



LUCAS ARMAND SOUZA ASSIS DE OLIVEIRA

**Otimização da Edificação de Grandes  
Blocos na Construção Naval  
Empregando Algoritmo Genético**

Prof. Claudio Luiz Baraúna Vieira, Ph.D.  
Orientador

Rio de Janeiro, Setembro de 2017

# Otimização da Edificação de Grandes Blocos na Construção Naval Empregando Algoritmo Genético

Lucas Armand Souza Assis de Oliveira

Projeto Final de Curso submetido ao Departamento de Engenharia Naval e Oceânica da Escola Politécnica da Universidade Federal do Rio de Janeiro como parte dos requisitos necessários para obtenção do grau de Engenheiro Naval.

Apresentado por:

---

Lucas Armand Souza Assis de Oliveira

Aprovado por:

---

Prof. Claudio Luiz Baraúna Vieira, Ph.D.

---

Prof. Luiz Felipe Assis, D.Sc.

---

~~Prof. Jean David Caprace, D.Sc.~~

RIO DE JANEIRO, RJ - BRASIL

Setembro de 2017

# Agradecimentos

Quero agradecer, em primeiro lugar, a Deus, pela força e proteção durante toda esta longa caminhada.

Agradeço também o apoio de todos os professores e colegas do departamento de Engenharia Naval, e de tantos outros departamentos da Escola Politécnica, que me acompanharam durante toda a graduação, e, em especial, aos amigos da Área de Transporte e Logística da Engenharia Naval e da equipe UFRJ Nautilus.

Ao meu amigo e orientador, Claudio Luiz Baraúna, que foi sempre um grande mestre durante todos os anos que trabalhamos juntos. Por todo tempo, atenção e paciência dedicados.

Dedico esta, bem como todas as minhas demais conquistas, aos meus amados pais (Jemoel e Glaucia), meus irmãos (Victor, Erika, Eric, Karyne e Mel) e a toda a minha grande família (vocês são um presente de Deus na minha vida!).

Dedico também à minha namorada, Isabele Benincasa, que foi minha grande ajudadora durante toda essa jornada. Por compreender os momentos em que não pude estar presente, por escutar as minhas lamentações e por me dar força e motivação para seguir em frente.

Eu não seria capaz de fazer tudo que fiz nesses últimos anos se não fosse por todos vocês.

## RESUMO

Otimização da Edificação de Grandes Blocos na Construção Naval Empregando

Algoritmo Genético

Lucas Armand Souza Assis de Oliveira

Setembro/2017

Orientador: Claudio Luiz Baraúna Vieira, Ph.D.

O trabalho apresenta uma análise realista do potencial do método e as dificuldades nesse tipo de implementação. Foi realizada uma revisão teórica do problema de edificação e das técnicas de otimização nele empregadas. Foi desenvolvido e implementado um modelo de algoritmo genético para o caso de ordenação da edificação e por fim foi feita uma discussão sobre as limitações e potencialidades do método.



# Lista de Figuras

Figura 2.1: Esquema das transformações na organização de estaleiros desde do pré-segunda guerra até meados dos anos 90. . . . .	7
Figura 2.2: Exemplo do layout de um estaleiro . . . . .	8
Figura 2.3: Exemplo do layout de um estaleiro. . . . .	9
Figura 2.4: Exemplo do layout de um estaleiro. . . . .	12
Figura 2.5: Esquema das etapas de um algoritmo genético. . . . .	18
Figura 2.6: Exemplo de construção de uma população em codificação binária: a) construção do cromossomo de um indivíduo; b) Composição da população inicial. . . . .	21
Figura 2.7: Esquema dos operadores de crossover one-point (a), two-points (b) e threeParents (c) . . . . .	22
Figura 3.1: Representação da embarcação usada como referência nesse trabalho	25
Figura 3.2: Representação dos tipos dos blocos numa plotagem de arestas. Da esquerda para direita,são adicionados, respectivamente: Blocos de fundo, blocos de costado, blocos de convés e blocos de cofferdam.	26
Figura 3.3: Representação 3D de uma vista lateral do casco, meia seção prin- cipal e o bloco de cofferdam . . . . .	27
Figura 3.4: Vista lateral dos blocos (com a numeração) da seção de corpo paralela . . . . .	28

Figura 3.5: Métodos de visualização:(a) Três imagens superiores: Visualização em blocos transparentes, utilizada dentro das animações (b) Três imagens inferiores: Visualização em mapa de cores . . . . .	30
Figura 3.6: Esquema demonstrando a ordem de precedência entre os blocos . .	31
Figura 3.7: Mapa de cor de uma ordenação gerada aleatoriamente (exemplo de DNA) . . . . .	33
Figura 3.8: Representação do resultado de otimização para tempo de edificação dos blocos constantes, é possível ver a progressão de cores o que indica a continuidade da construção. . . . .	36
Figura 3.9: Média de 1000 cromossomos gerados aleatoriamente. . . . .	38
Figura 3.10: Etapas da primeira proposta de metodologia de crossover . . . . .	39
Figura 3.11: Etapas do método do crossover implementado . . . . .	41
Figura 4.1: Resultado da otimização de ocupação. . . . .	48
Figura 4.2: Ilustração de dois cronogramas de construção e edificação de dois blocos, apresentando o efeito da ordenação no tempo total de edificação . . . . .	51
Figura 4.3: Resultado da otimização quando blocos de tipos iguais edificados em sequência são favorecidos (resultado obtido com função de penalização ainda constante). . . . .	52
Figura 4.4: Resultado da otimização quando blocos de tipos diferentes edificados em sequência são favorecidos (resultado obtido com função de penalização não constate). . . . .	54
Figura 4.5: Resultado da quando blocos de tipos diferentes edificados em sequência são favorecidos, resultado obtido com reagrupamento dos blocos por anéis gigantes. . . . .	55

Figura 4.6: Resultado da otimização quando blocos tem relações arbitrárias,  
resultado obtido com reagrupamento dos blocos por anéis gigantes. 57

Figura 5.1: Diagrama de possibilidade para uma edificação simplificada. . . . 61

# Lista de Tabelas

Tabela 4.1: Resultados para Função de Ocupação . . . . .	47
Tabela 4.2: Resultados para diferentes ordenações quando blocos de tipos iguais edificados em sequência são favorecidos . . . . .	52
Tabela 4.3: Resultados para diferentes ordenações quando blocos de tipos di- ferentes edificados em sequência são favorecidos . . . . .	54
Tabela 4.4: Resultados para blocos de tipos (A.G. com reagrupamento dos blocos por anéis gigantes) . . . . .	56
Tabela 4.5: Resultados para diferentes ordenações quando blocos de tipos di- ferentes edificados em sequência são favorecidos . . . . .	57



# Sumário

<b>Agradecimentos</b>	<b>i</b>
<b>Resumo</b>	<b>ii</b>
<b>Lista de Figuras</b>	<b>iii</b>
<b>Lista de Tabelas</b>	<b>vi</b>
<b>1 Introdução</b>	<b>1</b>
1.1 Contextualização . . . . .	1
1.2 Objetivo . . . . .	3
1.3 Justificativa . . . . .	4
1.4 Descrição dos capítulos . . . . .	4
<b>2 Revisão Teórica</b>	<b>5</b>
2.1 Revisão dos Aspectos da Construção naval . . . . .	5
2.1.1 Introdução a Construção naval . . . . .	5
2.1.2 O caso da construção naval brasileiro . . . . .	9
2.2 Introdução ao Problema da edificação . . . . .	12
2.3 Introdução aos Algoritmos Genéticos . . . . .	14

2.3.1	Codificação em um Alfabeto . . . . .	19
2.3.2	População Inicial . . . . .	20
2.3.3	Reprodução . . . . .	20
2.3.4	Mutação . . . . .	23
2.3.5	Seleção . . . . .	23
2.3.6	Critério de Parada . . . . .	24
<b>3</b>	<b>Metodologia</b>	<b>25</b>
3.1	Embarcação de Referência . . . . .	25
3.2	Transcrição genética do problema e visualização . . . . .	28
3.3	Geração dos Cromossomos . . . . .	29
3.4	Função de Mérito . . . . .	34
3.5	Técnica de Crossover . . . . .	37
3.6	Mutação . . . . .	42
3.7	Convergência . . . . .	43
<b>4</b>	<b>Análise dos Resultados</b>	<b>45</b>
4.1	Minimização da ocupação de um dique . . . . .	46
4.1.1	Integração da construção ao modelo . . . . .	47
4.2	Otimização para o caso de blocos similares . . . . .	52
4.3	Otimização para o caso de blocos de regiões diferentes . . . . .	53
4.4	Otimização para o caso de blocos com relações arbitrárias da ordena- ção com o tempo de edificação . . . . .	55
<b>5</b>	<b>Conclusões e Propostas</b>	<b>58</b>

5.1	Propostas . . . . .	59
-----	---------------------	----

# Capítulo 1



## Introdução

### 1.1 Contextualização

O processo de construção de uma grande embarcação é muito diferente da construção de uma caneta ou de um carro, basicamente porque se trata de uma mega estrutura, ou seja, não é possível pensar uma linha de montagem onde o bem produzido transita dentro de uma cadeia produtiva. Ao mesmo tempo um navio comercial, diferente das obras na construção civil, é um produto de natureza econômica, que surge para sanar uma necessidade econômica; Então sua construção deve ser rápida e barata, não só por uma questão de reduzir os gastos na produção, mas porque o custo de construção afetará a viabilidade da embarcação por toda sua vida útil [1]. É por isso que dentro da construção naval pensar a produtividade é algo tão relevante.

Uma das principais características da construção de um navio é o desafio de executar a geometria do casco. A forma do casco de um navio, dentro da engenharia naval, tem um significado muito particular por que esta, além de fornecer o deslocamento necessário para operação do embarcação, deve ter as propriedades hidrodinâmicas que garantam a viabilidade econômica do navio enquanto meio de transporte [2]. Do balanço entre esses requisitos surge a forma do casco que normalmente é uma geometria complexa e diferente a cada embarcação, o que dá a construção desses formatos tridimensionais um carácter quase que artesanal.

Devido a essas dificuldades (o tamanho, complexidade e número de operações) o planejamento da construção de um navio é um grande desafio. Mesmo que se tenha uma lista acurada de todas as tarefas que são necessárias para finalizar a construção, se executadas fora de uma ordem apropriada podem levar a um grande desperdício de horas/homem (por exemplo quando se faz solda sobre-cabeça sem necessidade) e até a retrabalho (por exemplo quando se apoia um bloco sem os mancais corretos, isso pode gerar deformações que depois deverão ser corrigidas).

