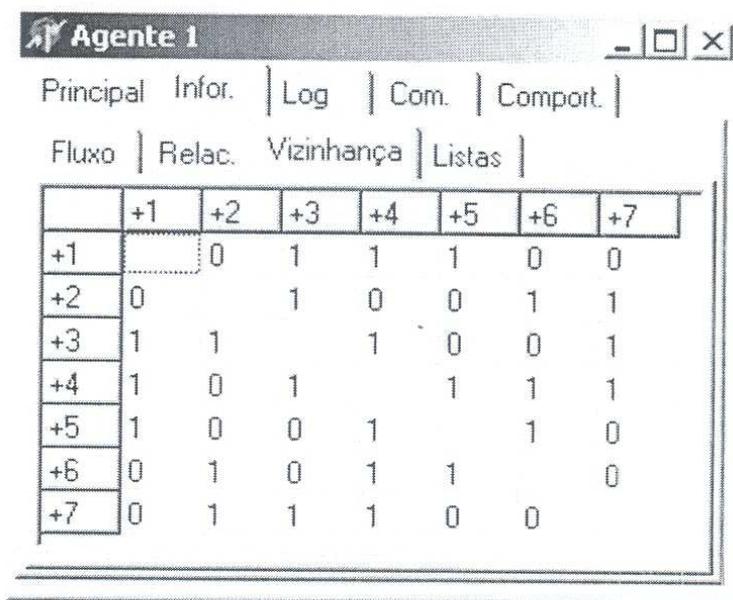


Figura 4-4 Informações que o agente tem da vizinhança atual do mundo

	+1	+2	+3	+4	+5	+6	+7
+1		5	2	4	1	0	0
+2	5		3	0	2	2	2
+3	2	3		1	0	2	5
+4	4	0	1		5	2	2
+5	1	2	0	5		10	0
+6	0	2	2	2	10		5
+7	0	2	5	2	0	5	

	+1	+2	+3	+4	+5	+6	+7
+1		0	0	0	0	0	0
+2	0		0	0	0	0	0
+3	0	0		0	0	0	0
+4	0	0	0		0	0	0
+5	0	0	0	0		0	0
+6	0	0	0	0	0		0
+7	0	0	0	0	0	0	

Figura 4-5 Informações que os agentes tem sobre os fluxos e relacionamentos de adjacências

Os agentes possuem a capacidade de troca de mensagens entre si e também com o mundo (Figura 4-6). Um agente pode matar um outro agente, enviado para tal o comando *KILL*, inclusive para próprio mundo. Uma terceira forma de eliminar um participante do sistema seria simplesmente fechar a sua janela. Isto vale inclusive para o próprio mundo. Entretanto, apenas o mundo tem a capacidade de criar um novo agente.

Agente 5

Principal | Infor. | Log | Com. | Comport.

DE: AG5

Para: WORLD

Mensagem: KILL

Enviar

Figura 4-6 Exemplo de comunicação de um agente

Para um agente desenvolver suas atividades é necessário que ele tenha informações sobre os fluxos e relacionamentos de todos os departamentos de um leiaute (Figura 4-5). Estas informações são informadas ao agente no evento de sua criação. O criador é o mundo. É importante notar que tais informações podem ser alteradas a qualquer momento, mesmo após os ciclos determinados terminarem.

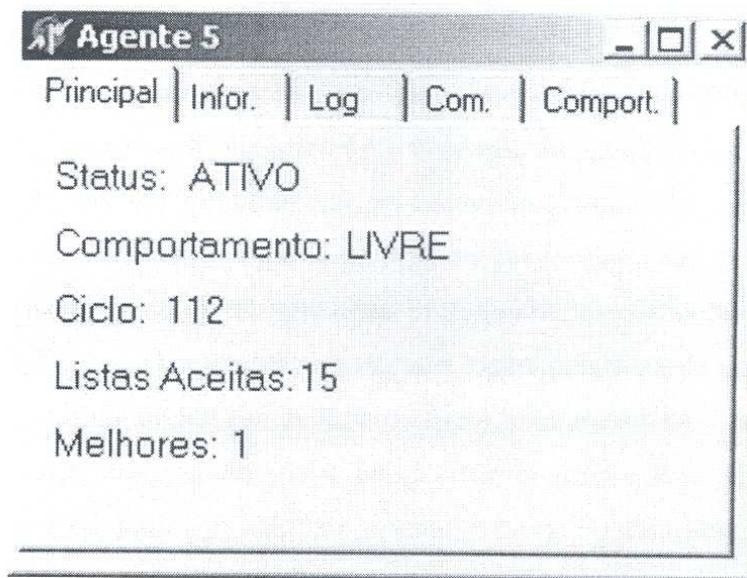


Figura 4-7 Acompanhamento do Agente

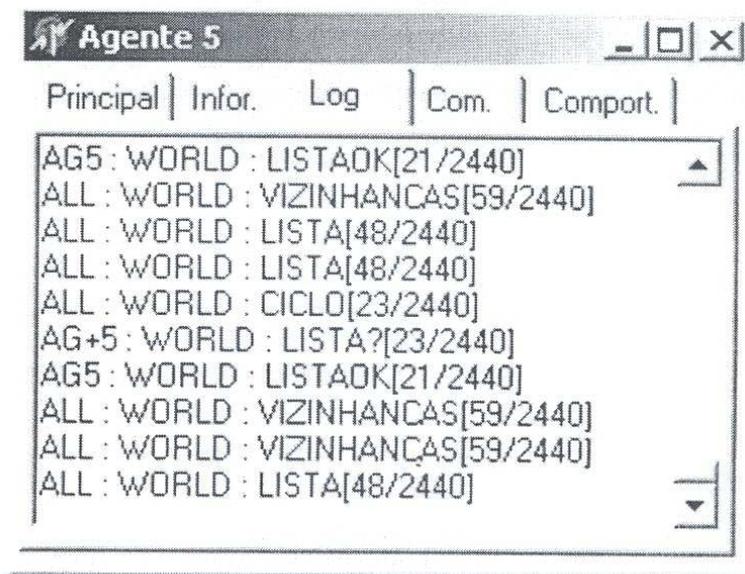


Figura 4-8 Log de comunicação do agente

O sistema AVOLI permite que acompanhe qualquer agente em tempo real (Figura 4-7), tomando conhecimento do seu comportamento. Pode-se também conhecer, se ele está ativo, que ciclo ele está avaliando, quantas listas sugeridas foram aceitas e quantas destas sugestões melhoraram efetivamente o leiaute. É possível também monitorar a sua comunicação (Figura 2-1).

4.3 Ambiente

O mundo ou ambiente também é considerado como um agente. Ele possui sua autonomia, desejos e percebe os outros agentes (Figura 4-9). O mundo é o único que controla diretamente o leiaute e suas mudanças. O comportamento do agente mundo diferente dos demais agentes. Ele sempre age em prol da coletividade e da melhora do leiaute final, e não por interesses individuais. I.e., o agente mundo sempre dará preferência à solução que tenha um maior ganho geral para o leiaute. O agente mundo optará assim pela lista de construção sugerida pelo agente que resultar na melhor utilidade total para o leiaute dentre as listas sugeridas num determinado ciclo. Assim sendo, com base na escolha da melhor lista do ciclo corrente, o agente mundo mudará o leiaute efetivamente, mesmo que piore a solução anteriormente feita. A escolha é dada pela função da utilidade total, que considera não apenas o fluxo, mas também as distâncias de todos os departamentos.

Entre as principais funções do agente mundo pode-se destacar:

- criação de uma solução inicial e sua lista construtiva na qual os agentes trabalharão;
- criação dos agentes no lançamento do sistema e passando-lhes as informações iniciais como ciclo inicial, lista construtiva, fluxos e relacionamentos de adjacências de todos os departamentos;
- atualização das informações sobre as mudanças no mundo, como lista de construção corrente, tabela de vizinhanças atuais e ciclo corrente; e
- avaliação e teste de todas as listas sugeridas pelos agentes e a escolha da melhor dentre elas para efetivamente mudar o leiaute.

O mundo é também responsável por guardar um histórico completo de todas as mudanças feitas e registrar a melhor de todas as soluções até então (Figura 4-10). Esta informação será útil para calcular os ganhos de qualidade do leiaute desde a solução inicial.

Aqui, os instantes são contados em ciclos. O tempo pré-determinado foi criado para atender a situação em que um agente pare de responder por algum motivo qualquer como, por exemplo, se ele encerrou suas atividades ou está com algum problema com sua comunicação. Assim sendo, o sistema como um todo pode continuar a funcionar sem a presença de alguns agentes. O mundo, por sua vez, pode funcionar mesmo sem nenhum agente, apesar de que neste caso, não haverá melhora alguma sobre a solução inicial.

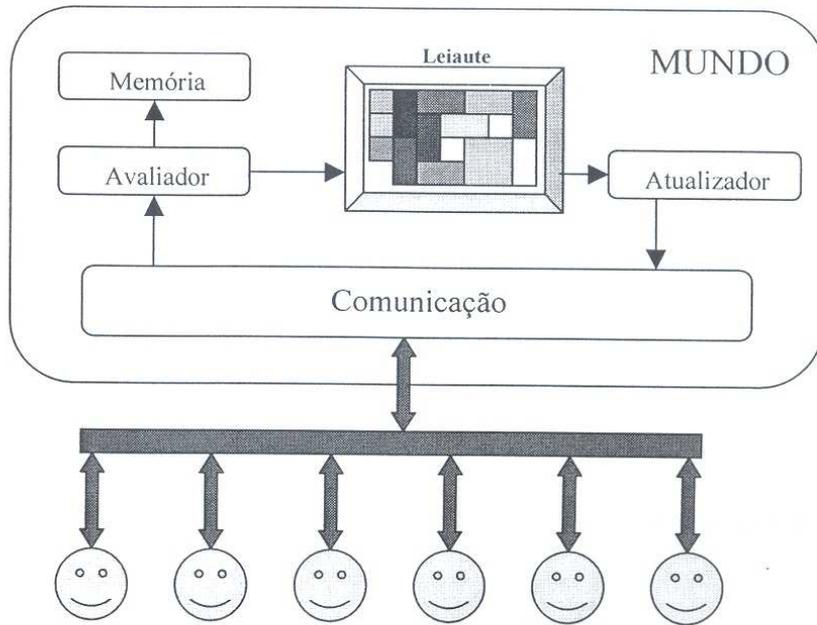


Figura 4-9 Representação em blocos do agente mundo

AVOLI 2.0 - D:\Meus documentos\testes\2 testes\Teste 15

Geral Ações Ferramentas Comunicação Tabu Agentes Ajuda

Ambiente | Configurações | Ocupados | Fixos | Móveis | Layout Manual | Etiquetas | Listas | Posições | Resultados | Layout

Vizinhança | Lista Automática | Força Bruta | Busca Tabu | Agentes

Geral Histórico Gráficos

Ciclo	Melhor	Atual	Agente	1	2	3	4	5	6	7	8	9	10	11	12	13
977	5426	5949	+15	+3	+10	+5	+4	+9	+12	+11	+14	+15	+13	+7	+1	+8
978	5426	5801	+12	+3	+10	+5	+4	+9	+15	+11	+14	+12	+13	+7	+1	+8
979	5426	5801	+8	+3	+10	+5	+4	+9	+15	+11	+14	+12	+13	+7	+1	+8
980	5426	5936	+8	+3	+10	+5	+4	+8	+15	+11	+14	+12	+13	+7	+1	+9
981	5426	5936	+8	+3	+10	+5	+4	+8	+15	+11	+14	+12	+13	+7	+1	+9
982	5426	5936	+15	+3	+10	+5	+4	+8	+15	+11	+14	+12	+13	+7	+1	+9
983	5426	5936	+13	+3	+10	+5	+4	+8	+15	+11	+14	+12	+13	+7	+1	+9
984	5426	5801	+9	+3	+10	+5	+4	+9	+15	+11	+14	+12	+13	+7	+1	+8
985	5426	5801	+9	+3	+10	+5	+4	+9	+15	+11	+14	+12	+13	+7	+1	+8
986	5426	5801	+14	+3	+10	+5	+4	+9	+15	+11	+14	+12	+13	+7	+1	+8
987	5426	5801	+12	+3	+10	+5	+4	+9	+15	+11	+14	+12	+13	+7	+1	+8
988	5426	5961	+12	+3	+10	+5	+4	+9	+15	+11	+14	+13	+12	+7	+1	+8
989	5426	5801	+13	+3	+10	+5	+4	+9	+15	+11	+14	+12	+13	+7	+1	+8
990	5426	5801	+10	+3	+10	+5	+4	+9	+15	+11	+14	+12	+13	+7	+1	+8
991	5426	5801	+9	+3	+10	+5	+4	+9	+15	+11	+14	+12	+13	+7	+1	+8
992	5426	5685	+13	+3	+10	+5	+4	+9	+15	+11	+14	+12	+7	+13	+1	+8
993	5426	5685	+14	+3	+10	+5	+4	+9	+15	+11	+14	+12	+7	+13	+1	+8
994	5426	5685	+11	+3	+10	+5	+4	+9	+15	+11	+14	+12	+7	+13	+1	+8
995	5426	5801	+13	+3	+10	+5	+4	+9	+15	+11	+14	+12	+7	+13	+1	+8
996	5426	5801	+8	+3	+10	+5	+4	+9	+15	+11	+14	+12	+13	+7	+1	+8
997	5426	5801	+9	+3	+10	+5	+4	+9	+15	+11	+14	+12	+13	+7	+1	+8
998	5426	5768	+8	+3	+10	+5	+4	+9	+15	+11	+14	+12	+13	+7	+1	+2
999	5426	5768	+14	+3	+10	+5	+4	+9	+15	+11	+14	+12	+13	+7	+1	+2
1000	5426	6163	+10	+3	+9	+5	+4	+10	+15	+11	+14	+12	+13	+7	+1	+2

Figura 4-10 Histórico completo de todas as mudanças feitas no leiaute pelos agentes

Assim como os agentes, o mundo também tem um dispositivo para evitar que as mesmas trocas sejam escolhidas de um mesmo agente. Esta memória é simplesmente uma variável que indica quantas vezes um determinado agente teve sua lista sugerida e aceita pelo

mundo. Este valor é somado a utilidade total tornando-o cada vez maior, assim sendo, nos ciclos subseqüentes, ele perderá um pouco da sua preferência no intuito de dar oportunidade a outros agentes de efetuarem suas listas de trocas sugeridas. Para tanto se usou a seguinte formula:

$$F_e = F + N_a$$

Equação 4-2 Critério de desempate do Mundo

onde:

- F_e é a função objetiva total que definirá qual a melhor lista sugerida pelos agentes no ciclo atual;
- F é a função objetiva total calculada a partir da lista fornecida por cada um dos agentes; e
- N_a é o número de vezes que um determinado agente já foi escolhido em ciclos anteriores e efetuou uma mudança no mundo.

Assim como em todos os agentes, o agente mundo também pode ser monitorado a qualquer momento no sistema AVOLI. Todas as listas sugeridas pelos agentes podem ser vistas em tempo real (Figura 4-11), bem como outras informações como ciclo atual, utilidade real gerada por cada lista sugerida, i.e., a variável N_a de cada agente.

AVOLI 2.0 - D:\Meus documentos\testes\2 testes\Teste 15

Geral | Ações | Ferramentas | Comunicação | Tabu | Agentes | Ajuda
 Ambiente | Configurações | Ocupados | Fixos | Móveis | Layout Manual | Tabelas | Listas | Posições | Resultados | Layout |
 Vizinhaça | Lista Automática | Força Bruta | Busca Tabu | Agentes |
 Geral | Historico | Gráficos |

Ciclo Atual: 254 Respostas: 0 / 15 Utilidade Inicial: 5426

AG	Na	Utilidade	Ciclo	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
+1	15	5875	254	+8	+11	+9	+14	+13	+1	+15	+6	+3	+12	+7	+5	+10	+4	+2
+2	22	5653	254	+8	+11	+9	+1	+13	+14	+15	+6	+3	+12	+7	+5	+10	+4	+2
+3	13	6612	254	+8	+11	+9	+1	+3	+14	+15	+6	+13	+12	+7	+5	+10	+4	+2
+4	28	449500	254	+8	+11	+9	+1	+13	+14	+15	+4	+3	+12	+7	+5	+10	+6	+2
+5	15	58030	254	+8	+11	+9	+1	+5	+14	+15	+6	+3	+12	+7	+13	+10	+4	+2
+6	17	57210	254	+6	+11	+9	+1	+13	+14	+15	+8	+3	+12	+7	+5	+10	+4	+2
+7	12	52790	254	+8	+11	+9	+1	+13	+14	+15	+6	+3	+12	+4	+5	+10	+7	+2
+8	25	5799	254	+11	+8	+9	+1	+13	+14	+15	+6	+3	+12	+7	+5	+10	+4	+2
+9	29	50570	254	+8	+11	+5	+1	+13	+14	+15	+6	+3	+12	+7	+9	+10	+4	+2
+10	17	5688	254	+8	+11	+9	+1	+13	+14	+15	+6	+10	+12	+7	+5	+3	+4	+2
+11	12	5653	254	+8	+11	+9	+1	+13	+14	+15	+6	+3	+12	+7	+5	+10	+4	+2
+12	19	5905	254	+8	+11	+9	+1	+12	+14	+15	+6	+3	+13	+7	+5	+10	+4	+2
+13	10	51970	254	+8	+11	+9	+13	+1	+14	+15	+6	+3	+12	+7	+5	+10	+4	+2
+14	12	56130	254	+8	+11	+9	+1	+13	+12	+15	+6	+3	+14	+7	+5	+10	+4	+2
+15	8	5908	254	+8	+11	+9	+1	+13	+14	+4	+6	+3	+12	+7	+5	+10	+15	+2

Figura 4-11 Contribuições dos agentes e suas listas sugeridas

5. Testes, Resultados e Comparações

Para verificar a performance do Sistema AVOLI foi tomando como base para comparação alguns problemas relatados na literatura ([CHI01], [NUG68], [MAR02]). No entanto, deve-se salientar que o único destes trabalhos que é realmente similar e possui todas as restrições realistas é o de pseudotrocas usando busca tabu [MAR02]. Vale aqui lembrar que alguns autores consideram tanto o fluxo como refluxo.

Os testes realizados para comparação foram repartidos em três etapas distintas com complexidades crescentes. A primeira bateria de testes usa os QAP, onde todos os departamentos possuem a mesma dimensão e lados iguais a uma unidade, vastamente utilizada na literatura, mas bem pouco ou nada realista.

A segunda bateria de testes usa departamentos de tamanhos variados e razões de aspectos também variadas, contemplando assim uma gama de casos reais. No entanto, a complexidade do problema, devido às variáveis adicionais, é exponencial. Relembarmos que as restrições de área e razão de aspecto dos departamentos jamais são violadas, tornando assim a solução acima de tudo prática e aplicável.

A terceira bateria de testes, possui variações de tamanho e razões de aspectos nos departamentos e também restrições de orientação e restrições de adjacências. As restrições de adjacências podem ser violadas. Se as restrições de adjacências forem violadas então a solução final será fortemente penalizada. Esta penalidade é calculada segundo os critérios adotados na formula de avaliação total do leiaute descrito no Capítulo III.

A força bruta também foi empregada quando possível, i.e., com problemas de até no máximo oito departamentos. Este método é impraticável devido aos vários fatores como uso gigantesco de memória, tempo de processamento, etc.

Os testes finais foram realizados em um computador Atlon K7 de 1.2GHz e 256Mbytes de memória RAM. O sistema AVOLI foi inteiramente desenvolvido em Delphi sobre o sistema operacional Windows, há estudos para convertê-lo para Kylix sobre plataforma Linux.

Todo o leiaute obtido nos testes bem como os seus fluxos, parâmetros dos departamentos (como tipo, área, razão de aspecto, etc.) e listas construtoras encontram-se descritos de forma detalhada no anexo D.

5.1 Usando QAP

Nesta primeira bateria de testes foram usados para fins comparativos os trabalhos propostos por [Chi01], [Nug68] e [Mar02]. Os QAP caracterizam-se por possuírem todos os departamentos iguais no formato de um quadrado de tamanho 1x1. O seu maior problema é que não é um ambiente realista. Sua principal vantagem é dada pela sua simplicidade, o que o torna rápido de ser calculado e posicionado no leiaute. Esta primeira bateria de testes esta dividida em nove fases com um número crescente de departamentos que variam de 5 até 40.

Os resultados obtidos e as comparações com outras metodologias relatados na literatura encontram-se na Tabela 5-1. Pode-se encontrar na Figura 5-1 um gráfico comparativo de tempos aproximados em segundos que se levou para realizar mil ciclos em cada fase dos testes. Nota-se que o tempo é diretamente proporcional ao número de agentes testados.

Pode-se ainda observar o número total de listas construtivas sugeridas por cada agente que foram efetivamente aceitas pelo mundo no período de mil ciclos nos gráficos das Figura 5-2 e Figura 5-3. O número de contribuições aceita pelo mundo de cada agente são bem variadas.

Encontra-se na Figura 5-4 quantas destas listas construtivas sugeridas pelos agentes e que foram aceitas pelo mundo, melhoraram efetivamente a qualidade final do leiaute. Nota-se que as melhorias reais no leiaute são poucas em relação ao número total de listas sugeridas pelos agentes, não ultrapassando seis listas sugeridas por agente.

