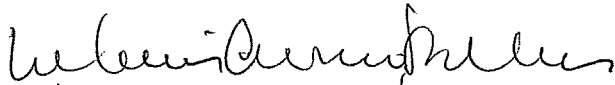


OTIMIZAÇÃO DO CONJUNTO DE PEÇAS
PARA AUTOMONTAGEM

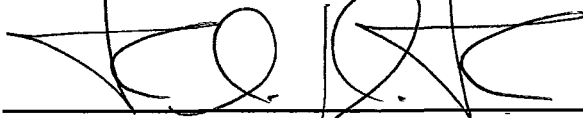
Fabio Rocha Jimenez Vieira

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓSGRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

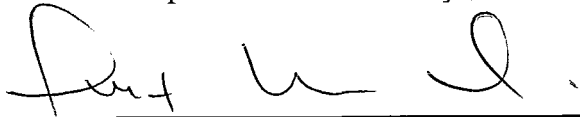
Aprovada por:



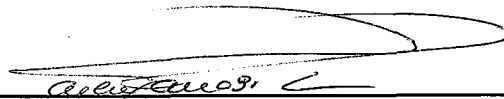
Prof. Valmir Carneiro Barbosa, Ph.D.



Prof. Felipe Maia Galvão França, Ph.D.



Prof. Luiz Satoru Ochi, D.Sc.



Profa. Marley Maria Bernardes Rebuzzi Vellasco, Ph.D.

RIO DE JANEIRO, RJ -BRASIL

SETEMBRO DE 2007

VIEIRA, FABIO ROCHA JIMENEZ

Otimização do conjunto de peças para Auto-
montagem [Rio de Janeiro] 2007

X, 100p. 29,7 cm (COPPE/UFRJ, M.Sc.,
Engenharia de Sistemas e Computação, 2007)

Dissertação - Universidade Federal do Rio de
Janeiro, COPPE

1. Algoritmos Evolucionistas
2. Otimização Multi-Objetivo
3. Nanotecnologia
4. Auto-montagem

I. COPPE/UFRJ II. Título(série)

Agradecimentos

Primeiramente agradeço aos meus pais por terem me dado os dois bens mais preciosos: a vida e o caráter.

Agradeço imensamente ao meu orientador, professor Valmir Barbosa. Por toda motivação, apoio científico e psicológico, durante o caminho nesta etapa da minha vida. Ele tem minha amizade incondicional.

Agradeço intensamente a minha esposa, por estar sempre ao meu lado. Por todo incentivo e amor. Por me fazer feliz e principalmente por sempre acreditar em mim.

A CAPES pelo apoio financeiro e a todas as pessoas que direta ou indiretamente possibilitaram a realização deste trabalho, meu muito obrigado!

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M.Sc.)

OTIMIZAÇÃO DO CONJUNTO DE PEÇAS PARA AUTO-MONTAGEM

Fabio Rocha Jimenez Vieira

Setembro/2007

Orientador: Valmir Carneiro Barbosa

Programa: Engenharia de Sistemas e Computação

Auto-montagem ("Self-Assembly") é o processo espontâneo pelo qual objetos simples unem-se para formar estruturas complexas. Posteriormente esses objetos, denominados ladrilhos, interagem para formação de macro-estruturas, como por exemplo, o DNA. A Auto-montagem propõem soluções eficientes para diversas áreas como: construção automatizada, integração de circuitos, computação em DNA e construção de nanorobôs.

Em 1998, Erik Winfree sugeriu um modelo matemático para este processo, denominado Auto-montagem Algorítmica, com a capacidade de construir formas geométricas em duas dimensões. Posteriormente, Adleman propôs um problema de otimização combinatória, denominado problema do conjunto mínimo de ladrilhos para a Auto-montagem. Neste trabalho estendemos o modelo bidimensional de Winfree, permitindo rotações nos ladrilhos. Esta extensão facilita a construção de formas geométricas em duas dimensões, reduzindo a variabilidade de ladrilhos e conseqüentemente diminuindo o espaço de busca do modelo original. Para criação de formas geométricas em três dimensões, criamos o modelo tridimensional de Auto-montagem. Para resolver o problema do conjunto mínimo de ladrilhos, desenvolvemos uma Heurística Evolucionária Multi-Objetivo, denominada Algoritmo Genético Multi-Objetivo Simples (AMS).

As soluções para o conjunto mínimo de tipos de ladrilhos são apresentadas para o modelo bidimensional, assim como para a extensão deste modelo e para o modelo tridimensional. As rotações permitiram melhores soluções para a extensão do modelo bidimensional, se comparado ao modelo original de Winfree. AMS obteve sucesso na solução do problema do conjunto mínimo de ladrilhos.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for degree of Master of Science (M.Sc.)

OPTIMIZATION OF THE TILE SET FOR
SELF-ASSEMBLY

Fabio Rocha Jimenez Vieira

September/2007

Advisor: Valmir Carneiro Barbosa

Program: Systems Engineering and Computer Science

Self-assembly is the ubiquitous process by which simple objects autonomously assemble into intricate structures. These structures can then grow into macrostructures, like DNA. Self-assembly suggests solutions for many fields, like autonomous construction, circuit fabrication, DNA computation, and nano-robotics.

In 1998, Erik Winfree proposed a mathematical model for this process, called the Tile Assembly Model. This model can build two-dimensional geometric forms. Later, Adleman introduced a combinatorial optimization problem, called the Minimum Tile Set Problem. In this study, we extend Winfree's model, to allow tile rotations. This extension facilitates the construction of two-dimensional geometric forms by decreasing the search space. We also introduce a three-dimensional model. We have created a heuristic multi-objective algorithm to solve the Minimum Tile Set Problem.

We have demonstrated that the two-dimensional model with rotation is capable of yielding better solutions than the original two-dimensional model. As for the three-dimensional model, the solutions we have found are the first ever to be obtained.

SUMÁRIO

Agradecimentos	iii
Lista de Figuras	viii
Lista de Tabelas	x
1 Introdução	1
1.1 Organização do trabalho	3
2 Auto-montagem	5
2.1 Aplicações	7
2.2 Modelos	10
2.2.1 Modelo bidimensional	11
2.2.2 Modelo bidimensional com rotação	15
2.2.3 Modelo tridimensional	18
2.3 O problema do conjunto mínimo de ladrilhos	23
2.3.1 Critérios	24
2.3.2 Soluções	29
2.4 O problema do conjunto mínimo de rótulos	38
3 Heurística para o PCML	40
3.1 Descrição do algoritmo	41
3.1.1 O indivíduo	42
3.1.2 Operadores genéticos	50
3.1.3 Recombinação	50
3.1.4 Mutação	52
3.1.5 Classificação do indivíduo	54

3.1.6	Composição das gerações	60
3.1.7	O Algoritmo	62
3.2	Espaço de busca	64
4	Resultados computacionais	70
4.1	Experimentos	70
4.1.1	Experimentos no modelo bidimensional	73
4.1.2	Experimentos no modelo bidimensional com rotação	83
4.1.3	Experimentos no modelo tridimensional	85
4.2	Paralelização do algoritmo	89
5	Conclusão	90
5.1	Contribuição	91
5.2	Trabalhos Futuros	92
	Referências Bibliográficas	94

LISTA DE FIGURAS

2.1	Construção de um quadrado de lado 2.	6
2.2	Transporte de objetos através de robôs.	8
2.3	Computação de um contador.	9
2.4	Representação gráfica de um ladrilho em duas dimensões.	12
2.5	Movimentos dos ladrilhos.	12
2.6	A rotação e a variabilidade de tipos de ladrilhos.	15
2.7	Rotação e polaridade.	16
2.8	Representação gráfica de um bloco em três dimensões.	19
2.9	Os três tipos básicos de posicionamento de um bloco.	20
2.10	Ordem de montagem.	25
2.11	A relação entre a rotação e a unicidade.	26
2.12	Forma completa para quadrados.	28
2.13	Solução para um quadrado 5×5 no <i>modelo 2D</i>	31
2.14	Solução para um quadrado 25×25 no <i>modelo 2D</i>	32
2.15	Forma objetivo para um quadrado 25×25 no <i>modelo 2D</i>	33
2.16	Solução para um quadrado 5×5 no <i>modelo 2D*</i> com $N + 3$ tipos de ladrilhos.	34
2.17	Solução para um quadrado 5×5 com $N + 1$ tipos de ladrilhos no <i>modelo</i> <i>2D*</i>	35
2.18	Solução para um cubo $5 \times 5 \times 5$	37
2.19	Solução para um quadrado 5×5 no <i>modelo 2D</i> com $N + 2$ rótulos distintos.	39
3.1	Distância mínima entre dois elementos das extremidades de uma forma no reticulado.	45
3.2	Sítio ativo do cromossomo.	46
3.3	Alinhamento entre duas formas.	48

3.4	Posições relativas entre duas formas no alinhamento.	48
3.5	Possibilidades da recombinação em um cromossomo com <i>sítio ativo</i> . . .	51
3.6	Exemplo de uma recombinação em um cromossomo com <i>sítio ativo</i>	53
3.7	Exemplo do teste de substituição.	55
3.8	Dominância no espaço de Pareto para o PCML.	58
3.9	Ponderação das camadas de Pareto.	59
3.10	Posicionamento e repetição dos rótulos no <i>modelo 2D*</i>	65
3.11	Tipos de blocos onde a vizinhança é singular.	67
3.12	Tipos de blocos com dois rótulos distintos em quantidades desiguais. . . .	67
3.13	Tipos de blocos com três rótulos distintos, onde três faces vizinhas possuem o mesmo rótulo.	67
3.14	Tipos de blocos com três rótulos distintos em quantidades iguais.	67
3.15	Tipos de blocos com três rótulos distintos, onde uma face possui vizinhas com a mesma quantidade de rótulos iguais.	67
3.16	Tipos de blocos com quatro rótulos distintos, onde existem pelo menos duas faces vizinhas com o mesmo rótulo.	67
3.17	Tipos de blocos com três rótulos distintos em quantidades iguais.	68
3.18	Tipos de blocos com quatro rótulos distintos, onde no máximo duas faces possuem o mesmo rótulo.	68
3.19	Tipos de blocos com cinco rótulos distintos.	68
4.1	Sementes <i>universais</i> dos experimentos.	72
4.2	Funções de avaliação do melhor indivíduo em cada geração no experimento 17.	76
4.3	Funções de avaliação do melhor indivíduo já visto ao longo das gerações no experimento 3.	78
4.4	Funções de avaliação do melhor indivíduo já visto ao longo das gerações no experimento 22.	80
4.5	Funções de avaliação do melhor indivíduo já visto ao longo das gerações no experimento 26.	82
4.6	Funções de avaliação do melhor indivíduo já visto ao longo das gerações no experimento 31.	84
4.7	Funções de avaliação do melhor indivíduo já visto ao longo das gerações no experimento 35.	86
4.8	Solução para um cubo $5 \times 5 \times 5$ com temperatura igual a 3.	88

LISTA DE TABELAS

2.1	Faixa de aplicação das soluções no <i>modelo 2D</i>	30
3.1	Exemplo de uma <i>tabela de rótulos</i> com tamanho igual a 8 e temperatura igual a 3.	43
3.2	Exemplo de função <i>ind.</i>	43
3.3	Variabilidade dos tipos de ladrilhos no <i>modelo 2D*</i>	65
3.4	Diferentes vizinhanças dos rótulos e as repetições dos tipos de blocos. . .	69
4.1	Exemplo de uma <i>tabela de rótulos</i> utilizada nos experimentos.	71
4.2	Experimentos de determinação dos parâmetros do AMS, para a construção de um quadrado 5×5 no <i>modelo 2D</i>	75
4.3	Experimentos para construção de um quadrado 5×5 no <i>modelo 2D</i>	77
4.4	Experimentos para construção de um quadrado 15×15 no <i>modelo 2D</i> . . .	79
4.5	Experimentos para construção de um quadrado 25×25 no <i>modelo 2D</i> . . .	81
4.6	Experimentos para construção de um quadrado 5×5 no <i>modelo 2D*</i>	83
4.7	Experimentos para construção de um cubo $5 \times 5 \times 5$	85
4.8	Tempo de execução dos experimentos.	89

CAPÍTULO 1

Introdução

Auto-montagem é o processo pelo qual objetos interagem automaticamente para formar objetos mais complexos. Este processo é observado tanto na formação de estruturas complexas como galáxias e sistemas planetários quanto no crescimento de cristais simples (Wang 1961). Auto-montagem é uma tecnologia emergente com potencial para substituir a Foto-Litografia convencional (Levinson 2005), utilizada na fabricação de circuitos integrados. Esta tecnologia provavelmente substituirá a forma eletrônica de armazenamento de informação, pois o fator de compactação da informação é maior que os meios de armazenamento atuais (Rothmund 2006). Em um futuro próximo, nanorobôs serão capazes de executar tarefas complexas em conjunto, através das regras de Auto-montagem (Tuci et al. 2006). A própria construção destes nanorobôs será regida pela Auto-montagem. Mas para que estas aplicações tornem-se realidade é necessário controlar o processo de Auto-montagem, através do controle da construção de formas geométricas geradas por este processo, como será visto na seção 2.1. Atualmente, a Auto-montagem é considerada uma área de importância estratégica pelos governos mundiais (Reif 2002; Majumder et al. 2007).

O estudo matemático do processo de Auto-montagem iniciou com o trabalho de Wang (Wang 1961), onde o reconhecimento de padrões no crescimento de cristais de carbono foi estudado. Wang observou que os cristais são construídos através da Auto-montagem de estruturas nanométricas, as quais ele denominou ladrilhos. Ele descobriu padrões na

união destes ladrilhos para formação dos cristais. Adleman (Adleman 2000) propôs um modelo matemático para o processo em uma dimensão. A partir deste modelo, Winfree propôs a primeira extensão em duas dimensões (Winfree 1998). O objetivo principal do modelo de Winfree é a execução de computação e não a construção de formas geométricas. Após a publicação do modelo de Winfree, surgiram diversos trabalhos com objetivo de estudar a complexidade e aumentar o poder computacional do modelo (Angelov, Khanna e Visontai 2006; Chen et al. 2004; Arbuckle e Requicha 2006). Entretanto, até o presente momento, não existem extensões do modelo de Winfree, com objetivo de construir formas geométricas em duas ou três dimensões. Trabalhos recentes procuram descobrir processos bioquímicos que facilitem e aumentem a precisão na construção de estruturas nanométricas (ladrilhos), que são usadas na Automontagem (Winfree et al. 1998; Yi et al. 2005; Pistol e Dwyer 2007). Outros estudos desenvolvem métodos para utilização de partes do DNA como estruturas nanométricas (Requicha e Arbuckle 2006; Rothmund 2006).

O principal objetivo deste trabalho é solucionar o Problema do Conjunto Mínimo de Ladrilhos (PCML), proposto por Adleman em (Adleman et al. 2002). Este problema consiste em determinar o conjunto de tipos de ladrilhos necessários para construção de formas geométrica em duas dimensões utilizando o modelo de Winfree. Neste contexto, estendemos este modelo, permitindo rotações dos ladrilhos em duas dimensões, já que o modelo original não permitia esse tipo de movimento. No entanto, alguns estudos comprovam a importância da construção de formas geométricas em três dimensões (Hogberg e Olin June 2006). Sendo assim, propomos o modelo em três dimensões a partir do modelo com rotação. Para realização de experimentos foi necessário desenvolver um simulador computacional para processo de Auto-montagem, visto que não havia simulador computacional definido na literatura para o modelo de Winfree. Dessa forma, desenvolvemos um simulador estocástico para o processo, obedecendo as regras dos modelos em duas e três dimensões.

Até o momento apenas soluções analíticas foram apresentadas para solucionar o PCML, e somente para formas geométricas restritas como quadrados, retângulos e linhas (Rothe-

mund e Winfree 2000). Entretanto, estas soluções permitem avaliar nosso algoritmo para resolver o PCML. Note que não existem provas analíticas de que as soluções do problema do conjunto mínimo de ladrilhos são ótimas. Neste trabalho propomos um método computacional utilizando Meta-Heurísticas Evolucionárias Multi-Objetivo (Gaspar-Cunha, Gonsalves e Covas 2006). Para implementar este método, desenvolvemos um algoritmo, denominado Algoritmo Genético Multi-Objetivo Simples (AMS), que possui particularidades originais.

Os resultados dos experimentos demonstraram que a extensão do modelo em duas dimensões mostrou-se capaz de construir as mesmas formas geométricas que o modelo de Winfree, porém com um espaço de busca menor. O modelo em três dimensões constrói formas geométricas com a mesma quantidade de tipos de ladrilhos que os modelos bidimensionais. Finalmente os resultados comprovaram a eficiência do algoritmo AMS na solução do problema do conjunto mínimo de ladrilhos, tanto no modelo em duas dimensões quanto no modelo em três dimensões.

1.1 Organização do trabalho

Os próximos capítulos desta dissertação estão organizados da seguinte forma.

No capítulo 2 são apresentados:

- O processo de Auto-montagem, suas aplicações e o modelo original em duas dimensões.
- A extensão do modelo em duas dimensões e o modelo em três dimensões. Todos os modelos apresentados são definidos matematicamente.
- O problema do conjunto mínimo de ladrilhos, suas soluções para os modelos bidimensionais e para o modelo em três dimensões.
- O problema do conjunto mínimo de rótulos.

No capítulo 3 são apresentados:

- A Heurística Evolucionária Multi-Objetivo AMS, desenvolvida neste trabalho, assim como suas configurações e parâmetros.
- O espaço de busca de AMS, para cada um dos modelos apresentados no capítulo 2.

No capítulo 4 são apresentados:

- Resultados de todos os experimentos realizados.
- A determinação dos parâmetros de AMS, de acordo com cada experimento.
- O desempenho de AMS, para cada um dos modelos apresentados no capítulo 2.

No capítulo 5 são apresentados:

- As conclusões deste trabalho.
- As contribuições do trabalho.
- Os trabalhos futuros.

CAPÍTULO 2

Auto-montagem

Auto-montagem ("Self-Assembly") é o processo espontâneo pelo qual objetos simples unem-se para formar estruturas complexas. Este processo é observado na natureza, inicialmente na reação entre os átomos para formar moléculas. Posteriormente essas moléculas interagem para formação de macro-estruturas, como por exemplo, o DNA. Em 1998, Erik Winfree sugeriu um modelo matemático para este processo, em duas dimensões, denominado Auto-montagem Algorítmica (Winfree 1998). Neste modelo cada molécula é representada por um ladrilho. Os lados deste ladrilho são rotulados. A cada rótulo está associada uma força. Dada uma temperatura, se dois ladrilhos possuem o mesmo rótulo, em lados opostos, e se a força deste rótulo é igual ou superior à temperatura, os ladrilhos são capazes de se unir. Basicamente o modelo sugerido por Winfree explica a maioria dos processos de Auto-montagem em duas dimensões. A figura 2.1 demonstra a Auto-montagem de um quadrado de lado 2. O rótulo x possui força igual a 2, os rótulos b e c possuem força igual a 1 e a temperatura é igual a 2. A figura 2.1-a apresenta a família de tipos de ladrilhos utilizada no processo de Auto-montagem. A figura 2.1-b ilustra as etapas de construção do quadrado. No início o ladrilho S é adicionado (etapa um). Em seguida, o ladrilho P é adicionado, unindo-se ao ladrilho S (etapa dois). Note que a adição do ladrilho N não é permitida, pois o rótulo b não possui força suficiente. Posteriormente, o ladrilho M é adicionado, unindo-se ao ladrilho S (etapa três). Finalmente, o ladrilho N é adicionado, pois a soma de suas uniões (rótulos b e c) é igual à temperatura (etapa quatro).

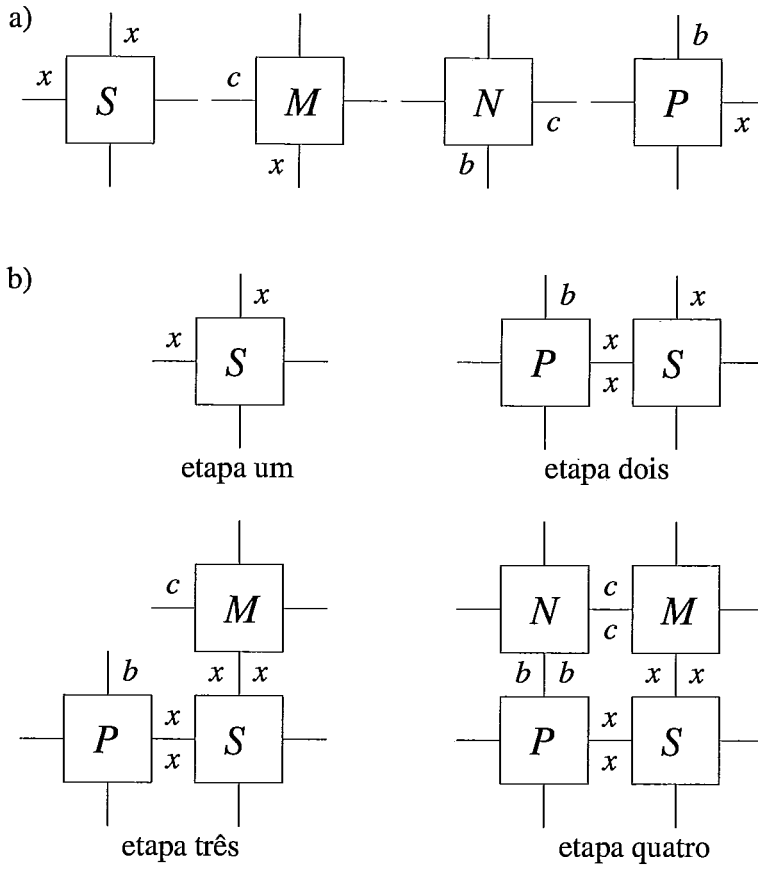


Figura 2.1: Construção de um quadrado de lado 2.

2.1 Aplicações

A Auto-montagem é considerada uma área de estudo da Nanociência, sendo aplicada em diversos setores, desde fabricação de circuitos eletrônicos à realização de computação. A seguir listamos as principais áreas de aplicação.

A construção de circuitos eletrônicos é conduzida e controlada por máquinas, que participam completamente do processo de construção do circuito. Este processo, denominado Foto-Litografia (Levinson 2005), possui custo elevado, pois depende de máquinas altamente especializadas. Por outro lado, a Auto-montagem propõe uma alternativa, onde o processo de construção é iniciado por uma máquina especializada e monitorado ao longo da Auto-montagem do circuito. Dessa forma, diminui-se o tamanho do circuito, reduz-se o consumo de energia e aumenta-se a eficiência da montagem, como demonstrado no trabalho de Pistol (Pistol, Lebeck e Dwyer 2006).

Outra área alcançada pelas técnicas da Auto-montagem é a construção utilizando robótica. O objetivo é aumentar a eficiência de construções automatizadas substituindo robôs, que realizam diversas tarefas, por um grupo de robôs autônomos, que realizam tarefas simples e trabalham cooperativamente. Neste contexto, a Auto-montagem é utilizada para definir as regras que governam as interações entre os robôs, tornando o trabalho em conjunto possível. Estes robôs são denominados robôs modulares. Como exemplo considere um grupo de robôs que devem transportar um objeto de um ponto a outro. Cada robô é uma esteira rolante. Para realizar a tarefa é necessário que os robôs unam-se ao longo da trajetória entre os pontos, evitando obstáculos. Na figura 2.2 vários robôs se unem para transportar o objeto M do ponto inicial ao ponto final, evitando o obstáculo, através de regras de Auto-montagem. A figura 2.2-a apresenta a família de robôs utilizada na tarefa. A figura 2.2-b ilustra o posicionamento final dos robôs. Esta união é guiada por regras de Auto-montagem, definidas de acordo com a trajetória (Tuci et al. 2006).

A construção automatizada também utiliza a Auto-montagem na fabricação de blocos de montagem dotados de informação sobre o seu encaixe. Se para construir uma ponte cada bloco contém a informação de como e onde será encaixado ou montado, um grupo

de robôs de construção simples é capaz de construir a ponte sabendo apenas quais são as regras de encaixe destes blocos. Cada robô não sabe o que está construindo, nem quais regras complexas regem a construção de uma ponte (Werfel et al. 2006). Toda a complexidade é transferida para a construção destes blocos, facilitando a programação dos robôs. Desta forma, a construção automatizada utilizando técnicas de Auto-montagem reduz os custos e aumenta a precisão do trabalho (Terada e Murata 2004).

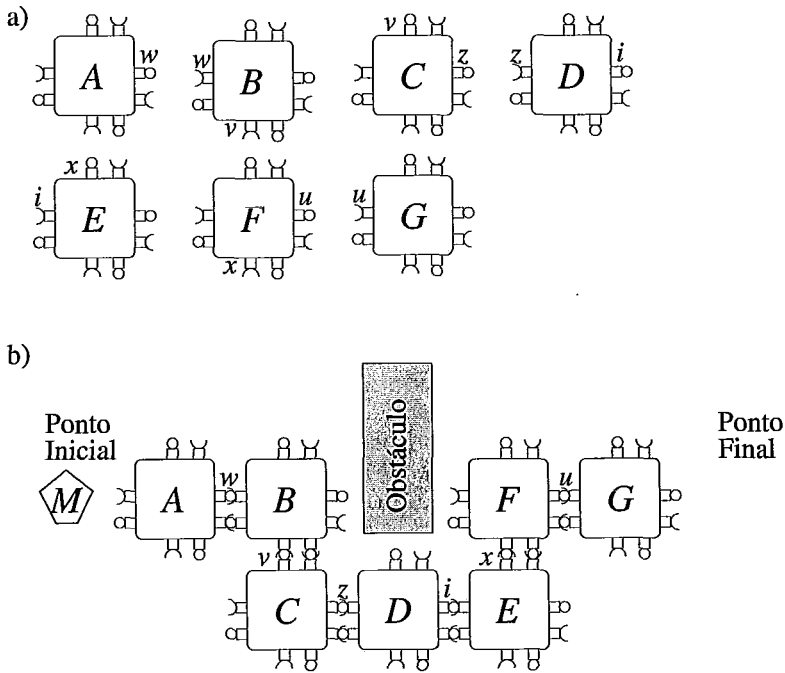


Figura 2.2: Transporte de objetos através de robôs.

A Auto-montagem também é aplicada à computação em DNA (Baryshnikov, Jr. e Momcilovic 2004; Winfree 1996; Brun 2007). Este novo paradigma determina que sistemas bioquímicos são capazes de desempenhar qualquer computação (Adleman et al. 2001). A computação é realizada à medida que as moléculas, DNA ou proteínas, se unem em um reticulado. A figura 2.3 ilustra a computação de um contador, onde cada molécula é representada por um ladrilho rotulado com 0 ou 1. Moléculas especiais rotuladas com as letras R, S e L são a base para o início da computação. Os rótulos a , b e n possuem força igual a 1. O rótulo x possui força igual a 2. A temperatura é igual a 2. A figura 2.3-a apresenta a família de tipos de ladrilhos utilizados no processo de Auto-montagem. A figura 2.3-b ilustra este processo, onde cada linha representa um número binário, exceto a primeira linha. Este número é acrescido de uma unidade a cada linha adicionada à forma

em construção. Este é o primeiro passo para a construção de um computador baseado em moléculas de DNA (Winfree 1996).

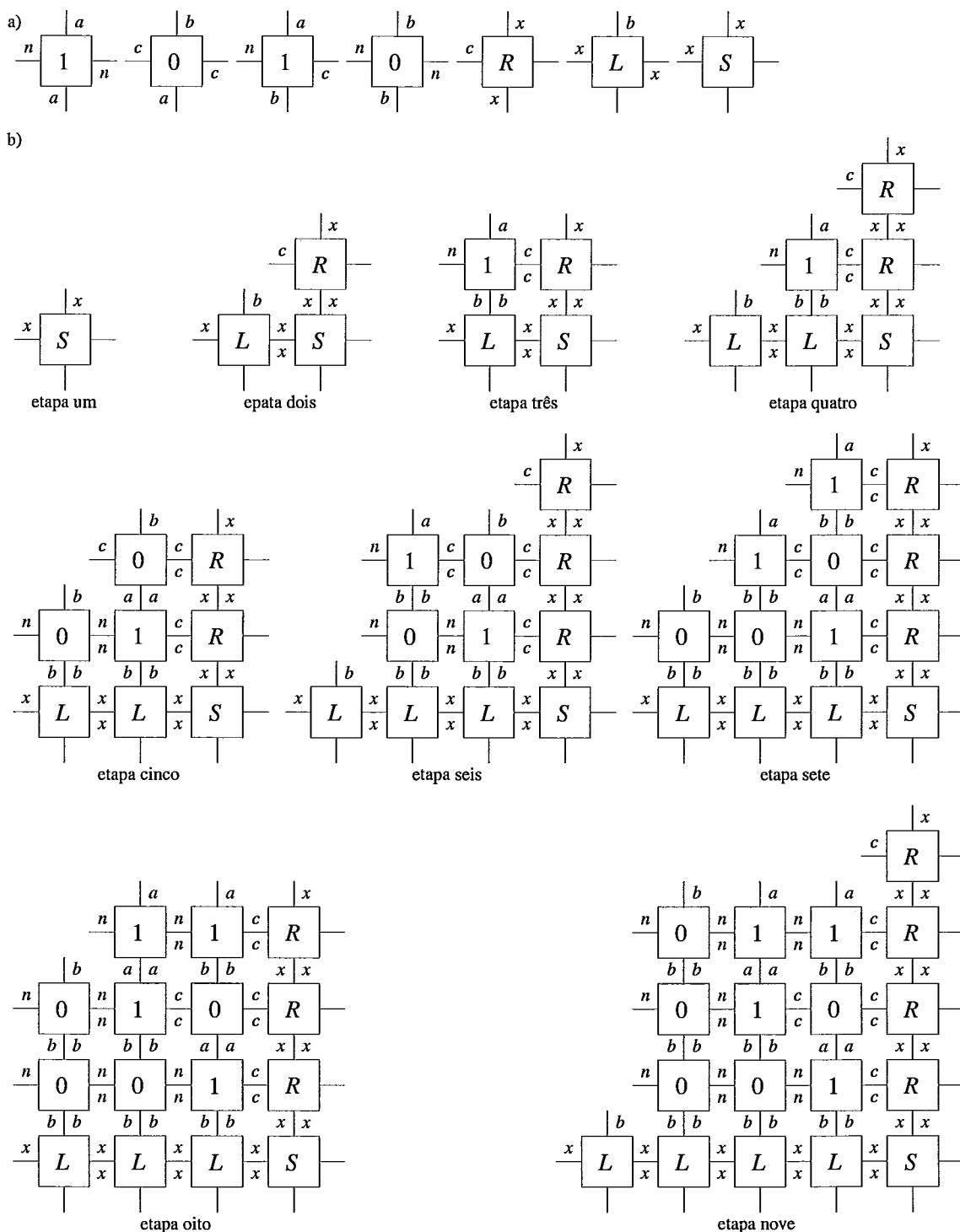


Figura 2.3: Computação de um contador.

As propriedades da Auto-montagem possibilitam a construção de padrões e formatos, permitindo o armazenamento de informações em escala nanométrica (Rothmund 2006). Outras técnicas atingem o mesmo objetivo, porém exigem ambientes específicos como vácuo e temperaturas criogênicas (Rothmund 2006). Embora a Auto-montagem tenha diversas aplicações, o controle deste processo é intrinsecamente árduo. Encontrar o conjunto mínimo de objetos, seus rótulos ou a regra que controla sua união, para realizar a computação correta ou construir a estrutura desejada, é um problema NP-Completo (Adleman et al. 2002). Conjectura-se que minimizar o tempo da computação, controlando a concentração destes objetos seja um problema #P-Árduo (Adleman et al. 2002).