A construção naval, sobretudo a de grandes embarcações, pode ser ilustrada como um "quebra cabeça" ou como jogo de blocos de montar. Partindo das chapas de aço, através de processos como corte e conformação, são criadas chapas curvas, perfis, reforçadores e chapas planas. Na pré edificação esses elementos são unidos para montar os painéis e depois, unindo chapas e painéis, são feitos os blocos existe ainda pré montagem e montagem, com o sem pré acabamento (e dependendo do método de edificação podemos fazer ainda mega blocos unindo vários blocos). Depois de edificados existe ainda a finalização e acabamento e por fim o lançamento. Para realizar a construção é necessário realizar uma serie de processos de fabricação, como cortes, conformações, soldas e ainda o controle dimensional (sem mencionar a logística). Dai o problema de edificação de navios depender da escolha de uma ordem de fabricação e edificação dos blocos de forma a tentar minimizar o tempo total entre a encomenda e a entrega do navio.

Por isso a última etapa de um projeto naval (o de maior nível detalhamento) é o projeto construtivo, que leva em consideração características do estaleiro de construção, tais como layout, oficinas, tecnologia e capacidade de movimentação de pesos. Preparar esse planejamento é uma tarefa muito complexa e os estaleiros mais produtivos utilizam softwares para serem capaz de fazê-lo (softwares de CAD/-CAE/CAM). A integração de tecnologia da informação no projeto detalhado tem se mostrado muito proveitosa e nas ultimas décadas tem mudado a forma como a produção de embarcações é controlada (principalmente por que com a expansão do emprego de máquinas autônomas no processo de construção de navios os comando podem ser gerados direto dos programas de planejamento e o controle da produção automatizado) [3].



Lima e Velasco [4] apontam que desde da década de 1970 o Brasil tem problemas de baixa performance na produção naval, principalmente se comparado com os players asiáticos que no mesmo período se destacavam como principais competidores. A situação atual não é muito mais animadora, o setor se encontra em uma crise de grandes proporções [6]. Apesar de ter passado por um grande ciclo de investimento recente a maioria dos estaleiros não conseguiu suportar a falta de encomendas internas, isso pode ser justificado pela falta de práticas modernas de gerenciamento de produção [5].

Nesse trabalho tratamos do problema edificação dentro da construção naval, que é um subproblema dentro da ordenação da sequência das etapas da construção. No problema de edificação se buscará encontrar qual a melhor ordem de se edificar os blocos no dique de edificação. É um problema particularmente interessante uma vez que lida com as restrições do tempo de dique (que é uma variável importante para o custo de produção em estaleiros produtivos) e com os equipamentos de movimentação, tais como guindastes pórticos e veículos transportadores. Outro aspecto é que, ao lidar com a ponta final do problema de ordenação, é possível, com algumas simplificações, estudar as etapas dos passos anteriores e enxergar o impacto no produto final.

## 1.2 Objetivo

O objetivo principal deste trabalho é introduzir o problema de otimização da ordenação das atividades na construção naval, o método de algoritmos genéticos como uma ferramenta para aumentar a produtividade da construção naval. Objetiva-se, também, explorar a versatilidade do método propondo novos operadores que o permitam lidar com problemas mais gerais ou obter soluções mais refinadas.

### 1.3 Justificativa

Já existiu uma época que a construção naval caracterizou-se pelo uso de grandes contingentes de mão de obra pouco qualificada, gerando baixa produtividade e qualidade como resultado. Atualmente não é possível afirmar isso sobre a indústria, porém ainda existem muitos casos de estaleiros que resistem as mudanças da maneira de produzir, inclusive no Brasil. Por privilegiar a quantidade de mão de obra em detrimento da qualidade, esses players acabam, para minimizar custos, utilizando-se de métodos organizacionais de trabalho ultrapassados. Nesses casos, qualquer esforço em pensar maneiras de tornar a produção mais eficiente pode significar uma melhora significativa.

Assim sendo, a implementação de tecnologias de controle e otimização poderia aumentar muito a qualidade do serviço e a produtividade dos estaleiros brasileiros.

### 1.4 Descrição dos capítulos

No primeiro capítulo apresenta-se a introdução. A seguir, traça-se o objetivo do trabalho, demonstrando a finalidade do mesmo, com a justificativa para tal abordagem. No segundo capítulo, é apresentada uma contextualização da Construção Naval e das várias técnicas de otimização que podem ser empregadas nesse tipo de problema, inclusive o Algoritmo Genético (A.G.). No terceiro capítulo é apresentada a Metodologia implementada no trabalho, são feitas algumas considerações sobre as limitações e dificuldades do método de otimização empregado e da sua implementação computacional. No quarto capítulo, são discutidos os resultados obtidos e são feitas comparações entre as ordenações obtidas através da otimização e as comumente utilizadas na indústria. No quinto capítulo são apresentadas as conclusões e as considerações finais da pesquisa.

# Capítulo 2

## Revisão Teórica

### 2.1 Revisão dos Aspectos da Construção naval

A construção de um navio é uma atividade complexa, é composta por um número muito grande de processos que, para tornar a embarcação economicamente viável, dão-se, grande parte, em paralelo. Muitas vezes mais de uma embarcação é construída ao mesmo tempo dentro de um estaleiro, o que torna esse problema de mais difícil trato. Ao mesmo tempo a pressão econômica leva a uma busca constante por eficiência, o que se realiza através da implementação de novas tecnologias dentro da linha de produção. Já houve um tempo onde o agendamento era realizado manualmente por um, suposto, experiente gerente de estaleiro, mas atualmente, nos melhores estaleiros do mundo, os softwares de CAD e simulação permitem um controle muito mais apurado e um planejamento preciso das etapas produtivas dentro da construção [8]. Nesse capítulo será realizado uma breve revisão do histórico da construção naval no Brasil e no mundo, também será recapitulado as bases da teoria de algoritmo genético.

#### 2.1.1 Introdução a Construção naval

A construção naval pode ser vista como um processo que começa quando um proprietário percebe a necessidade de um navio que realize algum conjunto de fun-



ções e que, através de vários estágios de projeto (design, contratação, planejamento, etc.), culmina em uma coleção de matéria prima, peças e componentes que através da fabricação tornam-se a embarcação desejada. A produtividade da construção naval é altamente dependente da capacidade de cada um dos agentes desse processo desempenhar de forma adequada o seu papel.

A etapa final do processo de construção naval é a construção real da embarcação. A construção do navio pode ser considerada como ocorrendo em quatro níveis. A primeira é a fabricação de peças, utilizando matérias-primas (como chapas de aço e seções, tubos, chapas e cabos) para fabricar peças individuais. A compra e manuseio de componentes pode ser considerada como parte desse nível de fabricação mais baixo. O próximo nível de fabricação envolve a união de peças e/ou componentes para formar subconjuntos ou unidades. Essas pequenas coleções de peças juntas são então combinadas no terceiro nível de fabricação para formar blocos de casco. Os blocos de casco são comumente as maiores seções de navios construídos longe do local de construção final. A montagem, o nível de fabricação final, envolve o desembarque e a união de blocos no local de construção (como formas de lançamento, doca de esgrima ou doca seca). A fase de construção real da construção naval é principalmente envolvida com a mesma forma, seja de peças, subconjuntos ou blocos, para formar um navio completo [7]. Uma importante parte da fase de construção é verificando que o navio está em conformidade com os requisitos convencionais. consequentemente, o navio é submetido a uma série de testes e ensaios antes da entrega ao proprietário

A indústria da construção naval tem séculos de idade e se desenvolveu em paralelo com a história da humanidade. As técnicas de construção naval mudaram em resposta a mudanças no design, materiais, mercados e métodos de construção de embarcações. A organização das empresas de construção naval também mudou para corresponder a esta progressão.

Ao longo de sua história inicial, a construção naval, como a maioria das indústrias iniciais, foi organizada. Como tal, era quase exclusivamente dependente das habilidades dos artesãos que faziam o trabalho. Pouco planejamento foi realizado

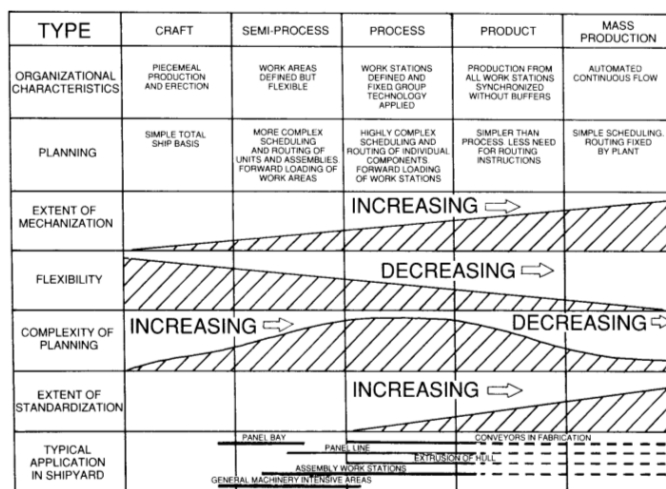


Figura 2.1: Esquema das transformações na organização de estaleiros desde do pré-segunda guerra até meados dos anos 90.

antes do início da construção. À medida que os proprietários se tornaram mais específicos na definição das características desejadas de um novo navio, os construtores de navios eram obrigados a fazer mais planejamento. No entanto, antes do uso de ferro e aço para navios, pouco mais do que um modelo em escala ou um simples desenho de um navio proposto foi usado para orientar a construção.

À medida que os processos industriais tornaram-se mais complexos e eficientes, os construtores navais acompanharam a mudança de tecnologia. A construção naval teve que ser subdividida em especialidades, como construção de casco, maquinário, equipamento e pintura. Mais recentemente, o desenvolvimento das técnicas de produção em massa e a soldagem tiveram profundos impactos na construção naval. Os estaleiros evoluíram drasticamente durante, e após, a Segunda Guerra Mundial de um, chamado padrão de primeira geração até uma geração 6 durante a última década. A Fig. 2.1 apresenta as principais alterações. Esta evolução resultou de uma implementação maciça das Tecnologia de Grupo e dos processos de automação dentro dentro da construção. [9].