Número de departamentos	Número total de ciclos	Melhor solução encontrada no ciclo	Solução inicial automática	Solução multi-agentes	Busca Tabu MAR02	Chiang 2001	Nugent 1968	Melhoria da solução inicial
5	100	47	58	50	50	50	50	13,79%
6	100	-	86	86	86	86	86	0%
7	100	5	146	128	128	144	148	4,11%
8	100	90	228	214	214	212	214	6,14%
12	500	278	678	578	578	578	578	13,57%
15	1000	183	1280	1150	1150	1110	1150	10,16%
20	1000	554	3036	2596	2570	2564	2570	14,49%
30	1000	488	7366	6178	6128	6094	6124	15,34%
40	1000	953	16880	14126	16880	-	-	16,32%

Tabela 5-1 Testes comparativos – [Chi01], [Nug68] e [Mar02]

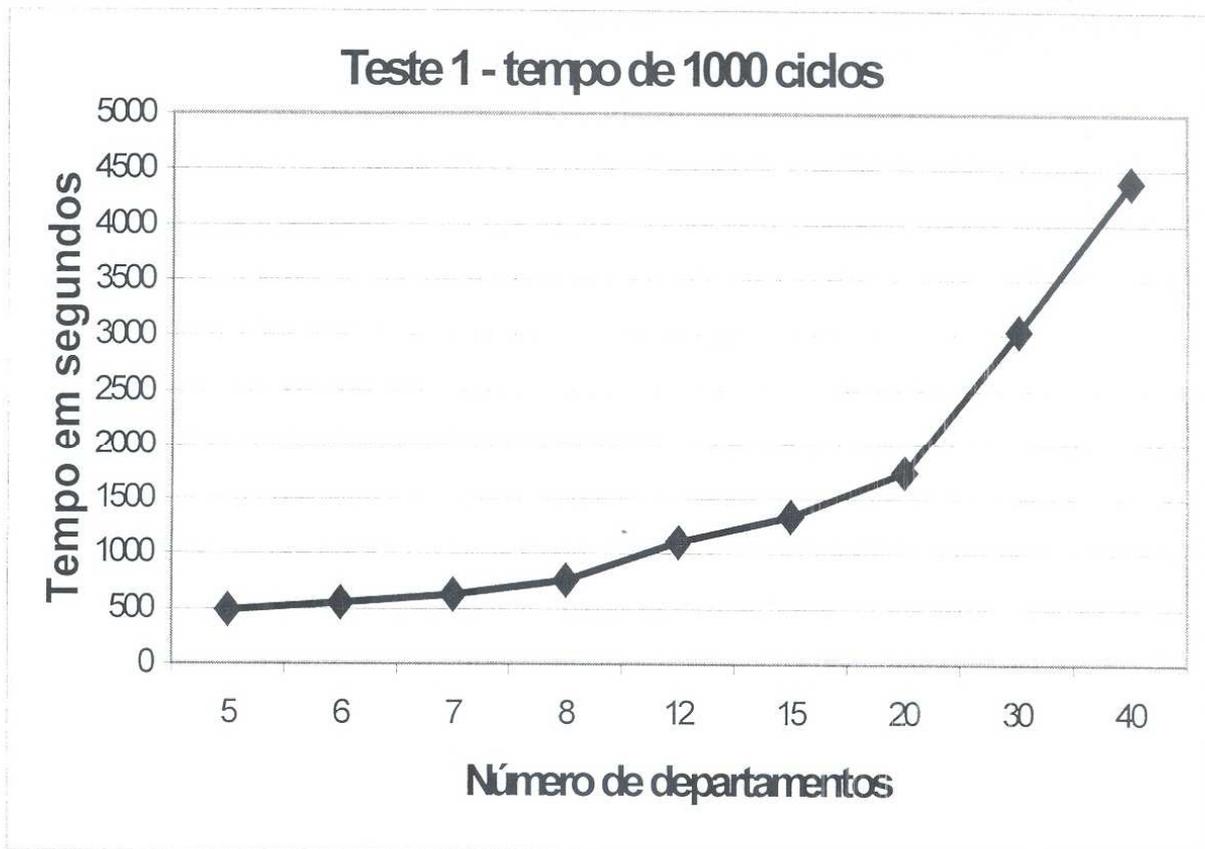


Figura 5-1 Tempo em segundos que levou cada teste da primeira bateria de testes

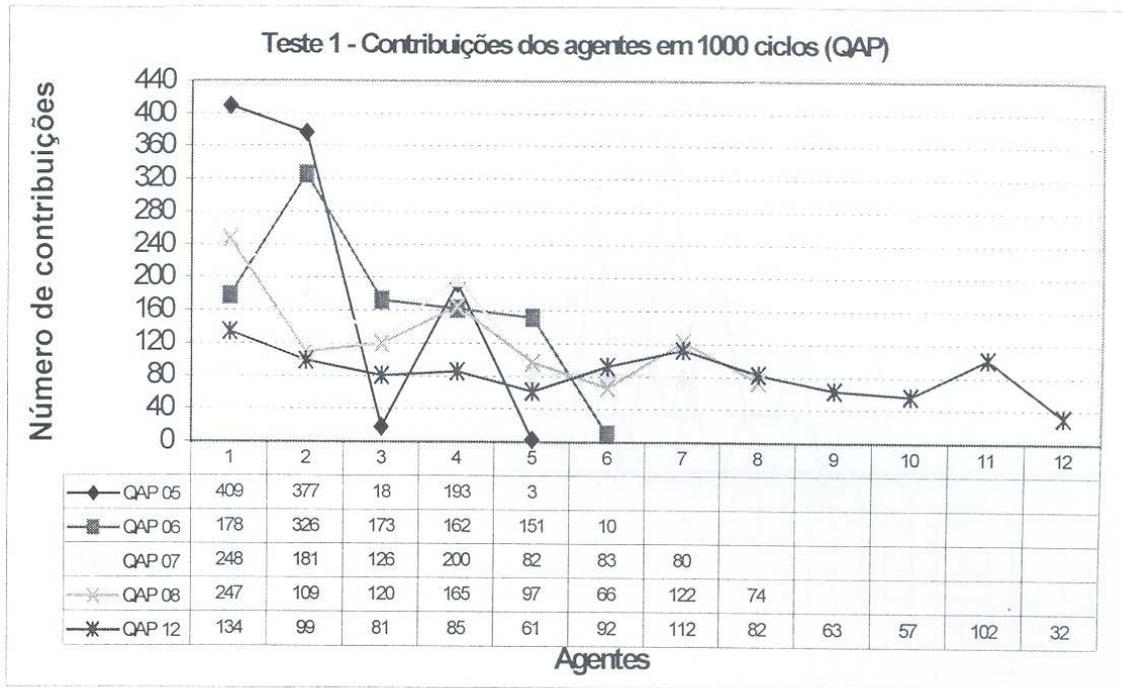


Figura 5-2 Contribuição dos agentes no teste 1 com 5,6,7,8 e 12 departamentos

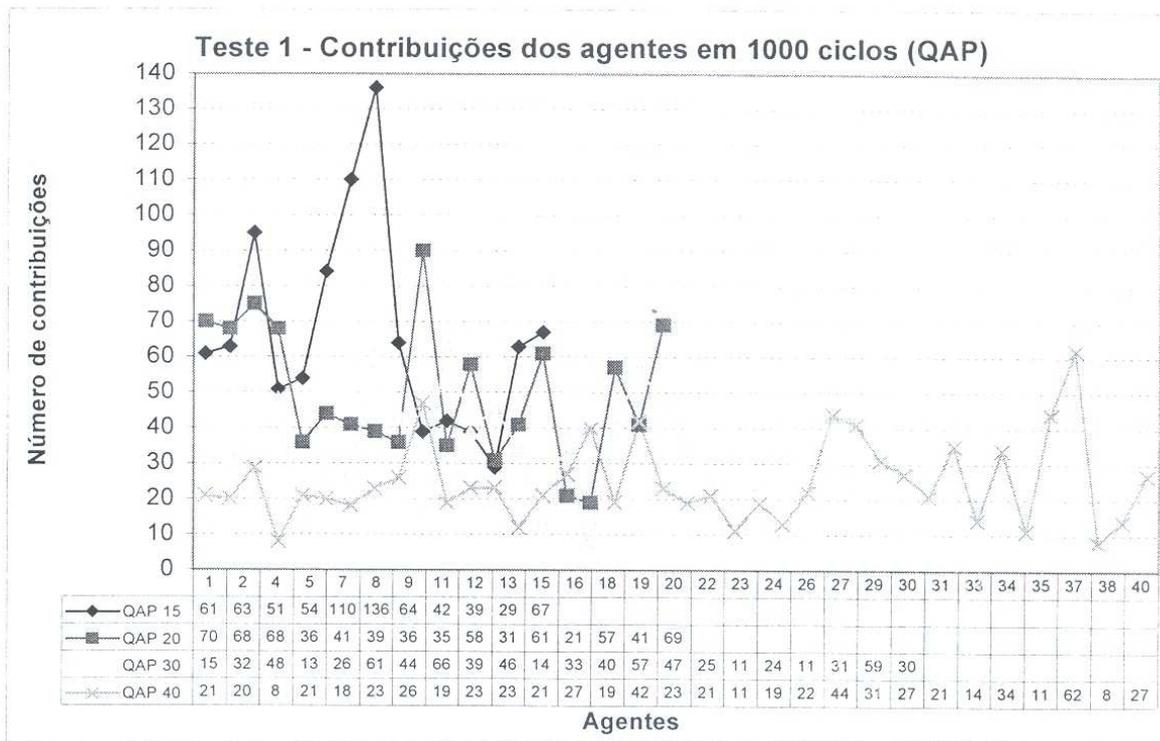


Figura 5-3 Contribuição dos agentes no teste 1 com 15, 20, 30 e 40 departamentos

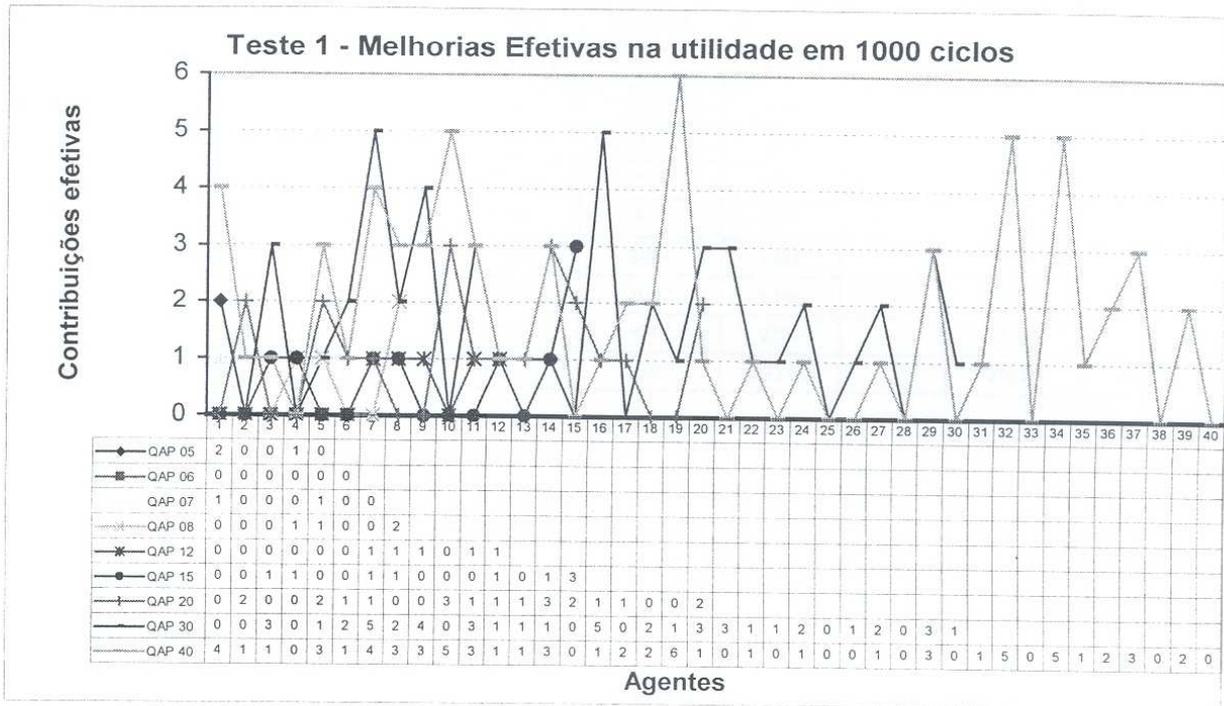


Figura 5-4 Melhorias efetivas na utilidade final do leiaute em 1000 ciclos

5.2 Usando Departamentos de Tamanhos Variados

Nesta bateria de testes formaram utilizadas as matrizes de fluxos dadas por [Nug68], as dimensões dos departamentos dadas por [Tam91] e as dimensões para o leiaute sugerido por [Mar02]. Este teste já contempla mais restrições reais do que o QAP e sua complexidade de combinações possíveis é consideravelmente maior. O resultado obtido e as comparações com outros métodos relatados na literatura podem ser observado na Tabela 5-2. Esta segunda bateria de testes esta dividida em oito fases com um número crescente de departamentos que variam de 5 até 30 departamentos.

Pode-se encontrar na Figura 5-5 um gráfico comparativo de quanto tempo, aproximadamente em segundos, que se levou para realizar mil ciclos em cada fase deste teste. Pode-se ainda observar o número total de listas construtivas sugeridas por cada agente que foram efetivamente aceitas pelo mundo no período de mil ciclos nos gráficos das Figura 5-6 e Figura 5-7. As listas construtivas sugeridas pelos agentes mostraram-se bem variadas indicando que o sistema não ficou em um ótimo local.

Número de departamentos	Número total de ciclos	Melhor solução encontrada no ciclo	Solução inicial automática	Solução multi-agentes	Busca Tabu Mar02	Solução Força Bruta	Tamanho do Leiaute	Melhoria da solução inicial
5	1000	118	272	232	232	232	14x09	14,71%
6	1000	114	456	376	376	376	13x11	17,54%
7	1000	113	898	676	676	676	16x10	24,72%
8	1000	976	1210	974	974	974	18x10	19,50%
12	1000	181	3228	2444	2444	-	19x13	24,29%
15	1000	232	7104	5022	4946	-	20x15	29,50%
20	1000	510	13778	9917	9722	-	21x15	28,03%
30	1000	996	34274	22090	23312	-	25x16	35,55%

Tabela 5-2 Departamentos de tamanhos variados

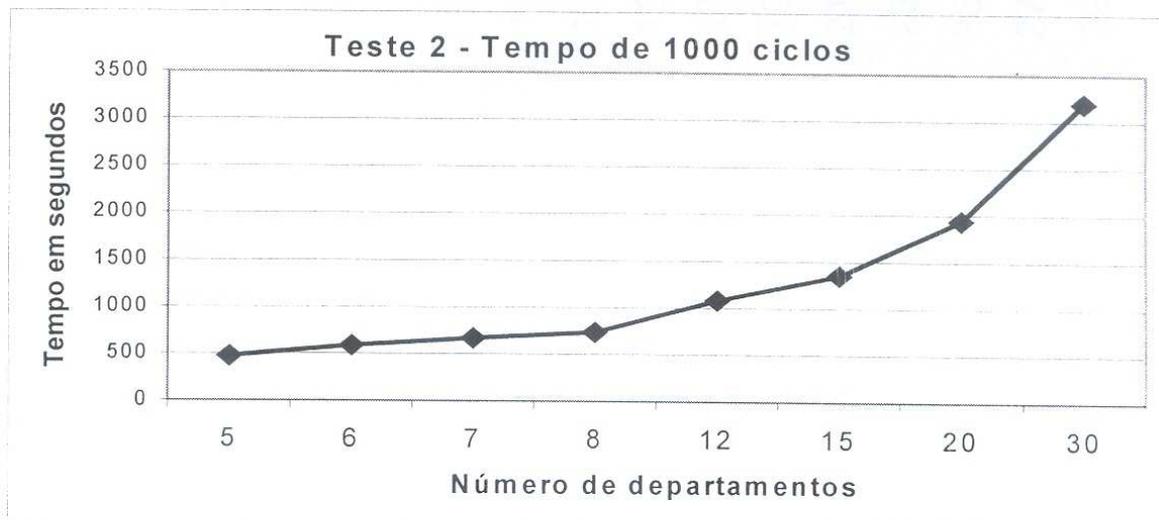


Figura 5-5 Tempo em segundos que levou cada teste da segunda bateria de testes

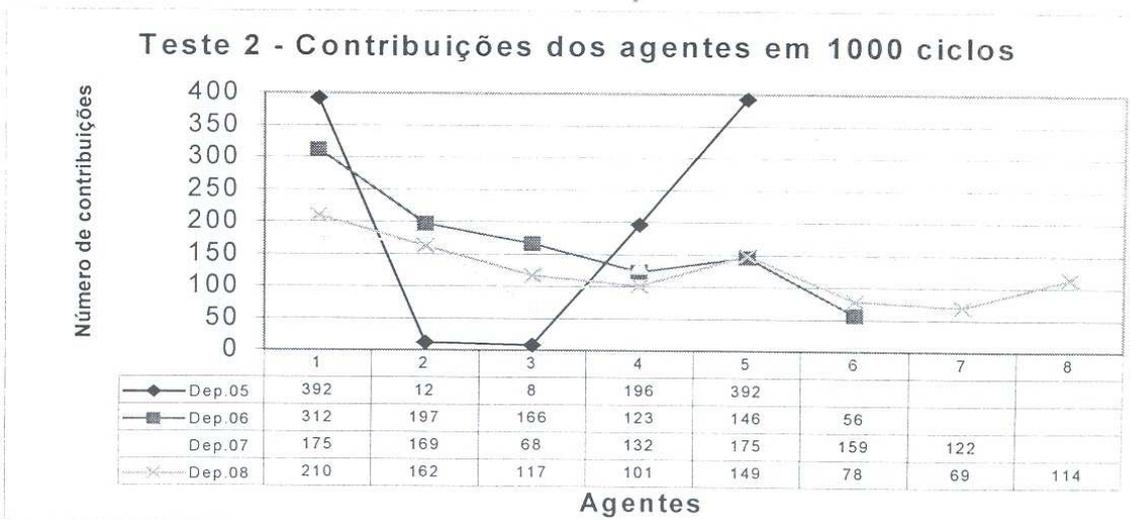


Figura 5-6 Contribuição dos agentes no teste 2 com 5, 6, 7 e 8 departamentos

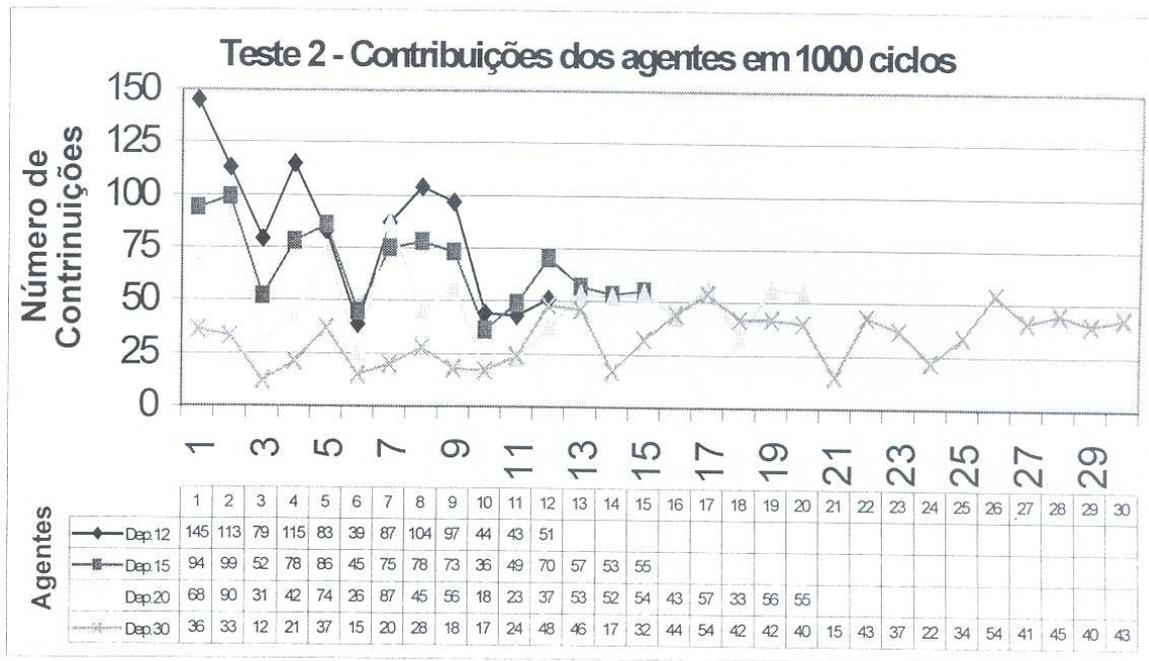


Figura 5-7 Contribuição dos agentes no teste 2 com 12, 15, 20 e 30 departamentos

As Figura 5-8 e Figura 5-9 apresentam os números de listas construtivas sugeridas pelos agentes foram aceitas pelo mundo e melhoraram efetivamente a qualidade final do leiaute. Nota-se que as melhorias reais no leiaute são pequenas em relação ao número total de listas sugeridas pelos agentes, não ultrapassando cinco listas sugeridas por agente.

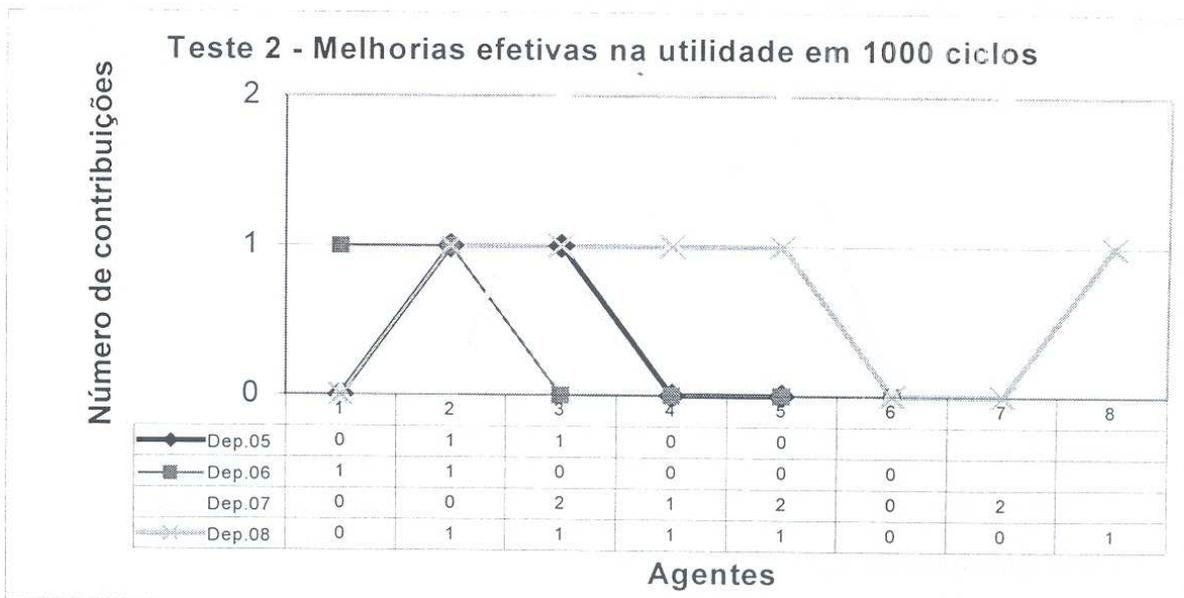


Figura 5-8 Melhorias efetivas na utilidade final do leiaute no teste 2 com 12, 15, 20 e 30 departamentos

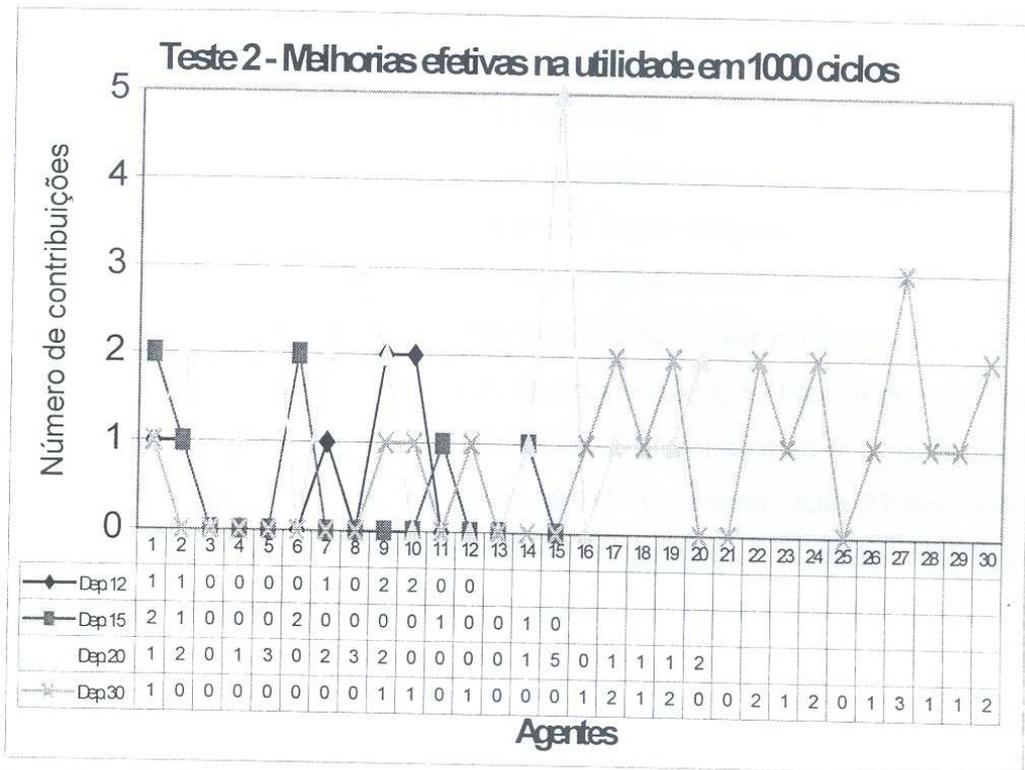


Figura 5-9 Melhorias efetivas na utilidade final do leiaute no teste 2 com 5,6,7 e 8 departamentos

5.3 Usando Departamentos de Tamanhos Variados com Restrições de adjacência e orientação

Nesta bateria de testes acrescentaram-se departamentos fixos (que são indicados pelos seus números negativos) e restrições realistas de adjacências e orientações como sugerido nos testes usados por [Mar02]. Os resultados encontram-se na Tabela 5-3 e as restrições são as seguintes:

- teste com 8 departamentos: os departamentos 3 e 4 devem ficar na vertical. O 7 deve ficar na horizontal;
- teste com 12 departamentos: o departamento 7 deve ficar na horizontal e o 10 deve ficar na vertical. O departamento 1 deve ficar perto do 7. O departamento 3 deve ficar perto do 11 porém longe do 4;
- teste com 15 departamentos: o departamento 6 deve ficar na vertical. O departamento 4 deve ficar longe do 10 e do 15. O departamento 10 deve ficar longe do 15;
- teste com 20 departamentos: o departamento 7 deve ficar na horizontal e o 8 deve ficar na vertical. O departamento 14 deve ficar longe do 8. O departamento 3 deve ficar perto do 13. Este teste possui ainda duas áreas ocupadas;

- teste com 30 departamentos: o departamento 2 deve ficar na horizontal e o 6 na vertical. O departamento 16 deve ficar longe do 21. O departamento 1 deve ficar perto do 20. Este teste possui duas áreas ocupadas e quatro departamentos fixos;
- teste com 40 departamentos: possui 3 áreas ocupadas. Os departamentos 8, 19 e 29 devem ficar na vertical e os 4, 10 e 17 devem ficar na horizontal. O departamento 5 deve ficar longe do 32. O departamento 2 deve ficar perto do 5 e 32;
- teste com 55 departamentos: os departamentos 8, 9, 19 e 29 devem ficar na vertical e os 10 e 17 devem ficar na horizontal. O departamento 9 deve ficar perto do 47. O departamento 15 deve ficar longe do 19. O departamento 23 deve ficar longe do -3 (departamento fixo). O departamento 14 deve ficar longe do -4 (departamento fixo). Este teste possui 5 áreas ocupadas e 5 departamentos são do tipo fixo.