Devido às dificuldades apresentadas, quase todos os estudos voltaram-se para a busca de métodos bioquímicos, mais simples para construção artesanal de nanodispositivos (Pistol, Lebeck e Dwyer 2006; Rothmund 2006; Schafmeister 2007) ou a busca por regras lógicas para programação de robôs (Terada e Murata 2004; Werfel et al. 2006).

2.2 Modelos

Neste trabalho adotamos o *modelo de Auto-montagem de ladrilhos* proposto originalmente em (Rothmund e Winfree 2000), no qual o processo de Auto-montagem ocorre em um reticulado, posicionando o primeiro ladrilho, denominado *semente* e prossegue com a adição de um ou mais ladrilhos ao reticulado. Os ladrilhos originam-se exclusivamente de cópias dos elementos de um conjunto de tipos de ladrilhos, disponíveis para a Auto-montagem. Não é permitido a um ladrilho mudar a sua posição após a sua adição ao reticulado. Este processo pode se tornar infinito, pois o suprimento de cópias do conjunto de tipos de ladrilhos é ilimitado. Como o reticulado também é infinito não há problema, porém em simulações computacionais, isto não é possível. Dessa forma, para simular o reticulado infinito empregamos uma versão heurística. Existe apenas uma ocorrência da *semente* no reticulado, pois esta não pertence ao conjunto de tipos de ladrilhos. A *semente* pode ser composta por mais de um ladrilho. Neste caso, a montagem da *semente* deve obedecer às regras do *modelo de Auto-montagem de ladrilhos* e a *semente* é denominada *semente complexa*.

Em (Rothemund 2001) foi demonstrada uma extensão deste modelo, onde são utilizadas várias *sementes* diferentes, assim como várias ocorrências da mesma *semente*. Esta extensão não foi estudada neste trabalho, porém criamos duas generalizações originais do modelo que flexibilizam a construção de formas e padrões geométricos em duas (seção 2.2.2) e três dimensões (seção 2.2.3). Outras extensões propostas visam o estudo matemático da complexidade e a aumentar o poder computacional da Auto-montagem (Aggarwal et al. 2005; Winfree 2006; Reif, Sahu e Yin 2005), não se aprofundando na construção de formas geométricas.

2.2.1 Modelo bidimensional

Definição informal

O modelo bidimensional é regido pelas regras descritas na seção anterior, onde o processo de Auto-montagem começa com a adição da *semente*. Quando um ladrilho t_1 é adicionado ao reticulado, este une-se a um ladrilho t_2 adjacente a t_1 , se o lado de t_1 adjacente a t_2 possui o mesmo rótulo r do lado de t_2 adjacente a t_1 . A força desta união é dada pela força de atração que o rótulo r possui, onde cada rótulo distinto possui uma força determinada. Dada uma determinada temperatura T , um ladrilho t é adicionado ao reticulado, se a soma de todas as uniões que t criou com cada adjacente a ele for maior ou igual a T .

Definição formal

A representação gráfica adotada para um tipo de ladrilho é mostrada na figura 2.4-a, onde S é o símbolo do ladrilho *semente*. Na figura 2.4-b vemos a numeração dos lados de um tipo de ladrilho, seguindo o sentido horário. Esta representação permite a criação da forma textual do mesmo, ilustrada na figura 2.4-c, onde a é o rótulo do lado 1, separado por uma vírgula do rótulo do lado 2 até o rótulo do lado 4. Não são permitidos nenhum dos movimentos de rotação mostrados na figura 2.5, nem mesmo quando o ladrilho é posicionado no reticulado.

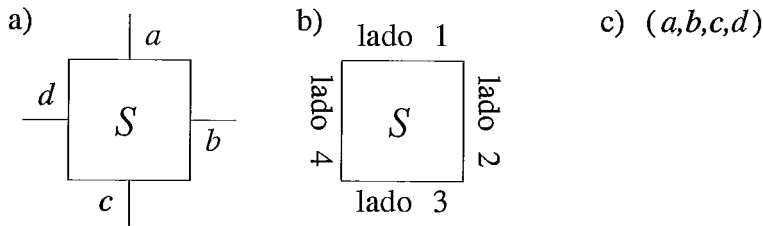


Figura 2.4: Representação gráfica de um ladrilho em duas dimensões.

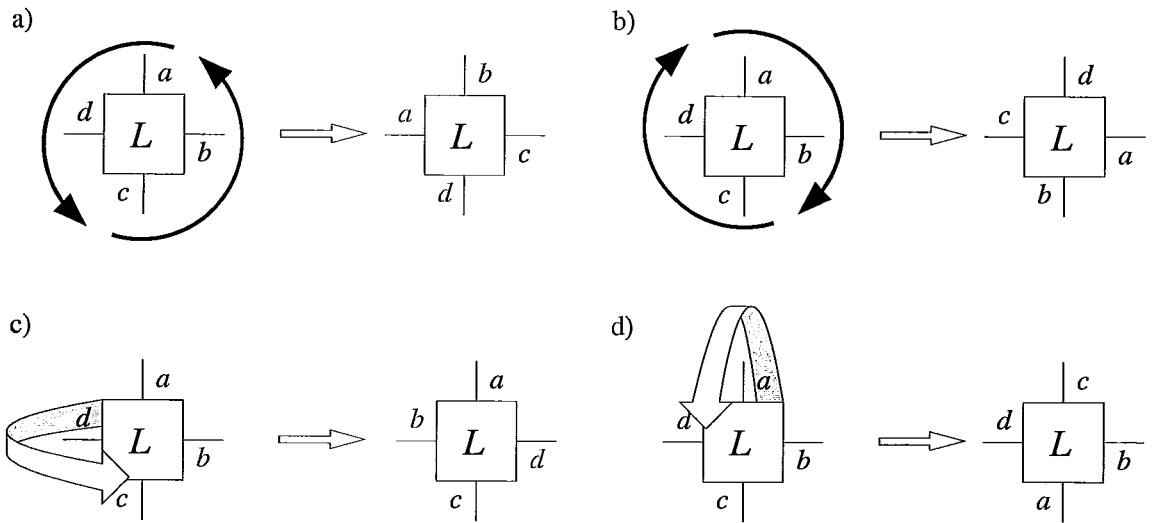


Figura 2.5: Movimentos dos ladrilhos.

Utilizamos as seguintes definições para facilitar a formulação matemática do *modelo de Auto-montagem de ladrilhos*, o qual denominamos *modelo 2D*. \mathbb{N} é o conjunto dos números naturais e \mathbb{Z} o conjunto dos números inteiros. O processo de Auto-montagem ocorre em um reticulado $\mathbb{Z} \times \mathbb{Z}$. $F = \{1,2,3,4\}$ é o conjunto de lados de um ladrilho, L é o conjunto dos tipos de ladrilhos e R é o conjunto de rótulos utilizados em um processo de Auto-montagem. Um tipo de ladrilho $\tau \in L$ é um quadrado, onde cada lado $x \in F$ possui um rótulo $r \in R$. Ladrilho é toda cópia de um tipo de ladrilho $\tau \in L$, posicionado no reticulado em (i,j) e representado pelo par $(\tau,(i,j))$. Para cada ladrilho t , copiado de um tipo de ladrilho τ , t^x é o rótulo do lado x , sendo $t^x = \tau^x$. O rótulo nulo $\eta \in R$ é um rótulo especial que representa a ausência de rótulo. O tipo de ladrilho $(\eta,\eta,\eta,\eta) \in L$ representa a ausência de ladrilho, denominada como ladrilho *vazio*. Não é permitido rotular lados de um tipo de ladrilho com o rótulo η , com exceção do tipo de ladrilho *vazio*. Uma *configuração* é uma forma montada por ladrilhos, representada por um conjunto C de ladrilhos não *vazios*, originados do conjunto L , onde não existam dois ladrilhos com a

mesma posição no reticulado. Para qualquer *configuração* C , $C_{(i,j)}$ representa o ladrilho $t \in C$ posicionado em (i,j) ou o ladrilho (*vazio*, (i,j)), caso não exista ladrilho na posição (i,j) . A função $Loc(t)$ retorna a posição (i,j) do ladrilho $t \in C$, onde $Loc : C \rightarrow \mathbb{Z} \times \mathbb{Z}$. Duas *configurações* C e D são ditas *idênticas*, quando possuem os mesmos ladrilhos nas mesmas posições. Se x é um lado de um ladrilho, seu lado oposto é aquele no lado oposto do ladrilho. O cálculo do lado oposto a x é dado pela função $FO_{2D} : F \rightarrow F$, tal que

$$FO_{2D}(x) = ((x + 1)\%4) + 1, \quad (2.1)$$

sendo $\%4$ o resto da divisão inteira por 4.

Se (i,j) é a posição de um ladrilho t no reticulado e x é um dos lados deste ladrilho, a função $P_{2D} : F \times \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z} \times \mathbb{Z}$, onde

$$P_{2D}(x, Loc(t)) = \begin{cases} (i, j + 1), & \text{se } x = 1 \\ (i + 1, j), & \text{se } x = 2 \\ (i, j - 1), & \text{se } x = 3 \\ (i - 1, j), & \text{se } x = 4, \end{cases} \quad (2.2)$$

calcula a posição adjacente ao lado x . Dois ladrilhos $t_1, t_2 \in C$, posicionados em (i_1, j_1) e (i_2, j_2) são ditos adjacentes quando $|i_1 - i_2| + |j_1 - j_2| = 1$ ou, equivalentemente, caso exista um lado x de t_1 , onde $P_{2D}(x, t_1) = Loc(t_2)$ ou, equivalentemente, $P_{2D}(FO_{2D}(x), t_2) = Loc(t_1)$.

Os rótulos dos lados regem a adição de ladrilhos na Auto-montagem. Cada rótulo $r \in R$, possui uma força de atração dada pela função $F(r)$, onde $F : R \rightarrow \mathbb{N}$. O rótulo zero $\emptyset \in R$ é definido como o único rótulo a possuir força igual a 0. Este rótulo não possui representação gráfica em um ladrilho, aparecendo somente na forma textual. Na figura 2.2-a o rótulo \emptyset está associado aos lados 1, 3 e 4 do robô A , porém não é representado graficamente. A forma textual do robô A é $(\emptyset, w, \emptyset, \emptyset)$. Vale ressaltar que o rótulo η descrito anteriormente, possui ausência de força, e não força zero. A função $FA_{2D} : R \times R \rightarrow \mathbb{N}$ define a força de atração entre os rótulos r_1 e r_2 , tal que

$$FA_{2D}(r_1, r_2) = \begin{cases} F(r_1), & \text{se } r_1 = r_2 \\ 0, & \text{se } r_1 \neq r_2. \end{cases} \quad (2.3)$$

Para todo ladrilho $t \in C$ no *modelo 2D*, podemos calcular a força de interação com a qual ele é atraído por um ladrilho adjacente, através da função $FI_{2D}^C : F \times C \rightarrow \mathbb{N}$, tal que

$$FI_{2D}^C(x, t) = FA_{2D}(t^x, (C_{P_{2D}(x, Loc(t))})^{FO_{2D}(x)}), \quad (2.4)$$

onde $(C_{P_{2D}(x, Loc(t))})^{FO_{2D}(x)}$ retorna o rótulo do lado $FO_{2D}(x)$ pertencente ao ladrilho posicionado em $P_{2D}(x, Loc(t))$ na configuração C . Este ladrilho t une-se a um ladrilho adjacente $P_{2D}(x, Loc(t)) \in C$, através do lado x , na *configuração C*, caso $FI_{2D}^C(x, t) > 0$, constituindo uma *ligação*. A função $FT_{2D}^C : F \rightarrow \mathbb{N}$, tal que

$$FT_{2D}^C(t) = \sum_{x \in F} FA_{2D}(t^x, (C_{P_{2D}(x, Loc(t))})^{FO_{2D}(x)}), \quad (2.5)$$

é a soma das forças de *ligação* de t com os ladrilhos adjacentes na *configuração C*.

Entretanto, o fato de t possuir $FI_{2D}^C(x, t) > 0$ ou mesmo $FT_{2D}^C(t) > 0$ não é suficiente para que t seja adicionado à *configuração C* em uma posição do reticulado. Como todo processo de Auto-montagem acontece sob a influência de uma temperatura T , é necessário que a soma das forças de *ligação* de qualquer ladrilho $t \in C$ seja sempre maior ou igual à temperatura, caso contrário as *ligações* seriam rompidas pela temperatura. Este fenômeno é denominado *disruptura termal*. A *configuração C* é dita *T-estável* se cada um de seus ladrilhos foi adicionado com $FT_{2D}^C \geq T$. Dessa forma, a *configuração C* não será desmontada pela força de *disruptura termal*. Em (Aggarwal et al. 2005; Kao e Schweller 2006) foram estudados modelos onde a temperatura varia ao longo da Auto-montagem, porém neste trabalho consideramos a temperatura sempre fixa.

Um *sistema de ladrilhos* é uma quádrupla $\langle T, S, L, FA_{2D} \rangle$, onde T é a temperatura, S é a *semente*, L é um conjunto finito de tipos de ladrilhos contendo o ladrilho *vazio* e FA_{2D} é a função que calcula a força de atração dos rótulos. Se SL é um *sistema de ladrilhos*, então o processo de Auto-montagem no *modelo 2D*, é representado pela relação $C \rightarrow_{SL} D$. Esta relação é válida se e somente se C e D são duas *configurações T-estáveis* e C e D tornam-se *idênticas* quando o ladrilho t , copiado de $\tau \in L$, é retirado de D . Desta forma, a Auto-montagem torna-se um processo incremental, onde cada relação

$C(n) \rightarrow_{SL} C(n + 1)$ representa um passo n e a adição de um ladrilho, a cada passo, transforma a *configuração* $C(n)$ em $C(n + 1)$.

2.2.2 Modelo bidimensional com rotação

A primeira extensão natural do modelo bidimensional que propomos, permite os movimentos de rotação dos ladrilhos, como mostrado nas figuras 2.5-a e 2.5-b. Como as moléculas representadas pelos ladrilhos normalmente rotacionam (Cooper e Hausman 2003), a rotação torna o novo modelo mais próximo da realidade. A extensão também diminui a variedade de ladrilhos, pois os dois ladrilhos da figura 2.6 são ladrilhos diferentes no *modelo 2D* e representam o mesmo ladrilho em um modelo que permite rotação. O ladrilho M é o ladrilho L , após uma rotação de M no sentido horário. A diminuição na variedade, a princípio parece uma desvantagem, mas proporciona a criação de ladrilhos com maior flexibilidade nas uniões. O ladrilho M jamais se ligaria ao ladrilho L obedecendo as regras do *modelo2D*. No entanto, o ladrilho M pode-se ligar ao ladrilho L , desde que M sofra uma rotação no sentido anti-horário.



Figura 2.6: A rotação e a variabilidade de tipos de ladrilhos.

A rotação agrega vantagens ao modelo, mas traz uma desvantagem crítica. Para ilustrar esta desvantagem observe a figura 2.7. A figura 2.7-a apresenta a *semente* S e um conjunto de tipos de ladrilhos, onde o objetivo é a simples construção de uma linha com três ladrilhos de comprimento. Este objetivo é alcançado pela montagem apresentada na figura 2.7-b. Porém, devido à possibilidade de rotação do ladrilho 0, a figura 2.7-c também representa uma montagem, no entanto indesejável, pois o comprimento da linha é infinito. Algo similar é observado na figura 2.7-d, onde o comprimento da linha é finito, porém quantidade de ladrilhos 0 utilizados é indeterminada. Dessa forma, o modelo com rotação torna o processo de Auto-montagem instável, mesmo para um conjunto de tipos

de ladrilhos e objetivo extremamente simples. Para solucionar este problemas, introduzimos o *modelo 2D**. Este novo modelo utiliza outro recurso das moléculas: a polaridade (Cooper e Hausman 2003), que acrescenta ao antigo rótulo do *modelo 2D* um sinal de + ou -. Devido à esta alteração, o lado de um ladrilho atrai o lado de outro ladrilho, se estes lados possuem o mesmo rótulo, porém com sinais diferentes. As figuras 2.7-e e 2.7-f ilustram a solução através do uso da polaridade. A figura 2.7-e apresenta a *semente S* e um conjunto de tipos de ladrilho do *modelo 2D**. A figura 2.7-f representa a única forma de unir os ladrilhos, a partir da *semente S*. Todos os rótulos apresentados na figura 2.7 possuem força suficiente para vencer a *disruptura termal*.

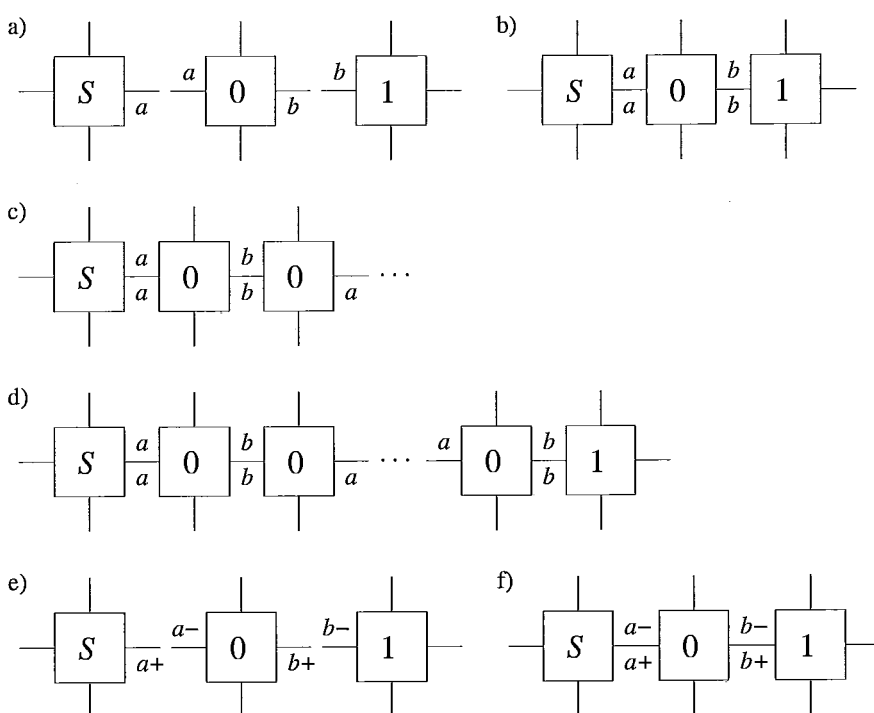


Figura 2.7: Rotação e polaridade.

Definição informal

Neste modelo o processo de Auto-montagem também começa com a adição da semente e prossegue com a adição de um ladrilho t_1 , originado de alguma rotação de um tipo de ladrilho τ_1 , onde t_1 une-se a um ladrilho t_2 adjacente a t_1 , se o lado de t_1 adjacente a t_2 possui o mesmo rótulo r do lado de t_2 adjacente a t_1 , porém com polaridades diferentes. Como no *modelo 2D*, a força desta união é dada pela força de atração que o rótulo r possui, onde cada rótulo distinto possui uma força determinada e um ladrilho t_1 é adicionado ao

reticulado, se a soma de todas as uniões que t_1 criou com cada adjacente a ele for maior ou igual a T .

Definição formal

Utilizamos a mesma formulação matemática do *modelo 2D*, porém atualizando e incluindo as seguintes definições. Os movimentos das figuras 2.5-c e 2.5-d requerem três dimensões e desta forma continuam proibidos. O rótulo $r \in R$ possui duas componentes (*identificador*, *polaridade*), onde *identificador* é equivalente ao rótulo do *modelo 2D* e *polaridade* assume um dos possíveis sinais $+$ ou $-$. Se $r \in R$ é um rótulo, r_i representa o *identificador* e r_p a *polaridade*. Se $\tau \in L$ é um tipo de ladrilho $(a+,b+,c+,d+)$, sua *rotação 0* é representada por $(a+,b+,c+,d+)$. As rotações: *rotação 1* $= (d+,a+,b+,c+)$, *rotação 2* $= (c+,d+,a+,b+)$ e *rotação 3* $= (b+,c+,d+,a+)$ são resultado de uma, duas e três rotações de t no sentido horário, respectivamente. $Rot = \{\text{rotação 0}, \text{rotação 1}, \text{rotação 2}, \text{rotação 3}\}$ é o conjunto de todas rotações possíveis em um tipo de ladrilho $\tau \in L$. Quando ocorre uma rotação, os rótulos mudam de lado, como observado nas figuras 2.5-a.e 2.5-b, porém a numeração dos lados é fixa em um ladrilho. Para calcular a novo lado de um rótulo, após a rotação rot , definimos a função $Rot_{2D*} : Rot \times F \rightarrow F$, tal que

$$Rot_{2D*}(rot, x) = ((rot + x - 1)\%4) + 1, \quad (2.6)$$

sendo $\%4$ o resto da divisão inteira por quatro e x é antigo lado do rótulo.

Ladrilho t é uma cópia de um tipo de ladrilho $\tau \in L$, posicionado no reticulado em (i,j) e representado pela tripla $(\tau, (i,j), rot)$, onde rot é uma das quatro rotações de τ . Desta forma, um tipo de ladrilho possui a capacidade de gerar quatro ladrilhos diferentes em uma mesma posição (i,j) . Uma vez escolhida a rotação e posicionado no reticulado, não é permitido ao ladrilho mais nenhum tipo de movimento. Em um tipo de ladrilho τ , τ^x representa o rótulo associado ao lado x . Para identificar o rótulo do lado x no ladrilho $t = (\tau, (i,j), rot)$, aplicamos a operação $t^x = \tau^{Rot_{2D*}(rot,x)}$. Para qualquer *configuração* C , $C_{(i,j)}$ representa o ladrilho $t \in C$ posicionado em (i,j) ou o ladrilho (*vazio*, (i,j) , *rotação 0*), caso não exista ladrilho na posição (i,j) . O cálculo do lado oposto a um lado x no *modelo 2D** é idêntico ao cálculo no *modelo 2D*. Dessa forma, $FO_{2D*}(x) = FO_{2D}(x)$.

Se (i,j) é a posição de um ladrilho t no reticulado e x é um dos lados deste ladrilho, a função $P_{2D^*}(x, Loc(t)) = P_{2D}(x, Loc(t))$, calcula a posição adjacente ao lado x . A função $FA_{2D^*} : R \times R \rightarrow \mathbb{N}$ define a força de atração entre os rótulos r_1 e r_2 , tal que

$$FA_{2D^*}(r_1, r_2) = \begin{cases} F(r_1), & \text{se } r_{1i} = r_{2i} \text{ e } r_{1p} \neq r_{2p} \\ 0, & \text{caso contrário,} \end{cases} \quad (2.7)$$

Para os rótulos \emptyset e η não é definida polaridade, pois não possuem força. A força com a qual um ladrilho $t \in C$ é atraído por um ladrilho adjacente à ele na *configuração* C , é dada pela função $FI_{2D^*}^C : F \times C \rightarrow \mathbb{N}$, onde

$$FI_{2D^*}^C(x, t) = FA_{2D^*}(t^x, (C_{P_{2D^*}(x, Loc(t))})^{FO_{2D^*}(x)}). \quad (2.8)$$

A função $FT_{2D^*}^C : F \rightarrow \mathbb{N}$ é a soma das forças de *ligação* de t com os seus ladrilhos adjacentes, representando a força que mantém o ladrilho t unido à *configuração* C , tal que

$$FT_{2D^*}^C(t) = \sum_{x \in F} FA_{2D^*}(t^x, (C_{P_{2D^*}(x, Loc(t))})^{FO_{2D^*}(x)}). \quad (2.9)$$

Um ladrilho $t \in C$ cria uma *ligação* com um ladrilho adjacente $P_{2D^*}(x, Loc(t)) \in C$, caso $FI_{2D^*}^C(x, t) > 0$. Em uma *configuração* C *T-estável*, o ladrilho t é adicionado se $FT_{2D^*}^C(t) \geq T$. Um *sistema de ladrilhos* SL no *modelo* $2D^*$ é representado pela quádrupla (T, S, L, FA_{2D^*}) . No *modelo* $2D^*$, o processo de Auto-montagem ocorre de forma idêntica ao *modelo* $2D$, através da relação $C \rightarrow_{SL} D$.

2.2.3 Modelo tridimensional

A programação de robôs, criação de blocos inteligentes para a construção automatizada, a montagem de peças para robôs e circuitos em escala nanométrica requerem naturalmente soluções em três dimensões, como visto na seção 2.1. Os *modelos* $2D$ e $2D^*$ não são capazes de construir formas geométricas em três dimensões. Para a montagem destas formas geométricas, estendemos o *modelo* $2D^*$ para três dimensões e denominamos esta nova extensão como *modelo* $3D$.

A primeira alteração do *modelo 2D** necessária é a transformação do ladrilho em um bloco com três dimensões. A figura 2.8-a ilustra a representação gráfica adotada para um bloco e sua representação textual, onde os rótulos das faces ocultas foram representados em cinza. No modelo tridimensional o bloco possui seis faces e conseqüentemente seis rótulos, onde cada face é delimitada por quatro arestas. A numeração das faces é apresentada na figura 2.8-c. Este padrão de numeração foi adotado segundo o conceito de vizinhança das faces em um cubo. Este critério define que uma face x é vizinha de uma face y se e somente se possuem uma aresta em comum. A figura 2.8-d mostra o grafo de vizinhança das faces de um bloco. Os nós do grafo representam as faces do bloco e são numerados no sentido horário do único ciclo hamiltoniano deste grafo (Chiba e Nishizeki 1989). Se duas faces são vizinhas, então seus nós são ligados por uma aresta no grafo.

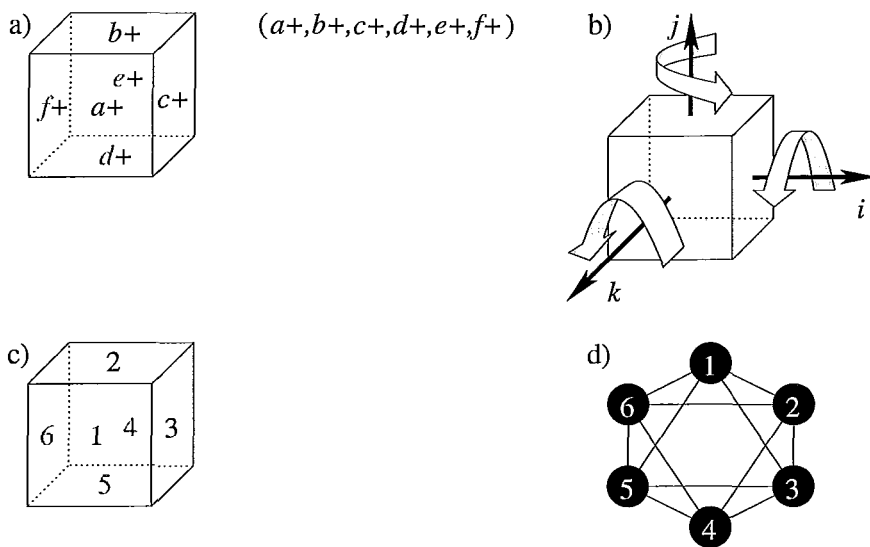


Figura 2.8: Representação gráfica de um bloco em três dimensões.

A liberdade para rotacionar o bloco proporciona as mesmas vantagens descritas no *modelo 2D** (seção 2.2.2). Note que o número de rotações possíveis do bloco é maior que o número de rotações possíveis do ladrilho. A figura 2.8-b mostra os três tipos de rotações possíveis em um bloco. Quando combinados, estes três tipos geram vinte e quatro posições distintas do bloco. Podemos comprovar isso, observando a figura 2.9, onde um bloco é posicionado de três formas distintas, fixando um de seus vértices na origem do sistema de coordenadas i, j, k . Como um bloco possui oito vértices, as três

formas geram vinte e quatro posicionamentos diferentes do mesmo bloco. O isomorfismo entre grafos é outra forma de visualização das rotações. Podemos enumerar todas as rotações através de todas as combinações isomórficas entre dois grafos de vizinhança de um bloco (figura 2.8-d).

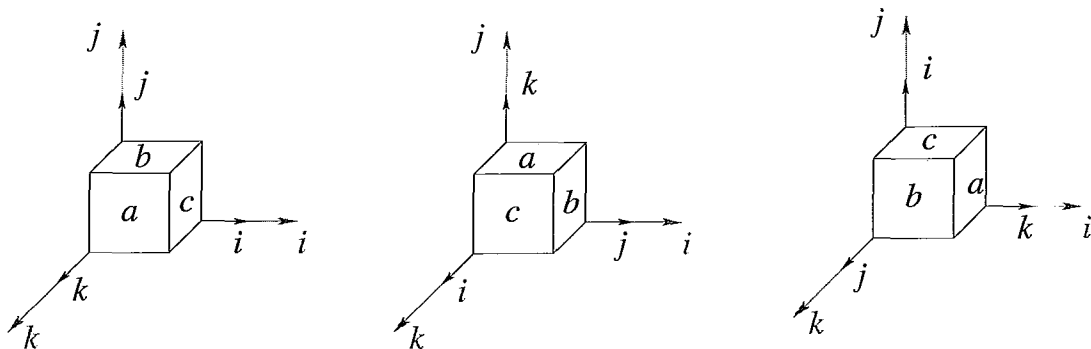


Figura 2.9: Os três tipos básicos de posicionamento de um bloco.

Definição informal

Analogamente ao *modelo 2D**, neste modelo o processo de Auto-montagem começa com a adição da semente e prossegue com a adição de um bloco t_1 , originado de alguma rotação de um tipo de bloco τ_1 , onde t_1 une-se a um bloco t_2 adjacente a t_1 , se a face de t_1 adjacente a t_2 possui o mesmo rótulo r da face de t_2 adjacente a t_1 , porém com polaridades diferentes. Como no *modelo 2D**, a força desta união é dada pela força de atração que o rótulo r possui, onde cada rótulo distinto possui uma força determinada e um bloco t_1 é adicionado ao reticulado, se a soma de todas as uniões que t_1 criou com cada adjacente a ele for maior ou igual a T .