Na Fig. 2.2, podemos ver os principais elementos dentro de um estaleiro. O padrão do fluxo básico de aço dentro de um estaleiro começa na área de armazenamento, onde a chapa de aço e as seções estruturais são mantidas após a entrega ao estaleiro. Em seguida, o aço vai para o "Plate Shop" onde é feita a preparação e

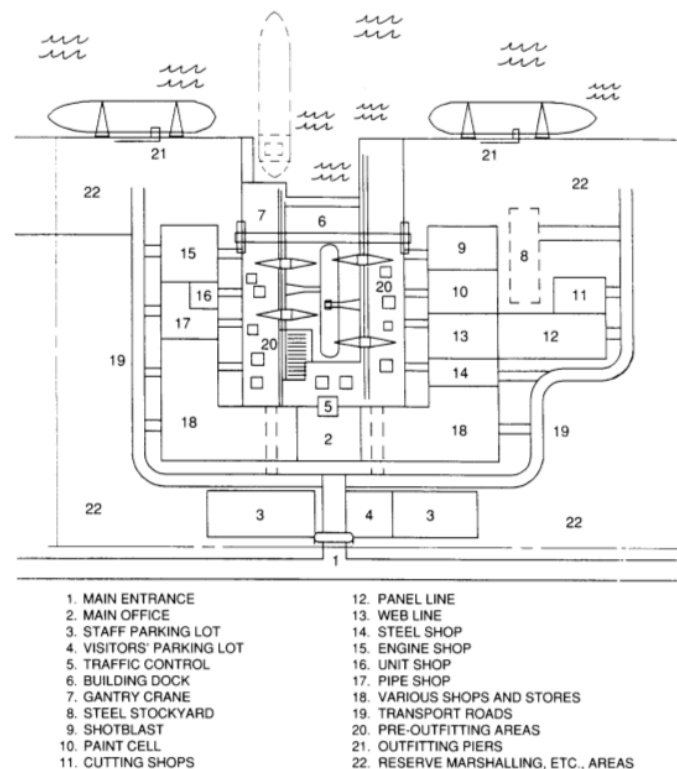


Figura 2.2: Exemplo do layout de um estaleiro .

revestimento da superfície inicial, seguido pelo corte de peças. Essas peças podem então passar diretamente pela linha do painelização ou ser estocadas para a submontagem. Através da composição de placas, painéis e peças são feitas as submontagens. As submontagens construídos seguem para montagem dos blocos. Após a conclusão dos blocos, eles são movidos para a aplicação de tinta e prime. Esses blocos são então movidos para a área de finalização, aonde são adicionado outfitting nos blocos e, em seguida, diretamente para a posição de construção. Em alguns estaleiro ao lado do deck são construídos grandes blocos, para diminuir o tempo de ocupação do deck, em todos os casos os blocos são unidos no deck tão bem finalizados quanto possível. As peças exteriores são fabricadas em shops específicos como, pipe shop, machine shop, sheet metal shop, e o electric shop. Essas unidades de equipamento, são então movidas para a área adjacente e entalados na montagem no bloco, ou para na posição de construção durante a montagem nos blocos ou a bordo. Segundo Storch [3] a organização dos elementos dentro estaleiro é pensado para minimizar a formação de estoque, engarrafamento de unidades entre os postos de trabalho

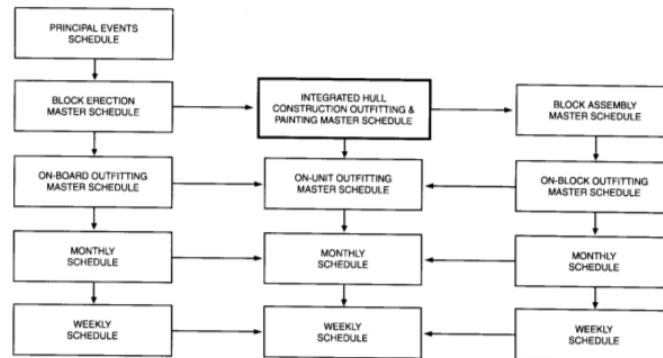


Figura 2.3: Exemplo do layout de um estaleiro.

(buffer) e reduzir o deslocamento dos materiais dentro do estaleiro. Outro aspecto importante na dinâmica da construção é o planejamento da construção. Como é apresentado na Fig. 2.3 o planejamento é feito em vários níveis, a partir dos eventos principais, que são definidos em contrato, até as tarefas diárias. Nesse trabalho focaremos no planejamento da edificação, que é o nível mais alto desse planejamento e por consequência influencia todos os níveis.

### 2.1.2 O caso da construção naval brasileiro

A história da construção naval no Brasil é extensa. Muitas vezes é dito que ela teve início em 1846, quando o Barão de Mauá inaugurou o primeiro estaleiro do País, mas mesmo antes disso, em épocas coloniais, já eram construídas embarcações dentro dos Arsenais de Marinha. Normalmente o estudo do desenvolvimento da indústria é dividido em três fases: Construção naval no Império (1822-1889, na República (pré Juscelino Kubitschek)(1889-1956) e pós Juscelino [5]. Como não é objetivo desse trabalho se aprofundar numa revisão histórica, mas tão somente recapitular o histórico recente afim de entender a situação atual da construção naval no país. Por isso discutiremos brevemente a seguir os principais marcos a partir do plano de metas de 1958.

O Plano de Metas foi um projeto encabeçado por Juscelino Kubitschek (1956-1961) com o objetivo de acelerar o desenvolvimento brasileiro. Foi um plano de cinco anos com intenso envolvimento do setor público no estímulo direto e indireto

a investimento em infraestrutura e na indústria de bens de capital. A política de desenvolvimento da indústria naval brasileira tem seu ponto de partida com a lei conhecida como "Lei do Fundo de Marinha Mercante"(FMM, Lei 3.381, de abril de 1958 ), uma lei que, através de taxas de proteção de reserva de mercado e subsídio a construção, objetiva prover recursos para a renovação, ampliação e recuperação da frota mercante nacional. Na época foram criados também o Grupo Executivo da Indústria de Construção

Naval (Geicon) e a Comissão de Marinha Mercante (CMM) que, posteriormente, viria a se transformar na Superintendência Nacional de Marinha Mercante (Sunamam).

Com as novas medidas a construção naval brasileira manteve uma atuação modesta e estável ao longo dos governos: Jânio Quadros (1961), João Goulart (1961-1964), dos interregnos de Ranieri Mazilli (em 1961 e 1964) e do governo Castelo Branco (1964-1967). Já em 1967, no governo Costa e Silva, são promovidos novos grandes investimentos nas indústrias de bens de capital e de bens de consumo duráveis. Nessa época os armadores tiveram acesso aos financiamentos e subsídios, sob a condição de concordarem com as especificações dos navios, orientadas pela Sunamam. Foi o período em que o Brasil se torna o segundo maior construtor naval mundial, durante um curto período. Os estaleiros com suas carreiras preenchidas de encomendas de navios através dos planos de construção naval, viveram momento de grande expansão. Mais adiante esse sistema de autarquia (controle de um segmento industrial através de um órgão público) se mostrou ineficiente em reagir as variações do mercado.

Pasin [10] vai nos dizer que a Sunamam apresentava em dificuldades financeiras desde o final da década de 1970. Quando em 1981, o país assiste à primeira recessão econômica desde o pós-guerra, os recursos se revelam escassos, os armadores não tiveram mercado onde utilizar os navios, situação agravada pela ingerência da Sunamam. Em 1983, a gestão financeira dos contratos de financiamentos à construção naval passa para o BNDE. Em 1987, a Sunamam foi extinta. A concessão de prioridades aos projetos passa a ser atribuição do conselho diretor do FMM, com a

diretoria do BNDES aprovando os financiamentos.

O ano de 1990, no governo Collor, marca a liberalização do transporte marítimo de longo curso. Isso expôs os armadores brasileiros à concorrência internacional. Ficou claro que as empresas locais não tinham porte para enfrentar grandes operadores em escala mundial. Juntamente aos subsídios cessaram as encomendas de embarcações e com essas a produção. Pasin [10] diz que o ciclo que levou a construção naval brasileira ao posto de segundo parque industrial naval mundial, em toneladas de porte bruto (TPB) construídas, e a empregar diretamente mais de 40 mil trabalhadores, chegou ao fim.

No final da década de 1990 ocorre a expansão da exploração de petróleo off-shore. A carteira de encomendas lotada de estaleiros mundiais favoreceu a decisão da Petrobras de criar a alternativa local para a construção dos novos navios. O programa Navega Brasil, lançado pelo governo federal em novembro de 2000, trouxe modificações nas condições do crédito aos armadores e estaleiros (expandindo as condições do FMM). Em 2003, o governo do presidente Luiz Inácio Lula da Silva, determina a prioridade para estaleiros locais de navios e equipamentos de exploração e produção de petróleo pela Petrobras. A Transpetro lança o Programa de Modernização e Expansão da Frota, licitando petroleiros de grande porte em estaleiros locais. Fazendo a carteira de encomendas brasileira ser a maior do mundo à época. Os estaleiros em fase de reativação ingressam em fase de consolidação e expansão. Grandes grupos empresariais brasileiros investem na construção de novos estaleiros. Esta fase (1997-2014), é denominada por muitos estudiosos, retomada da indústria naval brasileira e é caracterizada pelos investimentos, majoritariamente do setor offshore, e o crescimento do emprego com destaque para o forte empenho do governo federal nesse projeto.

Por fim, o momento atual, iniciado em 2015, com grande retração de investimentos, diminuição das encomendas e do apoio do estatal do governo federal. Além da diminuição de demandas da Transpetro/Petrobras concomitante aos problemas financeiros da Sete Brasil [6]. De maneira simples, talvez simplista, a crise atual no setor naval pode ser explicado pela ruína dos dois pilares que suportavam a retomada

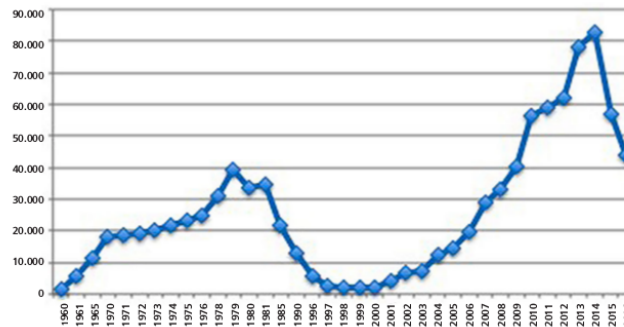


Figura 2.4: Exemplo do layout de um estaleiro.

do setor naval. Embora seja cedo para afirmar de forma mais concreta, não parece precipitado dizer que a crise institucional na Petrobras e no governo deixou órfão de pai e mãe o projeto político de uma indústria de construção naval madura no país, ao menos pelos próximos anos. Na Fig.2.4 podemos ver o reflexo dos ciclos de expansão e retração da indústria nas ultimas décadas que se refletem no número de pessoas trabalhando na área [6].

## 2.2 Introdução ao Problema da edificação

Como discutimos no capítulo anterior a construção naval brasileira viveu diversos ciclos de expansão e retração em toda sua história. Existem muitos fatores que podem ser relacionados com cada um desses movimentos, mas de forma geral podemos enxergar em cada ciclo, pelo menos, dois momentos que são representativos desse tipo de oscilação. O primeiro é o momento de aceleração, normalmente marcado pelo investimento do setor público na indústria, podendo ser feito através de subsídio, reserva de mercado, condições especiais em empréstimos para a construção naval e uma carteira de encomendas para a indústria nacional. O segundo momento é o de recessão da indústria, quando os recursos, que supostamente era abundantes, se mostram escassos. Nesse ponto a maioria dos estaleiros fecham suas portas, porque, sem condições que a suportem, a indústria naval nacional, de forma geral, não conseguiu, até hoje, ser competitiva quando comparada aos players do mercado mundial.