Número de departamentos	Número total de ciclos	Melhor solução encontrada no ciclo	Solução inicial automática	Solução multi-agentes	Busca Tabu Mar02	Solução Força Bruta	Tamanho do Leiaute	Melhoria da solução inicial
8	1000	62	1172	898	898	898	18x10	23,38%
12	1000	775	4174	2657	2744	-	18x13	32,44%
15	1000	18	6870	5270	5306	-	20x14	16,33%
20	1000	913	14502	9673	10338	-	21x15	28,27%
30	1000	591	28660	22787	22188	-	25x16	21,33%
40	1000	893	99546	75620	78024	-	60x42	22,40%
55	1000	951	182154	125393	132828	-	90x60	30,56%

Tabela 5-3 Departamentos de tamanhos variados com restrições e áreas ocupadas

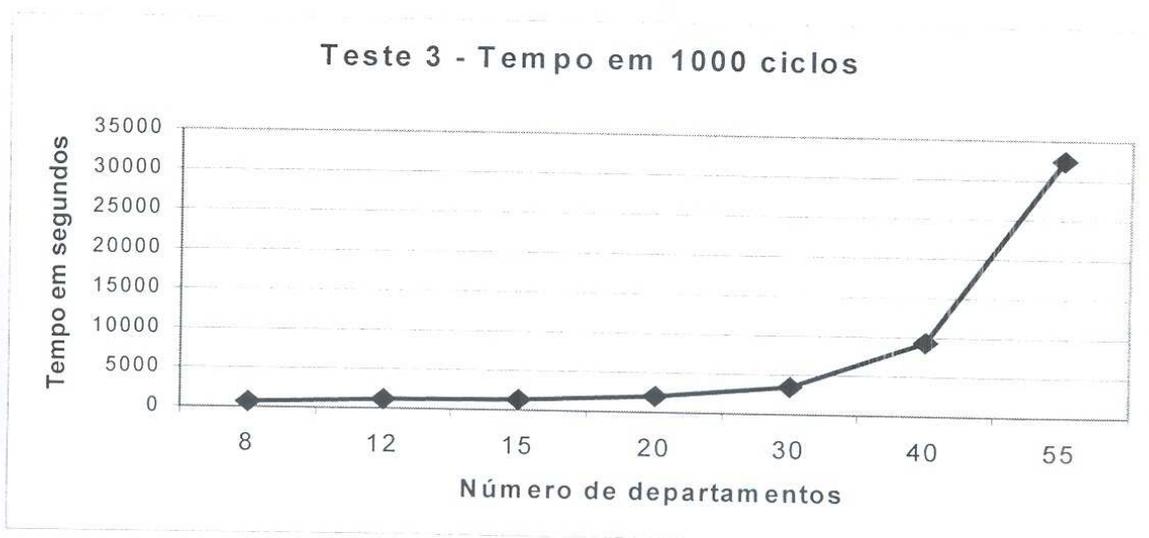


Figura 5-10 Tempo em segundos que levou cada teste da terceira bateria de testes

Pode-se encontrar na Figura 5-10 um gráfico comparativo de quanto tempo, aproximadamente em segundos, que se levou para realizar mil ciclos em cada fase deste teste. As Figuras 5-11, 5-13, 5-15 e 5-17 mostram o número total de listas construtivas sugeridas por cada agente que foram efetivamente aceitas pelo mundo no período de mil ciclos. Estas listas se mostraram bem variadas evidenciando assim que o sistema não ficou em um ótimo local.

Encontra-se nas Figuras 5-12, 5-14, 5-16 e 5-18 quantas destas listas construtivas sugeridas pelos agentes foram

aceitas pelo mundo e que melhoraram efetivamente a qualidade final do leiaute.

Nota-se que as melhorias reais no leiaute são pequenas em relação ao número total de listas sugeridas pelos agentes, não ultrapassando seis listas sugeridas por cada agente. Pode-se observar, por exemplo, o gráfico da Figura 5-17, cujo agente que teve o maior número de listas construtivas sugeridas aceitas, foi o departamento fixo de número -5. Este departamento teve um total de 53 listas aceitas em mil ciclos. No entanto, pode-se observar que de todas as listas aceitas deste departamento, nenhuma delas melhorou o leiaute inicial. Já o departamento fixo -1 melhorou o leiaute inicial duas vezes com suas 27 listas construtivas aceitas. Observando-se o departamento 37 deste mesmo teste (Figura 5-17), teve 19 listas construtivas aceitas em mil ciclos. Destas 19 listas construtivas aceitas, seis delas melhoraram efetivamente o leiaute inicial (Figura 5-18). Pode-se concluir que: (a) os departamentos fixos colaboram na melhoria do leiaute final, pois mesmo que eles não possam mudar de posição dentro do leiaute, eles podem, através de mudanças na lista construtiva, escolherem seus vizinhos e (b) nem sempre os agentes que tem um maior número de listas construtivas aceitas, são os responsáveis pelo maior número de contribuições que melhoram efetivamente o leiaute inicial.

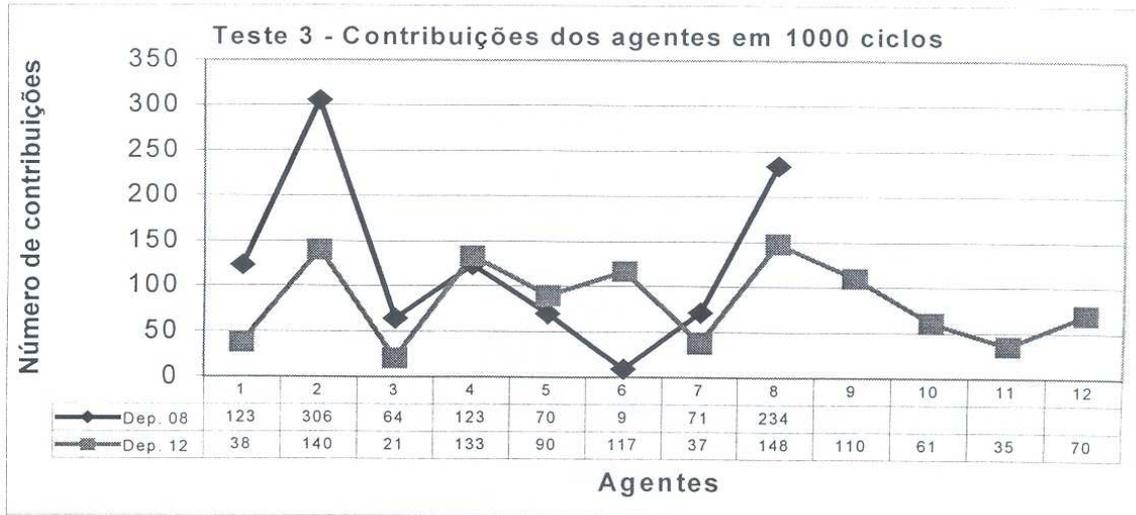


Figura 5-11 Contribuições dos agentes no teste 3 com 8 e 12 departamentos

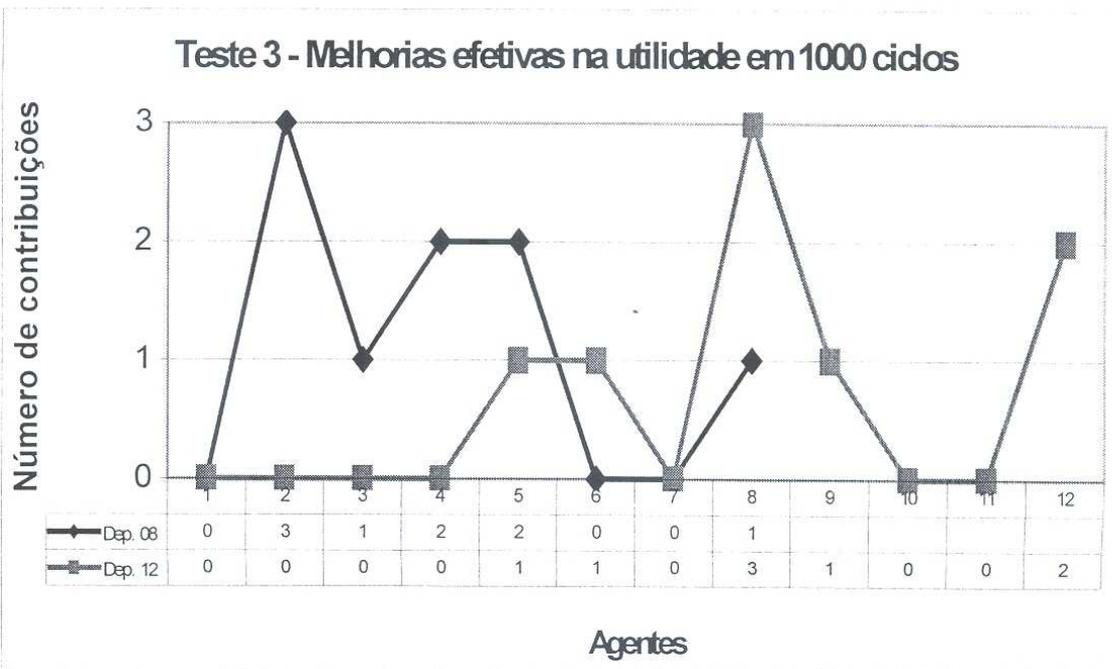


Figura 5-12 Melhorias efetivas na utilidade final do leiaute no teste 3 com 8 e 12 departamentos

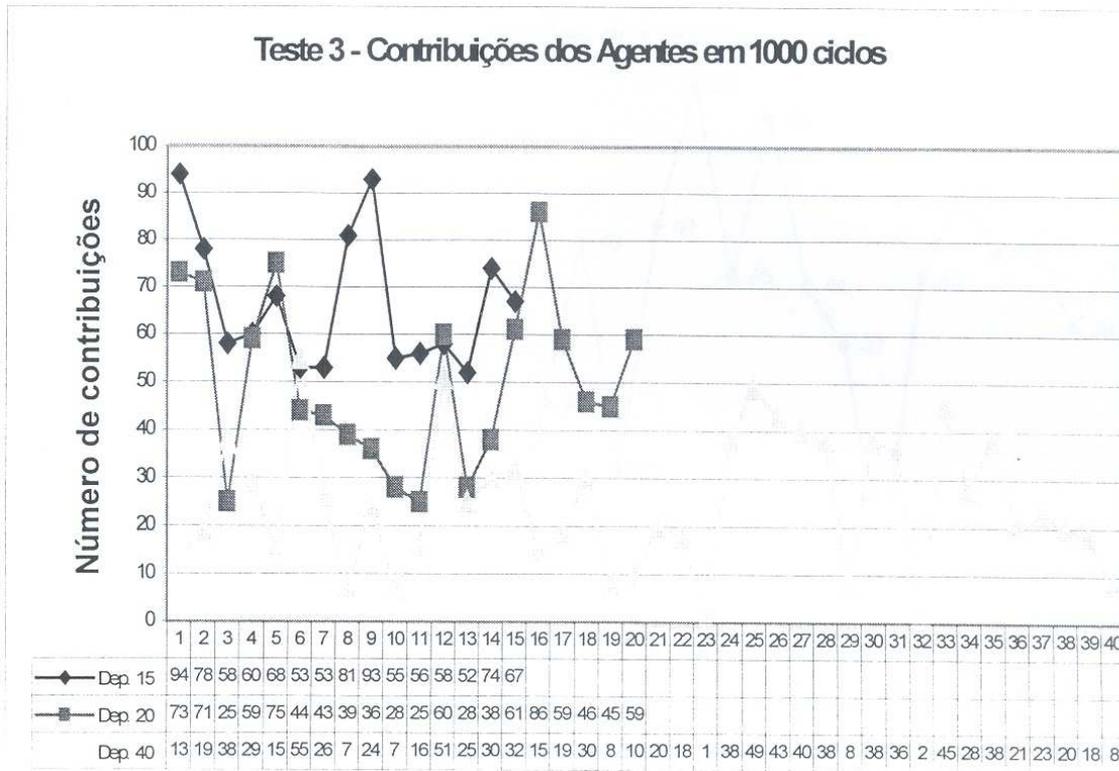


Figura 5-13 Contribuições dos agentes no teste 3 com 15, 20 e 40 departamentos

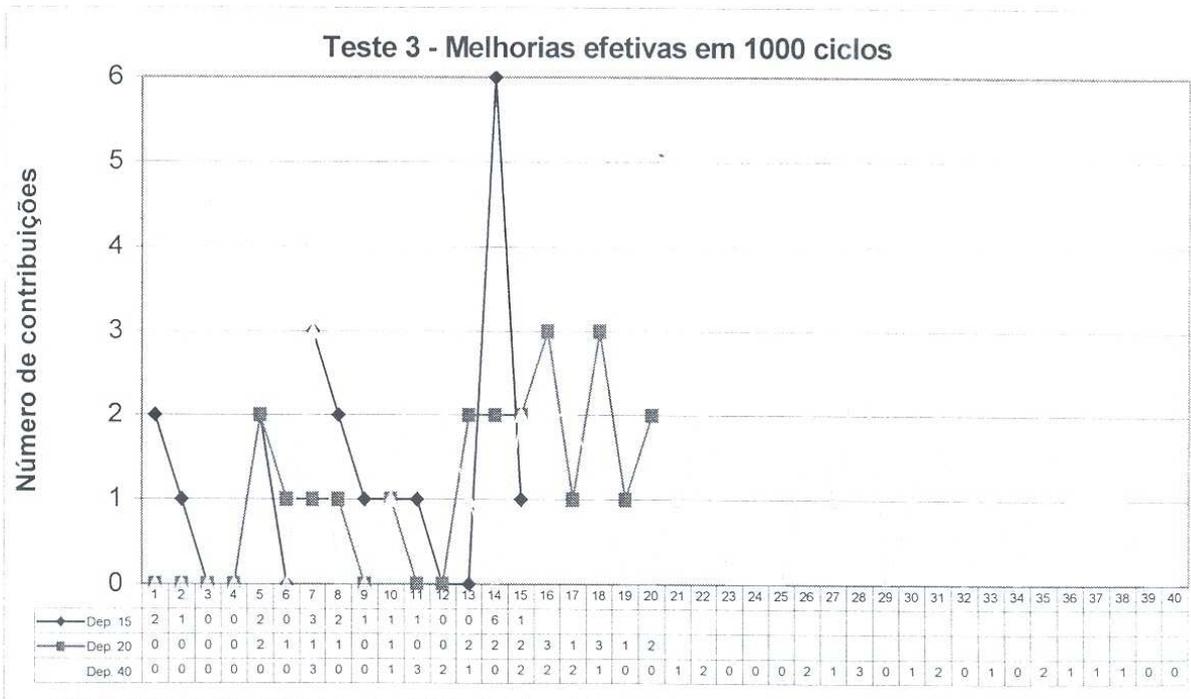


Figura 5-14 Melhorias efetivas na utilidade final do leiaute no teste 3 com 15, 20 e 40 departamentos

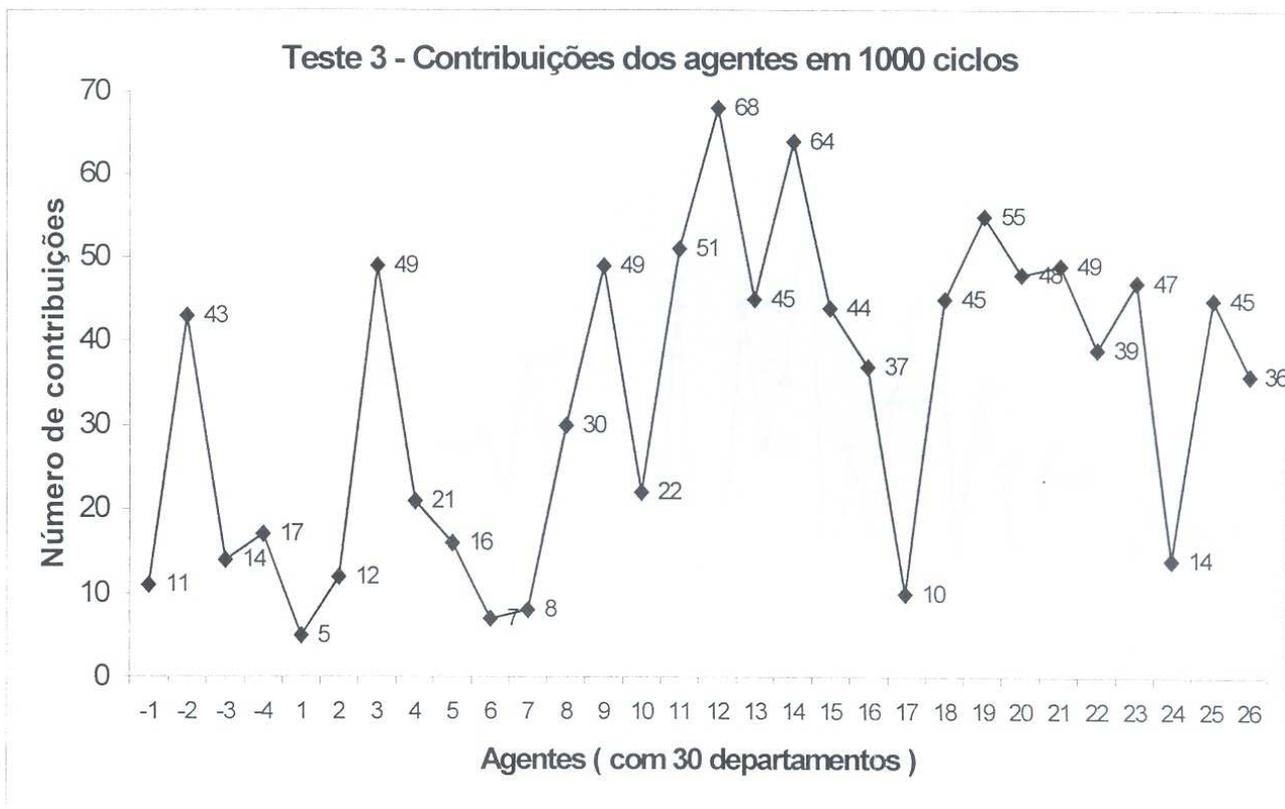
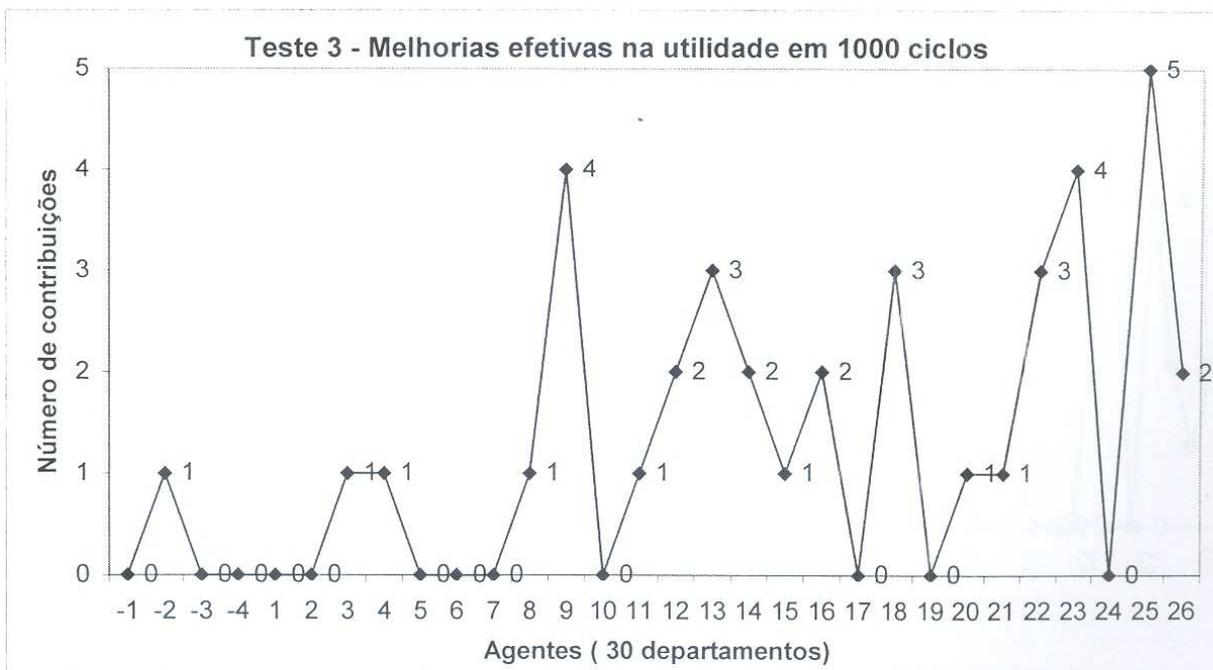


Figura 5-15 Contribuições dos agentes no teste 3 com 30 departamentos



5-16 Melhorias efetivas na utilidade final do leiaute no teste 3 com 30 departamentos

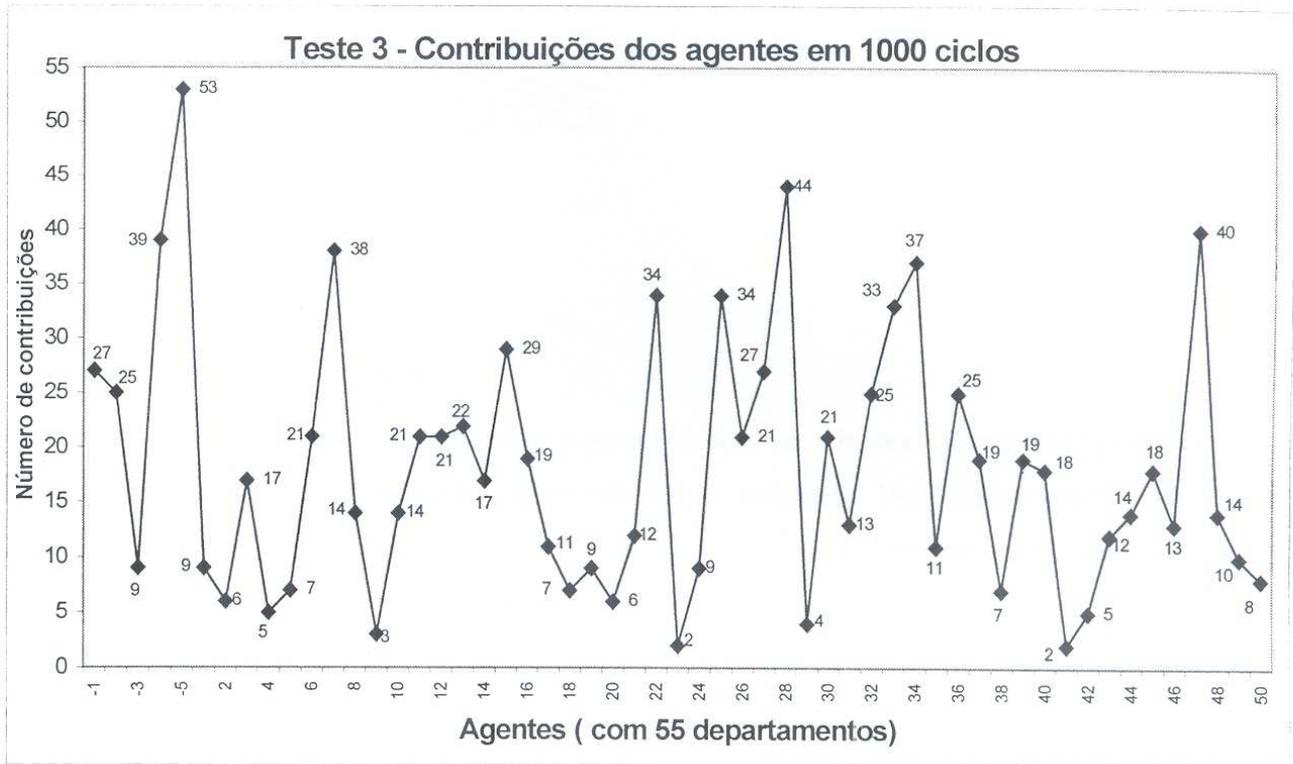


Figura 5-17 Contribuições dos agentes no teste 3 com 55 departamentos

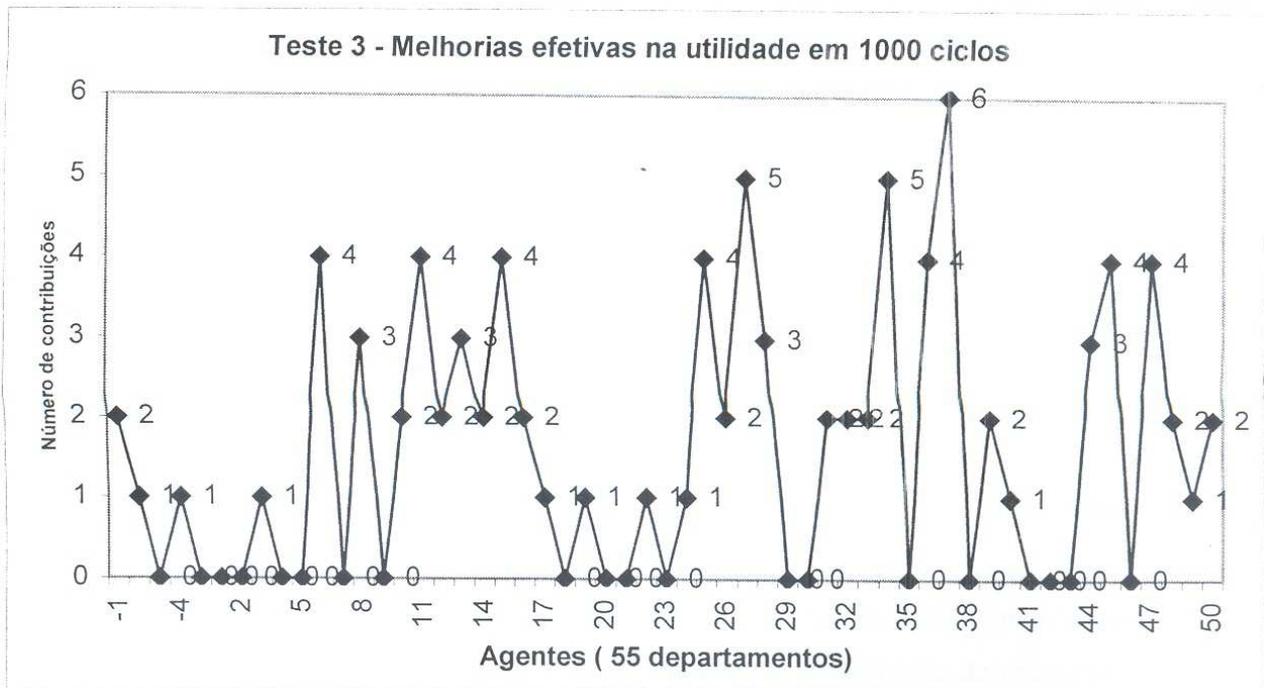


Figura 5-18 Melhorias efetivas na utilidade final do leiaute no teste 3 com 30 departamentos

6. Conclusões

6.1 Resultados obtidos

Os agentes mostraram-se um bom modelo para otimização de leiautes. Os resultados obtidos apresentam um ganho de qualidade surpreendente (da ordem de 19,27%) no leiaute final. Deve-se salientar que os agentes são simples e implementam comportamento reativo sem mudar de estratégia quando suas contribuições são inexistentes ou mínimas.