Definição formal

Novamente utilizamos a mesma formulação matemática do *modelo 2D**, porém atualizando as seguintes definições. A Auto-montagem ocorre em um reticulado cúbico $\mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$. L é o conjunto dos tipos de blocos utilizados no processo de Auto-montagem. O rótulo do *modelo 3D* é idêntico ao rótulo do *modelo 2D**. $F = \{1,2,3,4,5,6\}$ é o conjunto de faces de um bloco. Um tipo de bloco $\tau \in L$ é um cubo, onde cada face $x \in F$ possui um rótulo $r \in R$. Bloco é toda cópia de um tipo de bloco $\tau \in L$, posicionado no reticulado cúbico em (i,j,k) e representado pela tripla $(\tau,(i,j,k),rot)$, onde rot é uma

das vinte e quatro rotações de τ . Uma vez escolhida a rotação e posicionado no reticulado cúbico, não é permitido ao bloco mais nenhum tipo de movimento. $Rot = \{rotação\ 0, rotação\ 1, \dots, rotação\ 23\}$ é o conjunto de todas as vinte e quatro rotações possíveis em um tipo de bloco $\tau \in L$. Para calcular a nova face de um rótulo, após a rotação rot , definimos a função $Rot_{3D} : Rot \times F \rightarrow F$, tal que

$$Rot_{3D}(rot, x) = m_{rot,x}, \quad (2.10)$$

onde x é a antiga face do rótulo e $m_{rot,x}$ é um elemento da matriz M_{3D} , tal que

$$M_{3D} = \begin{pmatrix} 1 & 2 & 3 & 4 & 5 & 6 \\ 1 & 3 & 5 & 4 & 6 & 2 \\ 1 & 5 & 6 & 4 & 2 & 3 \\ 1 & 6 & 2 & 4 & 3 & 5 \\ 2 & 1 & 6 & 5 & 4 & 3 \\ 2 & 3 & 1 & 5 & 6 & 4 \\ 2 & 4 & 3 & 5 & 1 & 6 \\ 2 & 6 & 4 & 5 & 3 & 1 \\ 3 & 1 & 2 & 6 & 4 & 5 \\ 3 & 2 & 4 & 6 & 5 & 1 \\ 3 & 4 & 5 & 6 & 1 & 2 \\ 3 & 5 & 1 & 6 & 2 & 3 \\ 4 & 2 & 6 & 1 & 5 & 3 \\ 4 & 3 & 2 & 1 & 6 & 5 \\ 4 & 5 & 3 & 1 & 2 & 6 \\ 4 & 6 & 5 & 1 & 3 & 2 \\ 5 & 1 & 3 & 2 & 4 & 6 \\ 5 & 3 & 4 & 2 & 6 & 1 \\ 5 & 4 & 6 & 2 & 1 & 3 \\ 5 & 6 & 1 & 2 & 3 & 4 \\ 6 & 1 & 5 & 3 & 4 & 2 \\ 6 & 2 & 1 & 3 & 5 & 4 \\ 6 & 4 & 2 & 3 & 1 & 5 \\ 6 & 5 & 4 & 3 & 2 & 1 \end{pmatrix}. \quad (2.11)$$

O tipo de bloco $(\eta, \eta, \eta, \eta, \eta, \eta) \in L$ representa a ausência de bloco, denominada como bloco *vazio*. Não é permitido rotular faces de um tipo de bloco com o rótulo η , com exceção do tipo de bloco *vazio*.

Em um tipo de bloco τ , τ^x representa o rótulo associado à face x . Para identificar o rótulo da face x no bloco $t = (\tau, (i, j, k), rot)$, aplicamos a operação $t^x = \tau^{Rot_{3D}(rot, x)}$. Uma *configuração* é uma forma geométrica montada por blocos, representada por um conjunto

C de blocos não vazios, originados do conjunto L , onde não existam dois blocos com a mesma posição no reticulado cúbico. Para qualquer configuração C , $C_{(i,j,k)}$ representa o bloco $t \in C$ posicionado em (i,j,k) ou o bloco (vazio, (i,j,k) , rotação 0), caso não exista bloco na posição (i,j,k) . A função $Loc_{3D}(t)$ retorna a posição (i,j,k) do bloco $t \in C$ no reticulado cúbico, onde $Loc_{3D} : C \rightarrow \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$. Duas configurações C e D são ditas *idênticas*, quando possuem os mesmos blocos nas mesmas posições. Se x é uma face de um bloco, sua face oposta é aquela no lado oposto do bloco. O cálculo da face oposta a x é dado pela função $FO_{3D} : F \rightarrow F$, onde

$$FO_{3D}(x) = ((x + 2)\%6) + 1. \quad (2.12)$$

Note que, se uma face x é vizinha de um face y em um bloco, então $FO_{3D}(x) \neq y$.

Se (i,j,k) é a posição de um bloco t no reticulado cúbico e x é uma das faces deste bloco, a função $P_{3D} : F \times \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z} \times \mathbb{Z} \times \mathbb{Z}$ calcula a posição adjacente à face x , tal que

$$P_{3D}(x, Loc_{3D}(t)) = \begin{cases} (i, j, k + 1), & \text{se } x = 1 \\ (i, j + 1, k), & \text{se } x = 2 \\ (i + 1, j, k), & \text{se } x = 3 \\ (i, j, k - 1), & \text{se } x = 4 \\ (i, j - 1, k), & \text{se } x = 5 \\ (i - 1, j, k), & \text{se } x = 6. \end{cases} \quad (2.13)$$

Dois blocos $t_1, t_2 \in C$, posicionados em (i_1, j_1, k_1) e (i_2, j_2, k_2) são ditos adjacentes quando $|i_1 - i_2| + |j_1 - j_2| + |k_1 - k_2| = 1$ ou, equivalentemente, caso exista uma face x de t_1 , onde $P_{3D}(x, t_1) = Loc_{3D}(t_2)$ ou, equivalentemente, $P_{3D}(FO_{3D}(x), t_2) = Loc_{3D}(t_1)$.

A função $FA_{3D}(r_1, r_2) = FA_{2D*}(r_1, r_2)$, calcula a força de atração entre os rótulos r_1 e r_2 . A força com a qual um bloco $t \in C$ é atraído por um bloco adjacente a ele na configuração C , é dada pela função $FI_{3D}^C : F \times C \rightarrow \mathbb{N}$, tal que

$$FI_{3D}^C(x, t) = FA_{3D}(t^x, (C_{P_{3D}(x, Loc_{3D}(t))})^{FO_{3D}(x)}). \quad (2.14)$$

A função $FT_{3D}^C : F \rightarrow \mathbb{N}$, onde

$$FT_{3D}^C(t) = \sum_{x \in F} FA_{3D}(t^x, (C_{P_{3D}(x, Loc_{3D}(t))})^{FO_{3D}(x)}), \quad (2.15)$$

é a soma das forças de *ligação* de t com os seus blocos adjacentes, representando a força que mantém o bloco t unido à *configuração* C .

Um bloco $t \in C$ cria uma *ligação* com um bloco adjacente $P_{3D}(x, Loc_{3D}(t)) \in C$, caso $FI_{3D}^C(x, t) > 0$. Em uma *configuração* C T -estável, o bloco t é adicionado se $FT_{3D}^C(t) \geq T$. Um *sistema de Blocos SL* no *modelo 3D* é representado pela quádrupla (T, S, L, FA_{3D}) . No *modelo 3D*, o processo de Auto-montagem ocorre de forma idêntica ao *modelo 2D**, obedecendo à relação $C \rightarrow_{SL} D$.

2.3 O problema do conjunto mínimo de ladrilhos

A construção de formas e padrões geométricos através do processo de Auto-montagem consiste em determinar o *sistema de ladrilhos SL* necessário para compor a forma ou padrão objetivo. O final do processo deve ser pré-determinado e nenhuma interferência externa é aceitável. Várias soluções para formas bidimensionais mais simples, como quadrados e retângulos foram estudadas (Cheng e Espanes 2003; Rothmund e Winfree 2000). Porém, soluções para formas genéricas em duas dimensões ou qualquer forma em três dimensões são um problema em aberto. Inserida no mesmo contexto, a computação em DNA utiliza formas e padrões para executar determinada computação (Baryshnikov, Jr. e Momcilovic 2004). Encontrar o conjunto de regras necessárias à construção dessas formas auxiliaria a diminuição ou eliminação do erro na computação em DNA que existe atualmente (Ezziane 2006). Sendo assim, o problema do conjunto mínimo de ladrilhos (PCML) é um problema de otimização combinatória NP-Difícil (Adleman et al. 2002), cujo objetivo é encontrar a semente S , o conjunto de tipos de ladrilhos L e a função FA_D do *sistema de ladrilhos* $SL=(T, S, L, FA_D)$, que a partir de S construa a forma geométrica desejada, onde $|L|$ seja o menor possível, FA_D é a força que define a atração entre os rótulos, $D \in \{2D, 2D*, 3D\}$ e T é um parâmetro. Note que, ao tratarmos este problema no *modelo 3D*, queremos minimizar o conjunto de tipos de blocos. Para facilitar o enten-

dimento, a denominação elemento refere-se a um ladrilho nos modelos bidimensionais ou a um bloco no modelo tridimensional.

2.3.1 Critérios

Construir a forma geométrica desejada utilizando um *sistema de elementos SL* não é suficiente. Para considerarmos um *SL* como solução do problema, *SL* deve cumprir às restrições do PCML, representadas pelos seguintes critérios.

Terminação

Geralmente o processo de Auto-montagem produz uma série de *configurações* intermediárias a cada passo dado pela relação $C(n) \rightarrow_{SL} C(n+1)$. $Prod_{SL}(S)$ é o conjunto de todas as *configurações* intermediárias produzidas por *SL*, iniciando na semente *S*. Formalmente definimos $Prod_{SL}(S)$, tal que

$$Prod_{SL}(S) = \{B : \exists m > 0 : A \rightarrow_{SL}^m B\}, \quad (2.16)$$

onde *B* é uma *configuração* intermediária, *A* é *configuração* cujo único elemento é a *semente* e a relação $A \rightarrow_{SL}^m B$ representa a adição de *m* elementos à *configuração* *A*.

Se em um determinado passo finito n_f , não existe a possibilidade de adicionarmos mais elementos, então *SL* é um *sistema de elementos terminal*. Equivalentemente, *SL* é *terminal*, se $Prod_{SL}(S)$ é um conjunto finito.

Unicidade

O *sistema de elementos terminal SL* é *único*, se todas as *configurações* maximais $B \in Prod_{SL}(S)$ possuem a mesma forma geométrica, não importando a ordem ou quais tipos de elementos foram escolhidos para construir estas *configurações*. Vale ressaltar que não existe necessariamente uma ordem determinística de montagem de uma forma geométrica, porém um *sistema de elementos* é dito *terminal e único* se e somente se todas as ordem de montagem terminam construindo a mesma forma geométrica. Por exemplo, os tipos de elementos da figura 2.10-a, terminam sempre montando uma L com três elementos no *modelo 2D*. Entretanto, existem quatro ordens de montagem. Duas delas são apresentadas nas figuras 2.10-b e 2.10-c.

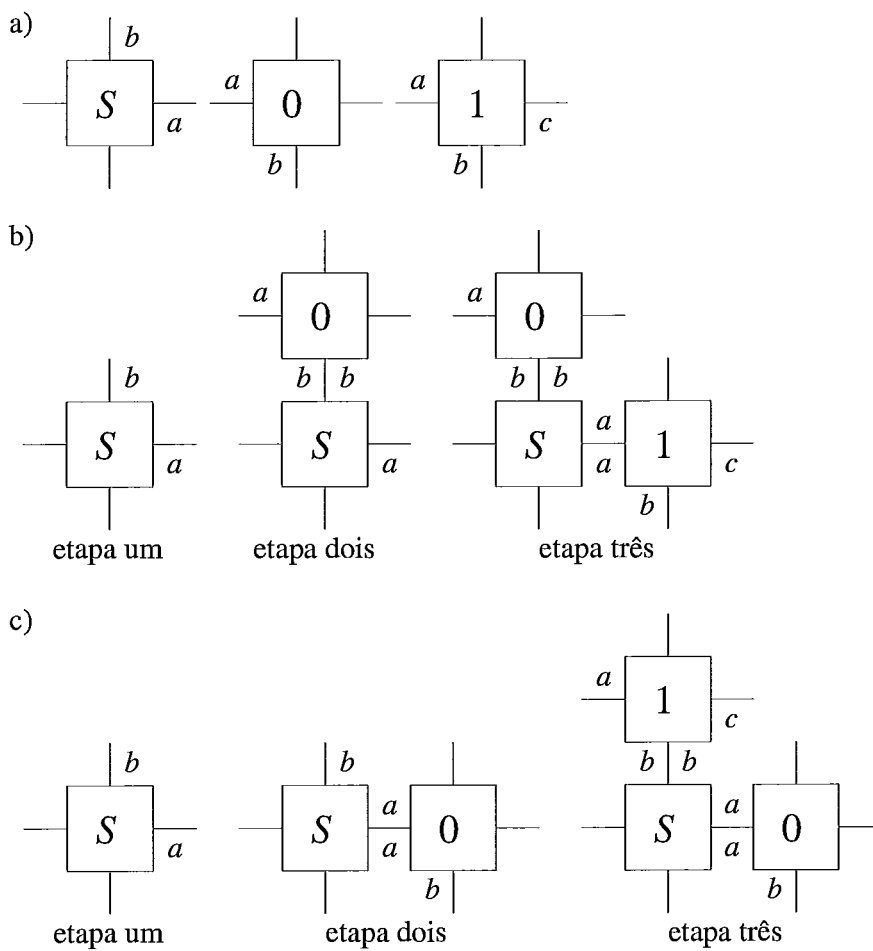


Figura 2.10: Ordem de montagem.

No *modelo 2D**, esta propriedade é observada na figura 2.11-a, onde a *semente S* e o conjunto de tipos de elementos constroem somente quadrados 3×3 . Note que, o *sistema de elementos SL* desta figura constrói duas *configurações* diferentes, pois o elemento $(4,(0,0),\text{rotação } 3)$ da figura 2.11-b é diferente do elemento $(4,(0,0),\text{rotação } 1)$ da figura 2.11-c. Os rótulos $a-, a+, b-, b+, x- e x+$ possuem força igual a 2, os demais rótulos possuem força igual a 1 e a temperatura é igual a 2.

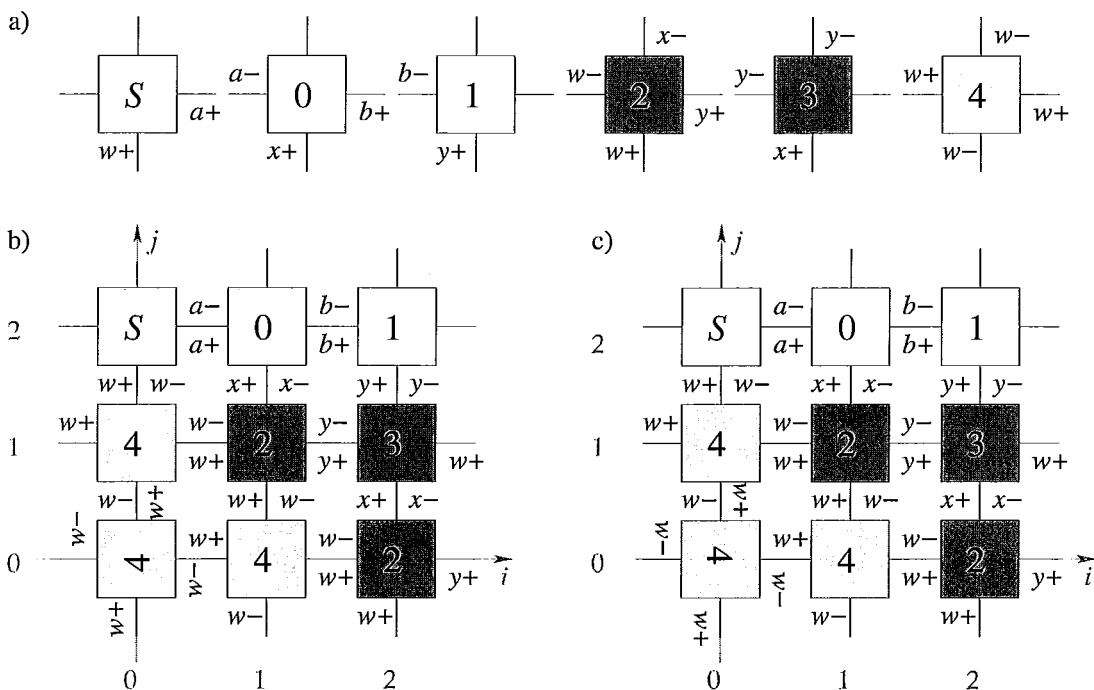


Figura 2.11: A relação entre a rotação e a unicidade.

Atingir a forma objetivo

A forma geométrica que desejamos construir é denominada como *forma objetivo*. Por suas utilidades na computação em DNA (Adleman et al. 2001), as formas mais estudadas são os quadrados, retângulos, linhas e árvores (Adleman et al. 2002). Uma *forma objetivo* é composta por elementos e possui tamanho definido. Ao especificarmos uma *forma objetivo* devemos escolher a forma geométrica, suas dimensões e a quantidade de elementos que a compõem. Um exemplo de *forma objetivo* foi apresentado na etapa quatro da figura 2.1-b, cuja forma geométrica é um quadrado, composta por quatro elementos com as dimensões 2×2 . Dizemos que o *sistema de elementos terminal e único SL* atinge a *forma objetivo*, se *SL* produz uma *configuração C* com forma geométrica idêntica à forma geométrica desejada.

Construir uma forma completa

Um *sistema de elementos terminal e único SL* é uma solução para o PCML, se *SL* atinge a forma desejada *C*, tal que *C* seja uma *forma completa*. *C* é uma *forma completa*, se cada elemento $t \in C$ possui uma *ligação* com todos os elementos adjacentes a ele. A figura 2.12-b ilustra uma *forma completa*, cuja *semente S* e o conjunto de tipos de elementos utilizados para construir esta forma são apresentados na figura 2.12-a. Observe que cada elemento da figura 2.12-b está ligado a todos os seus adjacentes. Por outro lado, a forma apresentada na figura 2.12-d não é completa, pois não há ligação entre os elementos identificados pelo número 3, como também não há ligação entre os elementos identificados pelo número 2. Devido à falta de ligações entre alguns dos elementos desta figura, a forma geométrica pode degenerar para a forma da figura 2.12-e, que não é um quadrado. A *semente S* e o conjunto de tipos de elementos utilizados para construir a forma da figura 2.12-d, são apresentados na figura 2.12-c. Todos os rótulos representados na figura 2.12 possuem força suficiente para vencer a *disruptura termal*, exceto o rótulo *w*. Este rótulo precisa criar duas ligações para vencer a *disruptura termal*.

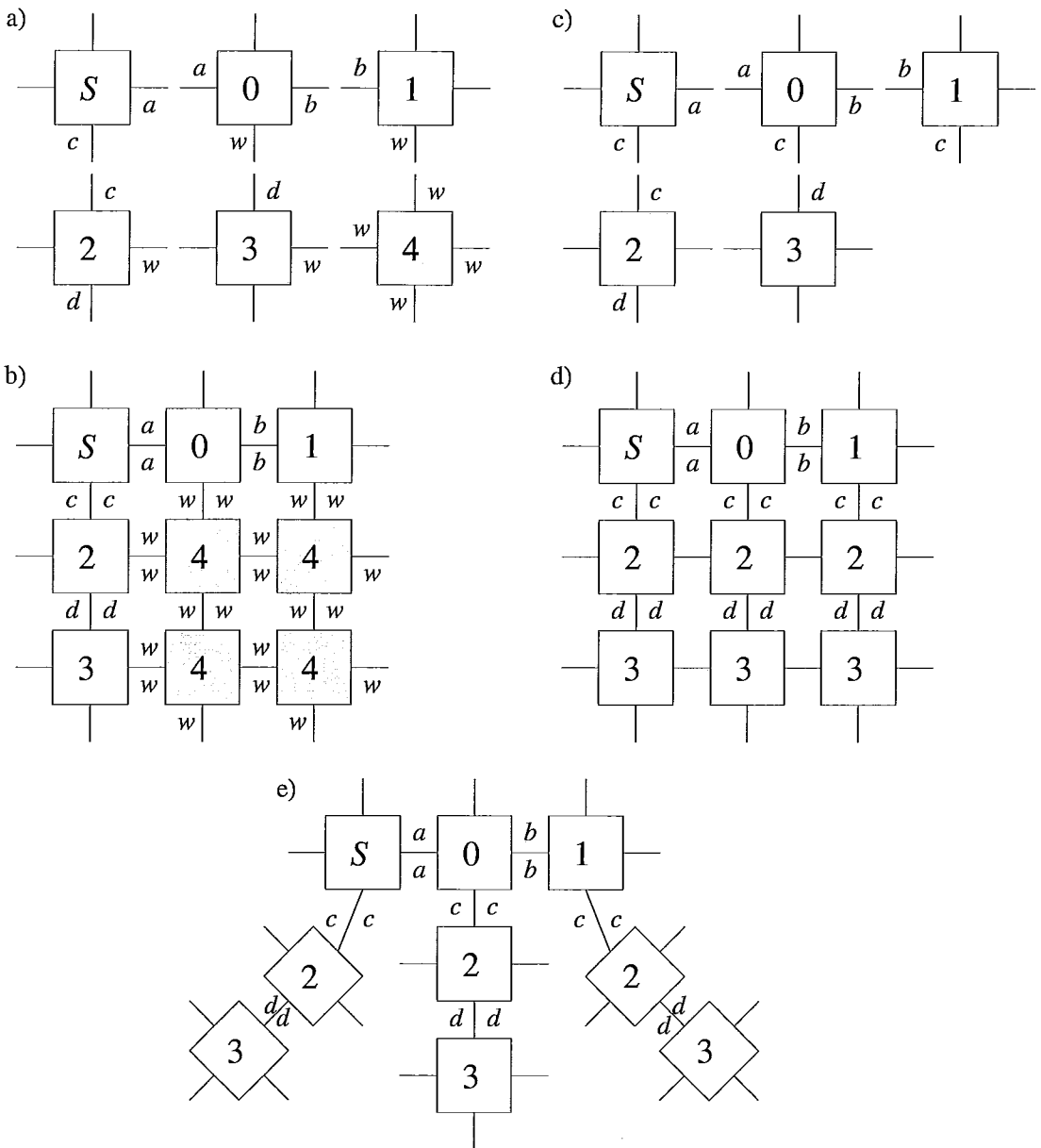


Figura 2.12: Forma completa para quadrados.

2.3.2 Soluções

Neste trabalho foram estudados quadrados $N \times N$ no *modelo 2D*, pois são as formas mais estudadas, sua solução já foi determinada e servem como parâmetro de comparação entre outros estudos do problema (Rothmund e Winfree 2000). Analogamente, apresentamos soluções para o novo *modelo 2D** em quadrados $N \times N$, proporcionando um ponto de comparação entre os dois modelos. Comparado aos modelos bidimensionais, o modelo tridimensional oferece um grau de liberdade maior para criação de formas geométricas. No *modelo 3D*, escolhemos estudar inicialmente formas cúbicas $N \times N \times N$. Estudos demonstram que é possível encontrar soluções melhores, programando a variação da temperatura e conseqüentemente diminuindo o tamanho do conjunto de tipos de ladrilhos (Kao e Schweller 2006). Porém, todas as soluções apresentadas neste trabalho possuem temperatura fixa, uma vez que o *sistema de ladrilhos* foi definido originalmente com apenas um valor de temperatura, como também não foi provada a otimalidade destas soluções. Vale destacar que, um *sistema de elemento* $SL_1 = (T, S_1, L_1, FA_{D1})$ é uma solução melhor que $SL_2 = (T, S_2, L_2, FA_{D2})$ para o PCML, se $|L_1| < |L_2|$. Observem que todas as soluções apresentadas nesta seção foram obtidas dos experimentos deste trabalho (seção 4).

Formas quadráticas no *modelo 2D*

As melhores soluções para formas quadráticas do *modelo 2D*, foram apresentadas originalmente em (Rothmund e Winfree 2000), que também descobriu que em um *sistema de ladrilhos* (T, S, L, FA_{2D}) , o tamanho do conjunto de tipos de ladrilhos L não é menor que N^2 , se a temperatura $T = 1$. Entretanto, Rothmund e Winfree mostraram que se a $T = 2$, o tamanho do conjunto de tipos de ladrilhos é $N + 4$.

Para faixas diferentes de N , são adotadas soluções diferentes, ou seja, dependendo do tamanho do quadrado desejado existem várias sementes diferentes e vários conjuntos de tipos de ladrilhos diferentes, cujo número de ladrilhos varia de acordo com N . Estas soluções foram divididas em três faixas de acordo com N e são apresentadas na tabela 2.1, onde $T = 2$ e $\log_2^* N$ é igual ao número de aplicações do operador \log_2 à N , até que o resultado seja igual ou menor que 1. Nesta tabela, a coluna Faixa possível significa que a

solução pode ser adotada para quadrados, cujo tamanho seja maior ou igual ao indicado. A coluna Faixa ótima indica o tamanho onde a solução é a melhor possível, e a coluna Tipos de ladrilhos necessários representa o tamanho de L acrescido da *semente*. Até o momento, nenhum estudo demonstrou soluções melhores com $T = 3$ para quadrados $N \times N$.

Tabela 2.1: Faixa de aplicação das soluções no *modelo 2D*.

Faixa possível	Faixa ótima	Tipos de ladrilhos necessários
$N \geq 3$	$3 \leq N \leq 23$	$N + 4$
$N \geq 5$	$23 \leq N \leq 22 + 2^{23} - 1$	$22 + \lceil \log_2 N \rceil$
$N \geq 22 + 2^{23}$	$22 + 2^{23} \leq N$	$22 \log_2^* N$

A figura 2.13 apresenta a solução para a faixa $3 \leq N \leq 23$, onde a figura 2.13-a representa a *semente* S e o conjunto L e a figura 2.13-b ilustra as etapas da construção da *forma objetivo* que é mostrada na figura 2.13-c. Os rótulos a, b, c, d e x possuem força igual a 2, os demais rótulos força igual a 1 e a temperatura é igual a 2. A solução ilustrada na figura 2.14, para a faixa $23 \leq N \leq 22 + 2^{23} - 1$, inicia a partir da solução para a faixa $3 \leq N \leq 23$. Os tipos de ladrilhos 7 a 25, do conjunto de tipos de ladrilhos da figura 2.14-a, possibilitam a construção de um quadrado 25×25 , mostrado na figura 2.15. A construção começa com a semente S e os tipos de ladrilhos 0 a 6, baseados nos tipos de ladrilhos da faixa $3 \leq N \leq 23$, como na figura 2.13. Os rótulos a, b, c, d, e, f, g, h e x possuem força igual a 2, os demais rótulos força igual a 1 e a temperatura é 2. A faixa de solução $22 + 2^{23} \leq N$ não foi estudada, pois o espaço requerido para a representação dos quadrados produzidos inviabilizam a computação das simulações deste trabalho.

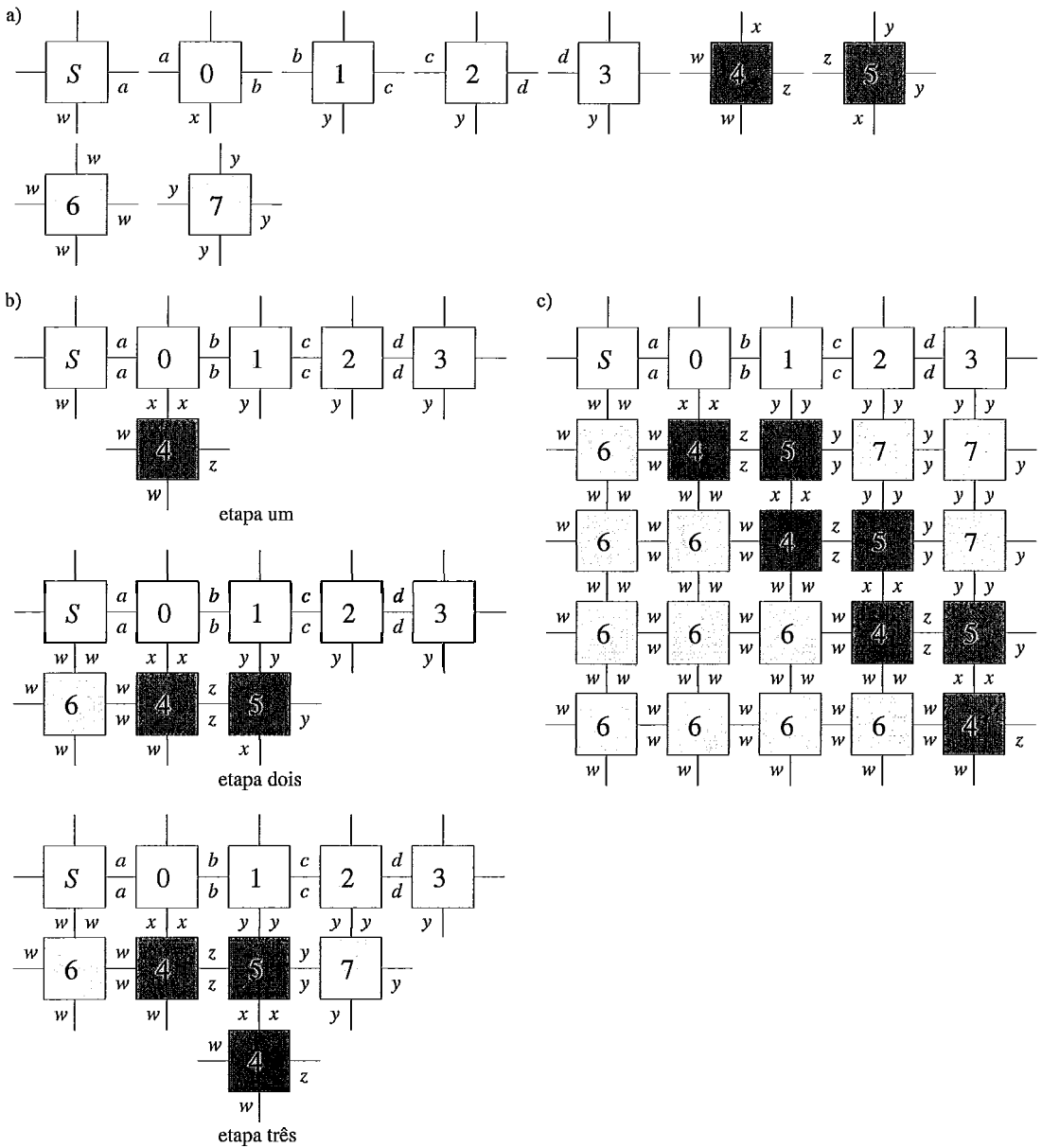


Figura 2.13: Solução para um quadrado 5×5 no *modelo 2D*.

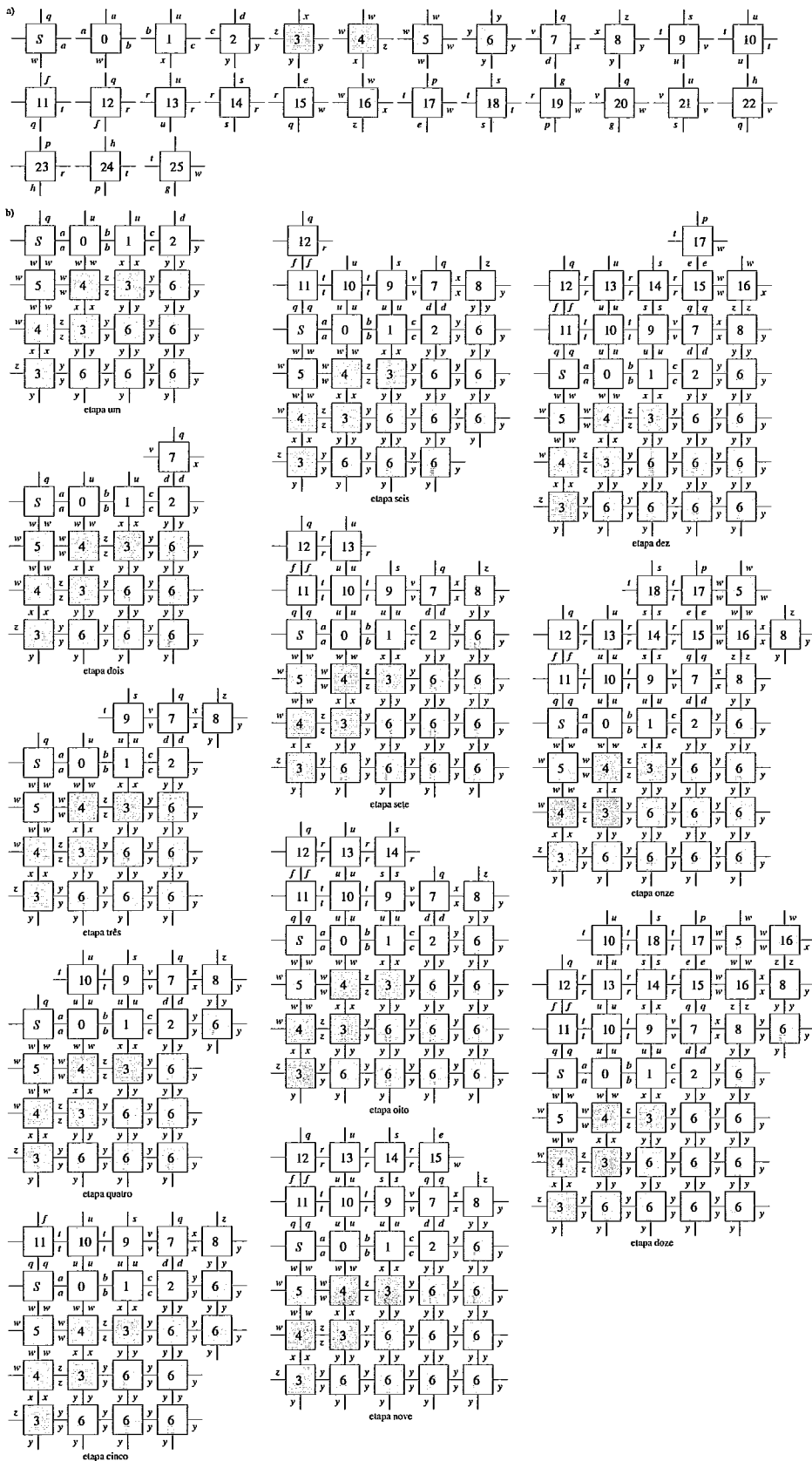


Figura 2.14: Solução para um quadrado 25×25 no *modelo 2D*.