Se traçarmos um paralelo entre a história da construção naval nacional e a dos países que detêm as maiores indústrias navais no mundo hoje (China, Japão e Coreia), poderemos perceber que, retrocedendo algumas décadas, não havia diferença significativa entre o nível de produção naval desses países e do nosso. Porém, o ganho de produtividade que alguns países asiáticos desempenharam ao longo desses anos, garantiu a eles o domínio da produção naval na atualidade [39]. A estratégia utilizada por esses players, foi o de ganho de produtividade através do planejamento do trabalho, e, com exceção da China, a extensa utilização da automação dos meios de produção.

Certamente existem muitas especificidades no caso brasileiro, e tentar indicar qual caminho a indústria deve seguir para alcançar uma maior grau de desenvolvimento não é uma tarefa trivial, e, nem tão pouco, é pretensão desse trabalho responder a essa questão. Mas, na visão do autor, qualquer tentativa de construir uma indústria naval mais competitiva, passa por uma melhora no planejamento da produção. Nesse sinto o desenvolvimento de técnicas e ferramentas que suportem análise da produtividade dos estaleiros é fundamental para o desenvolvimento da indústria naval brasileira.

Esse trabalho foca no desafio de planejar a edificação de blocos dentro da construção naval. A edificação de blocos é o processo no qual o casco de um navio é construído através da junção de blocos. Blocos são grandes estruturas de aço soldadas a partir de perfis, chapas e painéis. Quando os blocos estão prontos, eles são tipicamente organizados perto o local de lançamento, e o processo de edificação se dá com o içamento para o local de lançamento, o posicionamento do bloco na região adequada da embarcação e em seguida o bloco é soldado ao resto da estrutura já edificada. A sequência de montagem geralmente define a ordem de produção para as etapas anteriores também e isso faz da ordenação dos blocos na edificação um problema importante [11].

Historicamente as estratégias de edificação são definidas por gerentes de produção experientes. Existem algumas estratégias de edificação são amplamente utilizadas, as principais são: Edificação em grandes blocos, aonde é adotada uma estratégia de



construção verticalizada de maneira a finalizar a edificação de um anel gigante da embarcação de cada vez; Edificação em camadas, estratégia de construção horizontalizada, de maneira a finalizar a edificação de todos os blocos de um mesmo tipo de cada vez; Edificação em pirâmide, uma estratégia mista, onde são edificados blocos de tipos diferentes alternadamente, edificando de maneira que a base da edificação seja maior que o topo (o que dá o formato de "pirâmide" à edificação).

Por outro lado, é possível que uma ordenação específica, diferente das que são amplamente utilizadas, obtenha resultados melhores para um determinado estaleiro. Por isso, existem diversos trabalhos na área da otimização da ordem de edificação. Caprace et al [12] desenvolve uma otimização da edificação acoplada a simulação de estaleiros, Tokola e Assis [13] propõem uma otimização afim de realizar as soldas dos blocos o mais cedo possível dentro da edificação, Okumoto [15] propõem uma metodologia para o encontrar a melhor posição de um determinado bloco na edificação (quando existem incertezas dimensionais). Nesse trabalho utilizaremos algoritmos genético para buscar a melhor ordenação dos blocos a serem edificados.

## **2.3 Introdução aos Algoritmos Genéticos**

Nas décadas de 1950 e 1960, vários cientistas da computação estudaram independentemente "sistemas evolutivos" com a idéia de que a evolução poderia ser usada como uma ferramenta de otimização para problemas de engenharia. A ideia em todos esses sistemas era evoluir uma "população" de possíveis soluções para um determinado problema, usando operadores inspirados na variação genética natural e na seleção natural.

Essa ideia, de uma computação evolutiva, estava no imaginário dos pesquisadores de diversas áreas na época dos primeiros computadores eletrônicos, porém o primeiro a formalizar isso num arcabouço teórico mais completo foi John Holland o inventor dos chamados "Algoritmos Genéticos" (A.G.). Os primeiros A.G. foram inventados por Holland [17] na década de 1960 e foram desenvolvidos por ele, seus alunos e colegas da Universidade de Michigan nos anos 1960 e 1970. Em contraste com as

estratégias de evolução e programação evolutiva, o objetivo original da Holanda não era projetar algoritmos para resolver problemas específicos, mas sim estudar formalmente o fenômeno da adaptação, tal como ocorre na natureza e desenvolver maneiras pelas quais os mecanismos de adaptação natural podem ser importados em sistemas informáticos [16].

O livro de 1975 da Holanda, "Adaptation in Natural and Artificial Systems" [17], apresentou o algoritmo genético como uma abstração da evolução biológica e forneceu um quadro teórico para a adaptação no algoritmo genético. Este método foi popularizado por um de seus alunos, David Goldberg, que conseguiu resolver um problema difícil envolvendo o controle da transmissão de gasodutos para sua dissertação (Goldberg, 1989). Apesar de algum avanço na teoria apresentado por Goldberg sua maior contribuição deve ter sido a popularização dos A.G. com o seu sucesso do seu livro. Desde então, muitas versões de programação evolutiva foram testadas com diferentes graus de sucesso.

Algumas das vantagens de um A.G. incluem: Otimiza com variáveis contínuas ou discretas, não requer informações sobre derivadas, procura em uma ampla área do espaço de solução podendo convergir, para múltiplas soluções ótimo-locais em paralelo, atua com um grande número de variáveis, é adequado para computadores paralelos, otimiza variáveis com superfícies extremamente complexas (podem pular fora de um mínimo local), fornece uma lista de variáveis ótimas, não apenas uma única solução, pode codificar as variáveis para que a otimização seja feita com as variáveis codificadas, e funciona com dados gerados numericamente, dados experimentais ou funções analíticas.

Essas vantagens são intrigantes e produzem resultados impressionantes quando as abordagens de otimização tradicionais falham. Claro, o A.G. não é a melhor maneira de resolver todos os problemas. Por exemplo, existem métodos tradicionais que foram desenvolvidos para encontrar rapidamente a solução de uma função analítica convexa bem comportada de apenas algumas variáveis. Para esses casos, os métodos baseados em cálculos superam o A.G., encontrando rapidamente o mínimo, enquanto o A.G. ainda estaria gerando e analisando as primeiras populações. Além

disso, para problemas que não são excessivamente difíceis, outros métodos podem encontrar a solução mais rápido que um algoritmo genético. No entanto, muitos problemas realistas não se enquadram nesta categoria. O fato é que a grande população de soluções que dão ao A.G. seu poder também é prejudicial quando se trata de velocidade de processamento. No entanto, se um computador paralelo estiver disponível, cada processador pode avaliar uma função separada ao mesmo tempo. Assim, quando é possível empregar processamento paralelo, o A.G. deve ser considerado como uma possível solução [19].

Para a implementação A.G. codificamos as variáveis de decisão de um problema em cadeias de caracteres finitos com determinada cardinalidade. As listas de caracteres, que são candidatas a soluções ao problema, são referidas como cromossomos, os alfabetos são referidos como genes e os valores de genes são chamados de alelos. Em contraste com as técnicas de otimização tradicionais, os A.G. trabalham com a codificação de parâmetros e não os próprios parâmetros em si.

Para desenvolver boas soluções e implementar a seleção natural, precisamos de uma medida para distinguir boas soluções de soluções ruins. A medida pode ser uma função objetiva que é um modelo matemático ou uma simulação computacional, ou pode ser uma função subjetiva em que os seres humanos escolhem melhores soluções sobre pior. Em essência, a medida de aptidão deve determinar a aptidão relativa de uma solução candidata, que será posteriormente usado pelo A.G. para orientar a evolução de boas soluções.

Outro conceito importante dentro de algoritmos genéticos é a noção de população. Ao contrário dos métodos de pesquisa tradicionais, os A.G. dependem de uma população de soluções candidatas. O tamanho da população, que geralmente é um parâmetro especificado pelo usuário, é um dos fatores importantes que afetam a escalabilidade e o desempenho dos algoritmos genéticos. Por exemplo, pequenos tamanhos de população podem levar à convergência prematura e produzir soluções de qualidade inferior. Por outro lado, grandes populações demandam um grande empenho computacional [19]. Uma vez que o problema é codificado de forma cromossômica e uma medida de aptidão para discriminar boas soluções foi escolhida,

podemos começar a evoluir soluções para o problema. Um esquema da implementação do algoritmo está apresentado na Fig. 2.5 e as etapas do método são descritas a seguir:

1. Inicialização: a população inicial de soluções candidatas geralmente é gerada aleatoriamente em todo o espaço de busca. No entanto, conhecimento específico do domínio ou outro as informações podem ser facilmente incorporadas na geração da população inicial.

2. Avaliação: uma vez que a população é inicializada ou uma população de prole é criada, os valores de aptidão (resultados da função fitness) das soluções candidatas são avaliados.

3. Seleção: a seleção privilegia as soluções com melhores valores resultados e, assim, impõe um mecanismo de sobrevivência do mais apto. Muitos procedimentos de seleção foram propostos para realizar essa idéia, (roleta, estocástica, classificação, torneios, etc) alguns dos quais estão descritos na próxima seção.

4. Recombinação: Recombinação combina "partes" de duas ou mais soluções parentais para criar novas soluções possivelmente melhores (isto é, prole). Aqui também existem muitas maneiras de fazer isso, mas a idéia principal é que a prole não será idêntica a qualquer "pai" particular e, em vez disso, irá combinar os traços parentais de uma maneira nova.

5. Mutação: Enquanto a recombinação opera em dois ou mais cromossomos parentais, a mutação, aleatoriamente, modifica uma solução. Novamente, há muitas variações de mutação, mas geralmente envolve uma ou mais mudanças que são feitas para um traço ou traços de um indivíduo.

6. Substituição: Por fim a população de prole criada por seleção, recombinação e mutação substitui a população original .

7. Repita as etapas 2-6 até que um ou mais critérios de parada sejam atendidos.

Nos últimos anos, tem havido uma ampla interação entre pesquisadores que estudam vários métodos de computação evolutiva e os limites entre o A.G., estratégias

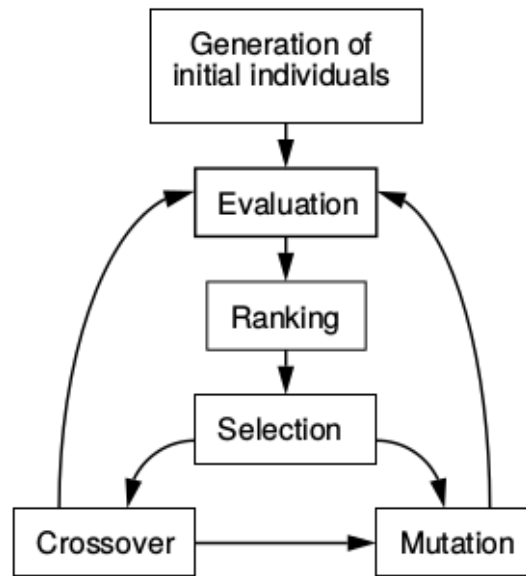


Figura 2.5: Esquema das etapas de um algoritmo genético.

de evolução, programação evolutiva, e outras abordagens evolutivas se desdobraram até certo ponto. Existem também aplicações de A.G. em diversas áreas do conhecimento tais como: Reconhecimento de padrões em dados médicos e clínicos [20], Design ótimo de um sistema de distribuição de água [21], Combate ao terrorismo [22], Controle de tráfego aéreo [23] e em conjunto com outras técnicas de aprendizado de máquinas como Redes Neurais [14].