Verificou-se que para problemas com até oito departamentos a abordagem multiagente tem pouca relevância. Para tal número podem-se aplicar métodos extremamente pesados como da *força bruta* para descobrir a melhor solução, ou qualquer outro método. É importante salientar que para casos onde o número de departamentos é pequeno, por exemplo, oito departamentos, o modelo agente não é recomendado. A complexidade ligada à comunicação entre agentes faz do sistema multiagente um péssimo exemplo do ponto de vista de performance.

Verificou-se que para problemas médios (de nove a vinte departamentos), os resultados baseados na abordagem multiagente permaneceram iguais ou pouco abaixo do método tabu com pseudotrocas com o mesmo número de ciclos. A solução baseada em agentes é sensivelmente mais lenta que o método tabu.

Para problemas com 20 ou mais departamentos, o modelo multiagente apresentou excelentes resultados, superando inclusive o método tabu. O único inconveniente é que o tempo final de cada ciclo elevou-se consideravelmente devido principalmente ao número de agentes e ao pesado algoritmo construtivo.

De forma geral, pode-se observar na Figura 6-1 que o computador quando executa os processos de otimização de leiaute ocupa quase toda a sua capacidade de processamento, independente do método de otimização. Pode-se notar então que a performance do sistema é diretamente proporcional à velocidade do computador e de seus recursos como memória e

sistema operacional. Recomenda-se o uso de computadores com múltiplos processadores quando se opta pela arquitetura de multiagente. Os picos de 100% de uso da CPU normalmente acontecem quando o computador esta executando o algoritmo construtivo.

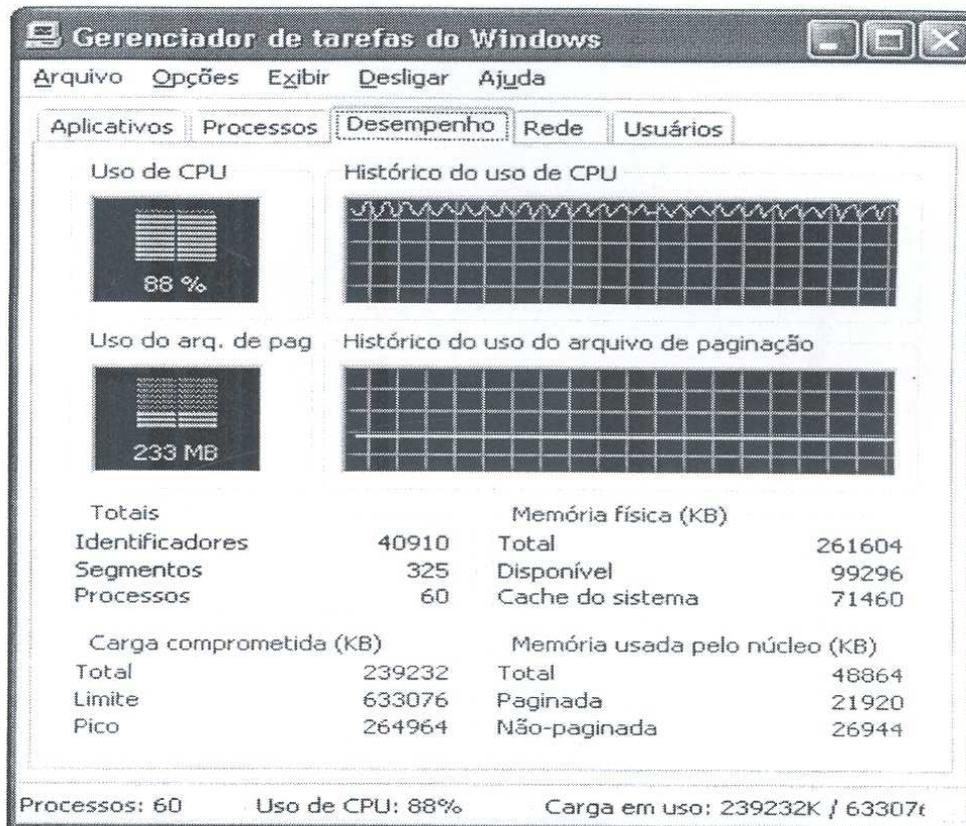


Figura 6-1 Demonstrativo do uso da CPU

Quanto ao ganho de qualidade total do leiaute, pode-se notar como mostra a Figura 6-2 que há um elevado ganho de qualidade no início do processo. Mas estes ganhos tendem a diminuir e estabilizar com o passar dos ciclos. As contribuições ficam mais raras e de percentuais cada vez mais baixos. Apesar de não se poder determinar exatamente qual a solução ótima com um elevado número de departamentos, acredita-se que as soluções apresentadas estão próximas das soluções ótimas.

Considerando todos os ciclos onde se encontrou a melhor solução para um leiaute em cada bateria de teste, obteve-se a média de 22,64. Considerando ainda uma margem de segurança, sugerimos que o número de ciclos inicial mínimo para uma boa solução na otimização de um leiaute seja igual ou superior ao número de departamentos a serem posicionados multiplicado por vinte e cinco.

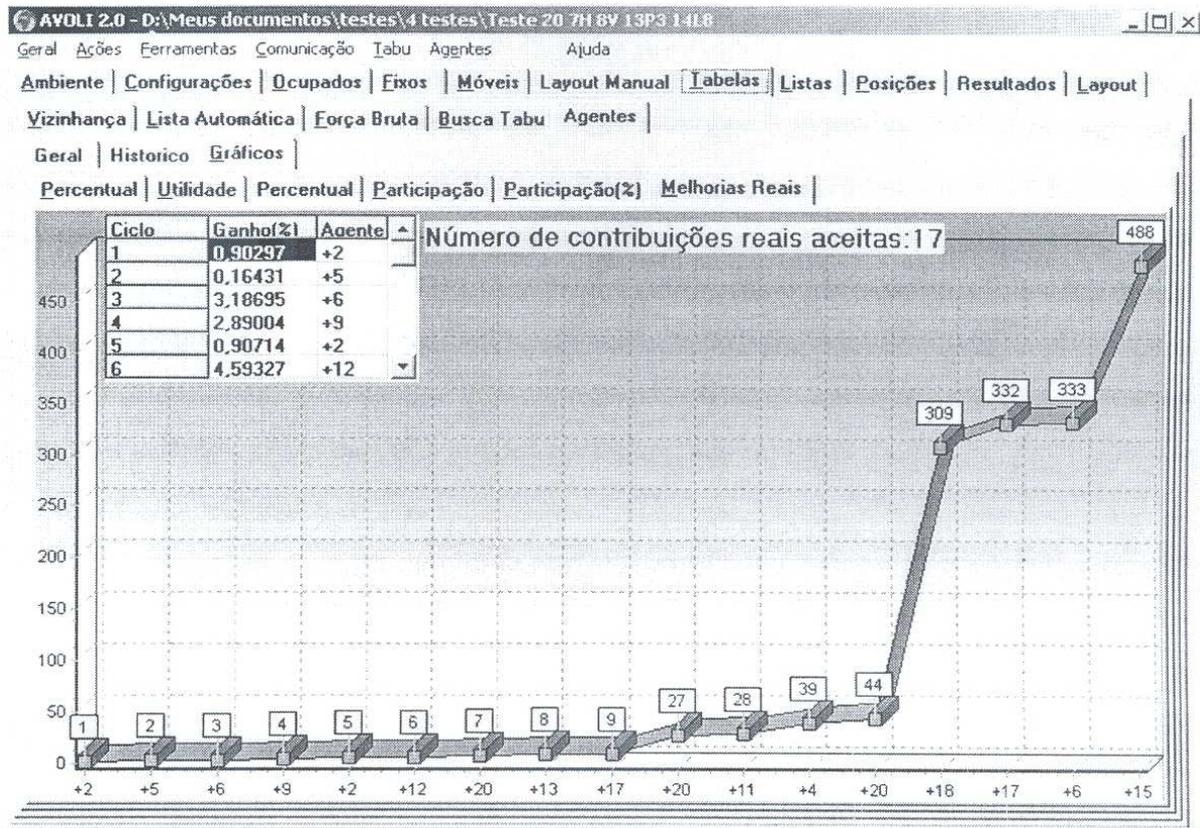


Figura 6-2 Melhorias efetivas no leiaute

A principal vantagem dos agentes é que eles aprendem e são persistentes. Eles podem ainda ter múltiplos comportamentos e mudar de comportamento no intuito de obter uma melhor solução, além do paralelismo.

6.2 Os agentes

Os agentes apresentaram uma surpreendente performance em um computador de um só processador. O processador usado para as experiências finais foi um microprocessador da marca AMD modelo Atlon a 1.2GHz, rodando sobre o sistema operacional Windows XP e com 128MBytes de RAM. Provavelmente os agentes teriam uma performance ainda melhor se o sistema es executasse em um computador multiprocessado.

O comportamento implementado nos agentes apresentou bons resultados. Ele é simples e requer pouca memória. Mesmo sem a presença de alguns agentes, que foram eliminados proposadamente durante o processo, o sistema teve bons ganhos. Percebeu-se, no entanto, que alguns agentes têm pouca ou nenhuma colaboração na melhoria do leiaute. Suas listas sugeridas raramente são aceitas. Pode-se verificar que mesmo no caso da falta de alguns

agentes o sistema funcionou satisfatoriamente. A eliminação dos mesmos não é trivial. Verificou-se, neste sentido, que alguns destes são vitais para a otimização do leiaute e outros simplesmente não fazem diferença significativa. A gestão (entrada e saída de agentes) é deixada para usuário do sistema AVOLI. O usuário pode acompanhar visualmente os agentes que estão contribuindo no processo de otimização.

Outra observação importante é sobre os comportamentos individualistas dos agentes num mundo coletivo. Nota-se um rápido ganho da qualidade final do leiaute já nos primeiros ciclos, mesmo quando os agentes têm comportamentos não cooperativos. Mas este ganho tende a se estabilizar rapidamente em poucos ciclos (Figura 6-3).

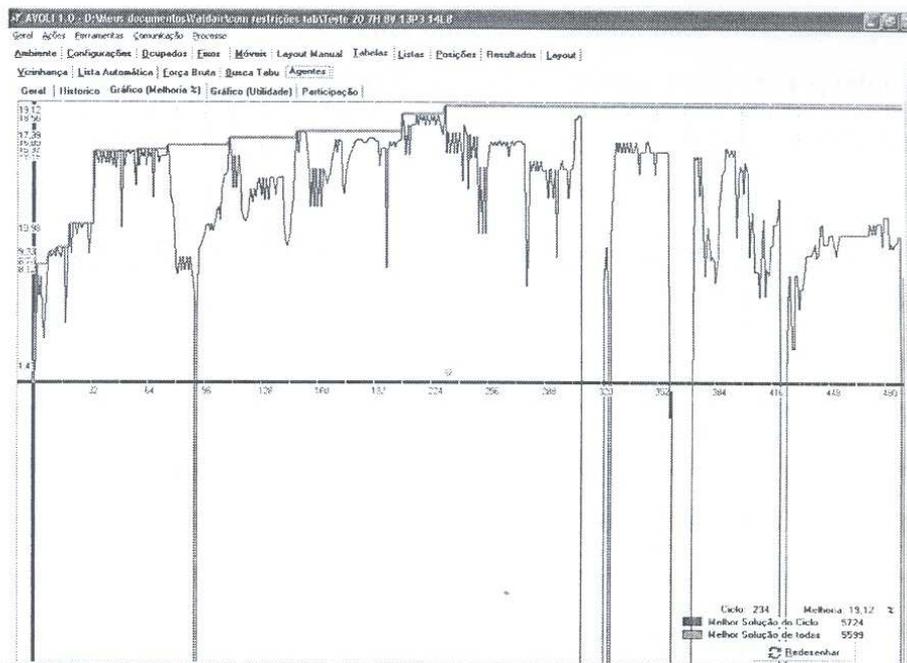


Figura 6-3 Variação da utilidade total durante 500 ciclos

6.3 O mundo

O agente mundo teve uma baixa performance, como já esperado. Isto é devido à sua complexidade natural e às inúmeras tarefas atribuídas a ele, atrasando o processo como um todo. Seu algoritmo de construção é pesado e é utilizado muitas vezes para avaliar todas as possíveis listas sugeridas pelos agentes. Além disto, ele ainda tem a tarefa de manter todos os agentes atualizados e informados sobre as mudanças do leiaute. O mundo deve também manter um histórico completo de todas as mudanças feitas no leiaute, para, no final,

apresentar a melhor solução de utilidade total. Conclui-se então que se deve tentar ao máximo deixar todas as tarefas básicas para os agentes e assim atribuir ao mundo só as tarefas e funções que realmente só ele pode executar, como, por exemplo, o histórico de todas as mudanças, acompanhamento dos agentes e suas performances, etc.

Observou-se que a memória de trocas N_a do mundo não é tão eficiente como as dos agentes. Chegou-se a resultados melhores sem esta variável. Ela influencia diretamente nas escolhas das listas construtivas sugeridas dos agentes. O que leva a concluir que o mundo não deve interferir nas sugestões das listas construtivas sugeridas pelos agentes. Constatou-se que realmente alguns agentes colaboram pouco ou mesmo com nenhuma melhoria no leiaute. No entanto a memória de trocas N_a mostrou-se útil quando se quer acompanhar a performance dos agentes (Figura 6-4). Ela fornece uma importante informação referente à participação de cada agente em um número pré-definido de ciclos. Assim, pode-se a qualquer momento, eliminar os agentes que tiveram pouco ou nenhuma participação na otimização do leiaute e continuar de onde se parou sem a perda de conhecimento dos agentes que realmente colaboram.

Houve melhorias da qualidade final do leiaute mesmo com a sua eliminação proposital de agentes que contribuem pouca ou nada. Pode-se notar, por exemplo, no gráfico da Figura 6-4 e Figura 6-5 que o agente +3 em 500 ciclos só colaborou duas vezes. Passaram-se na seqüência de mais mil ciclos e o mesmo agente não colaborou mais nenhuma vez. É importante notar que após a sua eliminação houve uma melhora na performance do sistema como um todo, principalmente no tempo de resposta que melhorou sensivelmente por ciclo. O motivo foi à redução do número de agentes, descongestionando o tráfego nas linhas de comunicação e o agente mundo. Este último não teria mais a carga de avaliar as listas sugeridas pelo agente suprimido.

6.4 A comunicação entre os agentes e o mundo

A comunicação é uma parte crítica do sistema multiagente. Ela se agrava proporcionalmente ao número de agentes que existem no sistema. O modelo multiagente não é tão eficiente como se recuperasse informações diretamente da memória do computador, como faz, por exemplo, o método tabu.

Para reduzir o atraso e a perda de informações provocadas pelo congestionamento sobre o canal de comunicação, criou-se mais um canal. Este canal adicional é utilizado exclusivamente pelo mundo para transmitir informações aos agentes.

Tal fato leva à conclusão de que a comunicação entre os membros do mundo deve ser a mais breve possível e apenas quando for realmente necessária. Eliminar informações redundantes na comunicação é uma questão essencial para a performance final de todo o sistema.

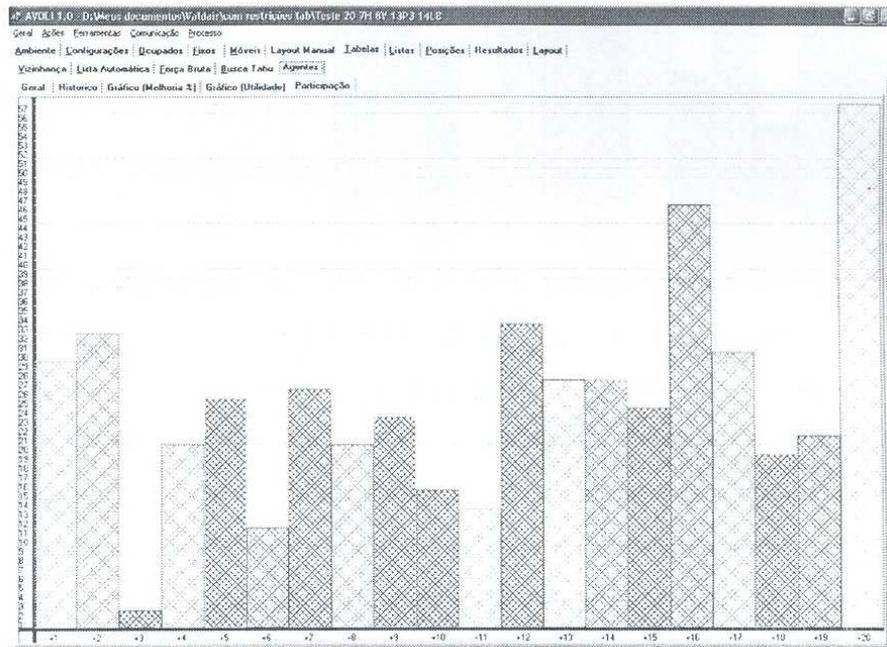


Figura 6-4 Participação dos agentes na melhoria geral do leiaute em 500 ciclos

6.5 Trabalhos Futuros

Em relação às funcionalidades do sistema proposto, ainda há muitas coisas que podem ser melhoradas e/ou acrescentadas. Podemos citar, por exemplo, o melhoramento do algoritmo construtivo, que é, com certeza, uma parte crítica e complexa, e que compromete sensivelmente a performance total do sistema. Pode-se também melhorar a função de construção realista. Ela opera apenas em duas dimensões, i.e., em uma única camada de leiaute. A sugestão seria que ela pudesse construir também os departamentos em múltiplos pisos, i.e., poder-se-ia dispor os departamentos em vários andares, como por exemplo, em prédios ou mesmo em múltiplos barracões.

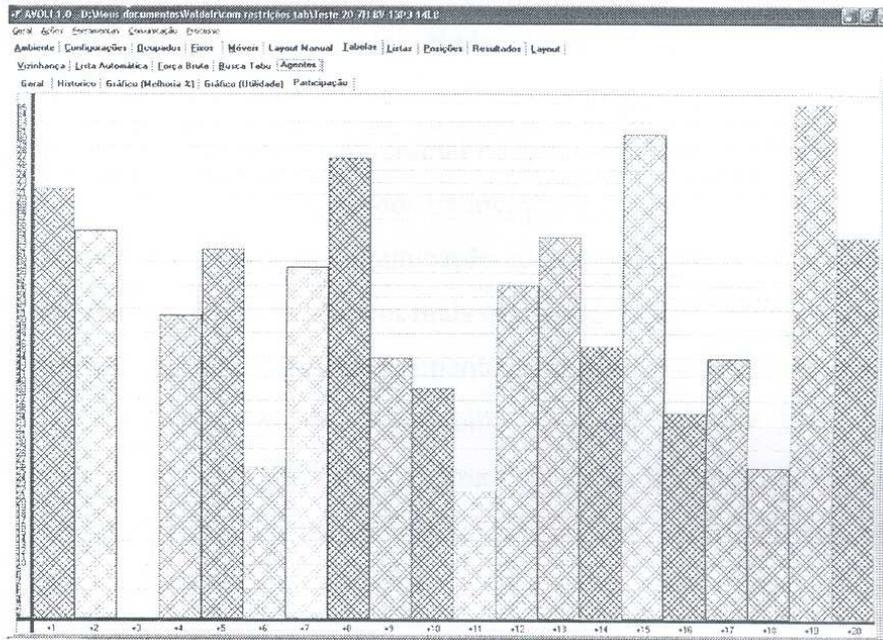


Figura 6-5 Depois de mais 1000 ciclos o departamento +3 nunca colaborou

Sobre o sistema básico do AVOLI, poder-se-ia implementar outros métodos para compará-los, como por exemplo, algoritmos genéticos e têmpera simulada. Alguns métodos poderiam ser acrescentados nos agentes como opções de comportamento.

Sobre os sistemas multiagente sugere-se um estudo para implementar formas de aprendizagem automática, assim como mecanismo para persistir os conhecimentos dos agentes. Atualmente, todo o conhecimento o agente é totalmente perdido quando este deixa de existir.

Cada departamento foi representado por um agente individual e autônomo. Uma forma de melhorar a performance seria distribuir estes agentes em uma rede local. Atualmente os agentes se executam todos em uma única máquina. Desta forma, se houver um número acentuado de departamentos serão criados inúmeros agentes, o que forçará o computador a alocar uma quantidade elevada de recursos para executar todos estes agentes. Outro efeito colateral é que há um congestionamento no tráfego de informações entre todos os membros da comunidade. Este congestionamento é diretamente proporcional ao número de agentes. O que se pode sugerir, neste caso, é que possa haver um agente cuidando de mais de um departamento de cada vez. O número de agentes então nunca deveria ultrapassar os limites aceitáveis de processos que o computador pode manipular e executar de uma só vez sem. Claro que isto depende diretamente do computador e seus recursos, como memória,

velocidade, sistema operacional e número de processadores que o computador possui. No entanto este número máximo de agentes ainda é empírico e talvez mereça um estudo separado.