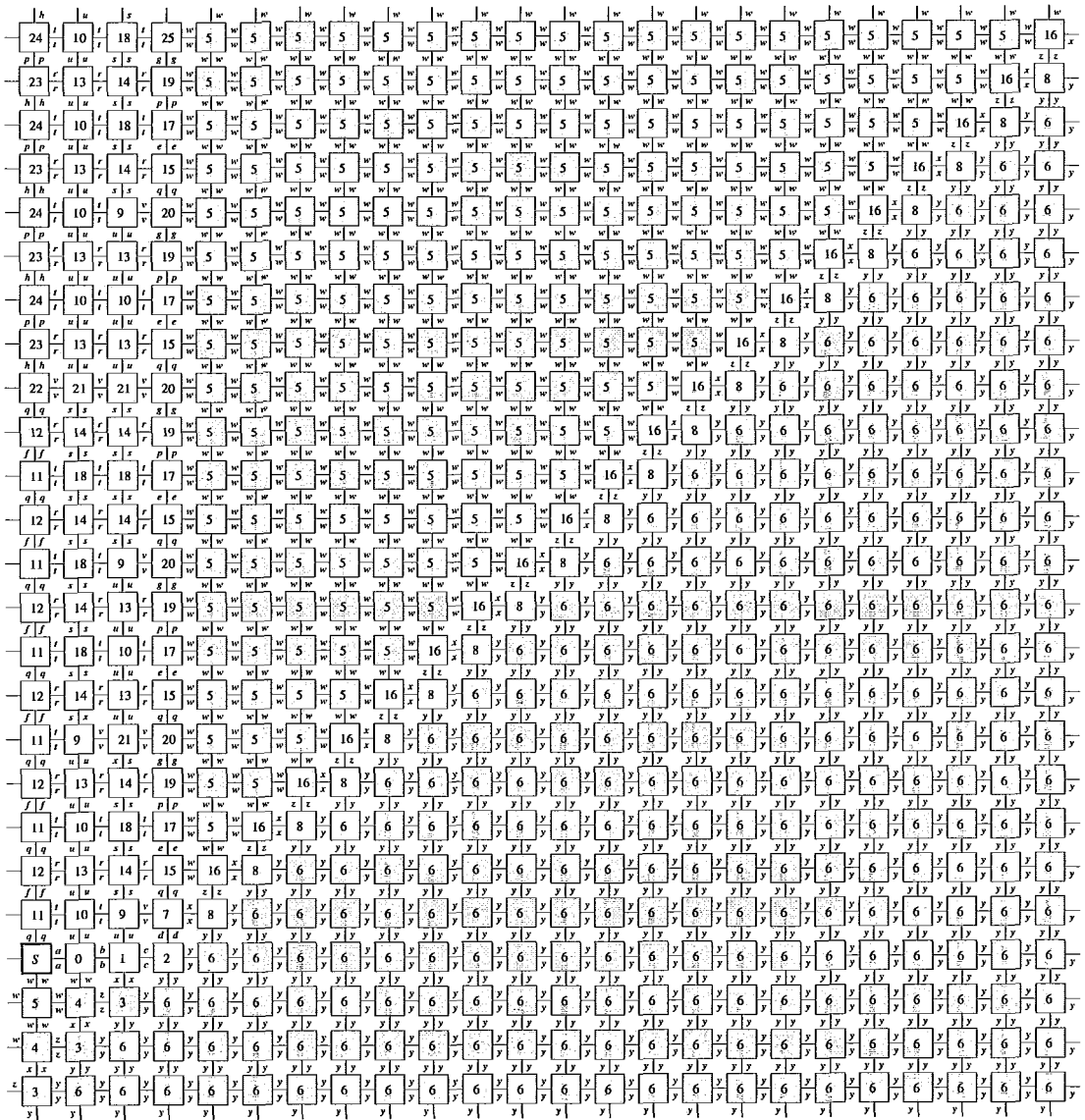


Figura 2.15: Forma objetivo para um quadrado 25×25 no *modelo 2D*.

Formas quadráticas no modelo 2D*

O modelo 2D* possui soluções melhores, se comparado ao modelo 2D, devido à diminuição na variedade dos tipos de ladrilhos e o aumento na flexibilidade das uniões (seção 2.2.2). Para observar a vantagem do modelo 2D* sobre o modelo 2D, compare as figuras 2.16 e 2.13. A figura 2.13 apresenta a solução para um quadrado 5×5 , onde o número de tipos de ladrilhos utilizado é $N + 4$. A mesma solução no modelo 2D* é apresentada na figura 2.16 e utiliza $N + 3$ tipos de ladrilhos. Nesta figura, a semente S e os tipos de ladrilhos utilizados na montagem do quadrado da figura 2.16-c são apresentados na figura 2.16-a. As etapas intermediárias são mostradas na figura 2.16-b. Os rótulos $a+$, $a-$, $b+$, $b-$, $c+$, $c-$, $d+$, $d-$, $x+$ e $x-$ possuem força igual a 2, os demais rótulos força igual a 1 e a temperatura é igual a 2.

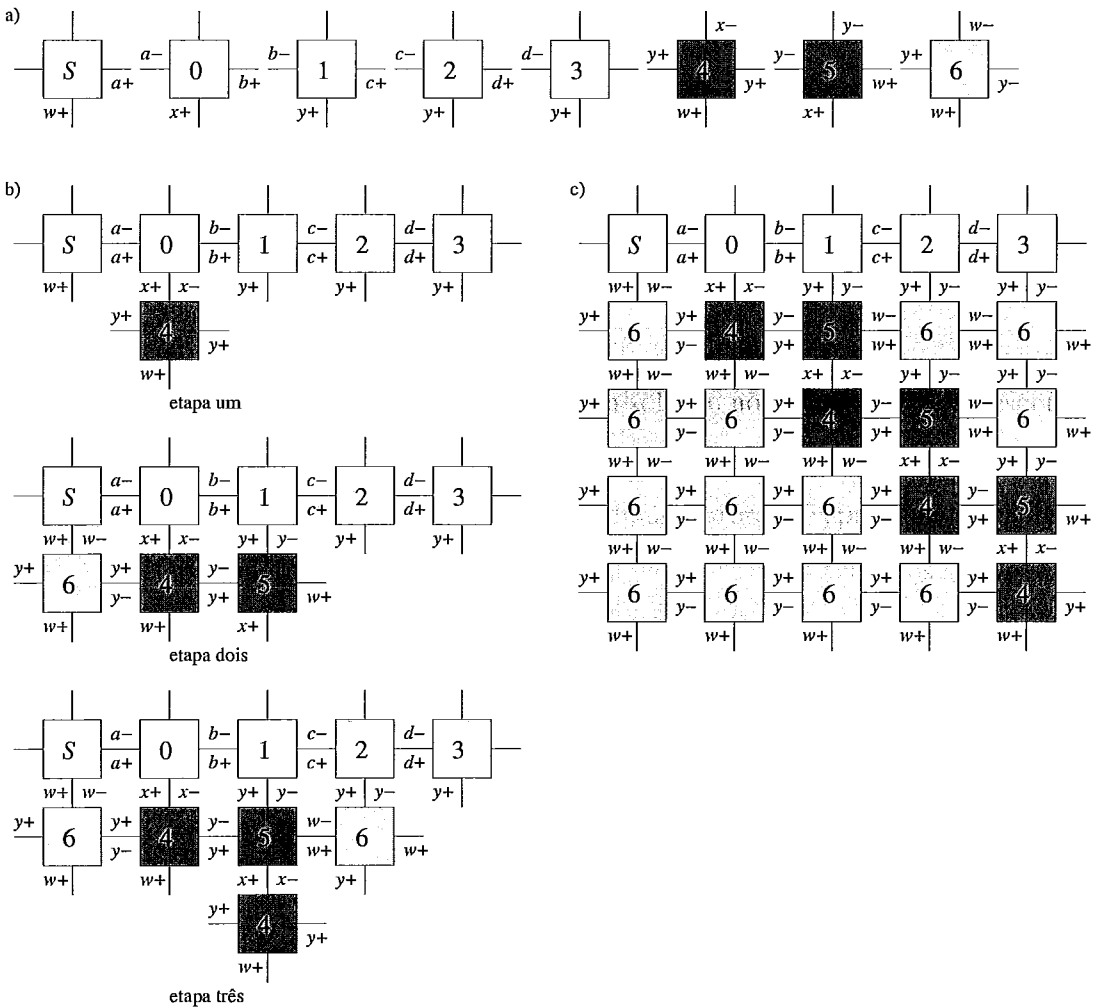


Figura 2.16: Solução para um quadrado 5×5 no modelo 2D* com $N + 3$ tipos de ladrilhos.

Entretanto, esta solução pode ser aprimorada. Na figura 2.17 ilustramos a melhor solução encontrada pelos experimentos realizados por este trabalho, onde a *semente* S e o conjunto de tipos de ladrilhos utilizados para construção da figura 2.17-c, são apresentados na figura 2.17-a. Na figura 2.17-b os ladrilhos 0 a 3 foram rotacionados em sentido horário, para que fossem encaixados na aresta vertical do quadrado. Os rótulos $a+$, $a-$, $b+$, $b-$, $c+$, $c-$, $d+$ e $d-$ possuem força dois, os demais rótulos possuem força igual a 1 e a temperatura é igual a 2. Para este modelo não foram encontradas soluções por faixa de N , pois este trabalho limitou-se a estudar quadrados 5×5 .

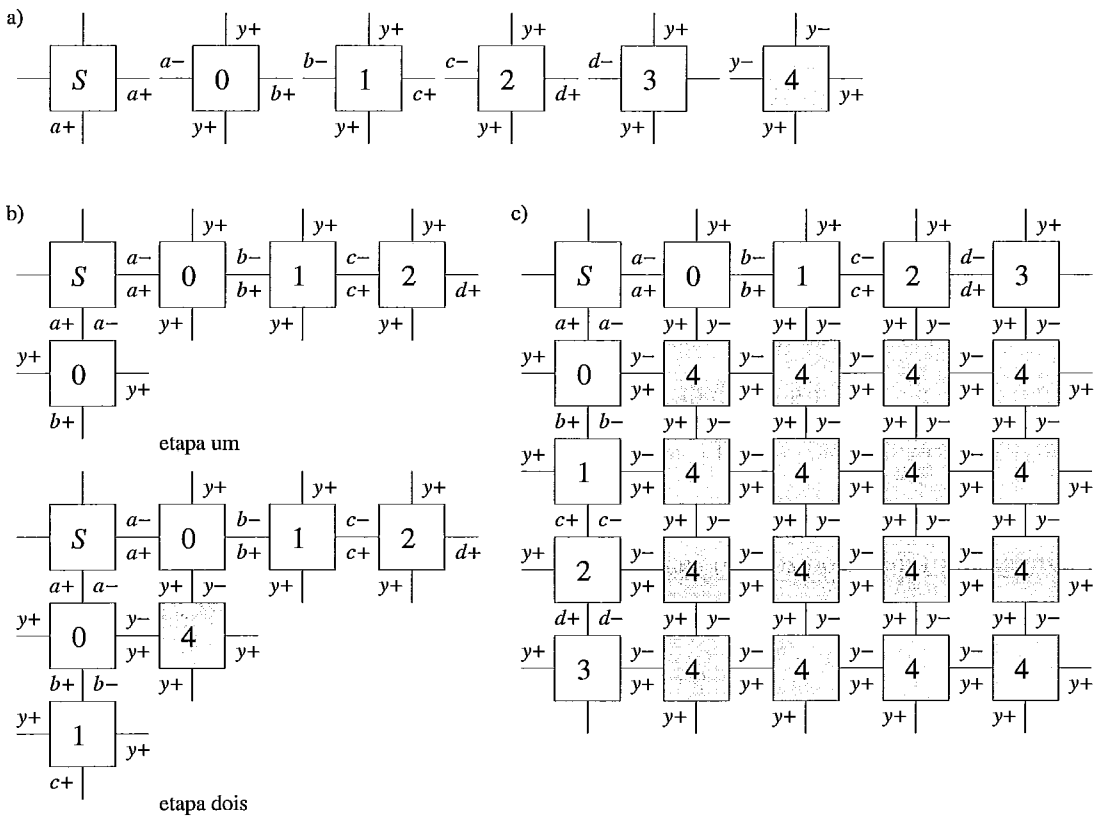


Figura 2.17: Solução para um quadrado 5×5 com $N + 1$ tipos de ladrilhos no *modelo* $2D^*$.

Formas cúbicas

O *modelo 3D* é o modelo que possui o maior grau de liberdade apresentado até o momento. Isto indica a possível existência de soluções melhores que as encontradas para os modelos bidimensionais. A figura 2.18 comprova essa melhora, pois o tamanho do conjunto L é o mesmo dos modelos bidimensionais. Nesta figura, todos os rótulos possuem força igual a 2, exceto o rótulo y que possui força um. A figura 2.18-b mostra as etapas de montagem do cubo, para a temperatura igual a 2. Nesta solução a quantidade de tipos de ladrilhos é a mesma da melhor solução para o *modelo 2D** (figura 2.17). Esta característica é muito interessante, pois aumentamos em uma dimensão a forma geométrica, mas não aumentamos a quantidade necessária de tipos de ladrilhos. O *modelo 3D* mostra-se promissor, pois ele torna possível a produção de formas tridimensionais com N^3 blocos e a mesma quantidade de tipos de ladrilhos dos modelos bidimensionais, que constroem formas com apenas N^2 ladrilhos. Não estudamos se o aumento da temperatura produz soluções melhores no *modelo 3D* que as apresentadas neste trabalho.

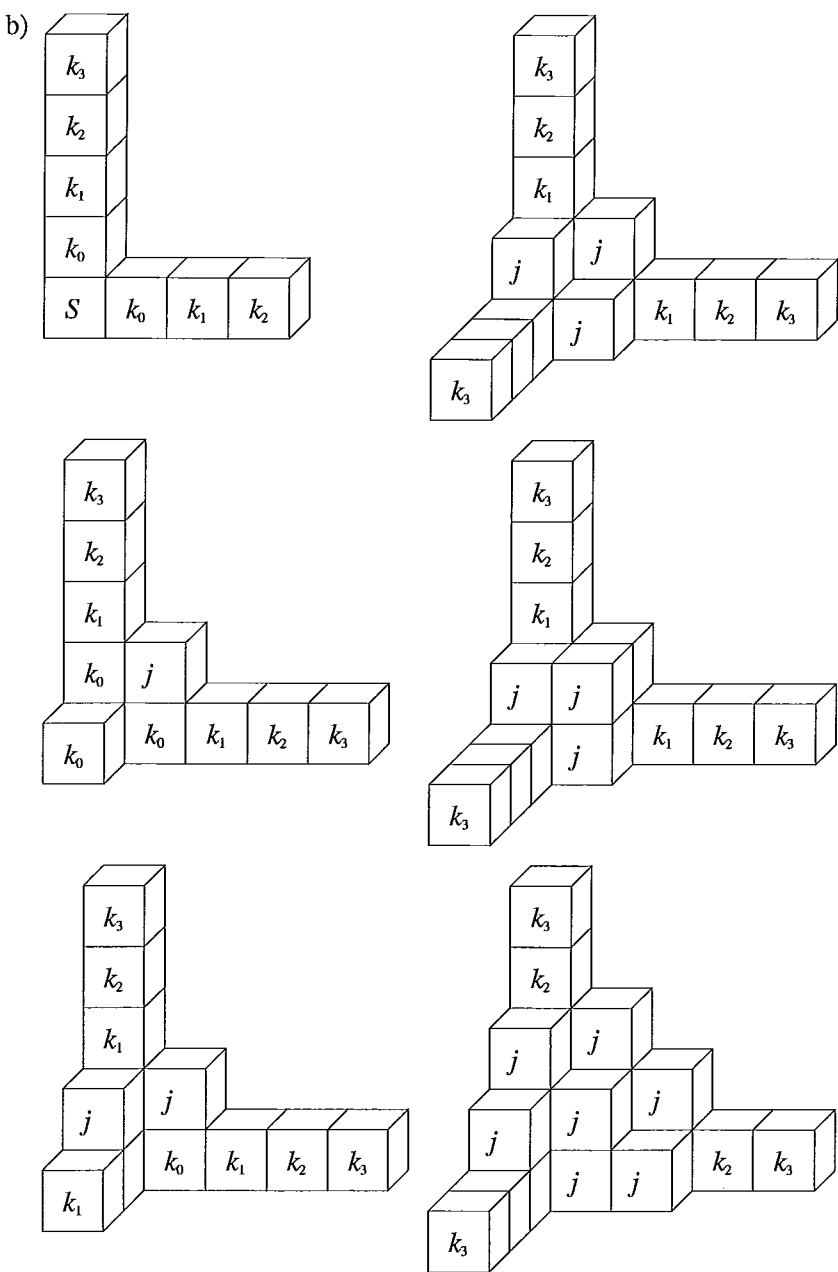
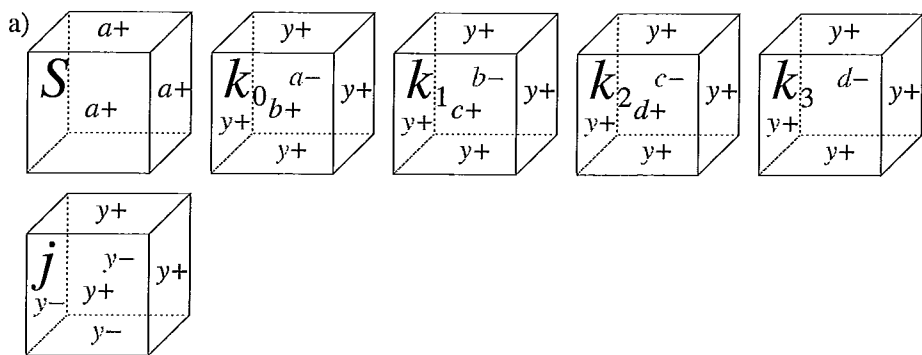


Figura 2.18: Solução para um cubo $5 \times 5 \times 5$.

2.4 O problema do conjunto mínimo de rótulos

Neste trabalho estudamos o *sistema de ladrilhos SL*, buscando minimizar o seu conjunto de tipos de ladrilhos. Em (Rothenmund e Winfree 2000) foi discutida a possibilidade de minimizar o tamanho do conjunto de rótulos R . Como a variedade de rótulos diferentes em sistemas reais é escassa (Cooper e Hausman 2003), seria interessante reduzir o número de rótulos diferentes utilizados em SL . As melhores soluções apresentadas em (Rothenmund e Winfree 2000), requerem no mínimo $N + 3$ rótulos distintos. Este trabalho encontrou uma solução melhor, onde são necessários no mínimo $N + 2$ rótulos distintos. A figura 2.19 ilustra esta solução. A figura 2.19-a representa a *semente S* e o conjunto de tipos de ladrilhos. A figura 2.19-b ilustra a construção da *forma objetivo* da figura 2.19-c. A montagem é semelhante a da figura 2.13, porém esta solução possui um rótulo a menos. Os rótulos a, b, c, d e x possuem força igual a 2 e os demais rótulos possuem força igual a 1.

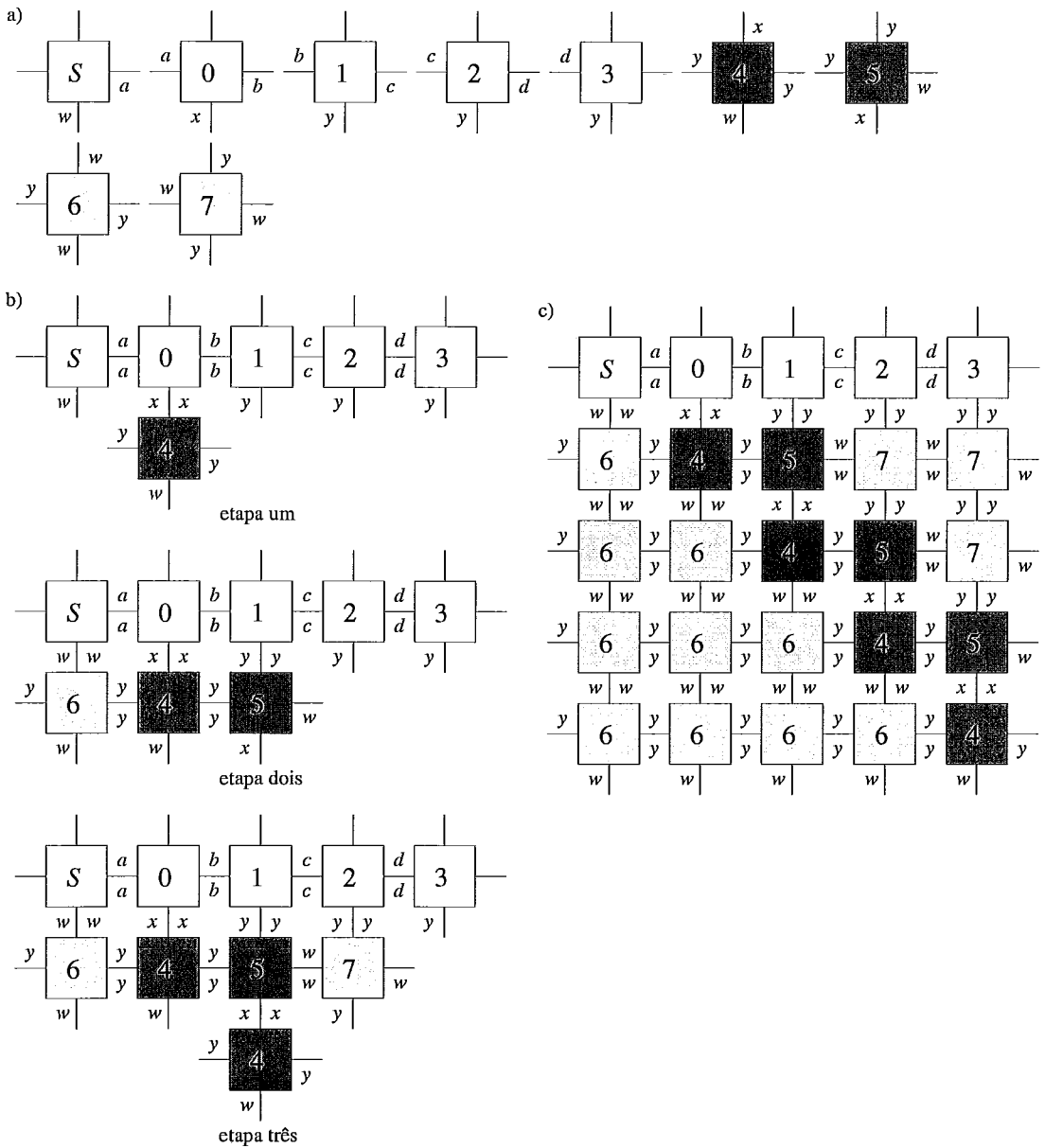


Figura 2.19: Solução para um quadrado 5×5 no *modelo 2D* com $N + 2$ rótulos distintos.

CAPÍTULO 3

Heurística para o PCML

O PCML tem sido tradicionalmente solucionado através de análises e deduções matemáticas (Adleman 2000; Adleman et al. 2001; Adleman et al. 2002; Cheng e Espanes 2003). Porém, este tipo de solução pode ser aplicada apenas a formas simples e em duas dimensões. Formas genéricas em duas ou três dimensões dificultam a solução do problema, pois contribuem para o aumento da complexidade. É importante observar que muitos trabalhos estão direcionados ao estudo da complexidade da computação que ocorre durante a construção de formas geométricas em duas dimensões (Rothmund e Winfree 2000; Soloveichik e Winfree 2004; Reif, Sahu e Yin 2005; Angelov, Khanna e Visontai 2006). Para formas genéricas em duas ou três dimensões é necessário adotar soluções que empreguem heurísticas com o objetivo de diminuir o espaço de busca. Neste sentido, a Universidade de Nottingham realizou um estudo utilizando Algoritmos Genéticos clássicos (Mitchell 1998) para a solução do problema em duas dimensões. Porém, a simulação do processo de Auto-montagem não está de acordo com a Auto-montagem Algorítmica proposta por Winfree (Winfree 1998) e adotada neste trabalho. A preocupação do trabalho da Universidade de Nottingham era simular o processo, o mais próximo possível dos processos bioquímicos que ocorrem na Auto-montagem, colocando em segundo plano a construção de formas geométricas em duas dimensões. Dessa forma, esse ainda é um estudo com uma abordagem preliminar sobre o processo de Auto-montagem (Terrazas et al. 2005). Para facilitar o entendimento, a denominação elemento refere-se a um ladrilho nos modelos bidimensionais ou a um bloco no modelo tridimensional, assim como a de-

nominação face refere-se ao lado de um ladrilho ou à face de um bloco, apenas na seção 3.1.

Inicialmente, tratamos o PCML como um problema de otimização combinatória com um único objetivo. Por ser um problema sem solução analítica, optamos por utilizar uma Meta-Heurística Evolucionária (Back, Hoffmeister e Schwefel 1991), como Algoritmos Genéticos (AG). Porém, uma análise mais apurada e estudos anteriores (Krasnogor e Gustafson 2005; Krasnogor et al. 2005) revelaram alguns problemas quando o PCML é tratado como um problema de otimização combinatória comum. Optamos por tratar o PCML como um problema de otimização multi-objetivo (Taboada et al. March 2007), utilizando Meta-Heurísticas Evolucionárias Multi-Objetivo (Gaspar-Cunha, Gonsalves e Covas 2006), pois um candidato a solução deve cumprir alguns critérios (seção 2.3.1) que estão indiretamente relacionados à minimização do conjunto de tipos de elementos. Cada critério é um objetivo a ser alcançado pelo candidato, além do objetivo principal definido anteriormente. Dessa forma, utilizamos uma abordagem inovadora para solucionar o PCML.

3.1 Descrição do algoritmo

As Meta-Heurísticas Evolucionárias Multi-Objetivo (MEMO) oferecem uma grande variedade de algoritmos, porém não existe nenhum estudo ou teste comparando o desempenho desses algoritmos (Coello 2000). Sendo assim, adotamos uma abordagem simplista. Inicialmente implementamos o MEMO mais simples possível: um Algoritmo Genético, onde a função de avaliação dos indivíduos possui várias componentes. Ao longo dos experimentos, agregamos novas particularidades que aumentam a eficiência e tornam este algoritmo singular, se comparado com o número considerável de algoritmos propostos atualmente (Coello, Pulido e Montes 2005). A seguir descrevemos estas particularidades, assim como os parâmetros do MEMO, o qual denominamos Algoritmo Genético Multi-Objetivo Simples (AMS).

3.1.1 O indivíduo

A estratégia principal de qualquer AG é a recombinação e alteração dos indivíduos. Em (Holland 1975) foi introduzida a idéia de *blocos de construção* (BC), onde um AG eficiente é aquele capaz de identificar os conjuntos de características bem sucedidas nos indivíduos, ou seja, os *blocos de construção* das soluções. Logo devemos viabilizar a criação e evolução dos *blocos de construção* nos indivíduos, através de seu cromossomo. No PCML, identificamos os *blocos de construção* como os subconjuntos de tipos de elementos do cromossomo capazes de construir formas geométricas próximas a *forma objetivo*.

O cromossomo

No AMS, um elemento ou alelo do cromossomo é um tipo de elemento do conjunto L em um *sistema de elementos* $SL=(T,S,L,FA_D)$, onde $D \in \{2D, 2D^*, 3D\}$. O cromossomo não é o conjunto L , mas sim uma lista ordenada de tipos de elementos, pois são permitidas repetições de tipos de elementos no cromossomo. O tamanho do cromossomo varia entre os valores $tamanhoMínimo_{indivíduo}$ e $tamanhoMáximo_{indivíduo}$, pois o número mínimo de tipos e elementos necessários para atingir uma *forma objetivo* é desconhecido. Logo não é possível estabelecer um tamanho ideal para o cromossomo. Os tamanhos mínimo e máximo do cromossomo foram pré-estabelecidos nos experimentos (seção 4.1).

A semente

Como a semente não pertence a L , cada indivíduo deveria criar sua própria semente, porém decidimos adotar uma semente global e desta forma simplificar o indivíduo e conseqüentemente o AMS. Entretanto, a *semente* deve ter suas faces rotuladas e isto pode privilegiar ou prejudicar indivíduos, dependendo de quais rótulos a semente possui e quais rótulos estão presentes nos elementos dos cromossomos. Este comportamento é indesejável e foi resolvido com a criação do rótulo especial *universal* $\in R$, para flexibilizar a rotulação da *semente*. Este rótulo constitui uma *ligação* com qualquer rótulo $r_1 \in R$, com força igual a $F(r_1)$. Assim, a *semente* é capaz de unir-se a outros elementos independente do rótulo, contanto que o rótulo possua força suficiente para vencer a *disruptura termal*. Vale destacar que, a *semente* aceita rótulos diferentes do rótulo *universal*, mas isto não é

aconselhável devido aos motivos descritos anteriormente.

A tabela de rótulos

Para calcular FA_D , são necessários os valores da função $F(r)$ (seção 2.2.1). Dessa forma, existe uma tabela global, denominada *tabela de rótulos*, que atribui um valor à força de atração de cada rótulo $r \in R$. Esta tabela é preenchida com valores da função $F : R \rightarrow \mathbb{N}$, tal que

$$F(ind(r)) = \begin{cases} 0, & \text{se } ind(r) \in \{0, 1, 2\} \\ ((ind(r) - 2) \% T) + 1, & ind(r) > 2, \end{cases} \quad (3.1)$$

onde $ind(r) : R \rightarrow \{0, 1, 2, 3, 4, 5, \dots, |tabela\ de\ rótulos| - 1\}$ e as três primeiras linhas desta tabela são reservadas aos rótulos especiais η , \emptyset e *universal*. Esta função garante a variabilidade de rótulos na tabela, produzindo periodicamente rótulos com força entre 1 e T . A tabela possui um tamanho pré-estabelecido nos experimentos (seção 4.1). Vale destacar que, por definição a temperatura é única para toda a população de indivíduos. Um exemplo de *tabela de rótulos* e da função ind correspondente são apresentados nas tabelas 3.1 e 3.2, respectivamente.