Mais próximas a nossa área de interesse existem aplicações de algoritmos genéticos ao problema de Job Shop Scheduling (JSP). A primeira aplicação de A.G. para o agendamento foi proposto por Davis [27]. Davis introduziu o princípio de codificação/decodificação para problemas de múltiplos pontos de atendimento. Um genótipo não contém uma solução completa, mas sim consiste em regras de decisão codificadas que são decodificadas em um fenótipo viável na avaliação da aptidão física. Assim, a descodificação do genótipo por meio de uma técnica de geração de futuros eventos desempenha um papel central na programação evolutiva.

Na literatura, encontramos inúmeras aplicações de algoritmos evolutivos na programação de produção. Werner [29] em seu trabalho de revisão do tempo lista 32 referências A.G. sobre problemas relacionados ao JSP. As várias abordagens não são diretamente comparáveis porque cobrem problemas diversos, incluindo restrições e

objetivos diferentes. Sobre aplicações de algoritmo genético dentro da construção naval existem diversos trabalhos como otimização do corte de chapa [24], otimização da construção [29] e otimização do projeto [30]. No presente trabalho tratamos da otimização da construção, mas especificamente da otimização da edificação dos grandes blocos da embarcação em estaleiros navais.

### 2.3.1 Codificação em um Alfabeto

Como qualquer método de busca e aprendizagem, a forma como as soluções candidatas são codificadas é um fator central no sucesso de um algoritmo genético. O método de codificação define o espaço amostral em que será realizada a busca, e tão importante quanto isso, define a forma desse espaço amostral. Um mesmo problema pode ser codificado de várias maneiras dentro de uma busca probabilística (ver o Paradoxo de Bertrand), cada maneira de abordar o problema pode privilegiar um tipo de solução em detrimento de outra e afetar diretamente o tempo de convergência do algoritmo. A maioria dos aplicativos A.G. usa "tamanho fixo" e "posição fixa" para codificar soluções candidatas. No entanto, nos últimos anos, houve muitos experimentos com outros tipos de codificação.

Os métodos mais comuns para codificação são Codificação Binária, Codificação Discreta ou Codificação Real. As codificações binárias (ou seja, cadeias de bits) são as codificações mais comuns por vários motivos. Um é histórico: Em seus trabalhos anteriores, Holland e seus alunos concentraram-se em tais codificações e a prática da A.G. tendeu a seguir este exemplo. Grande parte da teoria da A.G. existente baseia-se no pressuposto de tamanho fixo, e de posição fixa, em codificações binárias. Grande parte dessa teoria pode ser estendida para se aplicar a codificações não binárias, mas tais extensões não estão tão bem desenvolvidos como a teoria original. Além disso, heurísticas sobre configurações de parâmetros apropriadas (por exemplo, para taxas de cruzamento e mutação) foram geralmente desenvolvidos no contexto de codificações binárias.

A codificação não binária tem algumas vantagens, para muitas aplicações, é muito natural usar um alfabeto ou números reais para formar cromossomos. Hol-

land deu uma justificativa teórica para o uso de codificações binárias. Ele comparou duas codificações com aproximadamente a mesma capacidade de carga de informações, uma com um pequeno número de alelos e cordas longas e outra com grande número de alelos e pequenas cordas. A partir dessa análise ele conclui que uma “string” de binário era mais eficiente por possuir um maior paralelismo implícito. Porém vários testes empíricos mostraram o contrário. Parece seguro afirmar que a performance depende muito do problema e dos detalhes do A.G. que está sendo usado, e, atualmente, não é possível afirmar qual é a melhor prática de codificação a priori.

Lawrence Davis, um pesquisador com muito experimente a aplicação de A.G. a problemas do mundo real, defende fortemente o uso de qualquer codificação que seja mais natural para o seu problema e, em seguida, elaborar um A.G. que possa usar essa codificação [28].

### 2.3.2 População Inicial

A população inicial é um conjunto de soluções gerado aleatoriamente. Uma vez escolhido uma codificação é, normalmente, trivial gerar soluções coerentes. Por exemplo, em uma codificação binária um indivíduo pode ser gerado através de sucessivos sorteios de "zeros" e "uns" que, guardados e organizados numa string, formam o cromossomo do indivíduo.

Dessa maneira é possível gerar os indivíduos da primeira geração, como exemplificado na Fig.2.6 .

### 2.3.3 Reprodução

Dentro do A.G. o crossover é principal mecanismo de busca, ele recombina aleatoriamente as respostas afim de gerar possíveis novas soluções. A ideia central é que dentro da representação cromossômica da resposta existem "genes" que contribuem positivamente, e outros negativamente, para o resultado daquele individuo-resposta, quando recombinaando esses cromossomos existe a possibilidade de acumularmos ge-

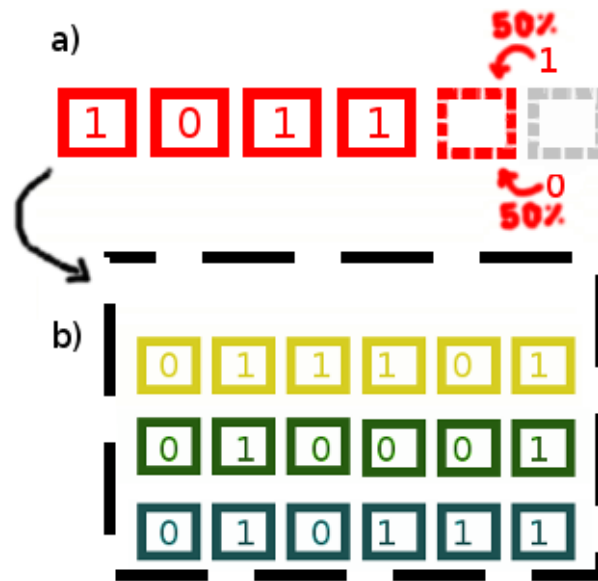


Figura 2.6: Exemplo de construção de uma população em codificação binária: a) construção do cromossomo de um indivíduo; b) Composição da população inicial.

nes bons e/ou genes ruins. Como na minha seleção eu valorizo aqueles que me dão melhor resultado, esperasse que o acumulo de genes bons ao longo das gerações encaminhem minhas soluções para a resposta ótima. Os principais exemplos de operadores crossover são:

\* One Point: Uma posição nas respostas é escolhida aleatoriamente e os dois indivíduos são recombinados de forma a manter a informação em cada indivíduo intacta até aquela posição e a informação a partir daquela posição é troca entre os dois indivíduos de forma que surjam duas novas repostas (que devem ser garantidas validas) que são a combinação dos indivíduos iniciais. Como mostra a Fig. 2.7 a.

\* Two Points: Similar ao método anterior, mas são escolhidos dois pontos aleatórios na resposta. A partir desses dois pontos determinasse o trecho que sera trocado entre os indivíduos e essa troca ocorre de maneira análoga ao método anterior. Como mostra a Fig. 2.7 b.

\* ThreeParents: Uma variação da forma convencional de combinação que utiliza três indivíduos como base para recombinar. Como mostra a Fig. 2.7 c.



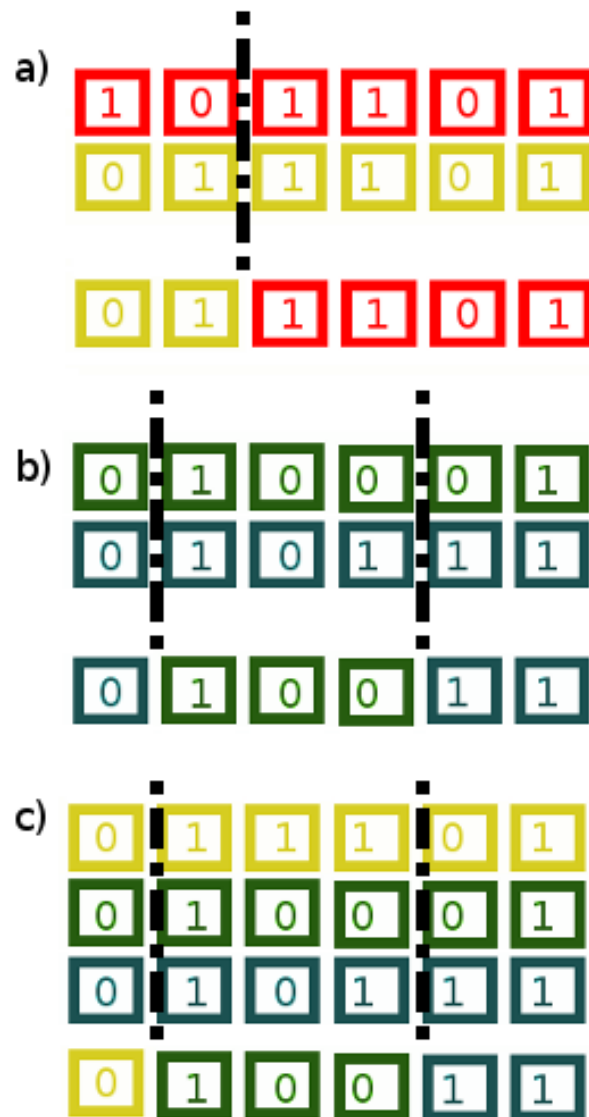


Figura 2.7: Esquema dos operadores de crossover one-point (a), two-points (b) e threeParents (c)

### **2.3.4 Mutação**

Existe basicamente duas maneiras de gerar novos indivíduos dentro de um A.G.. Através de Mutação e Crossover. O crossover já foi discutido e a mutação é ainda mais simples. Dado um indivíduo (cromossomo), uma mutação é uma pequena variação aleatória em um, ou mais, de seus genes.

Ao contrário do que se pode intuir a mutação não é um processo de menor importância dentro da lógica do algoritmo genético. David [29] apresenta que um algoritmo genético só com mutação é mais eficiente do que um só com operador de crossover. Isso ocorre por que durante o algoritmo genético, entre uma geração e a seguinte, varias respostas são eliminada pelo operador de seleção. Esse processo leva a uma redução da variabilidade genética, que por sua vez leva a uma estagnação do processo de otimização. O operador de mutação é a ferramenta do algoritmo para buscar novas possibilidades fora do espaço de soluções composto pelos genes da primeira geração.