O que nós leva ao próximo ponto crucial no desenvolvimento de sistemas multiagente que é a comunicação que existe entre todos os membros do mundo. A comunicação aqui se limitou a um único computador. Esta comunicação apresentou alguma demora que prejudica a performance do sistema como um todo. Por mais que os agentes sejam eficientes e rápidos em seus processos, eles são prejudicados imensamente no momento de transmitir suas conclusões e recuperar novas informações. Este problema pode se agravar se considerarmos a comunicação entre vários computadores em uma rede física qualquer. A complexidade dos dados envolvidos e os meios físicos utilizados podem contribuir negativamente na performance total dos sistemas multiagente.

Os agentes em sua essência devem ser simples e eficientes. Eles não devem provocar nenhuma sobrecarga ao mundo. O mundo é apenas um para todos os agentes. Estes últimos deveriam ter mais atribuições como, por exemplo, calcular o ganho real da utilidade e já passar este valor para o mundo. O mundo deveria repassar o algoritmo de construção, que tanto compromete a performance do sistema, para os agentes. Certamente, passar o algoritmo de construção e tudo o que ele requer para funcionar corretamente para os agentes, os tornaria bem mais volumosos e pesados. Considerando que eles são em grandes quantidades poderia haver sérios problemas quanto ao gerenciamento de memória e performance. O que nós leva a pensar em sistemas futuros distribuídos em vários computadores. No entanto, isto recai ao problema da lentidão da comunicação via rede.

7. Referências

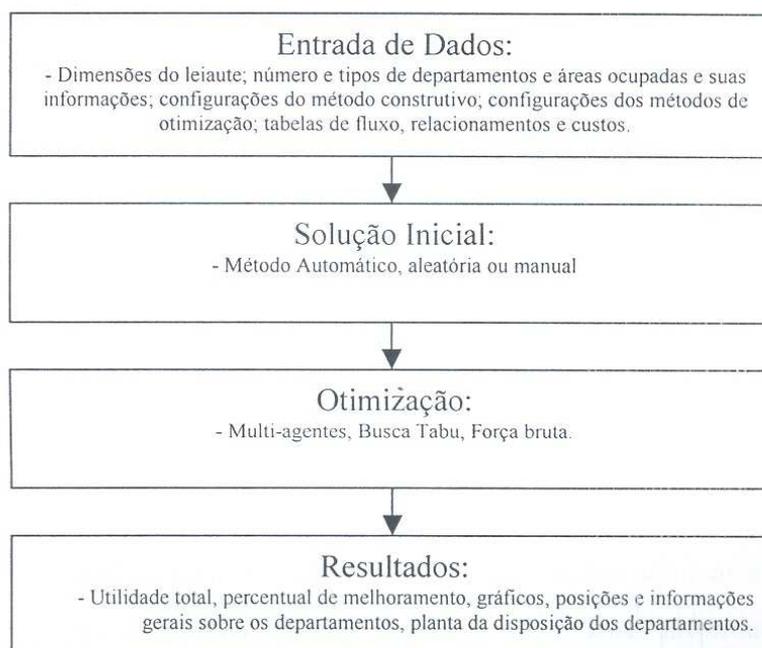
- [Alh91] Al-Hakim, L. A., "Two Graph-Theoretic Procedures for an Improved Solution to the Facilities Location Problem, *International Journal of Production Research*", 29, 8, 1701-1718, 1991.
- [Arm63] Armour, G.C., and Buffa, E.S., "A Heuristic Algorithm and Simulation Approach to Relative Allocation of Facilities," *Management Science*, 9, 294-309, 1963.
- [Ask93] Askin, Ronald G., Standridge, Charles R., "Modeling and Analysis of Manufacturing Systems", John Wiley & Sons Inc., 1993.
- [Aur99] Aurélio Buarque de Holanda Ferreira: "Dicionário Aurélio Eletrônico, Século XXI", versão 3.0, Lexikon Informática Ltda., 1999
- [Aza00] Azadivar, F., Wang, J., Facility Layout Optimization using simulation and Genetic Algorithms, *International Journal of Production Research*, 38, 17, 4369-4383, 2000.
- [Baz75] Bazaraa, M. S., "Computerized Location Design: A Branch and Bound Approach." *AIIE Transactions*, 7, 4, 432-437, 1975.
- [Bon88] Bond, A. H.; Gasser, L., "An Analysis of Problems and Research in DAI. E.: Readings in Distributed Artificial Intelligence." Editado por Alan H. Bond and Les Gasser, Morgan Kaufmann Publishers, inc, San Mateo, CA, 1988.
- [Chi01] Chiang, W. C., Visual Facility Location Design System, *International Journal Production Research*, vol, 39, nº9, 1811-1836, 2001.
- [Chi96] Chiang, W. C., and Kouvelis, P., "An Improved Tabu Search Heuristic for Solving Facility Location Design Problems" *International Journal Production Research*, 1996.
- [Cor94] Coraini, A. L. S.; Nolla, I. M., AUTOCAD release 12: Curso Básico e Prático, Makron Books, 1994.
- [Dav96] Davidsson, P., "Autonomous Agents and the Concept of Concepts.", Doctoral dissertation, Department of Computer Science, Lund University, Sweden, 1996.
- [Dur94] Durfee, E., H.; Rosenschein, J, S., "Distributed Problem Solving and Multi-Agent Systems: Comparisons and Examples. Em Proceedings of The International Workshop on Distributed Artificial Intelligence.", July 1994.
- [Fou78] Foulds, L. R., and Robinson, D. F., "Graph Theoretic Heuristics for the Plant Location Problem." *International Journal of Production Research*, 16, 1, 27-37, 1978.
- [Fur97] Furtado, J.C.; Lorena, L.A.N; "Otimização de localização usando busca tabu. *Gestão & Produção*.", vol. 4, no. 1, 88-107, 1997.
- [Gar79] Garey, M. R., and Johnson, D. S., "Computers and Intractability: A Guide to the Theory of NP-Completeness.", W. H. Freeman and Company, New York, New York, 1979.
- [Gil62] Gilmore, P. C., "Optimal and Suboptimal Algorithms for the Quadratic Assignment Problem.", *Journal of the Society for Industrial and Applied Mathematics*, 10, 305-313, 1962.
- [Glo89a] Glover, F., Tabu Search - Part I. *ORSA Journal on Computing*, vol. 1, no. 3, 190-206, 1989a.

- [Glo89b] Glover, F., Tabu Search - Part II, ORSA Journal on Computing, vol. 2, no. 1, 4-32, 1989b.
- [Glo97] Glover, F. e Laguna M., Tabu Search pp. 297-324. 8.9.2 Kluwer Academic Publishers, Massachusetts, 1997.
- [Gol96] Goleman, D.; "Inteligencia Emocional, Circulo de Leitores", 1996.
- [Hay95] Hayes-Toth, B., "Architecture for Adaptive Intelligent System. Em Artificial Intelligence: Special Issue on Agents and Interactivity.", 1995.
- [Her87] Heragu, S.S. e Kusiak, A., The facility layout problem. European Journal of Operational Research 29, 229-251, 1987.
- [Her91] Heragu, S. S., and Kusiak, A., "Efficient Models for the Facility Layout Problem.", European Journal Operational Research, 53, 1-13, 1991.
- [Hol75] Holland, J. H., Adaptation in Natural and Artificial Systems, University of Michigan, Ann Arbor, MI, 1975.
- [Huh98] Huhns, M.N., Munindar, P.s., editores, "Readings in Agents, Morgan Kaufmann, Inc.", 1998.
- [Kir83] Kirkpatrick, S., Gelatt, C. D. Jr., Vecchi, M. P. Optimization by Simulated Annealing, Science, vol 220, 671-680, 1983.
- [Koo57] Koopmans, T. C., and Beckman, M., "Assignment Problems and the Location of Economic Activities." Econometrica, 25, 53, 53-76, 1957.
- [Lee67] Lee, R.C.; Moore, J.M.; "CORELAP - Computerized Relationship Layout Planning, Industrial Engineering.", 18, 195-200, 1967.
- [Mar02] Martins, Valdair Candido, "Otimização de Layout em Ambientes Industriais Realistas Usando Busca Tabu.", Dissertação – PUC-PR, Março, 2002.
- [Mel96] Meller, Russell D., and Gau, Kai-Yin, The Facility Layout Recent and Emerging Trends and Perspectives. Journal of Manufacturing Systems, Vol. 15, nº 5, pp. 351-366, 1996.
- [Mic94] Microstation, Manuais do usuário, 1994.
- [Mon90] Montreuil, B., "A modelling framework for integrating layout design and flow network design," Proceeding of the Material Handling Research Colloquium, Hebron, KY, pp 43-58, 1990.
- [Mon93] Montreuil, B., Venkatadri, U., Ratliff, H. D., "Generating a layout from a Design Skeleton," IIE Transactions, 25, 1, 3-15, 1993.
- [Mou96] Moulin, M.; Chaib-Draa, B., "An Overview of Distributed Artificial Intelligence. Em Foundations of Distributed Artificial Intelligence.", Editado por G. M. P. O'Hara and N.R. Jennings, Foundations of Distributed Artificial Intelligence, John Wiley & Sons, inc., 1996.
- [Mae95] Maes, P., "Artificial Life Meets Entertainment: Life like Autonomous Agents. Em Communications of the ACM.", 38, 11, 1995.
- [Nug68] Nugent, C. E., Vollmann, T. E. and Ruml, J., An Experimental Comparison of Techniques for The Assignment of Facilities to Locations. Operations Research, 16, 150-173, 1968.
- [Rao95] Rao, A. S., Georgeff, M. P., BDI Agents: From Theory to Practice. Technical Note 56, Australian Artificial Intelligence Institute, April, 1995.
- [Rus95] Russel, S., Norvig P., "Artificial Intelligence – A Modern Approach. Prentice Hall, Inc." pg 4-5, 1995.
- [Scr85] Scriabin, M.; Vergin, R.C. "A cluster-analytic approach to facility layout". Management Science, 31(1), 33-49, 1985.
- [See67] Seehof, J.M.; Evans, W.O.; "Automated layout design program, Industrial Engineering.", 18, 690-695, 1967.
- [Shm99] Shmeil, Marcos Augusto Hochuli, "Sistemas Multiagente na Modelação da Estrutura e Relações de Contratação de Organizações", Tese de Doutorado, Faculdade de Engenharia da Universidade do Porto (Portugal), 1999.

- [Sim68] Simon, H.A., "The Sciences of The Artificial. Massachusetts.", The M.T.I. Press, pg. 1-22, 1968.
- [Smi94] Smith, D. C.; Cypher, A., Spohrer, J., KidSim, "Programming Agents without a Programming Language.", Communications of the ACM, 37, 7, 1994.
- [Tam91] Tam, Kar Yan, "A Hierarchical Approach to the Facility Layout Problem," Int. J. Prod. Res., Vol. 29, no. 1, 165-184, 1991.
- [Tam92] Tam, Kar Yan, "Genetic Algorithms, Function Optimization, and Facility Layout Design," European Journal of Operational Research, 63, 1992.
- [Tam98] Tam K. Y. and Chan S. K., "Solving Facility Layout Problems with Geometric Constraints Using Parallel Genetic Algorithms: Experimentation and Findings.", Int. J. Prod. Res., 1998.
- [Woo94] Wooldridge, M.; Jennings, N. R., "Intelligent Agents: Theory and Practice.", Knowledge Engineering Review, October, 1994.
- [Woo95] Wooldridge, M.; Jennings, N. R., "Agent Theories, Architectures, and Languages: A Survey.", Editado por Wooldridge. M. and Jennings, N. R., Berlin, Spring-Verlag, 1995.

ANEXO A - Usando o sistema AVOLI

Este anexo tem por finalidade explicar passo a passo todos os procedimentos para a operação e obtenção de resultados no sistema AVOLI. A concepção do sistema AVOLI foi feita para ser de fácil utilização, manuseio e rápida obtenção de resultados. No entanto é necessário algum conhecimento sobre a área de sistemas produtivos para o estudo e coleta de informações mínimas para a operação do sistema. Estas informações são as dimensões disponíveis ou desejáveis para o leiaute, departamentos, áreas, razões de aspectos mínimas e máximas, áreas ocupadas no leiaute, fluxos, adjacências e outras informações diversas. Estas informações básicas são comuns a todos as metodologias de construções iniciais e otimização do leiaute disponíveis neste sistema. Resumidamente o processo de construção e otimização de leiautes segue os seguintes passos no sistema AVOLI:



Devem-se informar inicialmente quais as dimensões do leiaute nos campos 'comprimento' e 'largura' na página 'Ambiente' (Figura A.1). Observe que a unidade de medida utilizada está bem a baixo e ela por padrão está normalmente em metros. Ainda se tem a opção de decímetros, centímetros e milímetros e pode ser mudado dinamicamente. A área total e a orientação do leiaute são automaticamente calculadas e mostradas logo a baixo da unidade

utilizada. Pode-se ainda deixar o sistema AVOLI escolher as dimensões do leiaute. Ele escolherá estas dimensões baseadas na somatória de todas as áreas dos departamentos e suas razões de aspectos.

O próximo passo é determinar quantos departamentos móveis e fixos haverá no leiaute no quadro 'departamentos'. Neste local também informamos quantas áreas ocupadas existem. Este campo dimensiona as grades de informações individuais destas áreas. Logo o próximo passo será o preenchimento das informações individuais dos departamentos e áreas ocupadas.

The screenshot shows the 'Ambiente' configuration window in AVOLI 2.0. The window title is 'AVOLI 2.0 - D:\Meus documentos\testes\1 testes\teste 15'. The menu bar includes 'Geral', 'Ações', 'Ferramentas', 'Comunicação', 'Tabu', 'Agentes', and 'Ajuda'. The menu bar also contains tabs: 'Ambiente', 'Configurações', 'Ocupados', 'Fixos', 'Móveis', 'Layout Manual', 'Tabelas', 'Listas', 'Posições', 'Resultados', and 'Layout'. The main area is divided into several sections:

- Área Total Disponível:**
 - Definida pelo sistema
 - Comprimento: 5 m
 - Largura: 3 m
 - Unidade: Metros
 - Área Total: 15 m²
 - Razão de Aspecto: 0,600
 - Orientação: Horizontal
- Departamentos:**

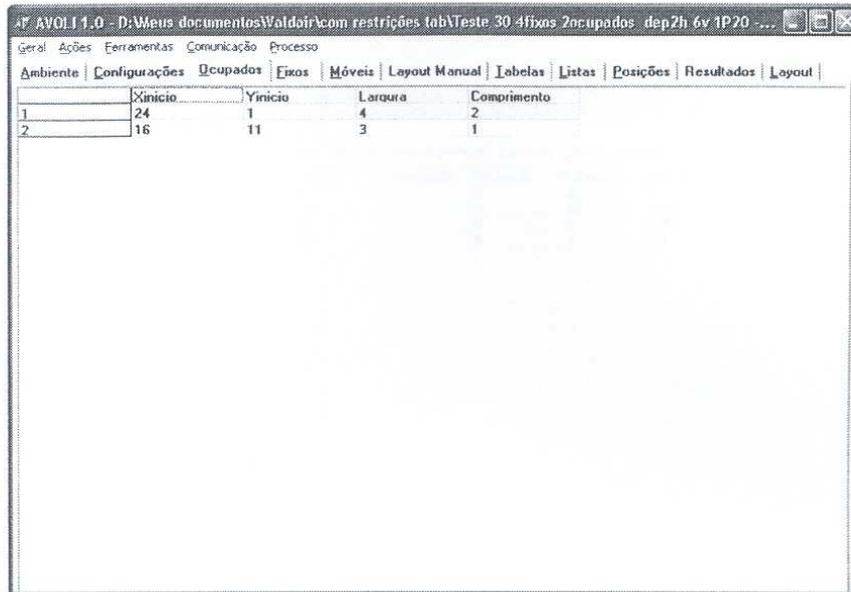
		Declarada / Usada	
Nº Áreas Ocupadas:	0	0 / 0	m ²
Nº Deptos Fixos:	0	0 / 0	m ²
Nº Deptos Móveis:	15	15 / 15	m ²
<input checked="" type="checkbox"/> Estimar Área		Total:	15 / 15 m ²
- Solução Inicial:**
 - Automática
 - Aleatória: 6 Listas
 - Lista Manual
 - Layout Manual
 - Inicialização
- Solução Otimizada (métodos):**
 - Multi-Agentes
 - Força Bruta
 - Busca Tabu
 - Tempera Simulada
 - Algoritmo Genético
 - Iniciar Processo

Buttons at the bottom right: Abrir Tabelas, Salvar Tabelas, Encerrar.

Figura A.1 Tela da configuração do ambiente

A página 'Ocupados' (Figura A.2) destina-se as informações sobre as áreas ocupadas e é a mais simples das áreas e departamentos de ser preenchida. Suas informações podem ser obtidas na simples observação do leiaute, se ele já existe. Nesta pagina teremos que informar a localização destas áreas no leiaute. Sua disposição no leiaute e dada por 'Xinicial' e 'Yinicial' que corresponde ao canto superior esquerdo da área ocupada no interior do leiaute. Observa-se que os pontos iniciais do leiaute localizam-se também no canto superior esquerdo do leiaute e começam inicialmente nas coordenada X igual a 1(um) e Y igual a 1(um). Outras informações sobre estas áreas correspondem a sua largura e comprimento que devem ser

iguais ou superiores a uma unidade de medida. Observa-se que todos os objetos do sistema em questão, departamentos, áreas ocupadas e leiaute, são na forma retangulares.



	Xinicial	Yinicial	Largura	Comprimento
1	24	1	4	2
2	16	11	3	1

Figura A.2 Informações sobre áreas ocupadas ou mortas

O próximo passo será o preenchimento da página 'Fixos' (Figura A.3) que correspondem aos departamentos fixos do leiaute. Nesta página o primeiro campo 'Área' corresponde à área que se deseja para o departamento fixo. A área é o produto da largura pelo comprimento. Os próximos campos 'Xinicial' e 'Yinicial' são de notação iguais às áreas ocupadas e correspondem as posições superiores esquerdas da posição deste departamento fixo no interior do leiaute.

O próximo campo 'Orientação' é opcional, pode ser deixado em branco, o que significará que a orientação do departamento é livre. Se, no entanto o departamento tiver uma orientação pré-estabelecida então use 'V' ou 'VERTICAL' para informar ao sistema que o departamento deverá necessariamente ter uma orientação na vertical (a largura é maior ou igual ao seu comprimento) ou pode-se definir com um 'H' ou 'Horizontal' para indicar que o departamento deverá estar necessariamente na horizontal (a largura é menor ou igual ao seu comprimento).

No campo 'RA Morta' é destinado a informar em porcentagem a razão da área morta do departamento. Ou seja, o máximo que o departamento pode perder de sua área útil com sobreposição de áreas ocupadas. No entanto sendo este um departamento fixo, não haverá

nenhum tratamento para respeitar estes limites máximos, pois isto é de responsabilidade do usuário que determinou sua posição. Apenas o sistema se limitará a avisar se tiver alguma irregularidade.

Ambiente	Configurações	Ocupados	Fixos	Móveis	Layout Manual	Labelas	Listas	Posições	Resultados	Layout
	Área	Xinicio	Yinicio	Orientação	RA Morta	R.Asp.Mín.	R.Asp.Máx.			
1	24	1	1		0	.8	1			
2	16	1	14		0	.75	1.15			
3	36	8	8		0	.6	1.85			
4	8	16	14		0	.3	1.1			

Figura A.3 Informações sobre os departamentos fixos

Os próximos campos a serem preenchidos são 'R.Asp.Mim' e 'R.Asp.Max.'; que correspondem a razão de aspecto mínima e a razão de aspecto máxima do departamento respectivamente. Sendo que o valor 1(um) significa que o comprimento e a largura são iguais.

Passamos agora para a próxima página 'Móveis' (Figura A.4), que define as informações sobre os departamentos móveis. Os campos são bem similares aos anteriores. Cujos campos "Área", "Orientação", "Perda A. Morta (%)", "R.Asp.Mín." e "R.Asp.Máx." são de notação iguais aos dos departamentos fixos anteriormente explicados. O que este grade não tem igual ao dos departamentos fixos é uma posição previamente estabelecida, pois será de responsabilidade do sistema posicioná-los de acordo com a lista de construção em algum lugar válido no interior do leiaute. O que esta grade tem a mais é o campo "Red.Máx.Área (%)" que informa ao sistema em porcentagem o quanto ele pode, se e somente se necessitar para posicionar o departamento dentro do leiaute, diminuir a área total declarada do departamento móvel para um melhor posicionamento do departamento no leiaute. O outro

campo “Área Mínima” não deve ser preenchido, pois o sistema se encarregará de preenchê-lo com base nas informações disponíveis, este campo e apenas informativo.

	Área	Orientação	Perda A. Morta (%)	R. Asp. Mín.	R. Asp. Máx.	Red. Máx. Área (%)	Área Mínima
+1	21		0	.9	1,18	0	
+2	17,5	h	0	.5	1	0	
+3	3,6		0	.3	1,4	0	
+4	15,4		0	.6	1,25	0	
+5	20		0	.9	1	0	
+6	19,5	v	0	1,2	1,8	0	
+7	16		0	.85	1,1	0	
+8	9		0	.9	1	0	
+9	9		0	.8	1,1	0	
+10	25		0	.92	1,05	0	
+11	4		0	.85	1,15	0	
+12	3		0	.9	1	0	
+13	4		0	1	1	0	
+14	9		0	1	1	0	
+15	4,5		0	.7	1,1	0	
+16	5		0	.5	1,5	0	
+17	16		0	.85	1,1	0	
+18	9		0	.9	1	0	
+19	9		0	.8	1,1	0	
+20	25		0	.92	1,05	0	
+21	4		0	.85	1,15	0	
+22	3		0	.9	1	0	
+23	4		0	1	1	0	
+24	9		0	1	1	0	
+25	4,5		0	.7	1,1	0	
+26	5		0	.5	1,5	0	

Figura A.4 Informações sobre os departamentos moveis

O próximo passo é preencher a tabela de fluxos na grade "Fluxos" na página configurações (Figura A.5) e a tabela de "Relacionamentos" que se encontra na pagina ao lado da de "Fluxos" dentro da pagina de "Configurações".

O relacionamento (Figura A.6) tem por notação apenas três possíveis valores: vazio ou “0”(zero), para departamentos cuja adjacências não tem relevância, “1”(um) para departamentos que necessariamente devem ser vizinhos um do outro e “-1”(menos um) para departamentos que não podem ser vizinhos.

Preenchido estas informações, o sistema já estará pronto para executar seus processos e métodos. Ainda há outras configurações possíveis que se encontram na página "Configurações" em "Gerais". Mas por padrão, elas estão todas marcadas com os valores mais comuns. Sendo esta parte comum a todos os processos e metodologias. Recomenda-se cautela na alteração destes itens, visto que qualquer mudança nestes itens, afeta diretamente na performance e resultados do sistema.

AVOLI 1.0 - D:\Meus documentos\Waldair\com restrições tab\teste 30 Afijos 2ocupados dep2h 6v 1P20 ...