Tabela 3.1: Exemplo de uma *tabela de rótulos* com tamanho igual a 8 e temperatura igual a 3.

r	$ind(r)$	Força
η	0	0
\emptyset	1	0
<i>universal</i>	2	indeterminada
a	3	1
b	4	2
c	5	3
d	6	1
e	7	2

Tabela 3.2: Exemplo de função ind .

$ind(r)$	r
3	a
4	b
5	c
6	d
7	e

Simulação

Todo AG baseia-se na classificação dos indivíduos para promover a evolução dos mesmos. A classificação de cada indivíduo é obtida comparando sua pontuação obtida em testes ou funções de avaliação. Sendo assim, se o indivíduo representa um conjunto de tipos de elementos, então devemos simular a adição de elementos, originada deste conjunto, para o cálculo de várias características. Através destas características, obteremos a pontuação do indivíduo e sua classificação. Desta forma, construímos um simulador computacional para processo de Auto-montagem obedecendo às regras do modelo de Winfree, visto que não havia simulador computacional definido na literatura. Este simulador também possui as opções de simular o processo de Auto-montagem obedecendo às regras dos *modelos 2D** e *3D*.

A simulação da Auto-montagem de um indivíduo executa a adição dos tipos de elementos que compõem o seu cromossomo, obedecendo às regras de um dos modelos apresentados na seção 2.2. Após a simulação, são estabelecidas algumas propriedades deste indivíduo como: terminação, número de elementos que compõem a forma geométrica montada e suas dimensões. Para simular um indivíduo são necessárias as seguintes informações: a *tabela de rótulos*, a *semente*, o reticulado, a temperatura e o cromossomo deste indivíduo. Substituímos o reticulado infinito por um reticulado circular, onde a simulação acontece, devido às limitações de memória do computadores. Devido à característica circular, as extremidades da forma em construção podem colidir neste reticulado. Impedimos a colisão estabelecendo uma distância mínima de uma unidade, entre as extremidades da forma, evitando a adição do elemento que viole este limite. A figura 3.1 ilustra como evitar a colisão, utilizando a distância mínima. Nesta figura, os elementos foram adicionados ao longo da ordenada 1, em um reticulado circular. Os elementos foram adicionados seguindo a ordem crescente de suas denominações, após a semente S . O elemento 4 está unido ao elemento 3. Não é permitida a adição de elementos na posição $(4,1)$, pois viola a distância mínima.

O crescimento da forma não é limitado somente pelas dimensões do reticulado e pela distância mínima entre suas extremidades. Além disso, o tamanho da forma em constru-

ção é limitado pelo número máximo de elementos ($máximo_{elementos}$) que são adicionados ao reticulado. As dimensões do reticulado circular e o número máximo de elementos adicionados foram pré-estabelecidas experimentalmente (seção 4.1).

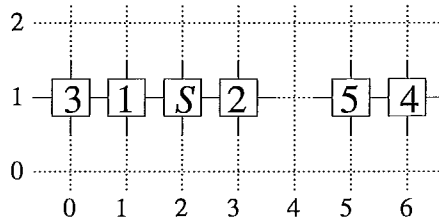


Figura 3.1: Distância mínima entre dois elementos das extremidades de uma forma no reticulado.

A simulação executa os seguintes passos:

1. Posicione a semente na origem do reticulado.
2. Inicialize a lista de posições vazias lpv do reticulado, vizinhas a *semente*.
3. Inicialize a lista de opções de montagem lom .
4. Para cada rotação rot de cada tipo de elemento τ do cromossomo faça:
 - (a) Para cada elemento pos de lpv faça:
 - i. Crie o elemento $t = (\tau, pos, rot)$, onde $t \notin lom$.
 - ii. Inclua o elemento t em lom se $FT_D^C(t) > T$.
5. Escolha um elemento de lom , através de um sorteio ponderado por $FT_D^C(t)$, de cada $t \in lom$.
6. Adicione o elemento sorteado ao reticulado.
7. Atualize a lista lpv , obedecendo às regras:
 - (a) Exclua a posição onde o elemento foi adicionado.
 - (b) Inclua as posições vazias $pos \notin lpv$, vizinhas ao elemento adicionado.
 - (c) Exclua as posições que não obedecem à distância mínima entre as extremidades da forma.

8. Atualize a lista *lom*, eliminando aqueles elementos cuja posição não está vazia.
9. Volte ao passo quatro, caso o reticulado tenha menos elementos que a quantidade $máximo_{elementos}$ e a lista de elementos não esteja vazia.

Como todo processo estocástico, a simulação não utiliza todas as possibilidades disponíveis e provavelmente alguns tipos de elementos do cromossomo não são utilizados. Desta forma, a simulação define outra propriedade importante do indivíduo: o *sítio ativo*. O *sítio ativo* contém todos os tipos de elementos localizados entre o primeiro e o último tipo de elementos utilizados na simulação. Sendo assim, o *sítio ativo* contém todos os tipos de elementos utilizados na simulação e possivelmente alguns tipos de elementos não utilizados na simulação. Os tipos de elementos localizados fora do *sítio ativo* não são necessariamente inúteis, mas são aqueles que não foram adicionados no reticulado durante a simulação e são os tipos de elementos com menor probabilidade de construir uma forma. O *sítio ativo* é a faixa com maior probabilidade de ocorrência dos BCs. Na figura 3.2 é representado o *sítio ativo* de um cromossomo, onde os tipos de elementos em cinza foram adicionados ao reticulado durante a simulação. Todos os tipos de elementos entre *i* e *f*, fazem parte do *sítio ativo*. O conjunto de tipos de elementos *L* é dado pelos tipos de elementos que estão representados em cinza na figura 3.2, retirando os tipos de elementos repetidos caso existam.

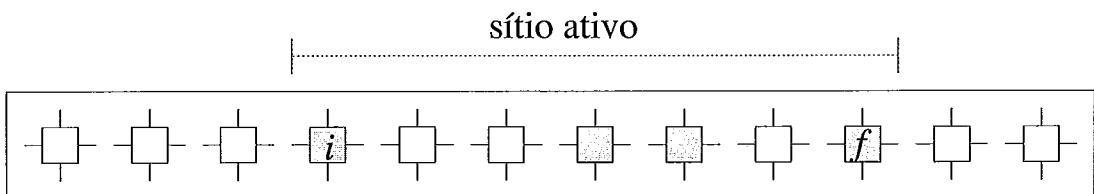


Figura 3.2: Sítio ativo do cromossomo.

Se a forma tenta crescer além dos limites do reticulado circular, talvez ela termine, porém isto é indeterminável. Então, ao final da simulação o cromossomo foi capaz de construir uma forma terminal se a lista de elementos a serem adicionados está vazia e nenhuma colisão foi evitada (passo oito da simulação). Note que, se uma colisão foi evitada, foi negado à forma crescer além dos limites do reticulado. Como a simulação

é um processo estocástico, existe a possibilidade da construção da forma geométrica não terminar. Isto pode ocorrer por dois motivos: cromossomo não possui os tipos e elementos corretos para a construção da forma ou cromossomo possui os tipos e elementos corretos, mas eles não foram selecionados. Para aumentar as chances do cromossomo, a simulação é executada até a criação de uma forma terminal ou até atingir um número máximo de tentativas pré-estabelecidas experimentalmente. Caso não seja construída nenhuma forma terminal, o cromossomo é associado à menor forma construída dentro das tentativas.

Alinhamento

Após a simulação do indivíduo, avaliamos se o mesmo atingiu a forma objetivo através do processo de alinhamento. Este processo procura o melhor posicionamento da forma menor, dentro dos limites da forma maior, procurando maximizar o número de elementos sobrepostos. A forma maior é colocada em uma posição fixa e a forma menor é deslocada utilizando rotações e movimentos na horizontal e vertical. A figura 3.3 ilustra este processo, onde as duas formas mostradas na figura 3.3-a são alinhadas. Na figura 3.3-b o alinhamento é executado em duas dimensões, deslocando a forma menor apenas na horizontal e na vertical. Por outro lado, a figura 3.4 mostra os movimentos de rotação utilizados no alinhamento das formas apresentadas na figura 3.3-a. Este mesmo processo é estendido em três dimensões, aumentando a liberdade com a qual a forma menor é movimentada, porém obedecendo os movimentos da figura 2.8-b.

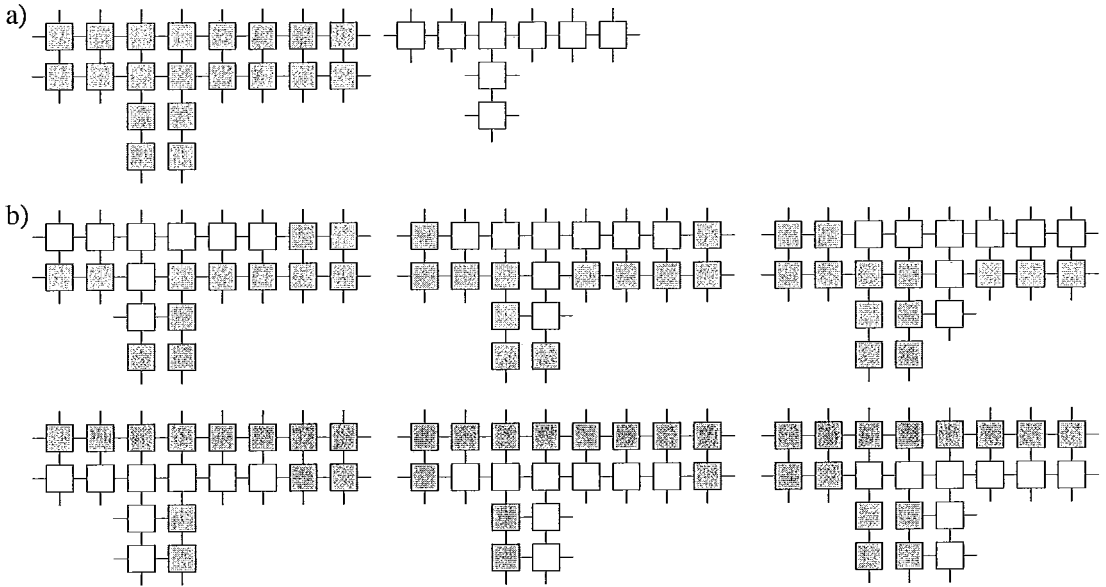


Figura 3.3: Alinhamento entre duas formas.

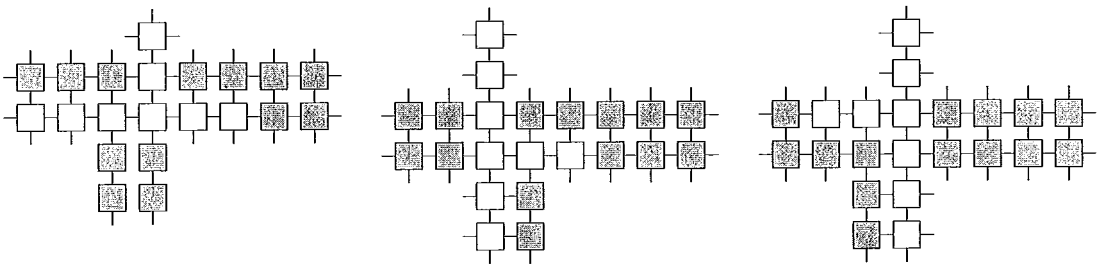


Figura 3.4: Posições relativas entre duas formas no alinhamento.

Substituição

Para testar se o conjunto de tipos de elementos L utilizados na simulação constrói sempre a mesma forma, devemos buscar todas as ordens de utilização dos elementos deste conjunto. Mas, esta busca demandaria $O(|L|^{N^2})$ passos em formas quadráticas (Adleman et al. 2002), então adotamos um teste aproximado, denominado teste de substituição, que comprova a unicidade sem enumerar todas as possibilidades, porém este teste não comprova a não unicidade. Isto ocorre porque existe a possibilidade de ocorrerem substituições, mas a construção terminar sempre na mesma forma geométrica, como exemplificado na figura 2.10. Para a execução deste teste, cada elemento adicionado na simulação é armazenado em uma fila f , quando a simulação termina, esta fila possui a ordem de montagem da forma construída.

O teste de substituição consiste nos seguintes passos:

1. Inicie a construção da forma novamente, adicionando a *semente*.
2. Atribua zero a ns .
3. Para $n = 1$ até o número de elementos de f faça:
 - (a) Tente adicionar o elemento $t \in L$, onde $t \neq f[n]$ e a representação textual de t não seja igual a de $f[n]$.
 - (b) Se t foi adicionado incremente ns de uma unidade.
 - (c) Retire t e adicione o elemento original $f[n]$.
4. Se ns for igual a 0, L constrói uma forma *única*.
5. Se não, se ns for maior que zero, L não constrói uma forma *única* e ns é o número de substituições de L .

3.1.2 Operadores genéticos

Um MEMO eficiente possui operadores genéticos, apropriadamente desenvolvidos e adaptados ao formato do cromossomo criado. Estes operadores manipulam as informações necessárias para evolução dos indivíduos ao longo das gerações. Dessa forma foram desenvolvidos operadores não usuais, pois o AMS possui um cromossomo não usual.

3.1.3 Recombinação

Existem vários tipos de operadores de recombinação ("crossover"), porém como os cromossomos não possuem tamanho fixo, uma recombinação simples não é o suficiente. Procurando promover maior troca dos BCs, escolhemos aleatoriamente uma posição de alinhamento entre um par de cromossomos escolhidos para recombinação. Assim a troca de BCs acontece independente de sua posição dentro do cromossomo. Este procedimento é normalmente adotado em AGs, quando os cromossomos possuem tamanho variável. A figura 3.5 mostra por que a recombinação simples não deve ser usada em cromossomos com tamanho variável. A figura 3.5-a apresenta uma recombinação simples. Note que o cromossomo maior (A) jamais troca seus últimos alelos com os últimos alelos do cromossomo menor (B), não importando a posição de corte α . Para evitar este problema é necessário alinhar o cromossomo B com o cromossomo A em uma posição sorteada, como mostrado na figura 3.5-b.

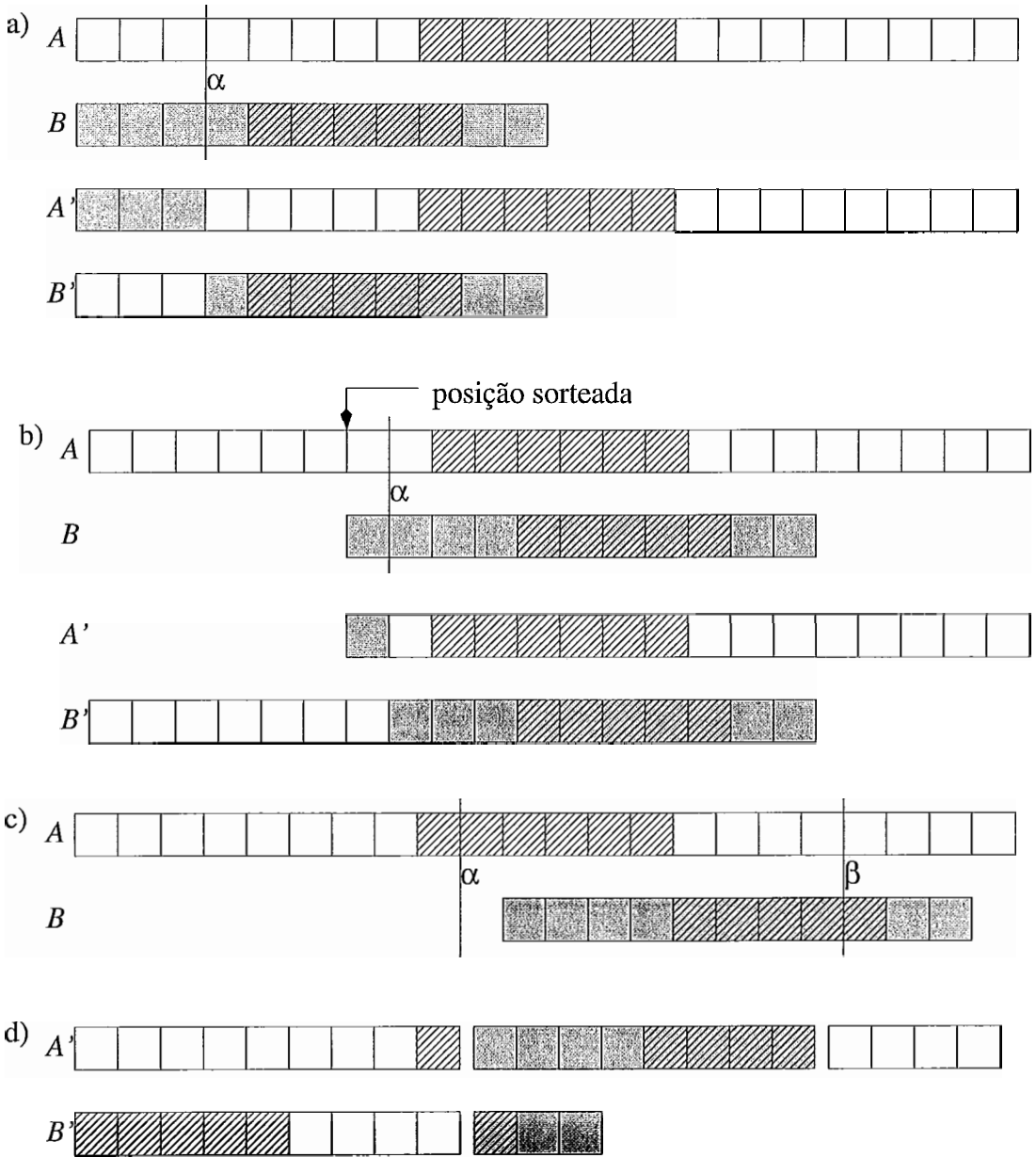


Figura 3.5: Possibilidades da recombinação em um cromossomo com *sítio ativo*

Entretanto, este tipo de recombinação ainda possui uma fraqueza. O cromossomo possui uma característica ignorada pela recombinação: o *sítio ativo*. Conseqüentemente este operador não promove obrigatoriamente a troca entre alelos que foram utilizados na simulação. Isto é observado na figura 3.5-b, onde os sítios ativos dos cromossomos A e B são representados pelos alelos hachurados; se qualquer ponto de corte α for escolhido fora destas áreas, a troca de BCs não ocorre. Para resolver este problema estabelecemos dois pontos de corte localizados obrigatoriamente dentro de cada *sítio ativo* dos cromossomos. Na figuras 3.5-c ilustra a escolha dos pontos de corte α e β . Estes pontos são escolhidos dentro dos sítios ativos dos cromossomos A e B , respectivamente e a figura 3.5-d apresenta o resultado da recombinação.

A figura 3.6 exemplifica o processo de recombinação adotado. Os sítios ativos dos cromossomos A e B estão localizados entre os alelos a_i e a_f , e, b_i e b_f . O ponto de alinhamento é escolhido aleatoriamente entre os alelos do maior cromossomo. Na figura 3.6-b, a posição do ponto de alinhamento a_j é escolhida, onde $j + m \leq n$. Os pontos de corte α , onde $i \leq \alpha \leq f$ e β , onde $i \leq \beta \leq f$ da figura 3.6-c são escolhidos aleatoriamente. Desta forma, este operador nunca criará um cromossomo maior que o maior cromossomo da população, assim como nunca criará um cromossomo menor que o menor cromossomo da população. Na figura 3.6-d são apresentados os dois filhos da recombinação.

3.1.4 Mutação

A recombinação promove a evolução da população gerando novos indivíduos, mas não gera alelos novos. A mutação foi desenvolvida para resolver este problema, aumentando a diversidade de alelos e a possibilidade de explorar todo o espaço de busca. O operador de mutação adotado escolhe aleatoriamente uma face de um tipo de elemento do cromossomo e troca seu rótulo por um dos rótulos da *tabela de rótulos*, exceto pelos rótulos η e *universal*. O novo rótulo escolhido é obrigatoriamente diferente do rótulo antigo.

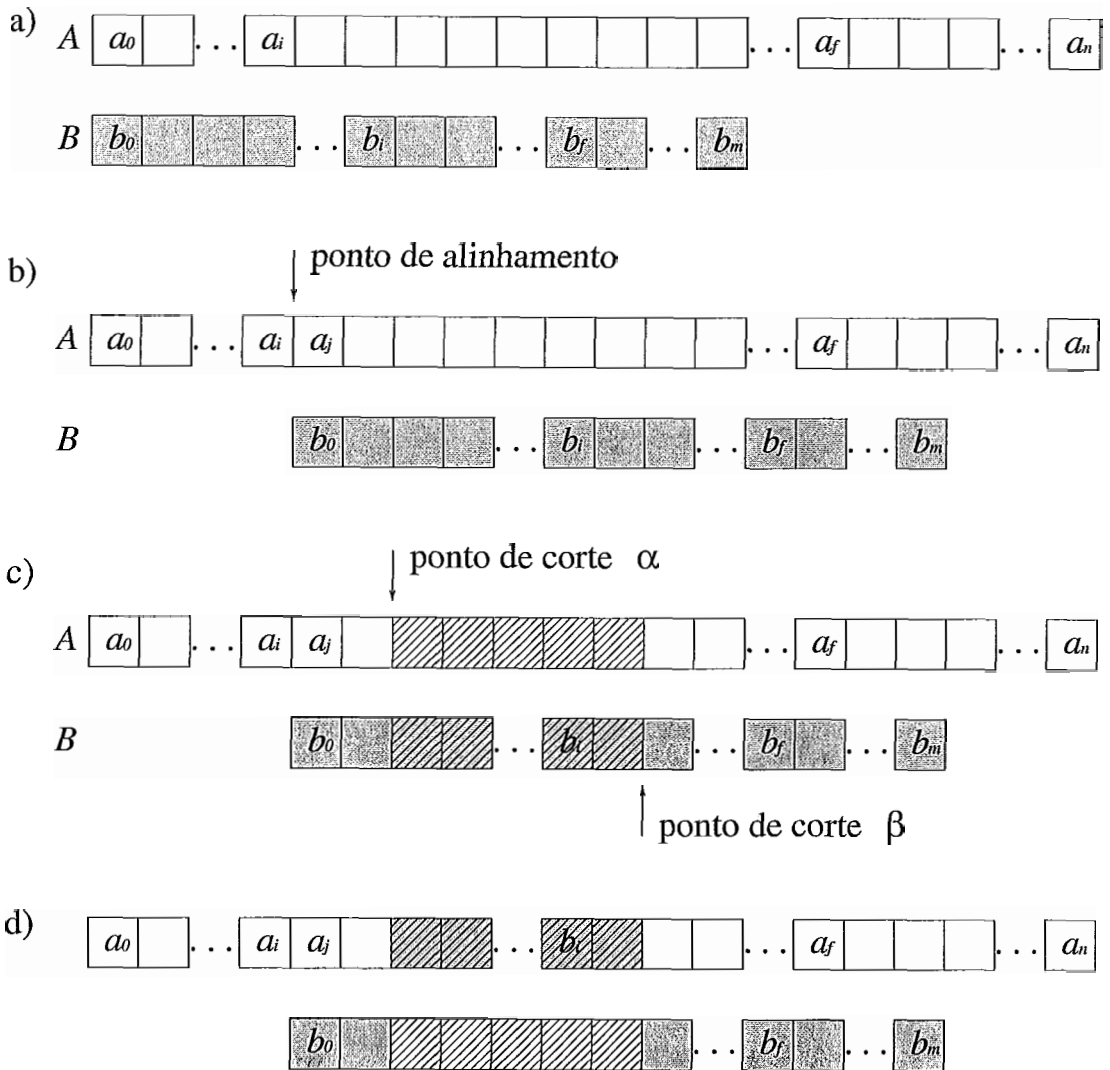


Figura 3.6: Exemplo de uma recombinação em um cromossomo com *sítio ativo*.

3.1.5 Classificação do indivíduo

Em um AG, cada indivíduo possui uma classificação, de acordo com a avaliação do seu desempenho. Esta avaliação mede o quanto o indivíduo se aproximou do objetivo. O PCML não é tradicionalmente tratado como um problema multi-objetivo. Entretanto, através de nossa abordagem identificamos vários objetivos a serem atingidos durante a minimização do conjunto de tipos de elementos. Geralmente existem duas forma de tratar problemas multi-objetivo: combinar os objetivos em um único objetivo ou obter as soluções não dominadas no espaço de Pareto (Coello 2000; Marler e Arora 2004). Combinar os objetivos é geralmente uma tarefa problemática, pois a natureza, grandeza e origem dos objetivos são incompatíveis, sendo impossível relacioná-los. Este é o caso do PCML.

As funções de avaliação do indivíduo

Inicialmente identificamos oito objetivos relacionados diretamente à minimização do conjunto de tipos de elementos. Ao eliminarmos os objetivos binários, como terminação e unicidade, e unirmos outros em um único objetivo, chegamos a três objetivos finais. Estes objetivos correspondem indiretamente às restrições do PCML, apresentadas na seção 2.3.1. A seguir apresentamos as funções de avaliação, que definem o desempenho do indivíduo em relação aos três objetivos. Estas funções foram normalizadas dentro de uma escala entre 0 (pior desempenho) e 1 (melhor desempenho).

- Fator de substituição - Esta função foi desenvolvida a partir do critério de unicidade (seção 2.3.1), pontuando melhor os indivíduos que mais se aproximam deste critério, através do teste de substituição, que determina se o indivíduo construiu uma forma única. Este teste é capaz de contar o número de substituições ns possíveis na forma construída. Note que, indivíduos que não conseguiram adicionar nenhum outro elemento ao reticulado, com excessão da *semente*, possuiriam $ns = 0$, pois o teste de substituição não substitui a semente. Nestes casos o teste de substituição retorna $ns = rots$ e dessa forma a função de avaliação fator de substituição, representada por $F1 : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R}$, atribui 0 a estes indivíduos, tal que

$$F1(ns, nf, ntf) = 1 - \frac{ns}{rots \cdot nf \cdot (ntf + 1)}, \quad (3.2)$$

mede quantitativamente a unicidade da forma, através do tamanho nf da forma construída em elementos e o número de tipos de elementos ntf utilizados na simulação do indivíduo, exceto a *semente*. A constante $rots$ representa o número de elementos obtidos pelas rotações e assume o valor 1 no *modelo 2D*, 4 no *modelo 2D** e 24 no *modelo 3D*. Para exemplificar o cálculo desta função, apresentamos a figura 3.7. Na figura 3.7-a apresenta a semente S e o conjunto de tipos de ladrilhos L utilizados na construção da forma representada na figura 3.7-b. Na figura 3.7-c foi feita a única substituição substituição possível, aplicando os passos apresentados na seção 3.1.1, onde o ladrilho 0 foi substituído pelo ladrilho 1. Desta forma, $ns = 1$, $nf = 4$, $ntf = 3$ e $F1(1, 4, 3) = 1 - \frac{1 + 1}{1 \cdot 4 \cdot (3 + 1)}$.

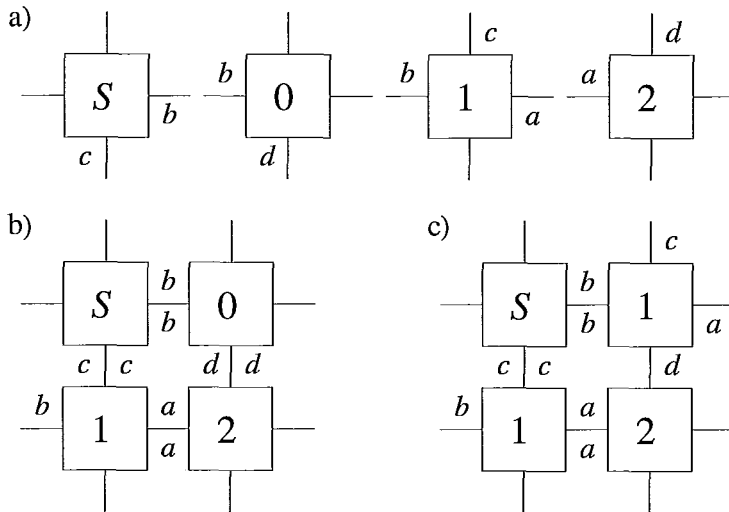


Figura 3.7: Exemplo do teste de substituição.

- Fator de sobreposição - Se um indivíduo atingiu a *forma objetivo*, ele possui a mesma quantidade de elementos nas mesmas posições da *forma objetivo* e terminou nela. Assim como existe a possibilidade do indivíduo terminar com menos elementos que a *forma objetivo*. Porém, se o indivíduo não termina, certamente possui mais elementos que a *forma objetivo*. Sendo assim, baseando-se nos critérios de terminação e atingir a *forma objetivo* (seção 2.3.1), a função fator de sobreposição $F2 : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R}$ pontua os indivíduos, tal que

$$F2(nm, no, nf) = 1 - \frac{(no - nm) + (nf - nm)}{no + nf} = \frac{2nm}{no + nf}, \quad (3.3)$$

onde nm é o número máximo de elementos sobrepostos, obtido através do alinhamento entre a *forma objetivo* e a forma construída na simulação pelo indivíduo, o tamanho no da *forma objetivo* em elementos além de nf especificado na função anterior, para medir a distância entre a forma construída e a *forma objetivo*. Como exemplo do cálculo desta função, observe o número máximo de sobreposições obtido na figura 2.9. Na figura 2.9-b obtemos o valor máximo de oito sobreposições entre a forma objetivo em cinza e a forma construída em branco. Neste caso $nm = 8$, $no = 20$, $nf = 8$ e $F2(8, 20, 8) = \frac{2 \cdot 8}{8 + 20}$.

- Fator do conjunto de tipos de elementos - A função $F3 : \mathbb{N} \times \mathbb{N} \times \mathbb{N} \rightarrow \mathbb{R}$, é derivada diretamente do objetivo principal do PCML: minimizar o conjunto de tipos de elementos, tal que

$$F3(nf, ntf) = 1 - \frac{ntf + 1}{nf}. \quad (3.4)$$

Um indivíduo é candidato a solução do PCML se obter $F1 = 1$ e $F2 = 1$. Para exemplificar o cálculo desta função, o cromossomo da figura 3.2 supostamente construiu a forma em branco da figura 2.9-a. Desta forma, $nf = 8$, $ntf = 4$ e $F3(8, 4) = 1 - \frac{4 + 1}{8}$.

A restrição para construção de formas completas 2.3.1 não possui uma função de avaliação correspondente. Isto deve-se a complexidade do teste que garante que esta restrição é atendida por candidatos a solução, pois todas as ordens de montagem possíveis devem terminar unicamente construindo formas completas. Entretanto, indivíduos que construíram formas completas são privilegiados pela simulação estocástica, pois quanto maior o número de ligações que um tipo de elemento deste indivíduo faz, maiores são as chances deste elemento ser adicionado ao reticulado durante a simulação.