### **2.3.5 Seleção**

A seleção é o processo de escolha dentro de uma determinada população dos indivíduos que realizaram crossover. É nessa etapa que a função de otimização é considerada. Existem diversas maneira de realizar a seleção, existem algoritmo que são chamados elitistas, porque permitem que os melhores indivíduos de uma geração passem para a próxima (sem sofrer mutação ou ser fruto de um crossover), em casos de funções multi-objetivo existem algoritmos dominantes e não dominantes [26].

A seguir estão exemplos dos operadores seleção, mas comumente utilizados:

\* roulette-wheel (roleta): É atribuída a cada individuo uma probabilidade proporcional à qualidade do seus resultado. Os individuos para serem recombinados são escolhidos tomando essas probabilidades como base.

\* tournament (torneio): Um pequeno grupo de indivíduos é selecionados dentro da população, o que tiver o melhor resultado é selecionado para ser recombinado. O

processo é repetido tantas vezes quanto for necessário para compor a nova população

\* truncation (truncada): São selecionados os melhores elementos da população para serem recombinados.

### **2.3.6 Critério de Parada**

Por fim é necessário estabelecer um critério de parada do algoritmo. Uma questão relacionada a esse tipo de algoritmo é que não é possível saber quando você atingiu a solução ótima, ou nem mesmo se você a atingiu. Os critérios mais comuns são o de um número máximo de gerações, ou seja, um limite máximo de processamento para a busca da solução ótima ou o de variabilidade genética, isso pode ser traduzido pela diferença entre os genes do primeiro e . Um dos problemas que se tenta evitar quando se aplica A.G. é o de convergência precoce. Para tentar garantir que um solução ótima ou perto-da-ótima os critérios que são modificados são o tamanho da população, a taxa de crossover, a taxa de mutação e os critérios de parada.

No próximo capítulo será discutido como todos esses operadores foram adaptados para a aplicação da construção naval.

# Capítulo 3

## Metodologia

A metodologia escolhida para se alcançar os resultados esperados foi de pensar um algoritmo genético que contemplasse o maior número possível de soluções. Isso é interessante porque limita as influências das hipóteses iniciais dos autores sobre os resultados, ao mesmo tempo que testa os limites da capacidade do método. Por causa disso nós usaremos somente restrições físicas para delimitar o nosso conjunto de soluções e para obter respostas factíveis utilizaremos penalizações dentro da função objetivo definida.

### 3.1 Embarcação de Referência

Para a implementação do método nesse trabalho foi escolhido uma embarcação de grande porte (um LNG), , de 220 000m<sup>3</sup>, com 319 m de comprimento total, 303 m de comprimento entre perpendiculares. Somente a parte prismática do navio (5

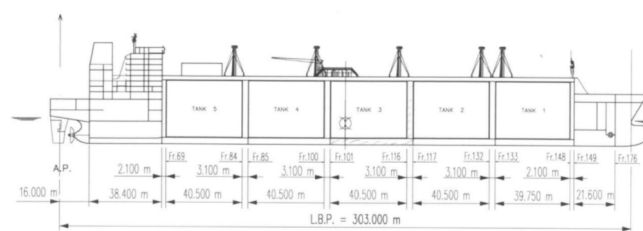


Figura 3.1: Representação da embarcação usada como referência nesse trabalho

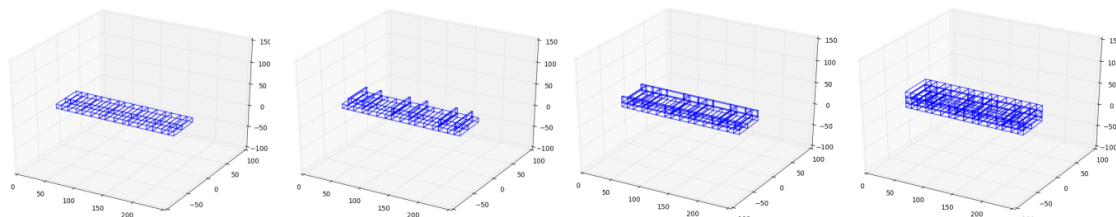


Figura 3.2: Representação dos tipos dos blocos numa plotagem de arestas. Da esquerda para direita, são adicionados, respectivamente: Blocos de fundo, blocos de costado, blocos de convés e blocos de cofferdam.

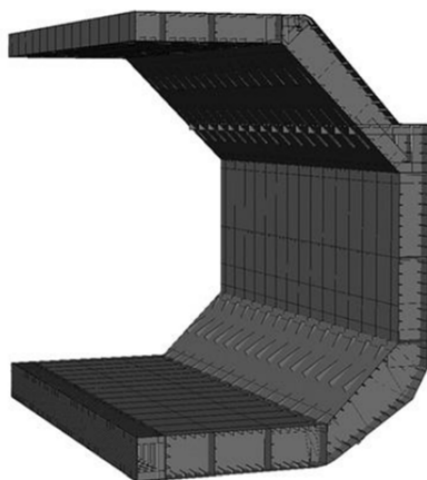
tanques, 218.95 m de comprimento, 22 000 tons) foi considerada nesse trabalho. A Fig.3.1 mostra a embarcação representada. O LNG utilizado corresponde ao utilizado nos estudos realizados por Caprace et. al [31] e [32] e por Baptista [35]. Em seus trabalhos foi explorado a simulação de toda a construção do navio. No capítulo final será discutido como um trabalho de simulação poderia ser integrado a um algoritmo de simulação afim de fornecer uma análise mais completa.

Sobre as divisões do blocos do navio, seu corpo paralelo foi considerado dividido em oito anéis, seis destes contendo anteparas transversais, da mesma maneira que fora feito nos trabalhos anteriores que usaram essa mesma embarcação. Na Fig. 3.3 é possível ver com mais detalhes a estrutura da embarcação, a imagem apresenta uma representação 3D do navio, da seção do navio e do bloco de cofferdam. A seção da embarcação pôde ser caracterizada pelo seu casco duplo, 50 metros de boca, 36 metros de pontal e 40,5 m de comprimento. A embarcação foi dividida em blocos de fundo, blocos de costado, blocos de convés e blocos de cofferdam.

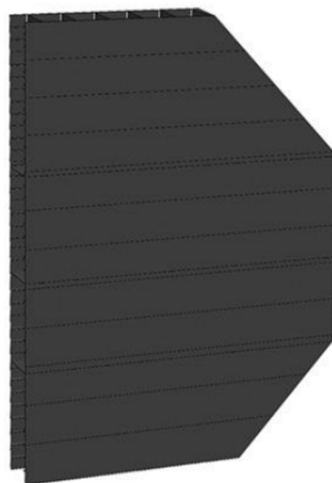
Na imagem da Figura 3.2 estão representados todos os blocos (cofferdam não é visível por fora) do navio com suas respectivas dimensões. Ao todo são 16 blocos de convés, 16 blocos de costado, 32 blocos de fundo e 6 de cofferdam totalizando 70 blocos. Cada bloco recebe um número, entre 1 e 70, para permitir sua identificação. A Figura 3.4 mostra a visão de um bordo da seção paralela do navio onde é possível ver metade dos blocos de fundo, metade dos blocos de costado e todos os blocos de convés com a respectiva sua numeração.



LNG outlines



LNG amidships section



LNG cofferdam

Figura 3.3: Representação 3D de uma vista lateral do casco, meia seção principal e o bloco de cofferdam

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17		19		21		23		25		27		29		31	
34	36	38	40	42	44	46	48	50	52	54	56	58	60	62	64

Figura 3.4: Vista lateral dos blocos (com a numeração) da seção de corpo paralela

Uma vez estabelecida a geometria da embarcação pode-se analisar as relações entre os blocos no que diz respeito a ordenação.

### 3.2 Transcrição genética do problema e visualização

Uma vez estabelecidas as características do problema é necessário construir o espaço de busca dentro de uma representação cromossômica. No problema de melhor ordenação para edificação é comum utilizar uma lista ordenada como o cromossomo. Ou seja, dado um conjunto de blocos **numerados**, se for definido uma ordenação, é possível organizar a lista de blocos de acordo com essa ordenação, isso seria uma lista ordenada. Nesse trabalho trataremos listas ordenadas dos blocos/index como cromossomos. Em outras palavras, cada cromossomos tem 70 genes, cada gene é um valor inteiro entre 1 e 70. Assim o alelo "**n**" no posição "**m**" do cromossomo representa que o bloco de index "**n**" será o "**m**"ésimo bloco a ser edificado dentro dessa ordenação.

Definido isso, um outro problema é como interpretar esses resultados. Uma lista de número não traduz diretamente o significado físico da ordenação, ao mesmo tempo que a visualização das resposta é fundamenta para implementação do método, uma vez que permite uma supervisão do funcionamento dos algoritmos e suporta uma análise crítica dos resultados. Nesse trabalhos utilizamos dois métodos principais para interpretar os resultados.

A Fig.3.5-a apresenta o primeiro método, onde a construção é representada por uma animação 3D onde, um á um, cada bloco é adicionado ao conjunto de blocos exibidos seguindo a ordenação. Esse tipo de representação é interessante porque traduz a sequência da edificação, isso torna mais fácil perceber se existe alguma

incoerência em algum ponto da ordenação (algo como um bloco de topo edificado antes de um bloco de costado ou cofferdam. Nesse tipo de visualização a utilização de "blocos transparentes" (representação do bloco só pelo seu o conjunto de arestas). Essa representação pode ser confusa em alguns casos, pela sobreposição de linhas, mas dentro da animação permiti uma fácil visualização do bloco que esta sendo edificado em cada etapa, mesmo para blocos como o de cofferdam que ficam no meio da estrutura.

A Fig.3.5-b mostra a representação em mapa de cores para os três métodos de edificação já discutidos na seção anterior. Da direita para esquerda eles são: Grandes Blocos, Camadas e Pirâmide. Esse tipo de visualização é interessante porque permite um rápido entendimento do processo de construção como o todo, também permite comparação de múltiplas respostas ao mesmo tempo. O A.G. é um processo de busca estocástico, ou seja, não necessariamente se obterá o mesmo resultado em uma otimização com os mesmos parâmetros. Em problemas onde não existe um único ótimo global, combinar diversos resultados pode dar informações sobre a região ótima de reposta. Em nossa proposta cada cromossomo guarda a posição de todos os blocos para uma determinada ordenação, assim, se comparamos diversas ordenações, podemos pensar em qual é posição (na média) que um determinado bloco ocupa. Porém, um ordinal é, por definição, um inteiro, mas se estendermos o entendimento de ordem para uma "ordem média", podemos construir uma ordenação média que é a média de diversas ordenações. Esse conceito é uma abstração porque cada bloco agora tem uma posição que é um valor real e dois blocos podem ter a mesma ordenação. Nesse caso a representação de ordem como uma animação sequenciada deixa de fazer sentido, mas como o espectro de cores é contínuo o mapa de cores ainda pode ser usado e permite uma rápida visualização desse tipo resultado.