Geral Ações Ferramentas Comunicação Processo

Ambiente Configurações Ocupados Fixos Móveis Layout Manual Tabelas Listas Posições Resultados Layout

Geral Fluxos Relacionamentos Pré-Processamento (Móveis) Pré-Processamento (Fixos)

	+1	+2	+3	+4	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12	+13	+14	+15	+16	+17	+18	+19	∑			
+1	x	x	x	3	2	0	0	2	10	5	0	5	2	5	0	0	2	0	5	6	3	0	1	10	0	11	
+2	3	x	x	x	4	0	10	4	0	0	2	2	1	0	5	0	0	0	0	2	0	1	6	1	0	1	
+3	2	4	x	x	x	3	4	0	5	5	5	1	4	1	0	4	0	4	0	6	3	2	5	5	2	1	
+4	0	0	3	x	x	x	0	0	0	2	2	0	6	0	2	5	2	5	1	1	1	1	1	2	2	4	0
+1	0	10	4	0	x	x	x	5	2	0	0	0	0	2	0	0	0	0	2	1	0	0	2	0	5	1	
+2	2	4	0	0	5	x	x	x	1	2	2	1	4	10	10	2	5	5	0	5	0	0	0	0	10	0	0
+3	10	0	5	0	2	1	x	x	x	10	10	5	10	10	6	0	0	10	2	1	10	1	5	5	2	3	
+4	5	0	5	2	0	2	10	x	x	x	1	3	5	0	0	0	2	4	5	2	10	6	0	5	5	2	
+5	0	2	5	2	0	2	10	1	x	x	x	10	2	1	5	2	0	3	0	2	0	0	4	0	5	2	
+6	5	2	1	0	0	1	5	3	10	x	x	x	5	5	6	0	1	5	5	0	5	2	3	5	0	5	
+7	2	1	4	6	0	4	10	5	2	5	x	x	x	0	0	1	2	1	0	2	0	0	0	6	6	0	
+8	5	0	1	0	2	10	10	0	1	5	0	x	x	x	5	5	2	0	0	0	0	2	0	4	5	11	
+9	0	5	0	2	0	10	6	0	5	6	0	5	x	x	x	2	0	4	2	2	1	0	6	2	1	5	
+10	0	0	4	5	0	2	0	0	2	0	1	5	2	x	x	x	2	1	0	5	3	10	0	0	4	2	
+11	2	0	0	2	0	5	0	2	0	1	2	2	0	2	x	x	x	4	5	1	0	1	0	5	0	2	
+12	0	0	4	5	0	5	10	4	3	5	1	0	4	1	4	x	x	x	0	3	0	2	2	0	2	0	
+13	5	0	0	1	2	0	2	5	0	5	0	0	2	0	5	0	x	x	x	2	2	0	0	0	6	5	
+14	6	2	6	1	1	5	1	2	2	0	2	0	2	5	1	3	2	x	x	x	5	1	2	10	10	4	
+15	3	0	3	1	0	0	10	10	0	5	0	0	1	3	0	0	2	5	x	x	x	0	5	5	1	0	
+16	0	1	2	1	0	0	1	6	0	2	0	2	0	10	1	2	0	1	0	x	x	x	5	2	1	3	
+17	1	6	5	2	2	0	5	0	4	3	0	0	6	0	0	2	0	2	5	5	x	x	x	4	0	1	
+18	10	1	5	2	0	10	5	5	0	5	6	4	2	0	5	0	0	10	5	2	4	x	x	x	5	0	
+19	0	0	2	4	5	0	2	5	5	0	6	5	1	4	0	2	6	10	1	1	0	5	x	x	x	0	
+20	10	1	1	0	1	0	3	2	2	5	0	10	5	2	2	0	5	4	0	3	1	0	0	x	x	0	
+21	2	2	0	2	0	0	5	5	0	2	4	1	5	0	0	5	3	0	5	1	0	4	4	5	5	0	
+22	1	2	0	0	2	4	0	0	5	10	5	0	0	0	0	0	5	0	2	5	0	4	4	5	5	0	

Figura A.5 Informações sobre os fluxos entre os departamentos

Depois de todas as informações preenchidas, recomenda-se salvar estes dados em alguma pasta qualquer usando a opção “Salvar Tabelas” na página inicial. Estas tabelas serão salvas com uma extensão “.tab” no local escolhido.

Agora o sistema está pronto para iniciar, para tanto basta escolher uma solução inicial e uma metodologia na página inicial e depois clicar no botão “Iniciar Processo”. Os resultados serão apresentados na página “Resultados” em “Geral”. Nesta mesma página em “Log”, descreve passo a passo o que aconteceu até o resultado final. A posição descritiva dos departamentos e outras informações dos mesmos encontra-se na página “Posições” e a disposição final do leiaute graficamente se encontra na página “Layout”. Outros resultados interessantes da metodologia escolhida podem ser encontrados na página “Tabelas”. Neste local podem-se encontrar gráficos e outras informações diversas sobre a metodologia escolhida.

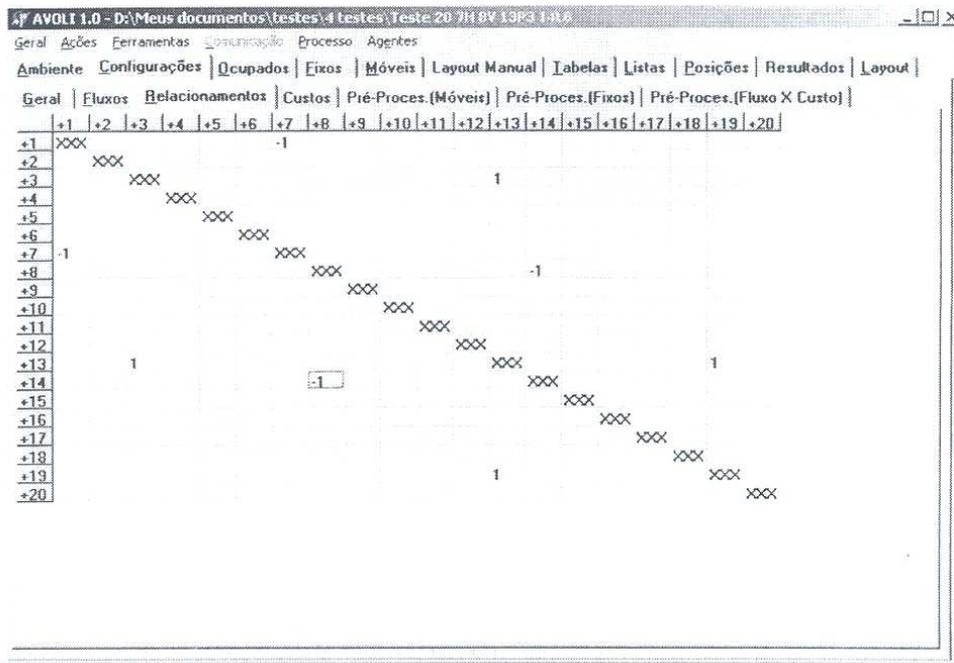


Figura A.6 Informações sobre relacionamento entre os departamentos

ANEXO B - Algoritmos e Fluxogramas

Neste anexo descrevem-se os algoritmos e fluxogramas dos principais processos do sistema AVOLI. Começamos pelo algoritmo que gera uma solução inicial viável. A partir deste leiaute inicial aplicam-se os métodos de otimização disponíveis no sistema AVOLI. Existem três tipos de metodologias para uma solução inicial sendo eles: método manual, método aleatório e método automático.

Na opção manual, a lista construtiva é inserida manualmente no sistema através da página 'Listas'. Este é a solução inicial mais simples, mas que não garante que seja uma solução inicial boa. A segunda opção seria a de listas aleatórias. Um pouco mais elaborada do que a primeira, mas que depende essencialmente do fator sorte. Nesta metodologia um número determinado pelo usuário de listas construtivas aleatórias é gerado. Destas listas o sistema escolhe automaticamente a que apresentar o melhor leiaute. A probabilidade de se encontrar uma boa lista inicial é diretamente proporcional ao número de listas construtivas geradas aleatoriamente.

A terceira opção é mais recomendada, pois oferece resultados iniciais melhores do que as outras metodologias de soluções manuais e aleatórias. Esta metodologia monta a lista construtiva pela simples ordenação dos departamentos considerando os seus fluxos. Ela possui um algoritmo mais complexo do que as anteriores e é mostrado na Figura B. 1.

Outro processo utilizado por todo o sistema é o algoritmo de construção realista (ACR). Este processo é sem dúvida alguma o mais complexo devido às inúmeras combinações que experimenta para tentar dispor todos os departamentos de uma melhor forma possível. Seu funcionamento é descrito passo a passo a seguir:

Passo 1:

Todos os departamentos fixos são posicionados de acordo com que foi previamente estabelecido no tabela dos departamentos fixos, definidos pelo usuário.

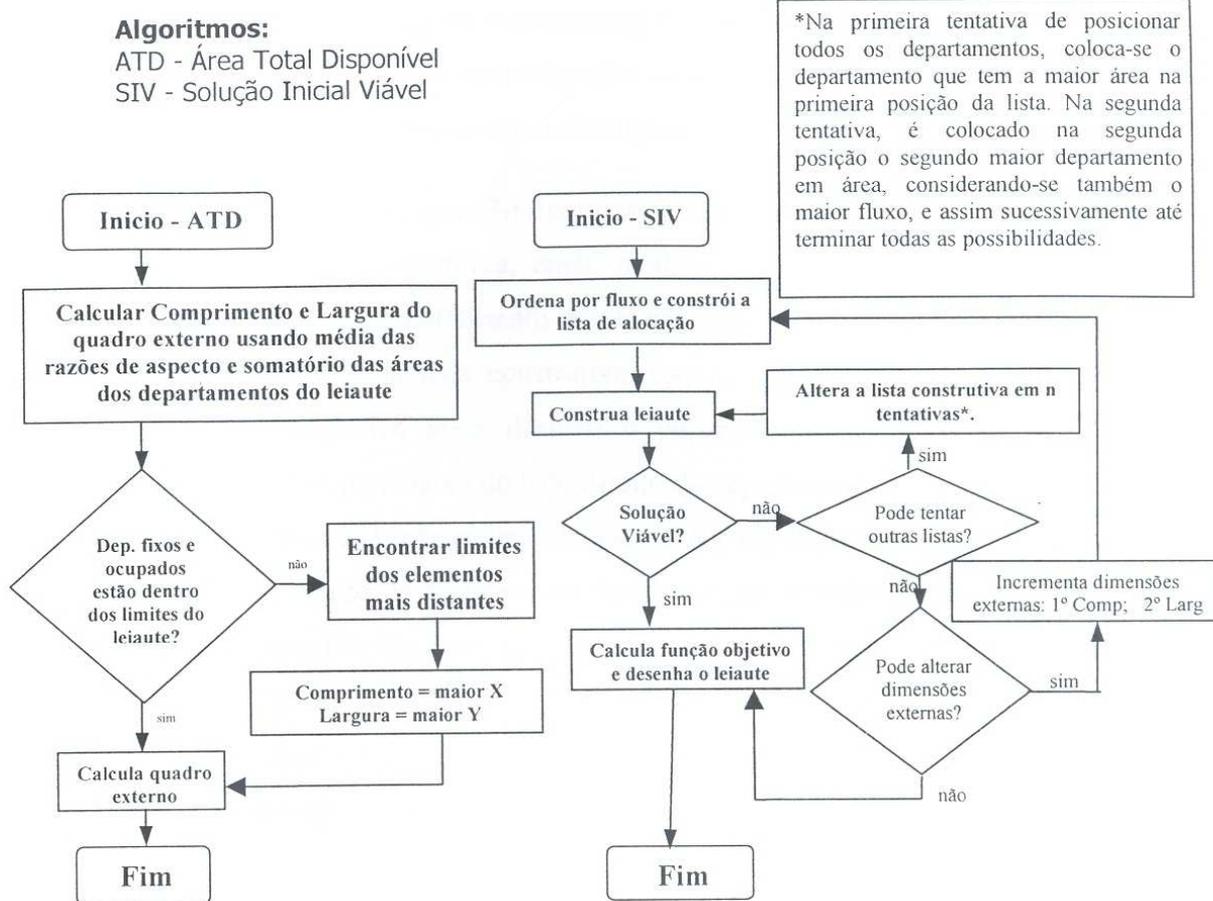


Figura B. 1 Fluxograma da solução inicial

Passo 2:

Se o departamento da lista da posição n é um departamento fixo então o algoritmo pula para o próximo item, pois este departamento já foi posicionado. Caso seja um departamento móvel então se tenta o seguinte:

- Se o elemento em questão for o primeiro da lista ou se o elemento à esquerda deste, não é válido ou nulo, então o procura-se o primeiro lugar vaga e válido no interior do leiaute para posicioná-lo, considerando todas as combinações de razão de aspecto e reduções possíveis previamente determinadas.
- Se o departamento em questão tiver outro departamento válido e já posicionado a sua esquerda na lista construtiva, então o algoritmo tenta posicionar em lugar válido dentro do leiaute o departamento móvel em questão no lado direito departamento que estava à esquerda na lista construtiva, tornando-os vizinho no leiaute. Para isto o algoritmo construtivo pode deslizar o departamento em questão de unidade em unidade de cima para baixo do lado direito do departamento que esta a sua esquerda na lista construtiva e já está posicionado até achar um lugar válido dentro do layout. Um lugar válido dentro do leiaute é um local que não se sobrepõe a outro departamento qualquer e que não faz com que o departamento a ser posicionado não perca mais de sua área útil do que permitida por sobreposição de áreas ocupadas. Para tanto, o algoritmo pode variar a razão de aspecto e a área total do departamento em cada deslizamento do mínimo até o máximo em n passos determinado na configuração inicial.
- Se mesmo assim o algoritmo construtivo não conseguiu posicionar o departamento em questão ao lado do departamento desejado, então ele tenta posicioná-lo em baixo do departamento desejado como vizinho. O processo segue os mesmos passos e combinações descritas anteriormente. Se mesmo assim falhar e não conseguir posicionar o departamento em questão, então tenta a esquerda e depois a cima.
- Se mesmo assim falhar na sua tentativa de posicionamento, então o algoritmo tenta achar o primeiro local vago válido no leiaute e tenta posicioná-lo, com todos os recursos de variação de razão de aspecto e reduções de áreas possíveis utilizados anteriormente.
- Se mesmo assim falhar, então outro algoritmo é acionado, o chamado Domino. Este algoritmo empurra todos os departamentos já posicionados para cima e para a esquerda, liberando espaços desperdiçados dentro do leiaute que ficavam entre os departamentos. Mostrou-se altamente eficiente e rápido, apresentando bons resultados.

Então repete todos os passos acima de tentar posicionar o departamento em questão começando pelo procedimento de posicionar a direita, seguindo a seqüência determinada: a baixo, à direita, à cima e finalmente em qualquer lugar.

- Se mesmo assim falhar nestas tentativas anteriores, então o algoritmo construtivo faz uma redução global em todos os departamentos posicionados em 1 passo dos n passos determinados na configuração inicial. E novamente tenta tudo novamente a partir do passo 1. O que implica que o algoritmo tem que começar do início da lista, ou seja, todos os departamentos da lista são novamente posicionados.
- Se mesmo assim não conseguir posicionar o departamento em questão, então o programa finalmente desiste de tentar posicionar o departamento móvel problemático. E simplesmente acrescenta um 'X' na frente do departamento na lista para indicar que não conseguiu posicioná-lo e não tenta mais posicioná-lo. E tenta posicionar o próximo da lista, repetindo todo o processo. Como o departamento não posicionado, ele passa para o status de departamento não válido e o departamento indicada a sua direita será posicionado no primeiro local vago válido no interior do leiaute.

Passo 3:

Depois que o algoritmo construtivo chega ao final da lista e posicionou todos os departamentos que conseguiu, então ele verifica a vizinhança para ver se não há nenhuma violação de adjacência. Ou seja, departamentos que deveriam estar longe um do outro, mas são vizinhos e vice-versa. Relação previamente estabelecida e configurada na tabela de relacionamentos.

Se houver alguma violação deste tipo, então o algoritmo tenta distanciar-los ou aproximá-los da lista em 1 passo dos n passos definidos na configuração inicial. Para tanto, o algoritmo construtivo, altera as posições dos departamentos em questão na própria lista construtiva e processa toda a lista novamente. Acredita-se que estes tipos de restrição entre os departamentos não sejam freqüentes ou em grande quantidade. Por isto o seu tratamento é de pouca complexidade e de fácil resolução, desde e claro, seja possível a separação.

Passo 4:

Depois de todos os processos acima citados, o algoritmo calcula a utilidade total e outros resultados relevantes. Bem como também chama o algoritmo que desenha graficamente o leiaute (Figura B. 2) para uma visualização gráfica.

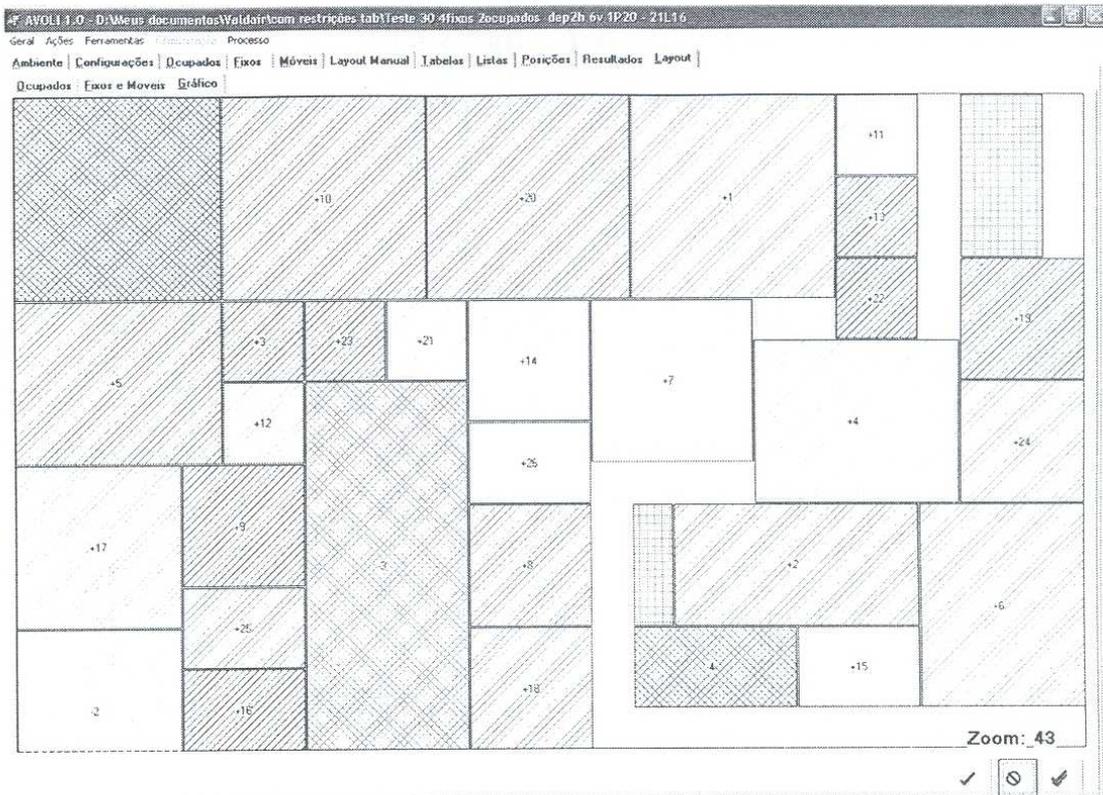
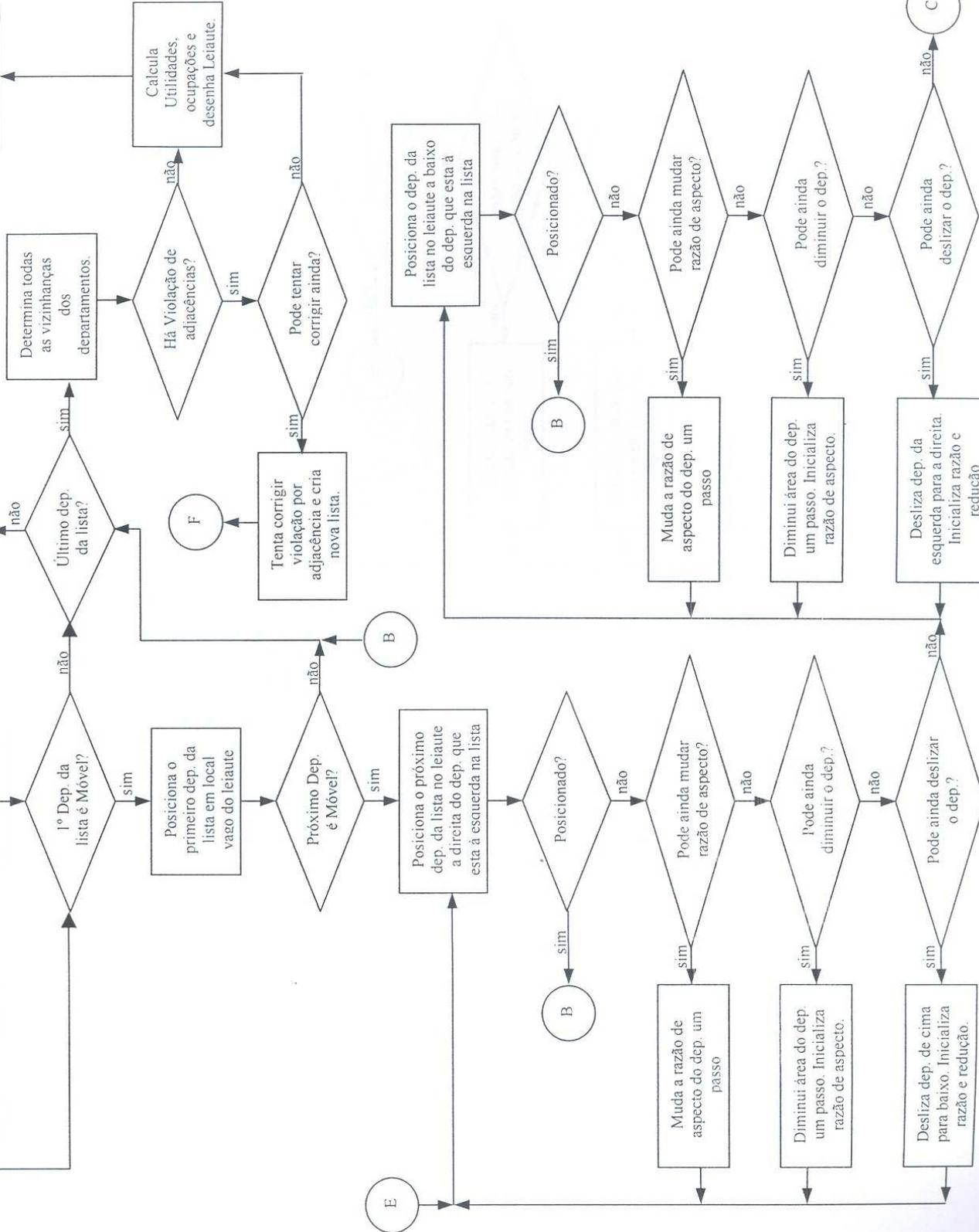
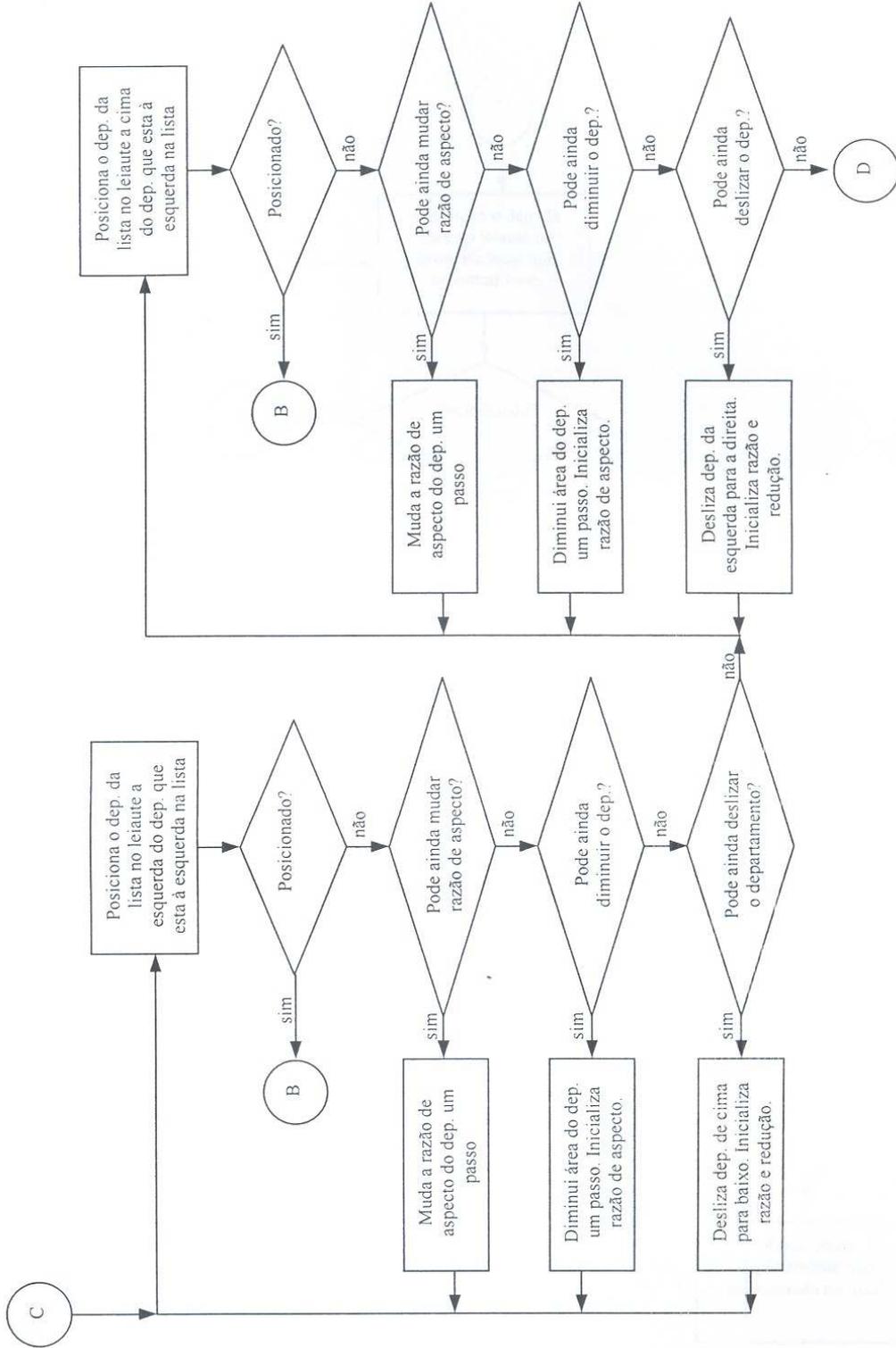