Dominância no espaço de Pareto e classificação

A classificação do indivíduo depende da sua posição no espaço de Pareto em relação aos demais indivíduos. Um indivíduo i_1 obtém classificação melhor que outro i_2 , se i_1 domina i_2 . Esta relação de dominância entre i_1 e i_2 é estabelecida se $F1_{i_1} > F1_{i_2}$, $F2_{i_1} > F2_{i_2}$, $F3_{i_1} > F3_{i_2}$, onde $(F1_{i_1}, F2_{i_1}, F3_{i_1})$ e $(F1_{i_2}, F2_{i_2}, F3_{i_2})$ são as respectivas posições

de i_1 e i_2 no espaço de Pareto (Kung, Luccio e Preparata 1975). Estas coordenadas são obtidas através dos valores das funções de avaliação de cada indivíduo. Caso não seja estabelecida esta relação, então os desempenhos de i_1 e i_2 são incomparáveis. Em uma população, ao retirarmos todos os indivíduos não dominados, estaremos retirando também as soluções denominadas *Pareto ótimas*, representadas por aqueles indivíduos que não são dominados por nenhum outro indivíduo da população. As soluções *Pareto ótimas* formam a primeira camada de Pareto. Ao retirar todos os indivíduos desta camada, surge um novo conjunto de indivíduos não dominados, pertencentes à segunda camada de Pareto. As camadas se sucedem até a retirada de todos os indivíduos da população. Desta forma, a classificação da população é dada por cada camada (Coello 2000; Marler e Arora 2004).

Segundo os critérios apresentados na seção 2.3.1, para que um indivíduo i_1 seja considerado uma solução do PCML, i_1 deve terminar unicamente construindo a *forma objetivo*. Para isso, $F1_{i_1} = 1$ e $F2_{i_1} = 1$, independente do valor de $F3_{i_1}$. Desta forma, a relação de dominância adotada neste trabalho foi alterada da seguinte forma: um indivíduo i_1 domina outro indivíduo i_2 , se i_1 é estritamente melhor que i_2 em uma das funções $F1$ ou $F2$ e pelo menos equivalente na outra função. Os valores da função $F3$ só são comparados caso ocorra um empate entre os dois indivíduos ($F1_{i_1} = F1_{i_2}$ e $F2_{i_1} = F2_{i_2}$). Equivalentemente, um indivíduo i_1 domina outro indivíduo i_2 , se uma das seguintes expressões for atendida:

- $F1_{i_1} \geq F1_{i_2}$ e $F2_{i_1} > F2_{i_2}$
- $F1_{i_1} > F1_{i_2}$ e $F2_{i_1} \geq F2_{i_2}$
- $F1_{i_1} = F1_{i_2}$ e $F2_{i_1} = F2_{i_2}$ e $F3_{i_1} > F3_{i_2}$

Esta relação foi alterada para acelerar a evolução e não privilegiar os indivíduos com as seguintes características:

- Cromossomo pequeno - se $F3 > 0.9$, este indivíduo não possui quantidade suficiente de tipos de elementos para se tornar uma solução, porém dificilmente será dominado por qualquer outro indivíduo. Isto foi descoberto experimentalmente.

- Se $F1_{i_1} = F1_{i_2} = 1$ e i_1 possuir $F2_{i_1} > F2_{i_2}$, o PCML considera i_1 como melhor solução, por definição.
- Se $F2_{i_1} = F2_{i_2} = 1$ e i_1 possuir $F1_{i_1} > F1_{i_2}$, o PCML considera i_1 como melhor solução, por definição.
- Se $F1_{i_1} = F1_{i_2} = F2_{i_1} = F2_{i_2} = 1$ e i_1 possuir $F3_{i_1} > F3_{i_2}$, o PCML considera i_1 como melhor solução, por definição.

A figura 3.8 ilustra a alteração da dominância usual, onde podemos observar três situações, de acordo com as regras descritas nos itens acima. a e b são dois indivíduos da população. Nas figuras 3.8-a e 3.8-b, b domina a . Na figura 3.8-c b domina a , mesmo que $F3_a > F3_b$.

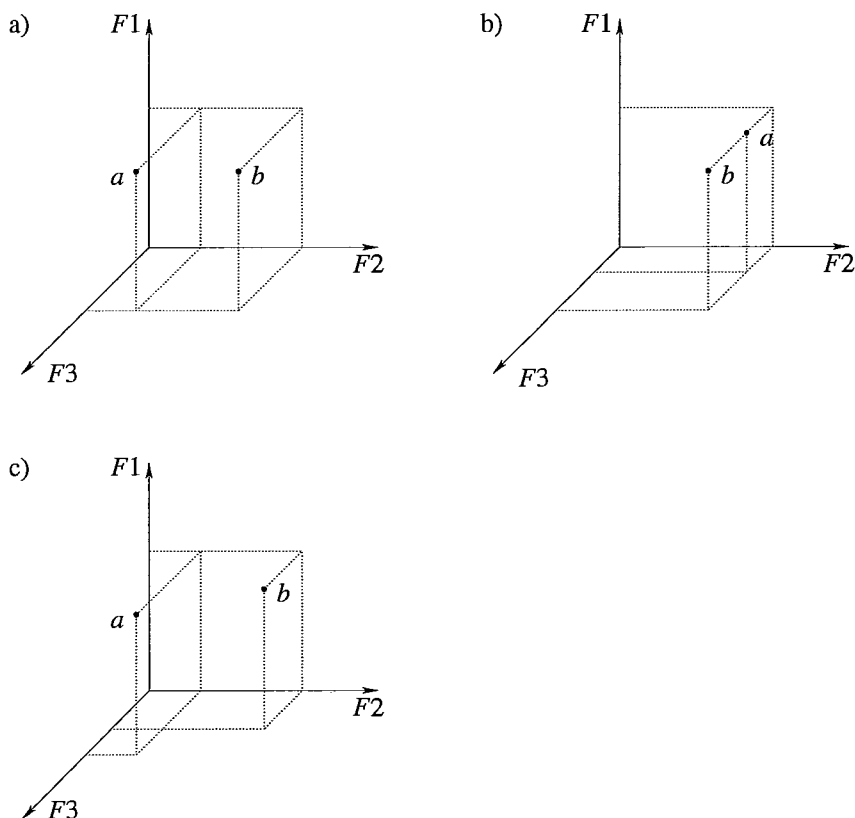


Figura 3.8: Dominância no espaço de Pareto para o PCML.

Através da classificação por camada no espaço de Pareto, obtemos o desempenho de cada indivíduo. Se os indivíduos pertencem à mesma camada, possuem desempenho igual. Entretanto, se dois indivíduos pertencem a camadas diferentes, torna-se necessário adotar algum método que determine a diferença de desempenho entre as camadas. Adotamos a ponderação linear ilustrada na figura 3.9, onde o eixo das abscissas representa as camadas de Pareto, o eixo das ordenadas representa os pesos das camadas e as retas i e m representam duas gerações. A ponderação linear de cada camada é dada pela altura até a reta i , onde o eixo do *peso* dá o valor do desempenho da *camada* e P_i é igual ao número de camadas na geração i , tal que

$$peso(camada) = L_i - \frac{L_i - 1}{P_i - 1}(camada - 1). \quad (3.5)$$

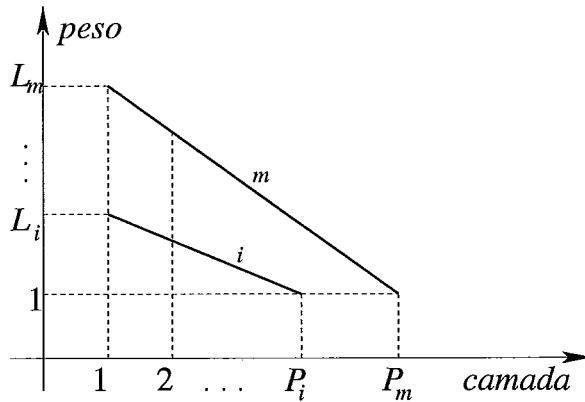


Figura 3.9: Ponderação das camadas de Pareto.

Desta forma, o desempenho de cada camada é dado pela altura da ordenada até atingir a reta i . Mas, o valor do *peso* L não é fixo ao longo das gerações do AMS. L é iniciado com valores próximos de 1, pois nas primeiras gerações, os indivíduos geralmente têm valores das funções de avaliação muito próximos e conseqüentemente a divisão em camadas não significa diferença de desempenho. Entretanto, conforme o passar das gerações, espera-se que a evolução aumente as diferenças nos valores das funções de avaliação, então L aumenta linearmente a cada geração, acentuando a diferença de desempenho entre as camadas. Este aumento linear foi definido experimentalmente (seção 4). Na figura 3.9, L_m é o valor final atingido por L na última geração m e P_m é a última camada da última geração m . O valor de L aumenta linearmente de L_i a L_m , da primeira geração a última

geração. A soma dos desempenho das camadas 1 a uma camada qualquer P' , sob a reta i , é dada por

$$\sum_{P=1}^{P'} peso(P) = \sum_{P=1}^{P'} \left(L_i - \frac{L_i - 1}{P_i - 1} (P - 1) \right) \quad (3.6)$$

$$= \sum_{P=1}^{P'} L_i - \left(\frac{L_i - 1}{P_i - 1} \right) \sum_{P=1}^{P'} P + \left(\frac{L_i - 1}{P_i - 1} \right) \sum_{P=1}^{P'} 1 \quad (3.7)$$

$$= P' L_i - \left(\frac{L_i - 1}{P_i - 1} \right) \left(\frac{P'(P' + 1)}{2} \right) + \left(\frac{L_i - 1}{P_i - 1} \right) P' \quad (3.8)$$

$$= P' L_i + \left(\frac{L_i - 1}{P_i - 1} \right) \left(\frac{P'(1 - P')}{2} \right). \quad (3.9)$$

$$(3.10)$$

Quando $P' = P_i$ sob a reta i ,

$$\sum_{P=1}^{P_i} peso = \frac{P_i(L_i + 1)}{2}. \quad (3.11)$$

3.1.6 Composição das gerações

O cromossomo de cada indivíduo da primeira população é criado aleatoriamente, sorteando os seguintes valores:

- O tamanho do cromossomo obedecendo a faixa pré-estabelecida.
- Um rótulo para cada face dos elementos do cromossomo. O rótulo deve pertencer à *tabela de rótulos* e não é permitida a escolha dos rótulos η e *universal* neste sorteio.

As gerações seguintes são compostas por:

- Elitismo - $taxa_{elite}$ % de indivíduos da camada *Pareto ótima* da geração anterior.
- Diversidade - $taxa_{aleatório}$ % de indivíduos sorteados aleatoriamente da geração anterior, exceto os selecionados pelo elitismo.
- Complemento - $(100 - taxa_{aleatório} - taxa_{elite})$ % de novos indivíduos (filhos), gerados pela aplicação dos operadores genéticos, escolhendo os pais da seguinte forma:

- Escolha da camada P' , através do sorteio aleatório de um q' , entre os valores 0 e $\frac{P_i(L_i + 1)}{2}$, onde P_i é o número de camadas da geração anterior, L_i é *peso* da primeira camada da geração anterior, e P' é o menor número inteiro que satisfaz a inequação

$$q' \leq \sum_{P=1}^{P'} \left(L_i - \left(\frac{L_i - 1}{P_i - 1} \right) (P - 1) \right) \quad (3.12)$$

- Sorteio aleatório dos indivíduos dentro da camada sorteada P' .

Vale destacar que, para determinar P' sem calcular o somatório no sorteio aleatório de q' , basta calcular a maior raiz da equação $P' L_i + \left(\frac{L_i - 1}{P_i - 1} \right) \left(\frac{P'(1 - P')}{2} \right) = q'$ ou, equivalentemente, $(P')^2(1 - L_i) - P'(1 - L_i(2P_i - 1)) - 2q'(P_i - 1) = 0$.

Aplicação dos operadores genéticos

A aplicação dos operadores genéticos adotada neste trabalho baseia-se na estratégia descrita em (Koza 1992). Nesta estratégia escolhe-se o operador de recombinação com probabilidade $taxa_{recombinação}$, caso faltem dois ou mais indivíduos para completar a nova geração ou escolhe-se o operador de mutação. Se a recombinação for escolhida, sorteiam-se dois cromossomos (pais), cuja distância euclidiana no espaço de Pareto seja maior que o valor $distância_{recombinação}$. Após a execução de um número de tentativas ($tentativas_{recombinação}$) de atingir esta distância, sorteiam-se dois cromossomos quaisquer. Esta distância evita que dois cromossomos semelhantes sejam recombinados e conseqüentemente aumenta a diversidade dos filhos. O valor $taxa_{recombinação}$ não é fixo, este valor cresce ao longo das gerações linearmente, pois no PCML precisamos aumentar a mutação nas primeiras gerações de forma a intensificar a diversidade de indivíduos e aumentar a exploração do espaço de busca. Porém, conforme a população evolui, a mutação pode fazer o processo divergir. Entretanto, como o valor $taxa_{recombinação}$ aumenta, as chances da mutação diminuem e este problema é evitado. O aumento linear da probabilidade de recombinação foi definido experimentalmente (seção 4).

3.1.7 O Algoritmo

O pseudo-código do AMS, desenvolvido neste trabalho, é apresentado no algoritmo 1, onde o número de gerações é o critério de parada do algoritmo e T é a temperatura. O tamanho da população e o número de gerações foram determinados experimentalmente.

Algoritmo 1 Algoritmo Genético Multi-Objetivo.

Inicialização:

$Tabela_{rótulos} = PreencheTabela(T, |Tabela_{rótulos}|)$
 $População = Randômico(|População|, tamanhoMínimo_{indivíduo},$
 $tamanhoMáximo_{indivíduo}, |Tabela_{rótulos}|)$

$L = L_{inicial}$
 $taxa_{recombinação} = recombinação_{inicial}$

Recursão:

Para $g = 0$ até $geração_{final}$

Para $i = 0$ até $|População|$

$Indivíduo_{simulacao}^{g,i} = Simula(T, Indivíduo^{g,i}, dimensões_{reticulado},$
 $máximo_{elementos}, tentativas_{simulação})$

$Indivíduo_{alinhamento}^{g,i} = Alinha(Indivíduo^{g,i}, FormaObjetivo)$
 $Indivíduo_{substituicao}^{g,i} = Substitue(T, Indivíduo^{g,i})$
 $Pareto_g = CalculaFunçõesAvaliação(Indivíduo^{g,i})$

Fim do Para

$Pareto_g = Classificação(Pareto_g)$
 $População_{g+1} = Elite(Pareto_g, taxa_{elite})$
 $População_{g+1} = Aleatório(Pareto_g, taxa_{aleatório})$

Enquanto $|População_{g+1}| < |População|$

Se $SorteioAleatório() > taxa_{recombinação}$ ou $|População| - |População_{g+1}| = 1$

$População_{g+1} = Mutação(SorteioPonderado(L, Pareto_g))$

Sé Não

Para $t = tentativas_{recombinação}$ até 0

$indivíduo_a = SorteioPonderado(L, Pareto_g)$
 $indivíduo_b = SorteioPonderado(L, Pareto_g)$

Se $DistânciaEuclidiana(indivíduo_a, indivíduo_b) \geq distância_{recombinação}$ ou $t = 0$

$População_{g+1} = Recombinação(indivíduo_a, indivíduo_b)$

Fim do Se

Fim do Para

Fim do Se

Fim do Enquanto

$taxa_{recombinação} = taxa_{recombinação} + \left(\frac{recombinação_{final} - recombinação_{inicial}}{geração_{final}} \right)$

$L = L + \left(\frac{L_{final} - L_{inicial}}{geração_{final}} \right)$

Fim do Para

Descrição do algoritmo:

- *PreencheTabela(temperatura, quantidade de linhas)* - utiliza a função 3.1 para preencher a *tabela de rótulos*.
- *Randômico(quantidade de indivíduos, tamanho mínimo do cromossomo, tamanho máximo do cromossomo, quantidade de linhas da tabela de rótulos)* - Preenche a população com indivíduos cujos cromossomos são gerados aleatoriamente, sorteadando o tamanho do cromossomo e os rótulos dos lados dos tipos de ladrilhos do cromossomo.
- *Simula(temperatura, indivíduo, dimensões do reticulado, máximo de elementos do reticulado, máximo de tentativas de simulação)* - Executa a simulação descrita na seção 3.1.1.
- *Alinha(indivíduo, forma objetivo)* - Alinha a forma construída pelo *indivíduo* com a *forma objetivo*, segundo os passos apresentados na seção 3.1.1.
- *Substitue(temperatura, indivíduo)* - Executa o teste de substituição, de acordo com os passos apresentados na seção 3.1.1.
- *CalculaFunçõesAvaliação(indivíduo)* - Calcula as funções de avaliação, apresentadas na seção 3.1.5.
- *Elite(camadas de Pareto, percentual)* - Preenche a nova geração com o *percentual* dos melhores indivíduos da geração anterior, segundo o posicionamento destes indivíduos nas *camadas de Pareto*.
- *Aleatório(camadas de Pareto, percentual)* - Preenche a nova geração com o *percentual* de indivíduos escolhidos aleatoriamente da geração anterior, exceto aqueles indivíduos já selecionados pelo elitismo.
- *SorteioAleatório()* - Retorna aleatoriamente uma fração entre 0 e 1.
- *SorteioPonderado(L, camadas de Pareto)* - Escolhe um indivíduo da geração anterior, segundo o sorteio ponderado descrito na seção 3.1.5.

- *Mutação(indivíduo)* - Aplica o operador de mutação, descrito na seção 3.1.2.
- *DistânciaEuclidiana(ponto a, ponto b)* - Calcula a distância Euclidiana entre *a* e *b*.
- *Recombinação(indivíduo a, indivíduo b)* - Aplica o operador de recombinação aos *indivíduo a* e *indivíduo b*, como descrito na seção 3.1.2.

3.2 Espaço de busca

O espaço de busca do AMS depende do tamanho do cromossomo e da variedade de tipos de ladrilhos que compõem o próprio cromossomo. A variedade de tipos de ladrilhos depende da quantidade de rótulos disponíveis na *tabela de rótulos*. No *modelo 2D*, se a tabela possui $|R|$ rótulos, o espaço de busca é $|cromossomo||R|^4$.

Como *modelo 2D** possui rotação, o cálculo do espaço de busca deve considerar estas rotações como repetições do mesmo tipo de ladrilho. Para efetuar este cálculo, é necessário considerar todas as configurações de tipos de ladrilhos de acordo com a quantidade de rótulos distintos, o número de repetições destes rótulos e o posicionamento destes nas faces, como mostrado pela figura 3.10. Na figura 3.10-a apresentamos os tipos de ladrilhos distintos com zero ou quatro rótulos diferentes. Observe que, no ladrilho 0 desta figura, não é possível produzir um tipo de ladrilho diferente mudando os rótulos de face. Porém, ao mudarmos os rótulos de face, no ladrilho 1 da figura 3.10-a, produziremos um tipo de ladrilho diferente, desde que esta mudança não represente uma rotação. Na figura 3.10-b, os tipos de ladrilhos possuem dois rótulos distintos em quantidades iguais e a figura 3.10-c em quantidades desiguais. A figura 3.10-d ilustra as possibilidades de tipos de ladrilhos com três rótulos distintos. A tabela 3.3 mostra o número de possibilidades diferentes de produzir tipos de ladrilhos, de acordo com a quantidades de rótulos distintos e seus posicionamentos nas faces, como mostrado na figura 3.10. Nesta tabela $A_{n,k} = \frac{n!}{(n-k)!}$ corresponde a um arranjo simples, $A_n^k = n^k$ corresponde a um arranjo com repetição e $C_{n,k} = \frac{n!}{k!(n-k)!}$ corresponde a uma combinação simples.

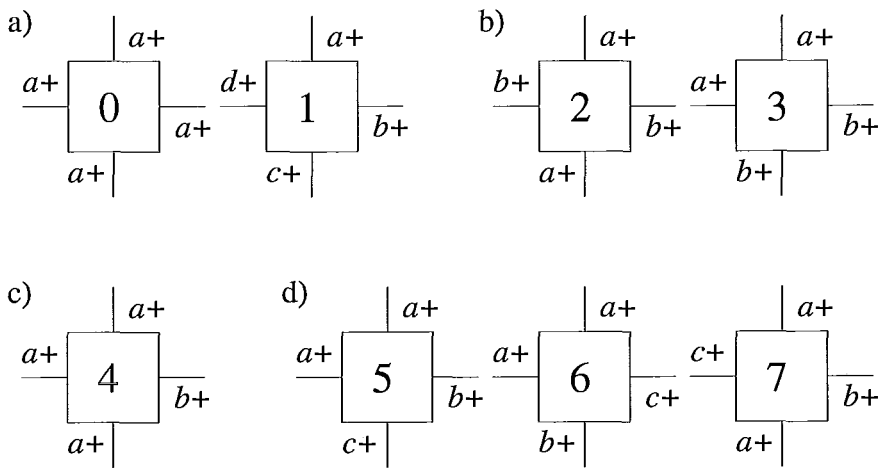


Figura 3.10: Posicionamento e repetição dos rótulos no *modelo 2D**.

Tabela 3.3: Variabilidade dos tipos de ladrilhos no *modelo 2D**.

Figura	Ladrilho	Possibilidades	Tipo
3.10-a	0	$ R $	$A_{ R ,1}$
3.10-a	1	$\frac{ R (R -1)(R -2)(R -3)}{4}$	$\frac{A_{ R ,4}}{rots2D^*}$
3.10-b	2	$\frac{ R (R -1)}{2}$	$C_{ R ,2}$
3.10-b	3	$\frac{ R (R -1)}{2}$	$C_{ R ,2}$
3.10-c	4	$ R (R -1)$	$A_{ R ,2}$
3.10-d	5	$\frac{ R (R -1)(R -2)}{2}$	$ R (C_{ R -1,2})$
3.10-d	6	$\frac{ R (R -1)(R -2)}{2}$	$ R (C_{ R -1,2})$
3.10-d	7	$\frac{ R (R -1)(R -2)}{2}$	$ R (C_{ R -1,2})$

Note que o espaço de busca é calculado somando todos os valores da coluna Possibilidades da tabela 3.3 e multiplicando pelo tamanho do cromossomo e multiplicando também por 2, pois o *modelo 2D** possui 2 valores distintos de polaridade. O espaço de busca é dado pela expressão

$$\begin{aligned} & 2|\text{cromossomo}|(A_{|R|,1} + A_{|R|,4} + 2C_{|R|,2} + A_{|R|,2} + 3|R|(C_{|R|-1,2})) = \\ & = |\text{cromossomo}||R|(2 + (|R| - 1))\{4 + (|R| - 2)[3 + \frac{|R| - 3}{2}]\}. \end{aligned} \quad (3.13)$$

Este modelo possui uma vantagem clara sobre o *modelo 2D*, pois seu espaço de busca é menor.

No *modelo 3D* o cálculo do espaço de busca depende do grafo de vizinhança do bloco. A variabilidade de blocos está relacionada com os tipos de vizinhos diferentes, de acordo com os rótulos das faces. As faces do bloco na figura 3.11-a possuem o mesmo rótulo, desta forma não existe outra associação entre rótulos e faces diferente da apresentada, pois toda face e sua vizinha possuem o mesmo rótulo. No bloco da figura 3.11-b também não existe outra associação que produza um bloco diferente. Entretanto, o caso do bloco representado na figura 3.11-c é diferente. O reposicionamento dos rótulos deste bloco produz um arranjo simples de diferentes blocos, porém cada bloco possui 24 rotações ou repetições que estão incluídas neste arranjo simples. Desta forma, o número real de blocos distintos é dado pelo arranjo simples dividido por vinte e quatro. A figura 3.12 apresenta dois blocos com as mesmas quantidades de rótulos distintos. Estes blocos são diferentes devido à diferença das vizinhanças. O primeiro bloco (figura 3.12-a) possui no máximo dois vizinhos com o mesmo rótulo, mas o segundo bloco (figura 3.12-b) possui três vizinhos com o mesmo rótulo. A tabela 3.4 relaciona o número de blocos distintos com as figuras 3.11 a 3.19.

O espaço de busca no *modelo 3D* é dado pela expressão

$$\begin{aligned} & |\text{cromossomo}||R|(1 + (|R| - 1)) \cdot \\ & \cdot \left\{ \frac{5}{2} + (|R| - 2) \left[\frac{5}{3} + (|R| - 3) \left(\frac{5}{12} + (|R| - 4) \left(\frac{5}{4} + \frac{|R| - 5}{24} \right) \right) \right] \right\}. \end{aligned} \quad (3.14)$$

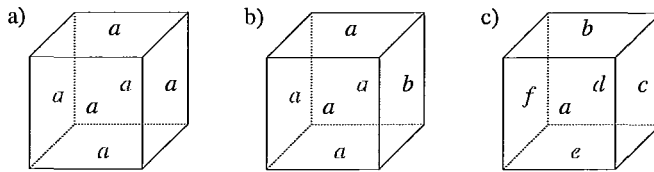


Figura 3.11: Tipos de blocos onde a vizinhança é singular.

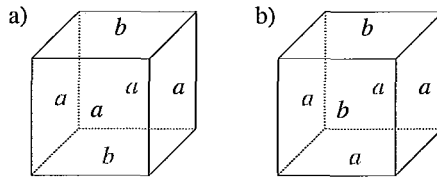


Figura 3.12: Tipos de blocos com dois rótulos distintos em quantidades desiguais.

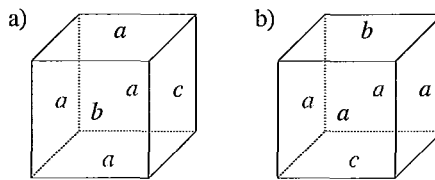


Figura 3.13: Tipos de blocos com três rótulos distintos, onde três faces vizinhas possuem o mesmo rótulo.

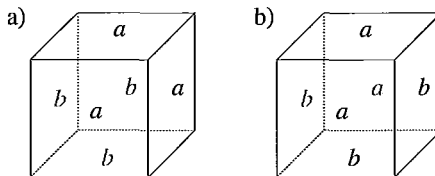


Figura 3.14: Tipos de blocos com três rótulos distintos em quantidades iguais.

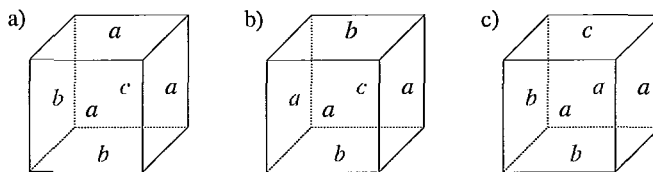


Figura 3.15: Tipos de blocos com três rótulos distintos, onde uma face possui vizinhas com a mesma quantidade de rótulos iguais.

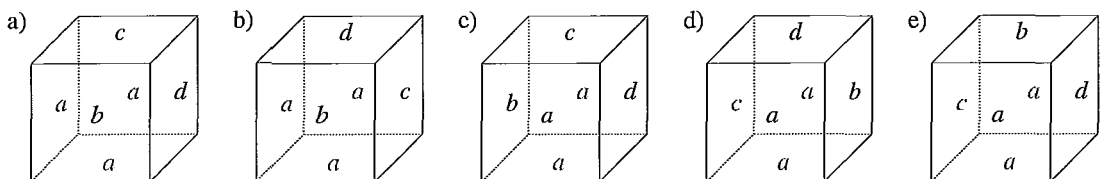


Figura 3.16: Tipos de blocos com quatro rótulos distintos, onde existem pelo menos duas faces vizinhas com o mesmo rótulo.

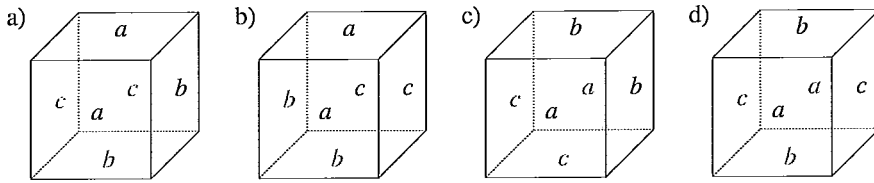


Figura 3.17: Tipos de blocos com três rótulos distintos em quantidades iguais.

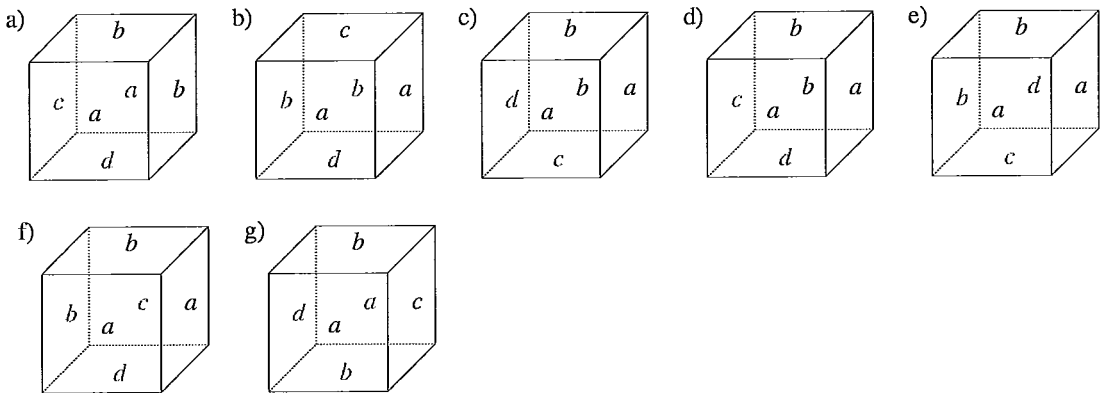


Figura 3.18: Tipos de blocos com quatro rótulos distintos, onde no máximo duas faces possuem o mesmo rótulo.

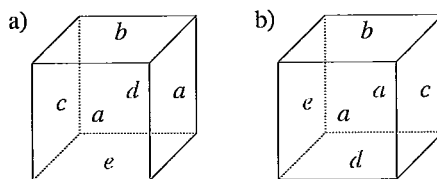


Figura 3.19: Tipos de blocos com cinco rótulos distintos.

Tabela 3.4: Diferentes vizinhanças dos rótulos e as repetições dos tipos de blocos.

Figura	Possibilidades	Tipo
3.11-a	$ R $	$A_{ R ,1}$
3.11-b	$ R (R - 1)$	$A_{ R ,2}$
3.11-c	$\frac{ R (R - 1) \dots (R - 5)}{24}$	$\frac{A_{ R ,6}}{rots3D}$
3.12-a, 3.12-b	$ R (R - 1)$	$A_{ R ,2}$
3.13-a, 3.13-b	$\frac{ R (R - 1)(R - 2)}{2}$	$ R (C_{ R -1,2})$
3.14-a, 3.14-b	$\frac{ R (R - 1)}{2}$	$C_{ R ,2}$
3.15-a, 3.15-b, 3.15-c	$ R (R - 1)(R - 2)$	$A_{ R ,3}$
3.16-a a 3.16-e	$\frac{ R (R - 1)(R - 2)(R - 3)}{6}$	$ R (C_{ R -1,3})$
3.17-a a 3.17-d	$\frac{ R (R - 1)(R - 2)}{6}$	$C_{ R ,3}$
3.18-a a 3.18-g	$\frac{ R (R - 1)(R - 2)(R - 3)}{4}$	$(C_{ R ,2})(C_{ R -2,2})$
3.19-a	$ R (R - 1)(R - 2)(R - 3)(R - 4)$	$ R (A_{ R -1,4})$
3.19-b	$\frac{ R (R - 1)(R - 2)(R - 3)(R - 4)}{4}$	$\frac{ R (A_{ R -1,5})}{rots2D*}$

CAPÍTULO 4

Resultados computacionais

Os resultados obtidos pelos experimentos comprovam o bom desempenho do algoritmo AMS. As soluções para o *modelo 2D* descritas na seção 2.3.2 e apresentadas originalmente em (Rothmund e Winfree 2000), foram atingidas pelos experimentos com o AMS. As soluções para o problema do conjunto mínimo de rótulos e para os novos *modelos 2D** e *3D* foram obtidas originalmente através dos experimentos deste trabalho. Note que, nenhum outro trabalho propôs soluções para o problema do conjunto mínimo de rótulos até o momento. O desempenho do AMS, com relação ao PCML, não foi comparado com nenhum outro algoritmo, pois trata-se de uma abordagem original deste problema.