### 3.3 Geração dos Cromossomos

O professor do MIT Patrick H. Winston [38], em suas aulas sobre algoritmo genético, quando apresenta para seus alunos o notório trabalho de Karl Sim "Envolving Virtual Creatures" [36] lembra que na primeira que teve contato com o trabalho



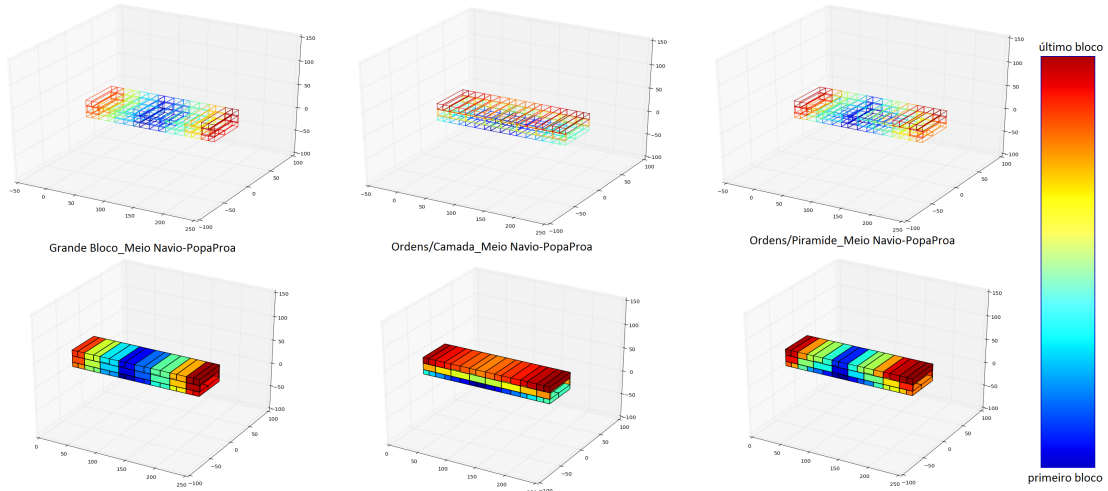


Figura 3.5: Métodos de visualização:(a) Três imagens superiores: Visualização em blocos transparentes, utilizada dentro das animações (b) Três imagens inferiores: Visualização em mapa de cores

disse "wow, this space is rich in solutions".

Talvez isso não seja notório para pessoas menos envolvidas com a área mas a arquitetura do espaço de busca é um fator muito importante (se não o mais importante) dentro dos A.G.s. Por isso, inicialmente, os autores desse trabalho objetivavam tornar o espaço de busca composto somente por soluções factíveis. Para uma solução ser considerada factível ela deve conter todas as 2 propriedades: (I) Fisicamente coerente, ou seja, nem um bloco pode ser edificado sem que haja blocos para suporta-lo fisicamente; (II) Construção continuada, ou seja, em qualquer etapa da edificação, com exceção do primeiro bloco, o próximo bloco a ser edificado tem que ser ligado a algum dos blocos já edificado. Essas propriedades foram estabelecidas com base nas boas práticas do mercado e em vista de que todas as edificações referencia seguem essas duas regras.

Determinar uma maneira de gerar ordenações aleatórias que tenham significado físico não é trivial. Um esquema das relações de precedência utilizados nesse trabalho esta apresentado na Figura 3.6. Essas relações garantem que a primeira propriedade (I) seja respeitada, uma vez que garantem uma precedência entre os blocos das

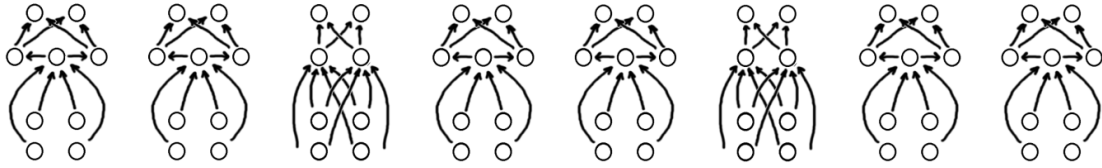


Figura 3.6: Esquema demonstrando a ordem de precedência entre os blocos

diversas regiões da embarcação, mas não dizem nada sobre a segunda relação. Na verdade se percebeu que não é possível assegurar as propriedades através de regras de precedência previamente estabelecidas. Isso porque para existir uma construção continuada um bloco do mesmo tipo ou camada só pode ser edificado caso algum outro ao lado (do mesmo tipo ou camada) já tenha sido edificado, com uma única, e óbvia, exceção, que é o primeiro bloco de cada camada. Mas se visamos formalizar isso no sentido de regras previamente estabelecidas de relações de precedências entre os blocos não existe uma maneira de construir essa "exceção" do primeiro bloco, ou seja, se todos os blocos dependem dos blocos ao lado para serem edificados nem um bloco pode ser edificado para começar. Existem algumas maneiras de contornar esse tipo de problema, como por exemplo, sortear, á principio, quais blocos de cada tipo seriam os primeiros a serem edificados e torna-los disponíveis para edificação sem a necessidade de blocos conectados, porém sem ignorar a propriedade (I). Porém quando dois cromossomos se recombina a organização dos blocos se perde e, sem relações de precedências gerais, não seria possível reconstruir a ordenação de forma coerente. Por fim, o autor desse trabalho decidiu flexibilizar a propriedade (II) para que a continuidade seja garantida através de uma penalização dentro da função a ser otimizada pelo algoritmo. Na conclusão desse trabalho é proposto uma nova metodologia que consegue garantir as propriedades (I) e (II) dentro do espaço de busca.

A Figura 3.6 é uma tentativa de formalizar a natureza física do problema, ou seja, representar graficamente as restrições que são implícitas ao problema. Para exemplificar pode-se tomar o primeiro grupo de nós e setas (da esquerda para direita). Cada nó representa um bloco e as setas as relações de precedência entre

eles, os quatro nós inferiores são quatro blocos de fundo, as setas partindo desses blocos até o nó central (no caso um bloco de cofferdam) significa uma condição de precedência. A leitura correta dessa parte do diagrama é que todos os blocos de fundo devem ser edificados antes (preceder) o bloco de cofferdam. Vale ressaltar que essas relações são evidentes dentro do processo racional de ordenação, uma vez que os blocos de fundo suportam os blocos de cofferdam, logo não é possível edificar um bloco de cofferdam antes dos blocos de fundo correspondentes a ele.

As regras apresentadas anteriormente são capazes de dizer quais blocos devem ser edificados antes dos outros porem ainda não são uma proposta formal de metodologia para gerar uma ordem de edificação aleatória e funcional. A seguir o Algoritmo 1 proposto para gerar de forma aleatória a ordem de edificação.

---

**Algorithm 1** Ordenação Aleatória do Blocos
 

---

```

1: procedure ORDENAÇÃO
2:   BlocosPermitidos  $\leftarrow$  BLOCOS-DE-FUNDO
3:   OrdemEdificação  $\leftarrow$  VAZIO
4:   loop:
5:     if BlocosPermitidos  $\neq$  VAZIO then
6:       BlocoEscolhido  $\leftarrow$  ALEATÓRIO(BlocosPermitidos)
7:       BlocosPermitidos  $\leftarrow$  BlocosPermitidos – BlocoEscolhido
8:       OrdemEdificação  $\leftarrow$  OrdemEdificação + BlocoEscolhido
9:       NovosBlocos  $\leftarrow$  PRECEDÊNCIA(BlocoEscolhido, OrdemEdificação)
10:      BlocosPermitidos  $\leftarrow$  BlocosPermitidos + NovosBlocos
11:      goto loop.
12:     else
13:       close
14:     end.
  
```

---



No Algoritmo 1 trabalhamos com os blocos variando entre duas listas: *BlocosPermitidos* e *OrdemEdificação*. A lista *OrdemEdificação* é a variável principal do método. Ela é construída ao longo do processo através de 70 interações aleatórias, e ao final deve ser um vector de blocos que contem todos os blocos da embarcação e que corresponde a uma ordenação de edificação viável. A ordem dos blocos dentro do vector

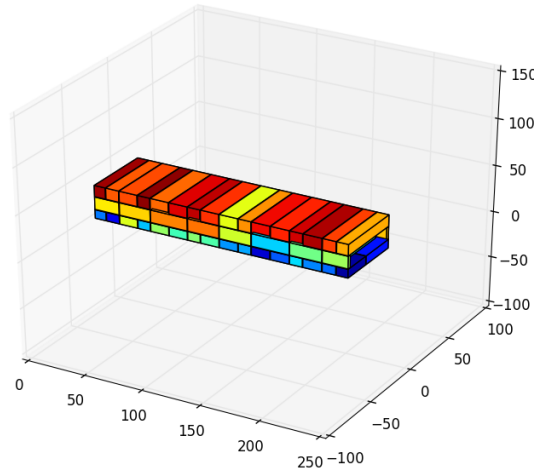


Figura 3.7: Mapa de cor de uma ordenação gerada aleatoriamente (exemplo de DNA)

corresponde a ordem de edificação. A lista *BlocosPermitidos* é, também, um vetor de blocos que corresponde, em cada ciclo do processo, aos blocos que podem ser edificados em seguida, ou seja, é, a cada iteração, o vetor deve conter os blocos que não tem precedentes ou que todos os precedentes já estão no vetor *OrdemEdificação*.

No início do processo a lista "blocos permitidos" é composta pelos blocos de fundo. A medida que ordenação é definida, e isso se dá através de sucessivas iterações aleatórias, novos blocos são adicionados a esta lista e assim o processo segue até o final. Ao mesmo tempo que a lista de "blocos edificados" começa vazia e, a cada, interação recebe um bloco sorteado aleatoriamente da *BlocosPermitidos*. No final a "OrdemEdificação" conterá todos os blocos e a "BlocosPermitidos" ficará vazia.

Para uma algoritmo genético é necessário fazer uma representação genética do domínio do espaço de solução. Dentro do nosso modelo de algoritmo genético a *OrdemEdificação* representa o "DNA" dos indivíduos. Esse DNA é uma lista de valores inteiros entre 1 e 70, cada valor é o index do seu bloco correspondente.