Figura B. 2 Leiaute mostrado graficamente





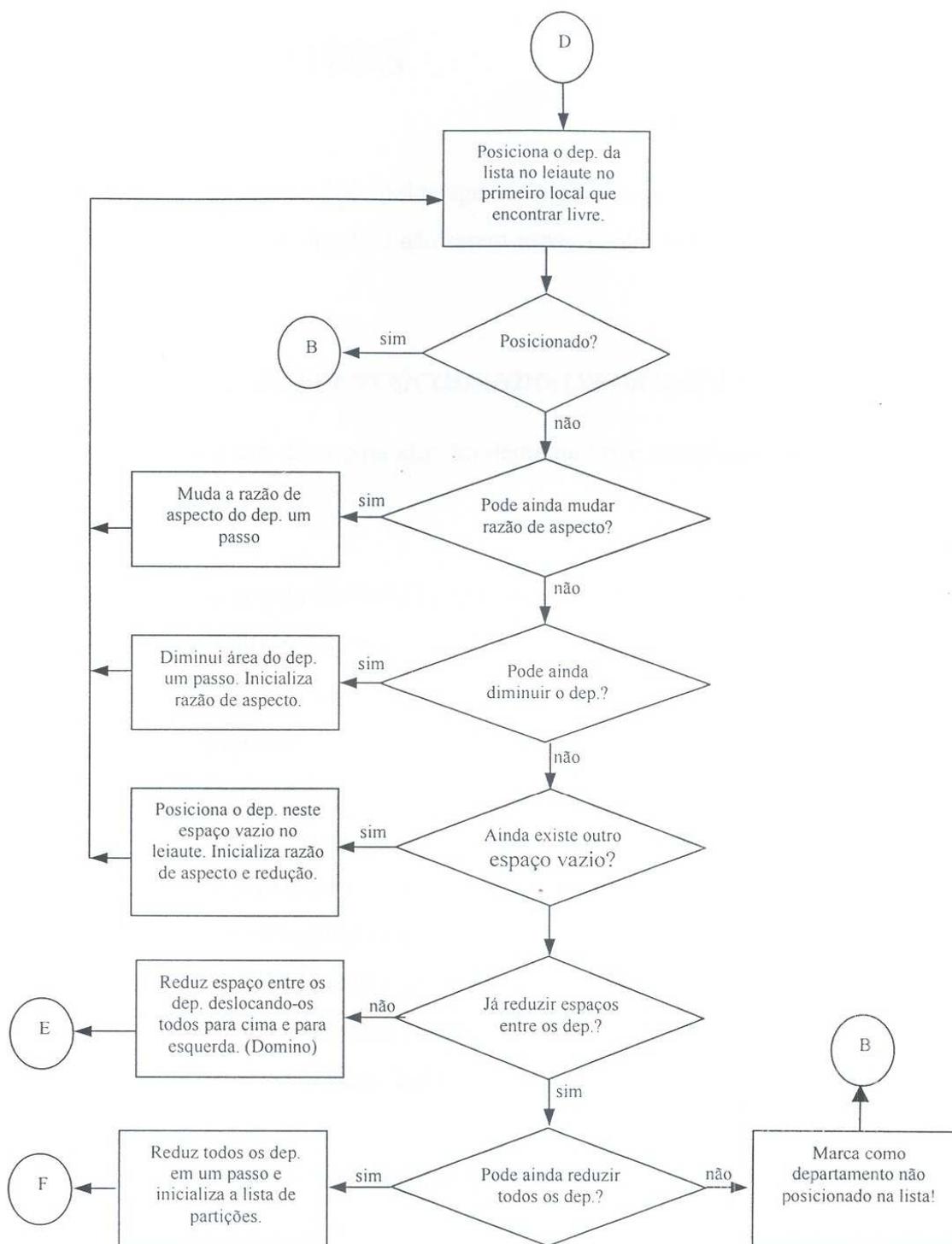


Figura B. 3 –Fluxograma do algoritmo construtivo

ANEXO C – Vocabulário Utilizado pelos Agentes

Segue os comandos reconhecidos pelos agentes e pelo mundo sendo que os comandos devem ser separados por ponto e vírgula e não terem espaço entre suas partes. Seu formato padrão é o seguinte:

DESTINÁRIO;REMETENTE;COMANDO;INFORMAÇÃO;...;INFORMAÇÃO;

A lista completa dos comandos implementados e reconhecidos pelos agentes e pelo mundo é a seguinte:

- ACEITA : o mundo (WORLD) informa ao agente destinatário que sua lista foi aceita no ciclo corrente e que ela modificará o mundo.
- CICLO : recupera o valor do ciclo atual do remetente e o copia no destinatário;
- CICLO? : o remetente pergunta ao destinatário qual é o atual ciclo que ele esta usando. O destinatário responde com o comando CICLO ao remetente.
- FLUXOS : recupera a tabela atualmente usada de fluxo do remetente na tabela de fluxos usada pelo destinatário, sobrescrevendo as informações;
- FLUXOS? : o remetente pergunta ao destinatário qual é sua tabela de fluxos atual. O destinatário responde com o comando FLUXO ao remetente;
- KILL : O remetente pode finalizar o destinatário. Este comando usado com o destinatário ALL finaliza todos os membros do mundo, menos o remetente em questão;
- LISTA : recupera a lista principal de construção do remetente e faz uma copia na lista usada pelo destinatário;
- LISTA? : o remetente pergunta ao destinatário qual é a atual lista que ele esta usando. O destinatário responde com o comando LISTA ao remetente;
- MELHOR : o mundo (WORLD) informa ao agente destinatário que além da sua lista ter sido aceita, também é a melhor solução encontrada até o presente ciclo.

- METODO : O remetente define qual a metodologia ou comportamento que o destinatário deve usar para solucionar o problema de leiaute ;
- RELACIONAMENTOS : recupera a tabela atualmente usada de relacionamentos do remetente e faz uma copia na tabela de relacionamentos usada pelo destinatário;
- RELACIONAMENTOS? : o remetente pergunta ao destinatário qual é sua tabela de relacionamentos atual. O destinatário responde com o comando RELACIONAMENTOS ao remetente;
- STATUS : O remetente por este comando ativa, desativa ou suspende as operações do destinatário;
- STATUS? : o remetente pergunta ao destinatário qual é o atual estado de atividade que ele se encontra. O destinatário responde com o comando STATUS! ao remetente.
- STATUS! : o remetente responde ao destinatário qual é o atual de atividade que ele se encontra. Este comando e puramente informativo e não modifica o destinatário.
- VIZINHANCAS : recupera a tabela de vizinhanças do remetente e faz uma copia na tabela de vizinhanças usada pelo destinatário;
- VIZINHANCAS? : o remetente pergunta ao destinatário qual é sua tabela de vizinhança atual. O destinatário responde com o comando VIZINHACAS ao remetente;
- ZERAMEMORIA : zera a memória de trocas. No caso do mundo a memória é a conhecida como N_a . Cuidado, pois a mundo perde todas as referencias sobre as participações dos agentes. No caso dos agentes a memória é a T_d .

ANEXO D - Os Melhores Leiautes

Obtidos Através de SMA

Neste anexo segue os demonstrativos gráficos conseguidos nas baterias de testes. Bem como as tabelas de fluxos utilizadas, as tabelas de relacionamentos de adjacências e as listas construtivas que geraram estes leiautes.

D.1. Primeira Bateria de Testes usando QAP (Quadratic Assignment Problem)

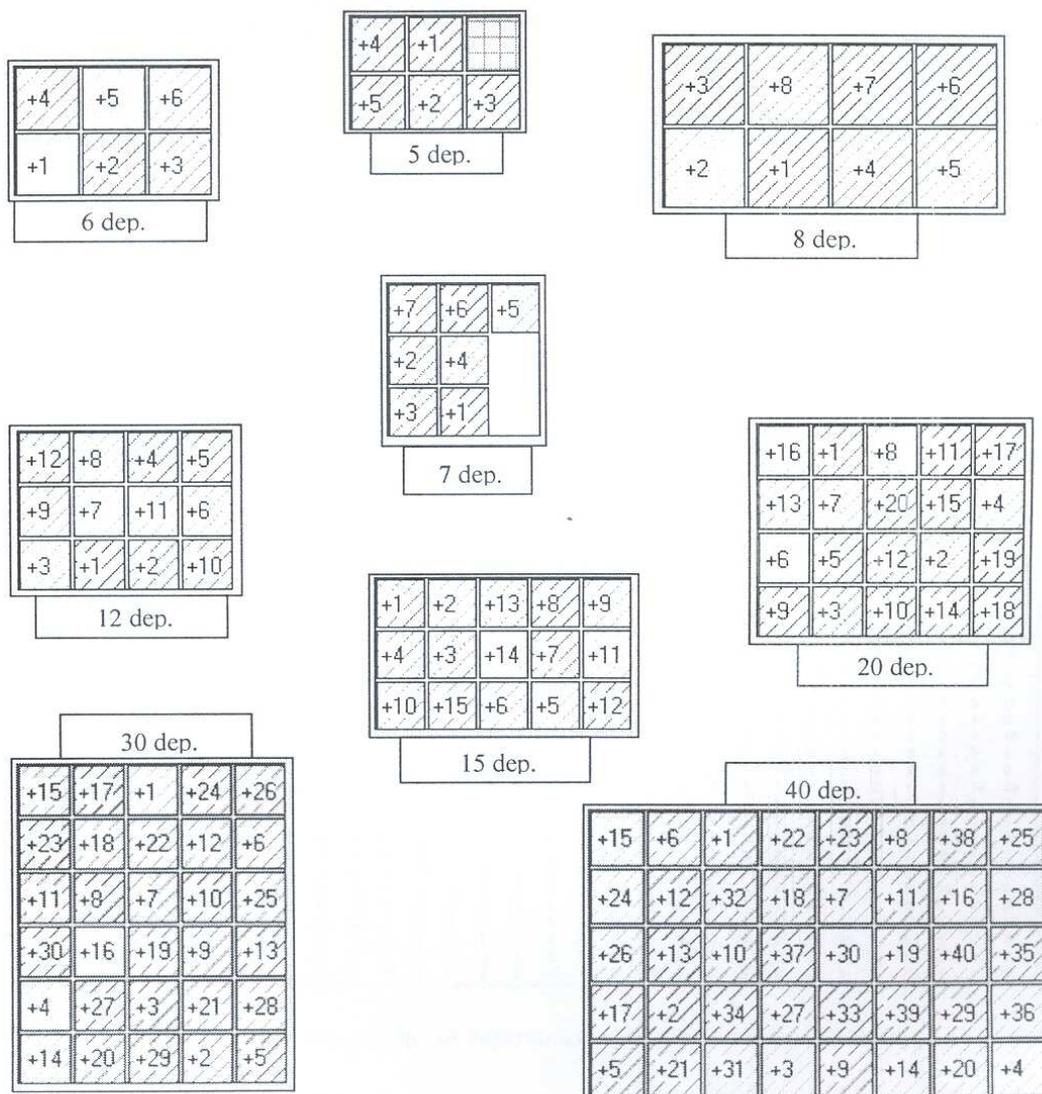


Figura D.1 - Leiaute dos melhores resultados encontrados pelos agentes na primeira bateria de teste

40 dep.

	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12	+13	+14	+15	+16	+17	+18	+19	+20	+21	+22	+23	+24	+25	+26	+27	+28	+29	+30	+31	+32	+33	+34	+35	+36	+37	+38	+39	+40		
+1	0	3	2	0	0	2	10	5	0	5	2	5	0	0	2	0	5	6	3	0	1	10	0	10	2	1	1	1	0	1	1	10	0	10	2	1	1	1	0	1		
+2	3	0	4	0	10	4	0	0	2	2	1	0	5	0	0	0	2	0	1	6	1	0	1	2	2	5	1	10	5	6	1	0	1	2	2	5	1	10	5	10	5	
+3	2	4	0	3	4	0	5	5	5	1	4	1	0	4	0	4	0	6	3	2	5	5	2	1	0	0	3	1	0	2	5	5	2	1	0	0	3	1	0	2	0	
+4	0	0	3	0	0	0	0	2	2	0	6	0	2	5	2	5	1	1	1	1	2	2	4	0	2	0	2	2	5	5	2	2	4	0	2	0	2	2	5	5	10	2
+5	0	10	4	0	0	5	2	0	0	0	0	2	0	0	0	2	1	0	0	2	0	5	1	0	2	1	0	2	1	0	2	1	0	2	1	0	2	1	0	2	1	
+6	2	4	0	0	5	0	1	2	2	1	4	10	10	2	5	5	0	5	0	0	0	10	0	0	0	4	0	10	1	1	0	10	0	0	0	4	0	4	0	10	1	1
+7	10	0	5	0	2	1	0	10	10	5	10	10	6	0	0	10	2	1	10	1	5	5	2	3	5	0	2	0	1	3	5	5	2	3	5	0	2	3	5	0	1	3
+8	5	0	5	2	0	2	10	0	1	3	5	0	0	0	2	4	5	2	10	6	0	5	2	5	0	2	5	0	5	5	0	2	0	5	5	2	5	0	5	5	0	2
+9	0	2	5	2	0	2	10	1	0	10	2	1	5	2	0	3	0	2	0	0	4	0	5	2	0	5	2	0	5	2	2	5	2	4	0	5	2	0	5	2	5	2
+10	5	2	1	0	0	1	5	3	10	0	5	5	6	0	1	5	5	0	5	2	3	5	0	5	2	10	10	1	5	2	3	5	0	5	2	10	10	1	5	2	10	
+11	2	1	4	6	0	4	10	5	2	5	0	0	0	1	2	1	0	2	0	9	0	6	6	0	4	5	3	2	2	10	0	6	6	0	4	5	3	2	2	10	10	
+12	5	0	1	0	2	10	10	0	1	5	0	0	5	5	2	0	0	0	2	0	0	2	0	4	5	10	1	0	0	0	1	0	4	5	10	1	0	0	0	0	1	
+13	0	5	0	2	0	10	6	0	5	6	0	5	0	2	0	4	2	2	1	0	6	2	1	5	5	0	0	1	5	5	6	2	1	5	5	0	0	1	5	5		
+14	0	0	4	5	0	2	0	0	2	0	1	5	2	0	2	1	0	5	3	10	0	0	4	2	0	0	4	2	5	5	0	0	4	2	0	0	4	2	5	5	10	
+15	2	0	0	2	0	5	0	2	0	1	2	2	0	2	0	4	5	1	0	1	0	5	0	2	0	0	5	1	1	0	0	5	0	2	0	0	5	1	1	0		
+16	10	0	4	5	0	5	10	4	3	5	1	0	4	1	4	0	0	3	0	2	2	0	2	0	2	0	5	0	5	2	5	10	2	0	2	0	5	0	5	2	5	10
+17	5	0	0	1	2	0	2	5	0	5	0	0	2	0	5	0	0	2	2	0	0	6	5	3	5	0	0	5	1	0	0	6	5	3	5	0	0	5	1	0	5	1
+18	6	2	6	1	1	5	1	2	2	0	2	0	2	5	1	3	2	0	5	1	2	10	10	4	0	0	5	0	0	2	10	10	4	0	0	5	0	0	5	0	0	
+19	3	0	3	1	0	0	10	10	0	5	0	0	1	3	0	0	2	5	0	0	5	1	0	5	2	1	2	10	10	5	1	0	5	1	0	5	2	1	2	10	10	
+20	1	2	1	0	0	1	6	0	2	0	2	0	10	1	2	0	1	0	0	5	2	1	3	1	5	6	5	5	3	5	2	1	3	1	5	6	5	5	3	0	2	
+21	1	6	5	2	2	0	5	0	4	3	0	0	6	0	2	0	2	5	0	4	0	1	0	0	5	0	0	0	5	0	0	0	0	0	0	0	1	0	1	0	1	
+22	10	1	5	2	0	10	5	5	0	5	6	4	2	0	5	0	0	10	5	2	4	0	5	0	4	4	5	0	2	5	0	2	0	0	1	10	3	0	2	0		
+23	0	0	2	4	5	0	2	5	0	6	5	1	4	0	2	6	10	1	1	0	5	0	0	4	4	1	0	2	2	0	3	0	3	0	1	1	5	0	3	0		
+24	10	1	1	0	1	0	3	2	2	5	0	10	5	2	2	0	5	4	0	3	1	0	0	0	5	5	0	1	0	0	0	4	0	0	0	1	0	7	0	4		
+25	2	2	0	2	0	0	5	5	0	2	4	1	5	0	0	5	3	0	5	1	0	4	4	5	0	1	0	10	1	0	0	5	0	3	0	1	2	9	0	5		
+26	1	2	0	0	2	4	0	0	5	10	5	0	0	0	0	0	5	0	2	5	0	4	4	5	1	0	0	0	0	0	6	0	0	0	1	5	0	0	6	0		
+27	1	5	3	2	1	0	2	5	2	10	3	0	0	4	5	5	0	5	1	6	0	5	1	0	5	0	2	2	0	2	0	5	1	2	8	5	4	7	6	3		
+28	1	1	1	2	0	10	0	5	2	1	2	0	1	2	1	2	0	2	5	5	0	0	1	10	0	0	0	2	2	3	8	5	6	7	8	9	0	1	8	0		
+29	0	10	0	5	2	1	1	0	5	5	2	0	5	5	1	5	0	10	5	0	2	2	0	1	0	0	2	0	2	0	5	1	2	8	5	4	7	6	3	0		
+30	1	5	2	5	1	1	3	2	2	10	1	5	5	0	10	1	0	10	3	0	5	2	0	0	0	10	2	2	0	1	2	3	4	5	6	7	8	9	0	1		
+31	1	6	5	2	2	0	5	0	4	3	0	0	6	0	2	0	2	5	0	0	0	0	0	0	0	0	0	9	3	0	1	0	0	0	1	0	2	0	0	1		
+32	10	1	5	2	0	10	5	5	0	5	6	4	2	0	5	0	0	10	5	2	1	2	3	4	5	6	7	8	5	2	0	0	1	5	0	2	1	0	5	2		
+33	0	0	2	4	5	0	2	5	0	6	5	1	4	0	2	6	10	1	1	0	0	0	0	0	0	7	5	1	3	0	1	0	9	0	2	0	2	9	3	0		
+34	10	1	1	0	1	0	3	2	2	5	0	10	5	2	2	0	5	4	0	3	3	0	3	0	3	0	6	6	2	4	1	5	9	0	0	0	2	5	0	7	4	
+35	2	2	0	0	2	0	0	5	5	0	2	4	1	5	0	0	5	3	0	5	1	1	1	1	1	1	1	8	5	6	2	2	2	2	0	1	0	3	6	0		
+36	1	2	0	0	2	4	0	0	5	10	5	0	0	0	0	0	5	0	2	5	1	1	1	1	1	1	8	5	6	2	2	2	2	0	1	0	2	4	7	0		
+37	1	5	3	2	1	0	2	5	2	10	3	0	0	4	5	5	0	5	1	6	0	10	1	0	2	5	3	9	4	7	0	1	0	5	0	1	0	2	4	7		
+38	1	1	1	2	0	10	0	5	2	1	2	0	1	2	1	2	0	2	5	1	3	5	1	3	5	7	9	0	2	0	7	8	0	0	2	0	1	0	2	0	1	
+39	0	10	0	5	2	1	1	0	5	5	2	0	5	5	1	5	5	0	0	0	0	0	0	0	1	1	6	9	1	5	9	7	5	3	4	1	0	9	0			
+40	1	5	2	5	1	1	3	2	2	10	1	5	5	0	10	1	0	10	3	1	2	3	4	5	6	7	8	3	0	1	2	3	4	5	6	7	8	9	0	0		

Tabela D.2 - Fluxos entre os departamentos da primeira bateria de testes (continuação)

Dep.	Utilidade	Lista construtora que gerou a melhor solução no teste 1
5	50	+4, +1, +2, +3, +5
6	86	+6, +5, +4, +1, +2, +3
7	128	+7, +6, +5, +4, +1, +2, +3
8	214	+6, +7, +8, +3, +2, +1, +4, +5
12	578	+5, +6, +10, +2, +1, +3, +9, +7, +12, +4, +8, +11
15	1150	+1, +2, +13, +8, +9, +11, +12, +5, +6, +15, +10, +4, +3, +14, +7
20	2596	+16, +1, +8, +11, +17, +4, +19, +18, +14, +10, +3, +9, +6, +5, +12, +2, +15, +20, +7, +13
30	6178	+15, +17, +1, +24, +12, +6, +13, +28, +2, +5, +21, +3, +4, +14, +20, +30, +19, +29, +25, +16, +8, +27, +23, +11, +7, +9, +26, +10, +22, +18
40	14126	+5, +24, +12, +6, +18, +23, +38, +15, +25, +16, +35, +4, +20, +14, +27, +36, +9, +31, +21, +2, +13, +10, +39, +19, +30, +29, +28, +37, +40, +33, +34, +3, +17, +26, +1, +7, +32, +22, +11, +8

Tabela D.3 - Llistas construtoras da primeira bateria de testes

D.2. Segunda Bateria de Testes usando Departamentos de Tamanhos Variados



Figura D.2 - Leiaute dos melhores resultados encontrados pelos agentes na segunda baterias de testes.

	+1	+2	+3	+4	+5	+6	+7	+8
+1	0	5	2	4	1	0	0	6
+2	5	0	3	0	2	2	2	0
+3	2	3	0	0	0	0	0	5
+4	4	0	0	0	5	2	2	10
+5	1	2	0	5	0	10	0	0
+6	0	2	0	2	10	0	5	1
+7	0	2	0	2	0	5	0	10
+8	6	0	5	10	0	1	10	0

8 dep.

	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12
+1	0	5	2	4	1	0	0	6	2	1	1	1
+2	5	0	3	0	2	2	2	0	4	5	0	0
+3	2	3	0	0	0	0	0	5	5	2	2	2
+4	4	0	0	0	5	2	2	10	0	0	5	5
+5	1	2	0	5	0	10	0	0	0	5	1	1
+6	0	2	0	2	10	0	5	1	1	5	4	0
+7	0	2	0	2	0	5	0	10	5	2	3	3
+8	6	0	5	10	0	1	10	0	0	0	5	0
+9	2	4	5	0	0	1	5	0	0	0	10	10
+10	1	5	2	0	5	5	2	0	0	0	5	0
+11	1	0	2	5	1	4	3	5	10	5	0	2
+12	1	0	2	5	1	0	3	0	10	0	2	0

12 dep.

	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12	+13	+14	+15
+1	0	10	0	5	1	0	1	2	2	2	2	0	4	0	0
+2	10	0	1	3	2	2	2	3	2	0	2	0	10	5	0
+3	0	1	0	10	2	0	2	5	4	5	2	2	5	5	5
+4	5	3	10	0	1	1	5	0	0	2	1	0	2	5	0
+5	1	2	2	1	0	3	5	5	5	1	0	3	0	5	5
+6	0	2	0	1	3	0	2	2	1	5	0	0	2	5	10
+7	1	2	2	5	5	2	0	6	0	1	5	5	5	1	0
+8	2	3	5	0	5	2	6	0	5	2	10	0	5	0	0
+9	2	2	4	0	5	1	0	5	0	0	10	5	10	0	2
+10	2	0	5	2	1	5	1	2	0	0	0	4	0	0	5
+11	2	2	2	1	0	0	5	10	10	0	0	5	0	5	0
+12	0	0	2	0	3	0	5	0	5	4	5	0	3	3	0
+13	4	10	5	2	0	2	5	5	10	0	0	3	0	10	2
+14	0	5	5	5	5	5	1	0	0	0	5	3	10	0	4
+15	0	0	5	0	5	10	0	0	2	5	0	0	2	4	0

15 dep.