4.1 Experimentos

O primeiro objetivo dos experimentos deste trabalho foi testar se o algoritmo AMS alcançaria as soluções do PCML, apresentadas em (Rothmund e Winfree 2000) no *modelo 2D*, pois são as melhores soluções encontradas até o momento, quando as *formas objetivo* são quadrados. Escolhemos *formas objetivo* quadráticas, pois são as formas mais estudadas e as únicas com solução demonstrada em estudos (Adleman et al. 2002). O segundo objetivo dos experimentos foi encontrar a melhor configuração do AMS para uma das instâncias mais simples do PCML: *formas objetivo* quadráticas, onde $N = 5$. O terceiro objetivo dos experimentos foi aplicar as melhores configurações do AMS, encontradas anteriormente, na solução do PCML para a extensão do *modelo 2D* e desta forma encontrar soluções viáveis ($F1 = 1$ e $F2 = 1$) para o *modelos 2D** na construção de quadrados.

Assim, foi possível comparar as soluções dos dois modelos bidimensionais. Os últimos experimentos tinham como objetivo encontrar soluções viáveis para a extensão em três dimensões de um quadrado: o cubo. Note que nenhuma solução apresentada para o PCML, quando as *formas objetivo* são quadrados, foi provada analiticamente como ótima.

Como a semente é global no AMS, definimos a semente de cada experimento como um dos vértices da *forma objetivo*, pois as sementes de todas as melhores soluções apresentadas em estudos anteriores são vértices das *formas objetivo*. Pelo mesmo motivo escolhemos a temperatura igual a 2. As sementes dos experimentos possuíam rótulos *universais* onde era necessário, para tornarem-se realmente vértices das formas geométricas em construção, com por exemplo, a semente da figura 2.19-c que possuía rótulos *universais* nos lados 2 e 3. A figura 4.1 apresenta as sementes utilizadas nos experimentos deste trabalho, onde o rótulo \bar{u} representa o rótulo *universal*. A figura 4.1-a ilustra a semente dos experimentos bidimensionais, exceto o experimento com quadrados 25×25 , cuja semente é apresentada na figura 4.1-c. A figura 4.1-b apresenta a semente utilizada nos experimentos tridimensionais. Vale destacar que nos experimentos, quando a *tabela de rótulos* possui x rótulos distintos com a temperatura igual a T , isto significa que a *tabela de rótulos* possui $x \cdot T$ rótulos. A tabela 4.1 exemplifica uma *tabela de rótulos*, onde a temperatura é igual a 2 e possui 5 rótulos distintos.

Tabela 4.1: Exemplo de uma *tabela de rótulos* utilizada nos experimentos.

r	$ind(r)$	Força
η	0	0
\emptyset	1	0
<i>universal</i>	2	indeterminada
a_1	3	1
a_2	4	2
b_1	5	1
b_2	6	2
c_1	7	1
c_2	8	2
d_1	9	1
d_2	10	2
e_1	11	1
e_2	12	2

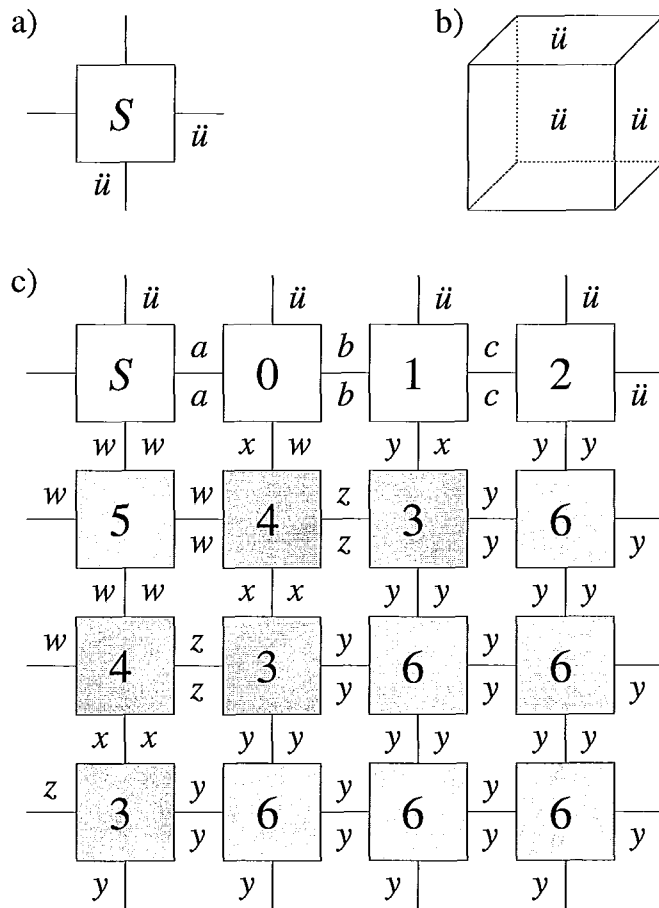


Figura 4.1: Sementes *universais* dos experimentos.

Escolhemos o tamanho da população, o número de gerações, o número de tentativas de simulação ($tentativas_{simulação}$) e o número de tentativas de aplicar o operador de recombinação ($tentativas_{recombinação}$) de acordo com a infra-estrutura disponível, ou seja, a quantidade de memória requerida para a simulação do indivíduo e a janela de tempo livre para execução do AMS. O número máximo de ladrilhos e as dimensões do reticulado devem proporcionar total liberdade de crescimento da forma geométrica, porém na simulação real a quantidade de memória finita limita esta liberdade. Desta forma, adotamos a seguinte estratégia:

- O número máximo de ladrilhos é igual ao número de ladrilhos da *forma objetivo* multiplicado por quatro.
- Cada dimensão do reticulado é igual a três vezes a maior dimensão da *forma objetivo*.

4.1.1 Experimentos no modelo bidimensional

Inicialmente o *modelo 2D* foi escolhido para a determinação dos parâmetros do AMS, por ser o modelo mais simples e com soluções conhecidas (seção 2.3.2). Foram realizados 19 experimentos para definição da melhor configuração do AMS no *modelo 2D*, de acordo com as seguintes características:

- A *forma objetivo* escolhida foi um quadrado 5×5 .
- Temperatura igual a 2 (T).
- População com 1000 indivíduos ($|População|$).
- Máximo de 100 ladrilhos no reticulado ($máximo_{ladrilhos}$).
- 10% de elitismo ($taxa_{elite}$).
- 5% de diversidade ($taxa_{elite}$).
- 1000 gerações ($geração_{final}$).
- Reticulado com as dimensões 30×30 ($dimensões_{reticulado}$).

O percentual de elitismo e diversidade basearam-se em valores iniciais, comumente utilizados em MEMOs (Back, Hoffmeister e Schwefel 1991). A faixa de variação do tamanho do cromossomo foi baseada na solução previamente conhecida (seção 2.3.2), que requer $N + 4$ tipos de ladrilhos. Assim o tamanho mínimo do cromossomo é nove, mas como a semente não está contida no cromossomo, este número foi reduzido para oito. O tamanho máximo foi obtido através da pior solução, representada por um tipo de ladrilho diferente para cada posição da forma 5×5 . Entretanto, como alguns ladrilhos não são utilizados na simulação (seção 3.1.1), o cromossomo diminui suas chances de construir uma forma geométrica, conforme diminui seu tamanho. Para evitar este problema, adotamos duas faixas de tamanho para aumentar a variedade de tipos de ladrilhos e proporcionar mais chances ao cromossomo. A quantidade de rótulos distintos também foi obtida da solução para a forma 5×5 , e requer oito rótulos distintos. Para observar a influência da variabilidade da quantidade de rótulos, adotamos dois valores diferentes: 10 e 20. A faixa de variação de L e da probabilidade de recombinação foram definidas arbitrariamente e mantidas devido ao sucesso dos experimentos. A tabela 4.2 apresenta os resultados dos experimentos de determinação dos parâmetros do AMS, onde é possível constatar a influência do tamanho do cromossomo no sucesso do AMS. Nos experimentos cujos cromossomos variaram de 8 a 24 ladrilhos a solução não foi encontrada. Nesta tabela e nas demais tabelas, L e a probabilidade de recombinação variam linearmente do valor inicial ao valor final apresentados, a cada geração.

Nos experimentos 17,18 e 19 procuramos descobrir o comportamento do AMS quando a probabilidade de recombinação não varia. O gráfico da figura 4.2 apresenta o valor das funções $F1$, $F2$ e $F3$ do melhor indivíduo a cada geração, segundo o maior valor da função $F1$, para o experimento 17. Caso ocorra um empate entre dois indivíduos, o melhor é aquele com maior valor da função $F2$ e caso o empate permaneça, o melhor indivíduo é escolhido pelo valor obtido na função $F3$. Devido à alta ocorrência de mutação os indivíduos não evoluíram, pois a probabilidade de recombinação foi fixada em 30%, e conseqüentemente a solução não foi encontrada. Sendo assim, as funções $F1$ e $F2$ não atingiram o valor 1, neste gráfico.

Tabela 4.2: Experimentos de determinação dos parâmetros do AMS, para a construção de um quadrado 5×5 no *modelo 2D*.

Experimento	Tamanho do cromossomo	Rótulos distintos	L	Probabilidade de recombinação	Distância mínima para recomb.	Solução encontrada na geração
1	8 a 24	10	1 a 30	30% a 70%	0	-
2	8 a 24	10	15	30% a 70%	0	-
3	25 a 50	10	1 a 30	30% a 70%	0	345
4	25 a 50	10	15	30% a 70%	0	173
5	8 a 24	20	1 a 30	30% a 70%	0	-
6	8 a 24	20	15	30% a 70%	0	-
7	25 a 50	20	1 a 30	30% a 70%	0	-
8	25 a 50	20	15	30% a 70%	0	-
9	8 a 24	10	1 a 30	30% a 70%	0.01	-
10	8 a 24	10	15	30% a 70%	0.01	-
11	25 a 50	10	1 a 30	30% a 70%	0.01	435
12	25 a 50	10	15	30% a 70%	0.01	148
13	8 a 24	20	1 a 30	30% a 70%	0.01	-
14	8 a 24	20	15	30% a 70%	0.01	-
15	25 a 50	20	1 a 30	30% a 70%	0.01	682
16	25 a 50	20	15	30% a 70%	0.01	142
17	25 a 50	10	3	30%	0	-
18	25 a 50	10	3	50%	0	-
19	25 a 50	10	3	70%	0	-

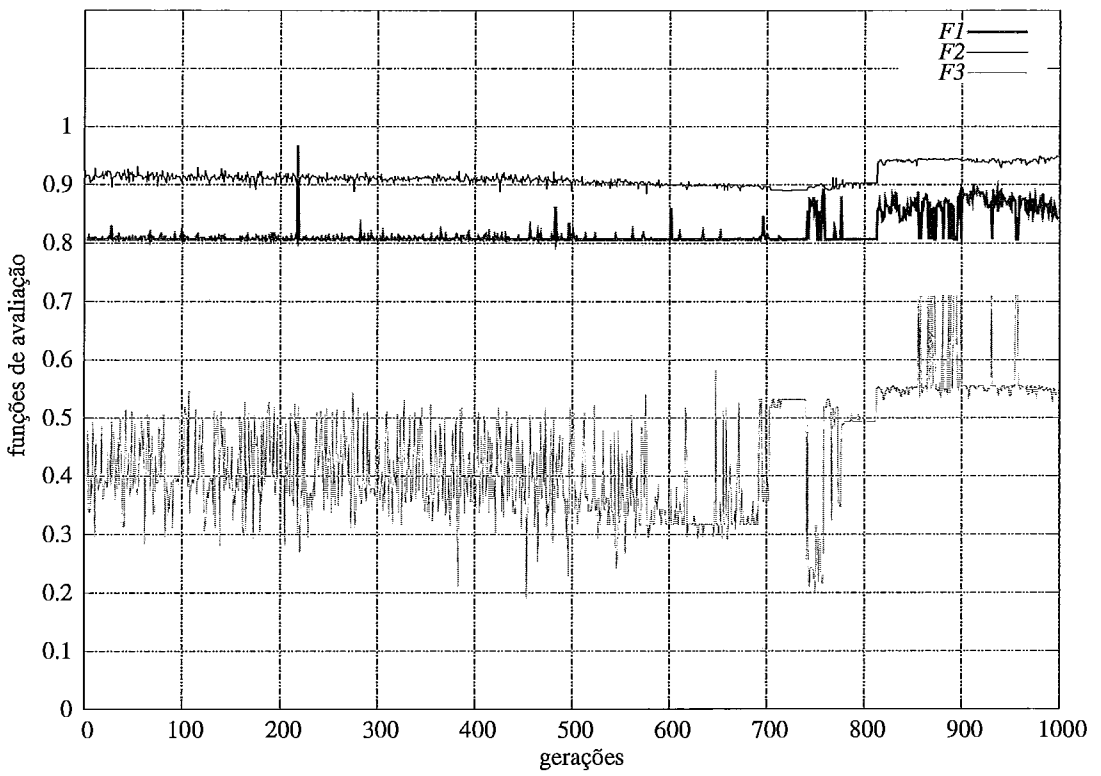


Figura 4.2: Funções de avaliação do melhor indivíduo em cada geração no experimento 17.

Para verificar a real influência do tamanho do cromossomo, selecionamos os experimentos com o tamanho do cromossomo variando entre 25 e 50, e realizamos novos experimentos com as mesmas configurações. Para aumentar a confiabilidade dos resultados, executamos cinco repetições de cada um destes experimentos, totalizando 40 experimentos. A tabela 4.3 apresenta os resultados destes experimentos, onde a coluna Geração representa em quais gerações a solução foi encontrada, cada coluna rodada representa uma repetição do experimento e a coluna Média apresenta o valor médio das gerações em seis rodadas. Note que, incluímos o resultado do experimento de determinação dos parâmetros do AMS na coluna rodada 0.

Os resultados apresentados na tabela 4.3 demonstram que a influência do tamanho do cromossomo não é determinante para que a solução seja encontrada, pois os experimentos 7 e 8 não encontraram a solução novamente. Entretanto, podemos observar que os experimentos cujo número de rótulos distintos é 20, obtiveram um desempenho menor que os demais. Vale destacar que, nesta nova série de experimentos, o experimento 3 encontrou a solução para o conjunto mínimo de rótulos, apresentada na figura 2.19.

Tabela 4.3: Experimentos para construção de um quadrado 5×5 no *modelo2D*.

Experimento	Geração						Média
	rodada 0	rodada 1	rodada 2	rodada 3	rodada 4	rodada 5	
3	345	346	931	527	152	715	502.7
4	173	167	550	434	755	521	433.3
7	-	-	-	-	-	-	-
8	-	-	-	-	-	-	-
11	435	434	753	-	-	-	541
12	148	149	791	-	-	-	462.7
15	682	633	-	-	-	-	657.5
16	147	405	-	-	-	-	276

A partir destes experimentos, observamos que o AMS não teve dificuldades para encontrar a solução nos experimentos 3 e 4, quando a *forma objetivo* é um quadrado 5×5 . O gráfico da figura 4.3 comprova esta facilidade, pois as funções de avaliação $F1$ e $F2$ iniciam com valores acima de 0.6 e ultrapassam 0.8 em menos de 200 gerações. Este gráfico descreve a evolução das funções de avaliação do indivíduo i_1 , onde i_1 é melhor indivíduo já visto na geração g se não existe outro indivíduo i_2 de uma geração menor

ou igual a g que domine i_1 , segundo a pontuação obtida por estes indivíduos na função de avaliação $F2$. Se ocorrer um empate entre i_1 e i_2 ($F2(i_1) = F2(i_2)$), então a pontuação obtida na função de avaliação $F1$ é utilizada para o desempate. Se mesmo assim for mantido empate entre i_1 e i_2 ($F1(i_1) = F1(i_2)$ e $F2(i_1) = F2(i_2)$), então a pontuação obtida na função de avaliação $F3$ é utilizada. Note que, o valor da função $F2$ é sempre crescente. Vale destacar que, quando o valor da função $F1$ decresce o valor de $F2$ cresce na mesma geração. Isto é devido a um novo indivíduo i_1 , cujo valor de $F2$ era melhor que o indivíduo da geração anterior i_2 , porém $F1(i_1) < F1(i_2)$. Também podemos observar que o valor da função $F3$ oscila por que $F3$ é usada somente como último recurso de desempate, além de $F1$ e $F2$ serem restrições do PCML enquanto $F3$ não. Este gráfico corresponde à execução com o pior desempenho do experimento 3, onde a solução foi encontrada no maior número de gerações.

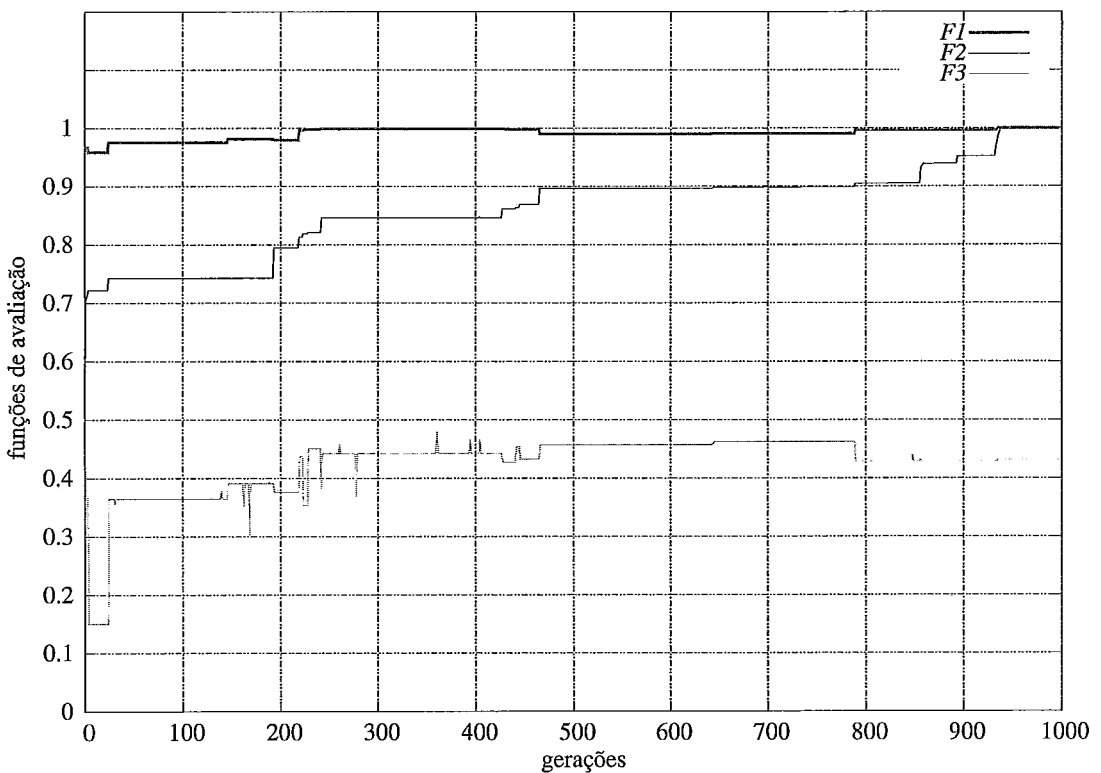


Figura 4.3: Funções de avaliação do melhor indivíduo já visto ao longo das gerações no experimento 3.

Para observar o comportamento do AMS conforme a dificuldade do PCML aumenta, escolhendo um quadrado 15×15 como *forma objetivo* e executamos cinco repetições de

quatro configurações diferentes do AMS. Estas configurações baseiam-se nas configurações dos experimentos 3, 4, 11 e 12, pois foram os experimentos mais bem sucedidos das execuções anteriores, porém com as seguintes diferenças:

- A *forma objetivo* é um quadrado 15×15 .
- Máximo de 900 ladrilhos no reticulado (*máximo₀ladrilhos*).
- Reticulado com as dimensões 45×45 (*dimensões₀reticulado*).

A tabela 4.4 apresenta os resultados destes 20 experimentos, onde a coluna Número de soluções encontradas mostra o número de experimentos que encontraram a solução para a mesma configuração. O gráfico da figura 4.4 comprova o aumento da dificuldade nestes experimentos, pois as funções de avaliação evoluem mais lentamente que no gráfico da figura 4.3, onde o melhor indivíduo já visto no experimento 22, foi escolhido com as mesmas regras do gráfico da figura 4.3, conforme descrito anteriormente. Neste gráfico observamos que os valores das funções não evoluem com a mesma velocidade do experimento 3, pois o espaço de busca é maior no experimento 22. Vale destacar que, as chances dos indivíduos atenderem ao critério de unicidade neste experimento são menores que as chances no experimento 3, pois tamanho da *forma objetivo* do experimento 22 é maior que no experimento 3 e isto aumenta as chances de ocorrerem substituições. Este fato é confirmado ao compararmos os gráficos das figuras 4.3 e 4.4, onde $F1$ (unicidade) atingiu o valor 1 depois de $F2$. Isto não aconteceu no experimento 3.

Tabela 4.4: Experimentos para construção de um quadrado 15×15 no *modelo2D*.

Experimento	Tamanho do cromossomo	Rótulos distintos	L	Probabilidade de recombinação	Distância mínima para recombin.	Número de soluções
20	50 a 100	20	1 a 30	30% a 70%	0	2
21	50 a 100	20	15	30% a 70%	0	2
22	50 a 100	20	1 a 30	30% a 70%	0.01	5
23	50 a 100	20	15	30% a 70%	0.01	5

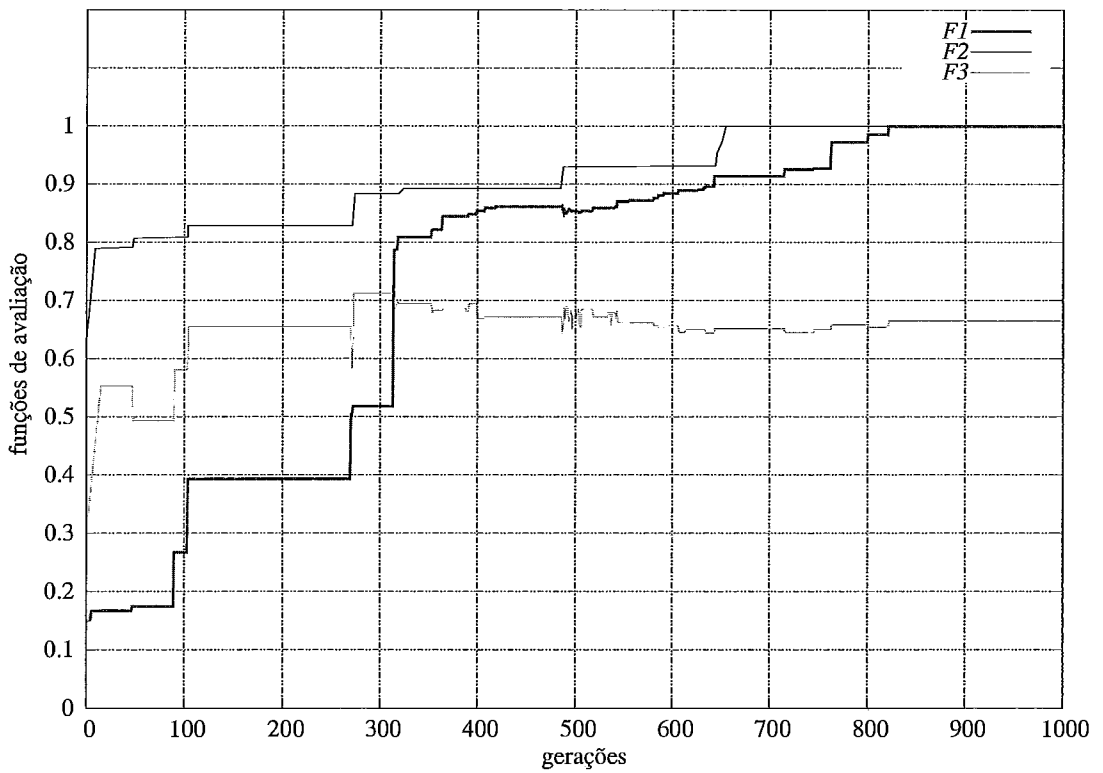


Figura 4.4: Funções de avaliação do melhor indivíduo já visto ao longo das gerações no experimento 22.

Com o objetivo de verificar a eficácia do AMS, escolhemos a *forma objetivo* correspondente à próxima faixa de solução para o *modelo 2D*: um quadrado 25×25 . Os experimentos não encontraram nenhuma solução para esta *forma objetivo*, porém ao adotarmos uma *semente complexa* (seção 2.2) similar a da etapa um da figura 2.14, o AMS obteve sucesso e encontrou a solução em dois dos quatro experimentos realizados. A utilização da *semente complexa* baseia-se no estudo apresentado em (Rothemund e Winfree 2000), onde a solução também origina-se de uma *semente complexa*.

O resultado deste experimento é apresentado na tabela 4.5, onde a seguinte configuração foi adotada:

- A *forma objetivo* é um quadrado 25×25 .
- Máximo de 2500 ladrilhos no reticulado (*máximo_{ladrilhos}*).
- Reticulado com as dimensões 100×100 (*dimensões_{reticulado}*).

Comparando os gráficos das figuras 4.3, 4.4 e 4.5, podemos notar o aumento do espaço de busca, a diminuição na velocidade da evolução da população e conseqüentemente o aumento da dificuldade do PCML, conforme aumentam as dimensões da *forma objetivo*. Vale destacar que, em todos os experimentos, a função F^3 sempre atinge valores maiores nas primeiras gerações que diminuem nas últimas gerações, pois conjuntos de tipos de ladrilhos menores não possuem a quantidade mínima necessária para tornarem-se soluções.

Tabela 4.5: Experimentos para construção de um quadrado 25×25 no *modelo 2D*.

Experimento	Tamanho do cromossomo	Rótulos distintos	L	Probabilidade de recombinação	Distância mínima para recomb. para recomb.	Solução encontrada na geração
24	100 a 150	20	1 a 30	30% a 70%	0.01	-
25	100 a 150	20	15	30% a 70%	0.01	-
26	100 a 150	20	1 a 30	30%	0.01	845
27	100 a 150	20	15	30%	0.01	773

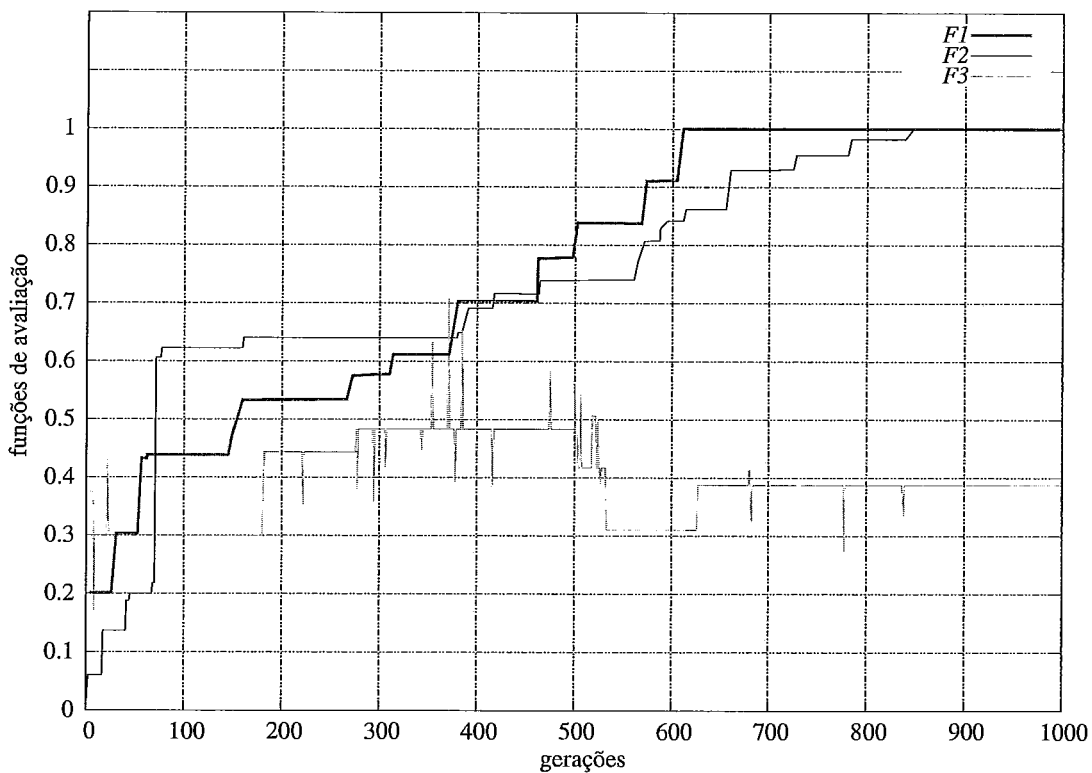


Figura 4.5: Funções de avaliação do melhor indivíduo já visto ao longo das gerações no experimento 26.

4.1.2 Experimentos no modelo bidimensional com rotação

Após a comprovação da eficácia do AMS para o *modelo 2D*, executamos cinco experimentos no *modelo 2D**, onde a configuração é idêntica a do experimento 3, pois foi o experimento executado com melhor desempenho. Como o espaço de busca do *modelo 2D** é menor que o espaço de busca do *modelo 2D*, o AMS não teve problemas para encontrar as soluções nos experimentos. Isto também é confirmado pelo gráfico da figura 4.6, onde a evolução é quase instantânea, se comparada a evolução do gráfico da figura 4.3 que corresponde a um experimento no *modelo 2D*. A tabela 4.6 mostra que todos os experimentos encontraram a solução, onde a coluna Geração representa em qual geração a solução foi encontrada. As soluções das figuras 2.16 e 2.17 foram encontradas nestes experimentos.

Tabela 4.6: Experimentos para construção de um quadrado 5×5 no *modelo2D**.

Experimento	Geração
28	853
29	942
30	763
31	327
32	877

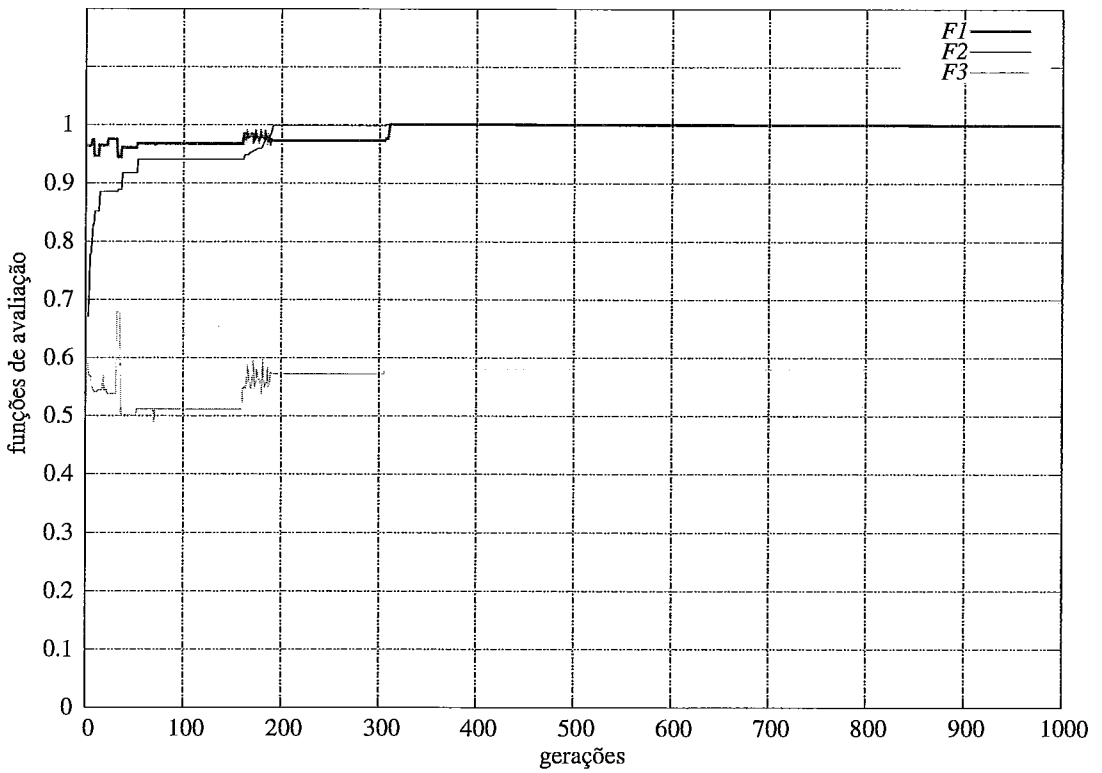


Figura 4.6: Funções de avaliação do melhor indivíduo já visto ao longo das gerações no experimento 31.

4.1.3 Experimentos no modelo tridimensional

Para os experimentos no *modelo 3D* escolhemos um cubo $5 \times 5 \times 5$ e executamos cinco repetições da configuração apresentada a seguir:

- Temperatura igual a 2.
- Máximo de 900 blocos no reticulado (*máximo_{ladrilhos}*).
- Reticulado com as dimensões $30 \times 30 \times 30$ (*dimensões_{reticulado}*).
- Tamanho do cromossomo entre 25 e 35.
- Vinte rótulos distintos.
- L variando de 1 a 30.
- Probabilidade de recombinação variando de 30% a 70%.
- Distância mínima para recombinação igual a 0.01.

As soluções encontradas pelo AMS com as configurações descritas, são apresentadas na tabela 4.7, onde todos os experimentos encontraram a solução. Devido à este fato não aumentamos a temperatura para três, pois a temperatura dois foi suficiente para alcançar as soluções. Nestes experimentos, era esperado o pior desempenho do AMS, pois o espaço de busca no *modelo 3D* é $O(N^2)$ vezes maior que os modelos bidimensionais. Entretanto, a possibilidade de rotação parece anular o aparente aumento de dificuldade do PCML. Isto é constatado comparando a velocidade de evolução do valor das funções de avaliação dos indivíduos no gráfico da figura 4.7 com todos gráficos das funções de avaliação apresentados anteriormente.

Tabela 4.7: Experimentos para construção de um cubo $5 \times 5 \times 5$.

Experimento	Geração
33	933
34	977
35	908
36	889
37	899

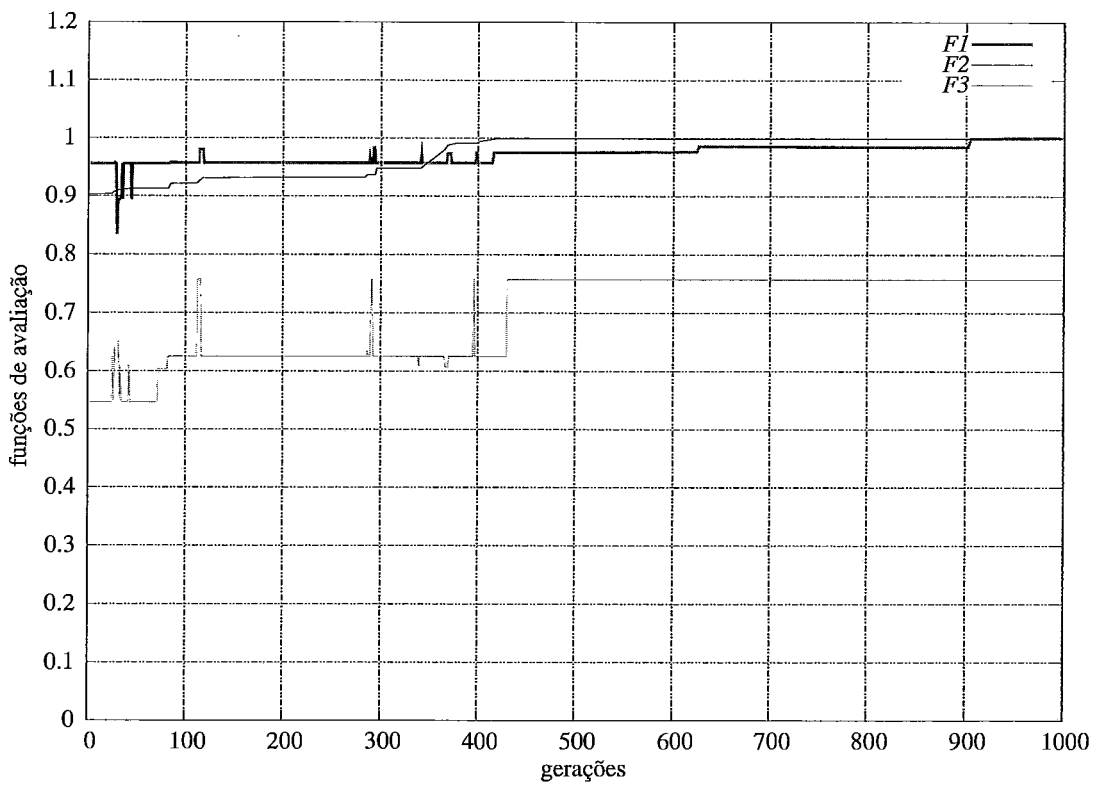


Figura 4.7: Funções de avaliação do melhor indivíduo já visto ao longo das gerações no experimento 35.

Apesar da solução encontrada nestes experimentos e apresenta na seção 2.3.2 ser uma forma completa, existem outras ordens de montagem utilizando os tipos de cubos desta solução que não terminam construindo formas completas. Desta forma, executamos mais um experimento com a mesma configuração descrita nesta seção, mas aumentando a temperatura para 3. Neste experimento foi encontrada uma solução que termina construindo formas completas independente da ordem de montagem. A figura 4.8 apresenta esta solução, onde o rótulo y possui força igual a 2, o rótulo z possui força igual a 1 e os demais rótulos possuem força igual a 3. A semente e os tipos de ladrilhos utilizados no processo de construção da forma (figura 4.8-b) são ilustrados na figura 4.8-a.

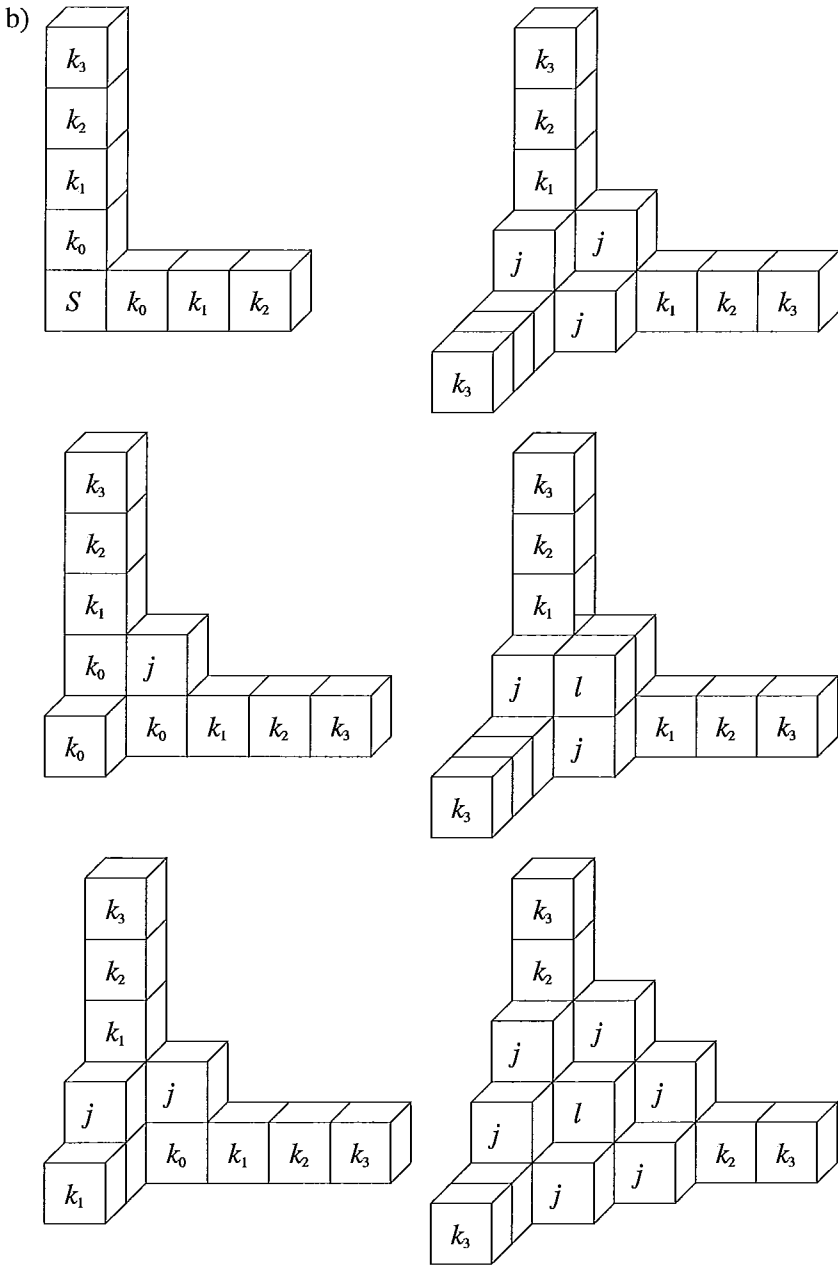
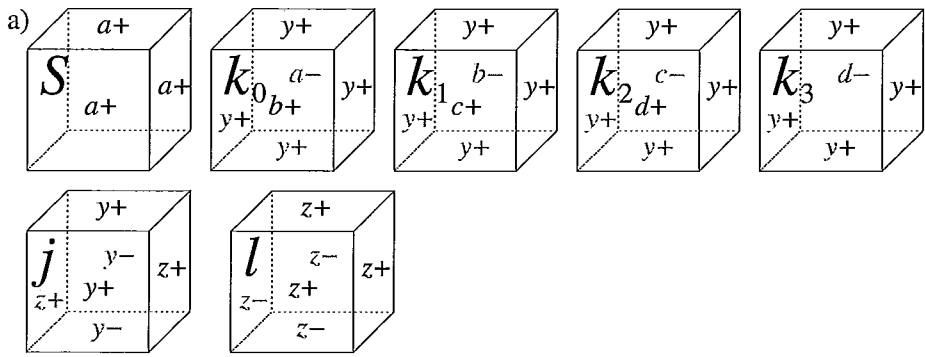


Figura 4.8: Solução para um cubo $5 \times 5 \times 5$ com temperatura igual a 3.

4.2 Paralelização do algoritmo

O AMS foi paralelizado através de um modelo mestre-escravo, onde o escravo executou os procedimentos *Simula*, *Alinha* e *Substitue*, apresentados no algoritmo 1, e o mestre executou todos os demais procedimentos. Apresentamos os tempos de execução, de acordo com os experimentos e CPUs disponíveis, na tabela 4.8. As CPUs são Intels Core 2 Dual 1.6GHz com 2GB de memória RAM e o Scientific Linux 4 smp é o sistema operacional.

Tabela 4.8: Tempo de execução dos experimentos.

Experimentos	CPUs	Tempo médio de execução
1 a 19	8	4 horas
20 a 23	10	7 horas e 40 minutos
24 a 27	10	10 horas
28 a 32	12	1 hora e 30 minutos
33 a 37	12	12 horas

CAPÍTULO 5

Conclusão

A construção de formas geométricas através do processo de Auto-montagem é vital para determinadas aplicações, como visto na seção 2.1. Nossos experimentos comprovaram que a Heurística Evolucionária Multi-Objetivo AMS, implementada neste trabalho obteve sucesso em todos os modelos: o *modelo 2D* proposto por (Winfrey 1998), a extensão do *modelo 2D*, denominada *2D**, e a extensão em três dimensões do *modelo 2D**, denominada *3D*, ambas proposta neste trabalho. Através destes experimentos e das soluções encontradas, podemos concluir que:

- Tratar o problema do conjunto mínimo de ladrilhos como um problema multi-objetivo é uma estratégia de sucesso, pois encontra soluções melhores que estudos analíticos, como mostrado na seção 2.3.2.
- O aumento do tamanho do cromossomo aumenta as chances de sucesso do AMS, pois cromossomos maiores aumentam a variedade de tipos de ladrilhos e conseqüentemente aumentam as chances do cromossomo montar a *forma objetivo*. Isto apenas provoca um aumento linear do espaço de busca, como visto na seção 3.2.
- O aumento do número de rótulos distintos piora o desempenho do AMS, pois o aumento do espaço de busca não é linear, como visto na seção 3.2.
- O *modelo 2D** é capaz de construir as mesmas formas geométricas que o *modelo 2D*, mas possui espaço de busca menor.

- O *modelo 3D* constrói formas geométricas em três dimensões, mas com o mesmo número de tipos de ladrilhos dos modelos bidimensionais, ou seja aumentamos em uma dimensão sem aumentar o número de tipos de ladrilhos necessários.

5.1 Contribuição

Este trabalho estudou o problema do conjunto mínimo de ladrilhos e a construção de formas geométricas utilizando o processo de Auto-montagem, do ponto de vista computacional. A seguir listamos nossas principais contribuições:

- Este estudo propôs a primeira abordagem computacional para resolver o problema do conjunto mínimo de ladrilhos. Além disso, desenvolvemos um algoritmo original, baseado em Meta-Heurísticas Evolucionárias Multi-Objetivo para solucionar o problema.
- O *modelo 2D** foi criado com objetivo de facilitar a construção de formas geométricas em duas dimensões. Este modelo requer $N + 1$ tipos de ladrilhos distintos para construir quadrados 5×5 , contra $N + 4$ tipos de ladrilhos distintos no *modelo 2D* proposto por Winfree (Rothmund e Winfree 2000), como visto na seção 2.3.2.
- O *modelo 3D* foi proposto para viabilizar a construção de formas geométricas em três dimensões. Embora este modelo construa formas geométricas em três dimensões, requer apenas $N + 1$ tipos de blocos. Esta é a mesma quantidade de tipos de ladrilhos do *modelo 2D**, que constrói somente formas geométricas bidimensionais.
- Os critérios de terminação e unicidade apresentados por Adleman em (Adleman et al. 2002), definem que um *Sistema de Ladrilhos SL* é *terminal* e *único*, se *SL* termina sempre construindo uma determinada *configuração C*. Os critérios de terminação e unicidade, foram estendidos para se adequar à construção de formas geométricas em duas e três dimensões, considerando o *modelo 2D** e o *modelo 3D*, pois para que um *Sistema de Ladrilhos SL* seja *terminal* e *único*, basta que *SL* construa sempre a mesma forma geométrica.
- Para o problema do conjunto mínimo de rótulos, foi encontrada uma solução melhor que as já apresentadas (Rothmund e Winfree 2000).

- Programas com o objetivo de simular a construção de formas geométricas utilizando as regras de Auto-montagem ainda não haviam sido desenvolvidos. Dessa forma criamos um simulador estocástico para realizar nossos experimentos.

5.2 Trabalhos Futuros

O AMS possui uma série de características únicas, porém não executamos testes comprovando a eficácia destas características em problemas clássicos que são usados geralmente para medir desempenho de algoritmos, como o problema do caixeiro viajante. Como trabalhos futuros, destacamos a realização desses testes para comparação com outras Heurísticas Evolucionárias Multi-Objetivo (Coello 2000) e para a verificação das seguintes características do AMS:

- Se a variação da probabilidade de recombinação contribui para evolução da população.
- Se a variação de L equilibra a classificação da população.
- Se a distância mínima para recombinação promove a diversidade da população.

Outra possibilidade é a implementação de um algoritmo que identifique os indivíduos semelhantes através de técnicas de agrupamento (Jain, Murty e Flynn 1999) no AMS, onde suas configurações serão regidas pelo número de grupos no espaço de Pareto, da seguinte forma:

- O operador de recombinação é aplicado a dois cromossomos se estes pertencem a grupos diferentes.
- O aumento da probabilidade de recombinação é proporcional ao aumento no número de grupos.
- A classificação de um indivíduo depende de sua posição no espaço de Pareto e também do número de indivíduos pertencentes ao grupo ao qual o indivíduo pertence.
- O aumento de L é proporcional ao aumento no número de grupos.

- O aumento do percentual de elitismo é proporcional ao aumento no número de grupos.
- O aumento do percentual de diversidade é inversamente proporcional ao aumento no número de grupos.

Não executamos experimentos com formas geométricas genéricas em duas ou três dimensões. Dessa forma, pretendemos realizar novos experimentos em larga escala, para comprovar a eficiência dos *modelos 2D** e *3D* na construção dessas formas, assim como na construção quadrados e cubos cujo lado é maior que 5.

Outro ponto não estudado até momento é a relação entre problema do conjunto mínimo de rótulos e o problema do conjunto mínimo de ladrilhos, ou seja, até que ponto a minimização do conjunto mínimo de rótulos influencia a minimização do conjunto mínimo de ladrilhos ou qual dos problemas é mais crítico para construção de formas geométricas. Sendo assim, gostaríamos de realizar experimentos com este objetivo.

Referências Bibliográficas

- ADLEMAN, L. *Toward a mathematical theory of self-assembly (extended abstract)*. [S.l.], 2000. Disponível em: <<http://www.usc.edu/dept/molecular-science/papers/fp-000125-sa-tech-report-note.ps>>.
- ADLEMAN, L. et al. Running time and program size for self-assembled squares. In: *STOC '01: Proceedings of the thirty-third Annual ACM Symposium on Theory of Computing*. New York, NY, USA: ACM Press, 2001. p. 740–748. ISBN 1-58113-349-9.
- ADLEMAN, L. et al. Combinatorial optimization problems in self-assembly. In: *STOC '02: Proceedings of the Thirty-Fourth Annual ACM Symposium on Theory of Computing*. New York, NY, USA: ACM Press, 2002. p. 23–32. ISBN 1-58113-495-9.
- AGGARWAL, G. et al. Complexities for generalized models of self-assembly. *SIAM J. Comput.*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, v. 34, n. 6, p. 1493–1515, 2005. ISSN 0097-5397.
- ANGELOV, S.; KHANNA, S.; VISONTAI, M. On the complexity of graph self-assembly in accretive systems. In: MAO, C.; YOKOMORI, T. (Ed.). *DNA*. [S.l.]: Springer, 2006. (Lecture Notes in Computer Science, v. 4287), p. 95–110. ISBN 3-540-49024-8.

- ARBUCKLE, D. J.; REQUICHA, A. A. G. Self-repairing self-assembled structures. In: *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA '06)*. Orlando, FL, USA: [s.n.], 2006.
- BACK, T.; HOFFMEISTER, F.; SCHWEFEL, H. P. A survey of evolution strategies. In: *Proceedings of the Fourth International Conference on Genetic Algorithms*. San Diego, CA, USA: [s.n.], 1991. p. 2–9.
- BARYSHNIKOV, Y.; COFFMAN, E.; MOMCILOVIC, P. Self assembly times in dna-based computation. *SIGMETRICS Perform. Eval. Rev.*, ACM Press, New York, NY, USA, v. 32, n. 2, p. 35–37, 2004. ISSN 0163-5999.
- BARYSHNIKOV, Y. M.; JR., E. G. C.; MOMCILOVIC, P. Dna-based computation times. In: FERRETTI, C.; MAURI, G.; ZANDRON, C. (Ed.). *DNA*. [S.l.]: Springer, 2004. (Lecture Notes in Computer Science, v. 3384), p. 14–23. ISBN 3-540-26174-5.
- BRUN, Y. Adding and multiplying in the tile assembly model. In: *Proceedings of the 4th Foundations of Nanoscience: Self-Assembled Architectures and Devices (FNANO07)*. Snowbird, UT, USA: [s.n.], 2007.
- CARBONE, A.; PIERCE, N. A. (Ed.). *DNA Computing, 11th International Workshop on DNA Computing, DNA11, London, ON, Canada, June 6-9, 2005*, v. 3892 de *Lecture Notes in Computer Science*, (Lecture Notes in Computer Science, v. 3892). [S.l.]: Springer, 2006. ISBN 3-540-34161-7.
- CHEN, H.-L. et al. Invadable self-assembly: combining robustness with efficiency. In: *SODA '04: Proceedings of The Fifteenth Annual ACM-SIAM Symposium on Discrete Algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 2004. p. 890–899. ISBN 0-89871-558-X.
- CHENG, Q.; ESPANES, P. M. de. *Resolving two open problems in the self-assembly of squares*. [S.l.], 2003. Disponível em: <<http://www.cs.usc.edu/Research/techreports/papers/03-793.pdf>>.

- CHIBA, N.; NISHIZEKI, T. The hamiltonian cycle problem is linear-time solvable for 4-connected planar graphs. *J. Algorithms*, Academic Press, Inc., Duluth, MN, USA, v. 10, n. 2, p. 187–211, 1989. ISSN 0196-6774.
- COELLO, C. A. An updated survey of ga-based multiobjective optimization techniques. *ACM Comput. Surv.*, ACM Press, New York, NY, USA, v. 32, n. 2, p. 109–143, 2000. ISSN 0360-0300.
- COELLO, C. C.; PULIDO, G.; MONTES, E. Current and future research trends in evolutionary multiobjective optimization. In: _____. *Information Processing with Evolutionary Algorithms*. [S.l.]: Springer Berlin Heidelberg, 2005. p. 213–231.
- COOPER, G. M.; HAUSMAN, R. E. *The Cell: A Molecular Approach*. Washington (DC), USA: ASM Press and Sinauer Associates, 2003.
- EZZIANE, Z. Dna computing: applications and challenges. *Nanotechnology*, v. 17, n. 2, p. R27–R39, 2006. Disponível em: <<http://stacks.iop.org/0957-4484/17/R27>>.
- FERRETTI, C.; MAURI, G.; ZANDRON, C. (Ed.). *DNA Computing, 10th International Workshop on DNA Computing, DNA 10, Milan, Italy, June 7-10, 2004*, v. 3384 de *Lecture Notes in Computer Science*, (Lecture Notes in Computer Science, v. 3384). [S.l.]: Springer, 2005. ISBN 3-540-26174-5.
- GASPAR-CUNHA, A.; GONSALVES, L.; COVAS, J. A. Application of evolutionary algorithms to the design of barrier screws for single screw extruders. In: _____. *Applied Soft Computing Technologies: The Challenge of Complexity*. [S.l.]: Springer Berlin Heidelberg, 2006. p. 763–774.
- HOGBERG, B.; OLIN, H. Programmable self-assembly - unique structures and bond uniqueness. *Journal of Computational and Theoretical Nanoscience*, v. 3, p. 391–397(7), June 2006. Disponível em: <<http://www.ingentaconnect.com/content/asp/jctn/2006/00000003/00000003/art00007>>.
- HOLLAND, J. H. *Adaptation in Natural and Artificial Systems*. Ann Arbor (MI), USA: University of Michigan Press, 1975.

- JAIN, A. K.; MURTY, M. N.; FLYNN, P. J. Data clustering: a review. *ACM Comput. Surv.*, ACM Press, New York, NY, USA, v. 31, n. 3, p. 264–323, 1999. ISSN 0360-0300.
- KAO, M.-Y.; SCHWELLER, R. Reducing tile complexity for self-assembly through temperature programming. In: *SODA '06: Proceedings of the seventeenth Annual ACM-SIAM Symposium on Discrete algorithm*. New York, NY, USA: ACM Press, 2006. p. 571–580. ISBN 0-89871-605-5.
- KLAVINS, E. Directed self-assembly using graph grammars. In: *Foundations of Nanoscience: Self Assembled Architectures and Devices*. Snowbird, UT: [s.n.], 2004. Invited Talk and Extended Abstract.
- KOZA, J. R. *Genetic Programming: On the Programming of Computers by Means of Natural Selection*. Cambridge (MA), USA: MIT Press, 1992. 24–25 p.
- KRASNOGOR, N.; GUSTAFSON, S. A family of conceptual problems in the automated design of systems self-assembly. In: *FNANO. 2nd Annual Conference on Foundations of Nanoscience: Self-Assembled Architectures and Devives*. Utah, USA: Sciencetechnica, 2005. Disponível em: <<http://www.cs.nott.ac.uk/nxk/PAPERS/fnan05.ps>>.
- KRASNOGOR, N. et al. A critical view of the evolutionary design of self-assembling systems. In: *Proceedings of the 2005 Conference on Artificial Evolution*. Lille, France: Springer, 2005. (Springer Lecture Notes in Computer Science, v. 3871), p. 179–188. ISBN 3-540-33589-7. Disponível em: <<http://www.cs.nott.ac.uk/nxk/PAPERS/crit2.pdf>>.
- KUNG, H. T.; LUCCIO, F.; PREPARATA, F. P. On finding the maxima of a set of vectors. *J. ACM*, ACM Press, New York, NY, USA, v. 22, n. 4, p. 469–476, 1975. ISSN 0004-5411.
- LEVINSON, H. J. *Principles of Lithography*. Washington, DC, USA: SPIE Press, 2005.

- MAJUMDER, D. D. et al. Current status and future trends of nanoscale technology and its impact on modern computing, biology, medicine and agricultural biotechnology. *iccta*, IEEE Computer Society, Los Alamitos, CA, USA, v. 0, p. 563–573, 2007.
- MAO, C.; YOKOMORI, T. (Ed.). *DNA Computing, 12th International Meeting on DNA Computing, DNA12, Seoul, Korea, June 5-9, 2006*, v. 4287 de *Lecture Notes in Computer Science*, (Lecture Notes in Computer Science, v. 4287). [S.l.]: Springer, 2006. ISBN 3-540-49024-8.
- MARLER, R.; ARORA, J. Survey of multi-objective optimization methods for engineering. *Structural and Multidisciplinary Optimization*, Springer Berlin Heidelberg, v. 26, p. 369–395, 2004. Disponível em: <<http://www.springerlink.com/content/63ruffy7gpw00gmve>>.
- MITCHELL, M. *An Introduction to Genetic Algorithms*. Cambridge (MA), USA: MIT Press, 1998.
- PISTOL, C.; DWYER, C. Scalable, low-cost, hierarchical assembly of programmable dna nanostructures. *Nanotechnology*, v. 18, n. 12, p. 125305 (4pp), 2007. Disponível em: <<http://stacks.iop.org/0957-4484/18/i=12/a=125305>>.
- PISTOL, C.; LEBECK, A. R.; DWYER, C. Design automation for dna self-assembled nanostructures. In: *DAC '06: Proceedings of The 43rd Annual Conference on Design Automation*. New York, NY, USA: ACM Press, 2006. p. 919–924. ISBN 1-59593-381-6.
- REIF, J. H. The emerging discipline of biomolecular computation in the us. *New Generation Comput.*, v. 20, n. 3, p. 217–236, 2002.
- REIF, J. H.; SAHU, S.; YIN, P. Complexity of graph self-assembly in accretive systems and self-destructible systems. In: CARBONE, A.; PIERCE, N. A. (Ed.). *DNA*. [S.l.]: Springer, 2005. (Lecture Notes in Computer Science, v. 3892), p. 257–274. ISBN 3-540-34161-7.

- REQUICHA, A.; ARBUCKLE, D. Cad/cam for nanoscale self-assembly. *IEEE Computer Graphics and Applications*, IEEE Computer Society, Los Alamitos, CA, USA, v. 26, n. 2, p. 88–91, 2006. ISSN 0272-1716.
- ROTHERMUND, P. W. K. *Theory and experiments in algorithmic self-assembly*. Tese (Doutorado) — University of Southern California, Los Angeles, California, USA, 2001.
- ROTHERMUND, P. W. K. Folding dna to create nanoscale shapes and patterns. *Nature*, Nature Publishing Group, v. 440, p. 297–302, 2006.
- ROTHERMUND, P. W. K.; WINFREE, E. The program-size complexity of self-assembled squares (extended abstract). In: *STOC '00: Proceedings of the thirty-second Annual ACM Symposium on Theory of Computing*. New York, NY, USA: ACM Press, 2000. p. 459–468. ISBN 1-58113-184-4.
- SCHAFMEISTER, C. E. Molecular lego. *Scientific American*, v. 296, n. 2, p. 64–71, 2007.
- SOLOVEICHIK, D.; WINFREE, E. Complexity of self-assembled shapes. In: FERRETTI, C.; MAURI, G.; ZANDRON, C. (Ed.). *DNA*. [S.l.]: Springer, 2004. (Lecture Notes in Computer Science, v. 3384), p. 344–354. ISBN 3-540-26174-5.
- TABOADA, H. A. et al. Practical solutions for multi-objective optimization: An application to system reliability design problems. *Reliability Engineering & System Safety*, v. 92, n. 3, p. 314–322, March 2007. Disponível em: <<http://www.sciencedirect.com/science/article/B6V4T-4K42DJ8-1/2/bb9ad2bd4369469c475f944b8bf19a28>>.
- TERADA, Y.; MURATA, S. Automatic assembly system for a large-scale modular structure - hardware design of module and assembler robot. In: *IEEE/RSJ International Conference on Intelligent Robots and Systems*. [S.l.]: IEEE, 2004. v. 3, p. 2787–2794.

- TERRAZAS, G. et al. Automated tile design for self-assembly conformations. In: CEC. *IEEE Congress on Evolutionary Computation*. IEEE, 2005. v. 2, p. 1808–1814. Disponível em: <<http://www.cs.nott.ac.uk/nxk/PAPERS/atd4sac.pdf>>.
- TUCI, E. et al. Cooperation through self-assembly in multi-robot systems. *ACM Trans. Auton. Adapt. Syst.*, ACM Press, New York, NY, USA, v. 1, n. 2, p. 115–150, 2006. ISSN 1556-4665.
- WANG, H. Proving theorems by pattern recognition. *Bell Systems Technical Journal*, v. 40, p. 1–42, 1961.
- WERFEL, J. et al. Distributed construction by mobile robots with enhanced building blocks. In: *IEEE International Conference on Robotics and Automation*. [S.l.]: IEEE, 2006. p. 2349–2355.
- WINFREE, E. On the computational power of dna annealing and ligation. In: _____. *DNA Based Computers*. Princeton University, NJ, USA: The American Mathematical Society, 1996. (DIMACS: Series in Discrete Mathematics and Theoretical Computer Science, v. 27), p. 199–221.
- WINFREE, E. *Algorithmic self-assembly of DNA*. Tese (Doutorado) — California Institute of Technology, Pasadena, California, USA, 1998.
- WINFREE, E. Self-healing tile sets. In: _____. *Nanotechnology: Science and Computation*. [S.l.]: Springer Berlin Heidelberg, 2006. p. 55–78.
- WINFREE, E. et al. Design and self-assembly of two-dimensional dna crystals. *Nature*, Macmillan Magazines Ltd., v. 394, p. 539–544, 1998.
- YI, H. et al. Patterned assembly of genetically modified viral nanotemplates via nucleic acid hybridization. *Nano Letters*, v. 5, n. 10, p. 1931–1936, 2005. ISSN 1530-6984.