	+1	+2	+3	+4	+5	+6	+7
+1	0	5	2	4	1	0	0
+2	5	0	3	0	2	2	2
+3	2	3	0	0	1	0	2
+4	4	0	1	0	5	2	2
+5	1	2	0	5	0	10	0
+6	0	2	2	2	10	0	5
+7	0	2	5	2	0	5	0

7 dep.

	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12	+13	+14	+15	+16	+17	+18	+19	+20
+1	0	5	0	5	2	10	3	1	5	5	5	0	0	5	4	4	0	0	1	
+2	0	0	3	10	5	1	5	1	2	4	2	5	0	10	10	3	0	5	10	5
+3	5	3	0	2	0	5	2	4	4	5	0	0	0	5	1	0	0	5	0	0
+4	0	10	2	0	1	0	5	2	1	0	10	2	2	0	2	1	5	2	5	5
+5	5	5	0	1	0	5	6	5	2	5	2	0	5	1	1	1	5	2	5	1
+6	2	1	5	0	5	0	5	2	1	6	0	0	10	0	2	0	1	0	1	5
+7	10	5	2	5	6	5	0	0	0	0	5	10	2	2	5	1	2	1	0	10
+8	3	1	4	2	5	2	0	0	1	1	10	10	2	2	0	2	5	2	2	10
+9	1	2	4	1	2	1	0	1	0	2	0	3	5	5	0	5	0	0	0	2
+10	5	4	5	0	5	6	0	1	2	0	5	5	0	5	1	0	0	5	5	2
+11	5	2	0	10	2	0	5	10	0	5	0	5	2	5	1	10	0	2	2	5
+12	5	5	0	2	0	0	10	10	3	5	5	0	2	10	5	0	1	1	2	5
+13	0	0	0	2	5	10	2	2	5	0	2	2	0	2	2	1	0	0	0	5
+14	0	10	5	0	1	0	2	0	5	5	5	10	2	0	5	5	1	5	5	0
+15	5	10	1	2	1	2	5	10	0	1	1	5	2	5	0	3	0	5	10	10
+16	4	3	0	1	1	0	1	2	5	0	10	0	1	5	3	0	0	0	2	0
+17	4	0	0	5	5	1	2	5	0	0	0	1	0	1	0	0	0	5	2	0
+18	0	5	5	2	2	0	1	2	0	5	2	1	0	5	5	0	5	0	1	1
+19	0	10	0	5	5	1	0	2	0	5	2	2	0	5	10	2	2	1	0	6
+20	1	5	0	5	1	5	10	10	2	2	5	5	5	0	10	0	0	1	6	0

20 dep.

	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12	+13	+14	+15	+16	+17	+18	+19	+20
+1	0	5	2	4	1	0	0	6	2	1	1	1								
+2	5	0	3	0	2	2	2	0	4	5	0	0								
+3	2	3	0	0	0	0	0	5	5	2	2	2								
+4	4	0	1	0	5	2	2	10	0	0	5	5								
+5	1	2	0	5	0	10	0	0	0	5	1	1								
+6	0	2	0	2	10	0	5	1	1	5	4	0								
+7	0	2	0	2	0	5	0	10	5	2	3	3								
+8	6	0	5	10	0	1	10	0	0	0	5	0								
+9	2	4	5	0	0	1	5	0	0	0	10	10								
+10	1	5	2	0	5	5	2	0	0	0	0	5	0							
+11	1	0	2	5	1	4	3	5	10	5	0	2	0							
+12	1	0	2	5	1	0	3	0	10	0	2	0	0							

6 dep.

	+1	+2	+3	+4	+5	+6
+1	0	5	2	4	1	0
+2	5	0	3	0	2	2
+3	2	3	0	0	0	0
+4	4	0	0	0	5	2
+5	1	2	0	5	0	10
+6	0	2	0	2	10	0

5 dep.

	+1	+2	+3	+4	+5
+1	0	5	2	4	1
+2	5	0	3	0	2
+3	2	3	0	0	0
+4	4	0	0	0	5
+5	1	2	0	5	0

30 dep.

	+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12	+13	+14	+15	+16	+17	+18	+19	+20	+21	+22	+23	+24	+25	+26	+27	+28	+29	+30
+1	0	3	2	0	0	2	10	5	0	5	2	5	0	2	0	5	6	3	0	1	10	0	10	2	1	1	1	0	1	
+2	3	0	4	0	10	4	0	0	2	2	1	0	5	0	0	0	2	0	1	6	1	0	1	2	2	5	1	10	5	
+3	2	4	0	3	4	0	5	5	1	4	1	0	4	0	4	0	6	3	2	5	5	2	1	0	0	3	1	0	2	
+4	0	0	3	0	0	0	0	2	2	0	6	0	2	5	1	1	1	1	1	2	2	4	0	2	0	2	2	5	5	
+5	0	10	4	0	0	5	2	0	0	0	0	2	0	0	0	2	1	0	0	2	0	5	1	0	2	1	0	2	1	
+6	2	4	0	0	5	0	1	2	2	1	4	10	10	2	5	5	0	5	0	10	0	10	0	0	4	0	10	1	1	
+7	10	0	5	0	2	1	0	10	10	5	10	10	6	0	0	10	2	1	10	1	5	5	2	3	5	0	2	0	1	3
+8	5	0	5	2	0	2	10	0	1	3	5	0	0	0	2	4	5	2	10	6	0	5	5	2	5	0	5	5	0	2
+9	0	2	5	2	0	2	10	1	0	10	2	1	5	2	0	3	0	2	0	4	0	5	2	0	5	2	2	5	2	
+10	5	2	1	0	0	1	5	3	10	0	5	5	6	0	1	5	5	0	5	2	3	5	0	5	2	10	10	1	5	2
+11	2	1	4	6	0	4	10	5	2	5	0	0	0	1	2	1	0	2	0	0	0	6	6	0	4	5	3	2	2	10
+12	5	0	1	0	2	10	10	0	1	5	0	0	5	5	2	0	0	0	0	2	0	4	5	10	1	0	0	0	0	1
+13	0	5	0	2	0	10	6	0	5	6	0	5	0	2	0	4	2	2	1	0	6	2	1	5	5	0	0	1	5	5
+14	0	0	4	5	0	2	0	0	2	0	1	5	2	0	2	1	0	5	3	10	0	0	4	2	0	0	4	2	5	5
+15	2	0	0	2	0	5	0	2	0	1	2	2	0	2	0	4	5	1	0	1	0	5	0	2	0	0	5	1	1	0
+16	0	4	5	0	5	10	4	3	5	1	0	4	1	4	0	0	3	0	2	2	0	2	0	5	0	5	2	5	10	10
+17	5	0	0	1	2	0	2	5	0	5	0	0	2	0	5	0	2	2	0	0	6	5	3	5	0	0	5	0	5	1
+18	6	2																												

Dep.	Utilidade	Lista construtora que gerou a melhor solução no teste 2
5	232	+5, +2, +1, +3, +4
6	376	+3, +2, +1, +5, +6, +4
7	676	+3, +1, +4, +6, +2, +7, +5
8	974	+3, +1, +6, +8, +2, +5, +4, +7
12	2444	+3, +1, +10, +2, +11, +8, +9, +5, +6, +4, +12, +7
15	5022	+3, +12, +9, +11, +8, +1, +5, +14, +6, +13, +7, +4, +15, +10, +2
20	9917	+5, +18, +1, +3, +6, +10, +14, +16, +8, +20, +7, +11, +19, +2, +4, +17, +15, +12, +13, +9
30	22090	+5, +3, +18, +10, +12, +24, +6, +15, +8, +11, +9, +21, +28, +14, +29, +23, +27, +2, +13, +19, +25, +7, +30, +4, +22, +26, +1, +20, +16, +17

Tabela D.5 - Listas construtoras da segunda bateria de testes

Móveis						
	Área	Orientação	Perda A. Morta (%)	R.Asp.Mín.	R.Asp.Máx.	Red.Máx.Área (%)
+1	24		0	,8	1	0
+2	16		0	,75	1,15	0
+3	36		0	,6	1,85	0
+4	8		0	,3	1,1	0
+5	21		0	,9	1,18	0
+6	17,5		0	,5	1	0
+7	3,6		0	,3	1,4	0
+8	15,4		0	,6	1,25	0
+9	20		0	,9	1	0
+10	19,5		0	1,2	1,8	0
+11	16		0	,85	1,1	0
+12	9		0	,9	1	0
+13	9		0	,8	1,1	0
+14	25		0	,92	1,05	0
+15	4		0	,85	1,15	0
+16	3		0	,9	1	0
+17	4		0	1	1	0
+18	9		0	1	1	0
+19	4,5		0	,7	1,1	0
+20	5		0	,5	1,5	0
+21	16		0	,85	1,1	0
+22	9		0	,9	1	0
+23	9		0	,8	1,1	0
+24	25		0	,92	1,05	0
+25	4		0	,85	1,15	0
+26	3		0	,9	1	0
+27	4		0	1	1	0
+28	9		0	1	1	0
+29	4,5		0	,7	1,1	0
+30	5		0	,5	1,5	0

Tabela D.6 - Informações sobre departamentos móveis da segunda bateria de testes

D.3. Terceira Bateria de Testes usando Departamentos de Tamanhos Variados com Restrições de adjacência e orientação



Figura D. 3 Leiaute dos melhores resultados encontrados pelos agentes na terceira baterias de testes.

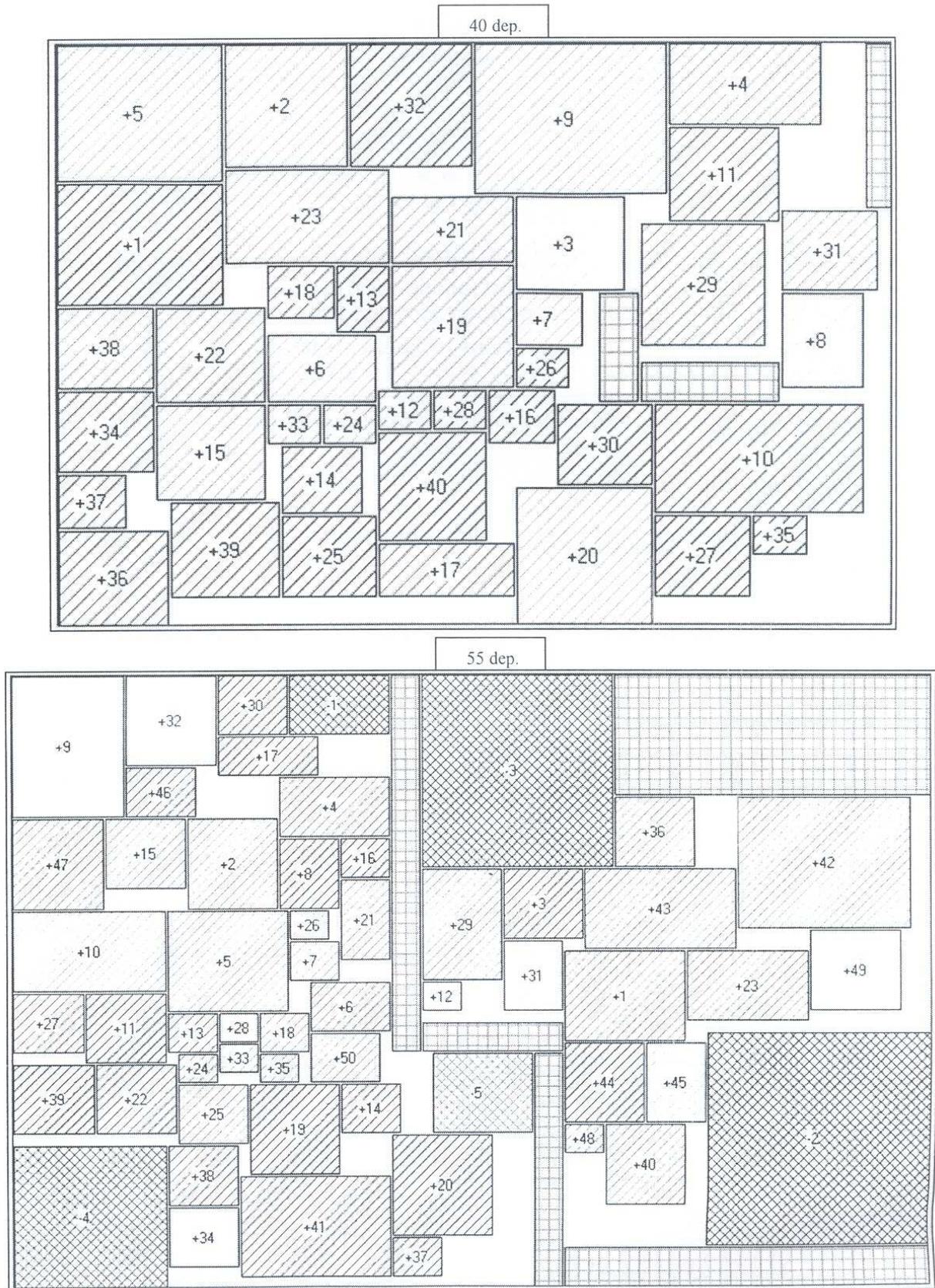


Figura D. 4 Leiaute dos melhores resultados encontrados pelos agentes na terceira baterias de testes (cont.)

Dcp.	Utilidade	Lista construtora que gerou a melhor solução no teste 3
8	898	+3, +1, +8, +7, +4, +5, +2, +6
12	2657	+5, +2, +1, +8, +7, +6, +10, +11, +3, +9, +12, +4
15	5270	+9, +11, +12, +6, +10, +2, +1, +3, +5, +8, +14, +13, +4, +15, +7
20	9673	+3, +14, +18, +10, +5, +17, +8, +11, +6, +9, +20, +12, +13, +2, +16, +1, +19, +15, +4, +7
30	22548	-3, +5, +6, +25, +18, +2, +7, +17, +24, +4, +15, +1, +20, +22, +13, -4, +19, +14, +10, +23, +12, +3, +11, +9, +16, -1, +8, +26, +21, -2
40	75620	+5, +2, +32, +9, +4, +11, +31, +1, +23, +21, +19, +3, +38, +22, +18, +6, +12, +28, +16, +30, +10, +20, +27, +35, +34, +15, +33, +24, +40, +17, +25, +39, +36, +37, +14, +13, +7, +26, +29, +8
55	125393	+47, -3, +42, +23, +41, +9, +49, +1, +19, +20, +5, +2, +25, -2, +10, +4, +43, +44, -4, +27, +29, +13, -1, +39, +31, +6, +21, +50, +34, +37, +17, +12, +28, +35, +26, +24, +11, +22, +3, +15, +16, +46, +45, +36, -5, +8, +33, +30, +40, +32, +38, +7, +48, +18, +14

Tabela D.7 - Listas construtoras da terceira bateria de testes

Fixos							
	Área	Xinício	Yinício	Orientação	RA Morta	R.Asp.Mín.	R.Asp.Máx.
-1	60	28	1		0	,5	1
-2	450	69	36		0	1	1
-3	360	41	1		0	1	1
-4	200	1	47		0	1	1
-5	80	42	38	h	0	,6	1,2

Tabela D.8 - Informações sobre departamentos fixos da terceira bateria de testes

Ocupados				
	Xinício	Yinício	Largura	Comprimento
1	38	1	37	3
2	41	35	3	14
3	60	1	12	31
4	52	38	23	3
5	55	57	4	36

Tabela D.9 – Informações sobre áreas ocupadas da terceira bateria de testes

Móveis						
	Área	Orientação	Perda A. Morta [%]	R.Asp.Mín.	R.Asp.Máx.	
+1	100		0	0,7	1	
+2	80		0	1	1	
+3	50		0	0,7	1,3	
+4	60	h	0	0,5	0,8	
+5	120		0	0,9	1	
+6	40		0	0,6	1	
+7	20		0	0,7	1,4	
+8	40	v	0	1	1	
+9	150	v	0	0,8	1,1	
+10	120	h	0	0,5	1,5	
+11	50		0	0,7	1,1	
+12	10		0	0,8	1,2	
+13	20		0	0,95	1,5	
+14	30		0	0,75	1,25	
+15	50		0	0,9	1,1	
+16	20		0	0,8	1,5	
+17	40	h	0	0,4	1,4	
+18	20		0	0,9	1,9	
+19	80	v	0	1	1	
+20	100		0	0,95	1,15	
+21	40		0	0,5	1,5	
+22	50		0	1	1,1	
+23	80		0	0,6	1	
+24	10		0	0,9	1	
+25	40		0	0,8	1,1	
+26	10		0	0,5	1,2	
+27	40		0	0,8	1	
+28	10		0	0,5	1,3	
+29	80	v	0	,7	1,05	
+30	40		0	0,9	1,1	
+31	40		0	,9	1,2	
+32	80		0	,9	1,5	
+33	10		0	,9	1,1	
+34	40		0	,8	1	
+35	10		0	,8	1,3	
+36	50		0	,8	1,2	
+37	20		0	,8	1,5	
+38	40		0	,9	1,6	
+39	50		0	,9	1,1	
+40	60		0	1	1,8	
+41	150		0	,7	1,6	
+42	210		0	,3	1	
+43	110		0	,5	1	
+44	60		0	,85	1,6	
+45	45		0	,9	1,2	
+46	30		0	,7	1,05	
+47	80		0	,95	1,15	
+48	10		0	,8	1,1	
+49	65		0	,75	1,2	
+50	30		0	,6	1	

Tabela D.10 – Informações sobre departamentos móveis da terceira bateria de testes

40 dep.

+1	+2	+3	+4	+5	+6	+7	+8	+9	+10	+11	+12	+13	+14	+15	+16	+17	+18	+19	+20	+21	+22	+23	+24	+25	+26	+27	+28	+29	+30	+31	+32	+33	+34	+35	+36	+37	+38	+39	+40	
3	2	0	0	2	10	5	0	2	1	0	5	0	2	0	2	0	5	6	3	0	1	10	0	10	2	1	1	1	0	1	1	8	0	0	0	0	0	0	0	1
3	0	4	0	10	4	0	0	2	1	0	5	0	0	0	0	2	0	1	6	1	0	1	0	1	2	2	5	1	10	5	2	4	5	0	0	0	0	0	2	
2	4	0	3	4	0	5	5	1	4	1	0	4	0	4	0	6	3	2	5	5	2	1	0	0	3	1	0	2	3	7	0	0	0	0	0	0	0	0	0	
0	0	3	0	0	0	2	2	0	6	0	2	5	2	5	1	1	1	2	2	4	0	2	0	2	2	2	5	5	4	8	2	0	0	0	0	0	0	0	1	
0	10	4	0	0	5	2	0	0	2	0	0	0	0	0	2	1	0	0	2	0	5	1	0	2	1	0	2	1	5	1	0	0	0	0	0	0	0	5		
2	4	0	0	5	0	1	2	2	1	4	10	10	2	5	5	0	5	0	5	0	10	0	0	0	4	0	10	1	1	6	1	5	0	0	0	0	0	0		
10	5	0	5	2	1	0	10	10	5	10	6	0	0	2	10	10	6	0	5	5	2	3	5	0	5	0	2	0	1	3	7	1	0	0	0	0	0	8		
5	0	5	2	0	2	10	0	1	3	5	0	0	2	4	5	2	10	6	0	5	5	2	5	0	5	5	0	2	9	0	1	0	0	0	0	0	0	0		
2	5	2	0	2	10	1	0	10	2	1	5	2	0	3	0	2	0	4	0	5	2	0	5	2	2	2	5	2	8	2	0	4	0	0	0	0	0	7		
10	5	2	1	0	0	1	5	3	10	0	5	6	0	1	5	5	0	5	2	3	5	0	5	2	10	10	1	5	2	9	2	1	5	0	0	0	0	5		
12	1	4	6	0	4	10	5	2	5	0	0	1	2	1	0	2	0	0	0	6	0	4	5	3	2	2	10	8	3	0	2	0	0	0	0	0	1			
15	0	1	0	2	10	10	0	1	5	0	5	5	2	0	0	0	2	0	4	5	10	1	0	0	0	0	1	5	5	1	1	0	0	0	0	0	0	2		
13	0	5	0	2	0	10	6	0	5	6	0	5	0	2	0	4	2	1	0	6	2	1	5	5	0	0	1	5	6	0	5	2	5	4	8	0	0	0		
14	0	0	4	5	0	2	0	0	2	0	1	5	2	0	2	1	0	5	3	10	0	0	4	2	0	0	4	2	5	3	1	0	6	0	0	0	0	9		
15	2	0	0	2	0	5	0	2	1	2	2	0	4	5	1	0	1	0	1	0	5	0	2	0	0	5	1	0	0	4	7	9	0	0	0	0	0	0		
16	0	0	4	5	0	5	10	4	3	5	1	0	4	0	0	3	0	3	0	2	0	2	0	5	0	5	2	5	10	0	8	8	0	0	0	0	0	5		
17	5	0	1	2	0	2	5	0	5	0	2	0	5	0	2	2	0	2	0	0	6	5	3	5	0	0	5	0	5	2	5	2	5	0	0	0	0	0		
18	6	2	6	1	1	5	1	2	2	0	2	5	1	3	2	0	5	1	2	10	10	4	0	0	5	0	0	5	0	0	2	1	4	0	1	2	3	6	2	
19	3	0	3	1	0	0	10	10	0	5	0	1	3	0	0	2	5	0	0	5	5	1	0	5	2	1	2	10	10	0	0	2	0	0	0	0	0	7		
20	0	1	2	1	0	0	1	6	0	2	0	2	0	10	1	2	0	1	0	0	5	2	1	3	1	5	6	5	5	3	0	0	4	1	0	0	2	0	0	
21	1	6	5	2	2	0	5	0	4	3	0	0	6	0	0	2	2	5	5	0	4	0	1	0	0	0	5	0	0	0	0	5	0	0	0	0	0	0	1	
22	10	1	5	2	0	10	5	5	0	5	6	4	2	0	5	0	10	5	2	4	0	5	0	4	4	5	0	2	5	0	0	4	0	4	0	0	0	0	0	
23	0	0	2	4	5	0	2	5	5	0	6	5	1	4	0	2	6	10	1	1	0	5	0	4	4	1	0	2	2	0	4	0	0	4	0	0	0	0	4	
24	10	1	1	0	1	0	3	2	2	5	0	10	5	2	2	0	5	4	0	3	1	0	0	0	5	0	1	0	0	0	0	7	1	0	0	0	0	5		
25	2	2	0	2	0	0	5	5	0	2	4	1	5	0	0	5	3	0	5	1	0	4	4	5	0	1	0	1	0	0	2	0	0	0	0	0	0	7	8	
26	1	2	0	2	4	0	0	5	10	5	0	0	0	0	0	5	0	2	5	0	4	4	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	8	
27	1	5	3	2	1	0	2	5	2	10	3	0	0	4	5	5	0	5	1	6	0	5	1	0	0	0	0	0	0	10	0	0	1	0	0	0	0	0	9	
28	1	1	2	0	10	0	5	2	1	2	0	1	2	1	2	0	0	2	5	5	0	0	1	10	0	0	0	2	2	0	0	0	0	0	0	0	0	0	0	0
29	0	10	0	5	2	1	1	0	5	2	0	5	1	5	5	0	10	5	0	2	5	0	2	0	1	0	0	2	0	2	0	0	0	0	0	0	0	0	0	
30	1	5	2	5	1	1	3	2	2	10	1	5	5	0	10	1	0	10	3	0	5	2	0	0	1	0	0	2	0	2	0	0	0	0	0	0	0	0	2	
31	1	2	3	4	5	6	7	8	9	8	5	6	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	
32	1	4	7	8	1	1	0	2	2	3	5	0	1	4	0	5	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	
33	8	5	0	2	6	5	0	1	0	1	0	1	5	6	7	8	2	1	6	4	5	4	4	7	2	0	0	0	0	0	0	0	0	0	0	0	0	0	4	
34	0	0	0	0	0	0	0	0	4	5	2	1	5	6	9	8	5	4	2	1	0	4	0	1	2	0	1	0	0	0	0	0	0	0	0	0	0	0	1	
35	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
36	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
37	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	2	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
38	0	0	0	0	0	0	0	0	0	0	0	8	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	
39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	
40	1	2	0	1	5	0	8	0	7	5	1	2	0	9	0	5	0	2	7	0	1	0	4	5	7	8	9	0	0	2	4	1	4	0	1	0	2	5	4	0

Tabela D.12 - Fluxos entre os departamentos da terceira bateria de testes (continuação)

55 dep.

Table with columns labeled from +1 to +50 and rows labeled from +1 to +50. The table contains numerical data representing fluxes between departments. Each cell contains a number representing the flux between two specific departments.

Tabela D.13 - Fluxos entre os departamentos da terceira bateria de testes (continuação)