A Figura 3.7 apresenta a representação de uma ordenação levando em consideração as regras de precedência, analisando a escala de cores é possível perceber que nem um bloco é mais avermelhado do que o bloco que sustenta. Isso significa uma ordenação coerente dentro do que foi proposto. A leitura do "DNA" dessa solução é

"56, 37, 49, 42, 41, 46, 54, 36, 61, 55, 58, 51, 52, 59, 64, 53, 35, 33, 43, 34, 47, 62, 50, 68, 28, 25, 60, 27, 63, 70, 57, 11, 65, 38, 26, 44, 48, 22, 18, 21, 45, 67, 23, 39, 40, 10, 9, 66, 6, 20, 19, 5, 31, 4, 69, 12, 32, 17, 24, 2, 1, 8, 7, 16, 15, 3, 30, 29, 14, 13"que representa a ordenação dos blocos.

### 3.4 Função de Mérito

A função de mérito é o objetivo da otimização, é a função que se deseja maximizar ou minimizar. Dentro desse trabalho foram explorados algumas variações de funções de mérito, variando de acordo com o contexto analisado. Nessa seção será discutido como introduzir dentro da função de mérito uma penalidade afim de se garantir uma construção continuada.

A ideia de introduzir uma penalidade é simples, o A.G. é um método para o mínimo da função objetivo, se as soluções que tiverem alguma descontinuidade sofrem um acréscimo na função de mérito e assim naturalmente esses "indivíduos"são descartados.

O caso mais simples, para referência é uma otimização de tempo total da edificação com tempos de edificação dos blocos constantes, isso significa, a cada bloco é definido um tempo de edificação especificado a priori (no caso todos os blocos foram definidos com uma unidade de tempo) e dessa maneira o tempo de edificação de cada bloco não depende da ordem dele na sequência de edificação. Pode-se perceber que o essa "otimização"não faz sentido, por que, como já foi discutido, nosso algoritmo só pode alterar a ordem dos blocos e se o tempo de edificação já não depende da ordem não há nada para ser "otimizado". Por outro lado, se ao modelo for adicionarmos a chamada "função de penalidade"já é possível aplicar métodos de busca de solução ótima. Nesse caso, o único parâmetro a otimizar seria a função de penalidade e o resultado esperado é simplesmente uma construção continuada. Esse caso foi implementado para explorar o funcionamento da penalidade dentro da função de mérito e discussão dos resultados esta a seguir.

$$\sum_{i=1}^N T(b_i) = f(B) \quad (3.1)$$

$$T(b_i) = \begin{cases} C, & \text{if } \forall b_j \in B^*(i) : b_i \text{ connected to } b_j \\ C + p, & \text{if } \forall b_j \in B^*(i) : b_i \text{ is not connected to } b_j \end{cases} \quad (3.2)$$

A Função de mérito " $f(B)$ ", proposta para o caso descrito anteriormente, está apresentada nas equações 3.3 e 3.4. Aonde o tempo de edificação ( $T(b_i)$ ) de cada bloco ( $b_i$ ) é dado na equação 3.4. Este tempo de é descrito como constante ( $C$ ) para o caso de já existir algum bloco edificado imediatamente ao lado dele, caso contrário esse tempo é acrescido de uma penalidade ( $p$ ). Uma notação relevante é  $B^*(i)$  que corresponde a sequência de bloco ( $B$ ) até imediatamente antes do bloco " $i$ ".



Essa execução, onde se conhecia a priori o resultado esperado, nos permitiu identificar alguns problemas e melhorar o método.

Nas primeiras versões do programa nos usávamos uma função de penalidade onde se o bloco fosse edificado sem nenhum bloco vizinho (ou seja, uma descontinuidade na construção) ele recebia uma penalidade de tempo constante, porém isso se mostrou pouco eficiente e o algoritmo convergia de forma prematura para edificações com algumas descontinuidades. Mittu Mittal et. al [37] quando falam sobre as limitações do algoritmo genético apontam que, caso a função fitness retornar uma única medida de certo/errado, o algoritmo genético pode ser tão eficiente quanto uma busca aleatória. A luz disso foi repensada a função de penalização para considerar a distância entre o bloco colocado e o bloco mais próximo (ou seja o quão longe esse bloco está de onde ele deveria estar) e assim a convergência se tornou muito melhor.

$$\sum_{i=1}^N T(b_i) = f(B) \quad (3.3)$$

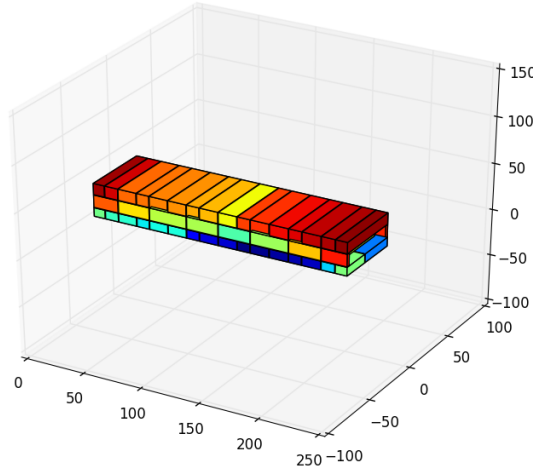


Figura 3.8: Representação do resultado de otimização para tempo de edificação dos blocos constantes, é possível ver a progressão de cores o que indica a continuidade da construção.

$$T(b_i) = \begin{cases} C, & \text{if } \forall b_j \in B^*(i) : b_i \text{ connected to } b_j \\ C + \alpha * d(i, B^*(i)) & \text{if } \forall b_j \in B^*(i) : b_i \text{ is not connected to } b_j \end{cases} \quad (3.4)$$

Na Fig.?? nos mostra o resultado médio de dez execuções da versão final do algoritmo (quando aplicado tempo constante). Nessas dez execuções o tempo médio total das edificações foi de "setenta unidades de tempo", ou seja, como são setenta blocos, isso significa que todas as execuções convergiram á região de ótimo global (uma sequência de edificação que garanta continuidade ao longo da construção).



Esse resultado é muito interessante porque nos indica que o método privilegia sequência partindo do centro. Isso faz sentido, uma vez que, se compreende a resposta (resultado final) como fruto de um processo de recombinação e mutação. Uma forma simplificada de visualizar isso é pensar que, durante a execução do algoritmo, se por exemplo uma resposta começa a sua edificação pelos blocos centrais, esta tem vantagem em relação a uma sequência que começa a edificação pelos blocos de popa, isso porque, durante a fase de crossover, a primeira pode se combinar com uma outra resposta para produzir tanto um resultado que começa no centro e progride para a popa, quando um resultado que comece no centro progrida em direção a proa. Ou

seja, como o algoritmo genético tem características aleatórias, existe uma vantagem de começar pelos blocos centrais e esse resultado reflete isso.

Outra característica que deve nos chamar a atenção é a forte assimetria na progressão horizontal com a progressão vertical das edificações. Obviamente, como não é possível edificar o bloco de topo sem primeiro edificar os seus respectivos blocos de fundo, é esperado que os blocos de fundo sejam, em média, edificados primeiro que os blocos de topo, mas na “Fig.4.1” a diferença parecia ser muito grande e nós decidimos estudar mais o porque.

A “Fig.3.9” mostra a média das ordens de edificação entre mil cromossomos gerados aleatoriamente. Observando esse novo resultado, e sabendo a maneira como nos geramos esses cromossomos é possível entender porque essa tendência de uma construção horizontal. Como o nos propomos a criar um método somente com restrições físicas para a geração dos cromossomos, inicialmente todos os blocos de fundo estão "disponível" para que sejam edificados, o problema disso é que a probabilidade de que os blocos do fundo sejam os primeiros é muito grande (obviamente os primeiros quatro blocos são sempre de fundo, mas esse efeito faz com a probabilidade de aparecer um bloco de cofferdam ou de costado logo no inicio fique diluída no meio de todos os blocos de fundo). Isso explica a tendência de ordenação horizontal que observamos. É importante ressaltar que, enquanto o efeito que centralizava o inicio das edificações estava relacionado a natureza do algoritmo genético em si, esse segundo efeito esta mais relacionado a como nós estamos gerando os cromossomo aleatório, ou seja, como nós pesamos a aleatoriedade do problema.

### 3.5 Técnica de Crossover

O algoritmo genético é um, dentre muitos, método de otimização baseado em busca do ótimo, ou seja, a partir da valor da função para diversos pontos dentro do espaço amostral são escolhidos novos pontos para analisar (e assim sucessivamente). O crossover, dentro do A.G., é o principal mecanismo de busca da solução.

Nesse trabalho foi aplicada uma técnica de crossover denominada crossover de



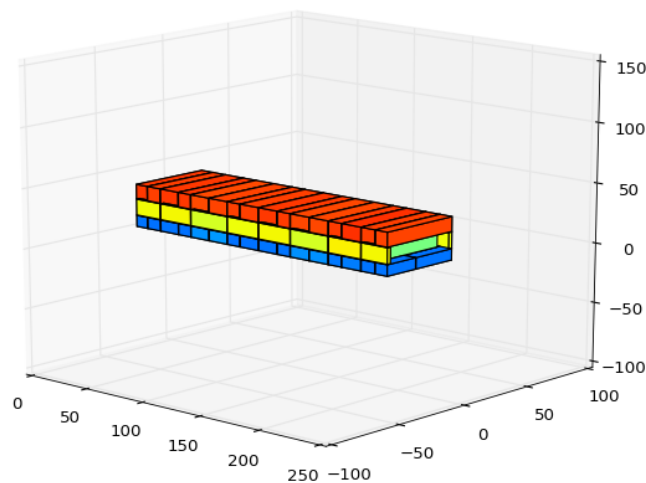


Figura 3.9: Média de 1000 cromossomos gerados aleatoriamente.

um único ponto. Nela dois indivíduos são selecionados para serem combinados, então um ponto no DNA é escolhido como ponto do crossover, o DNA dos dois indivíduos é separado em duas seções no ponto de crossover e então a segunda seção de DNA dos dois indivíduos é trocada e por fim os blocos errados são reordenados.

A seguir serão apresentados duas metodologias distintas de crossover, ambas foram desenvolvidas ao longo desse trabalho e a segunda representa uma melhoria da primeira uma vez que é um método equivalente porem demanda menos comparações e trabalho computacional. As duas serão apresentadas através de um exemplo simplificado mas análogo ao **caso d estudo**. Nesse exemplo tratamos de uma **"edificação"** de 9 blocos onde os DNAs são representados por uma sequência de letras entre "A" e "I" dentro do sequência alfabética. Nessa proposta, para representar as restrições físicas do nosso problema, será considerada a restrição de que os blocos "A", "E" e "I" devem sempre ser edificados nessa ordem (restrição de precedência). É necessário frisar que na versão final da metodologia somente o segundo método foi implementado, porem a apresentação de ambos é relevante uma vez que suporta uma melhor compriensão dos desafios da recombinação de sequências de edificação.

Na Figurá 3.10 estão representadas todas as etapas do processo de crossover do primeiro método proposto. No recorte "I" estão duas sequências aleatórias propostas para esse exemplo, onde estão sinalizados os blocos "vogais" para indicar que a

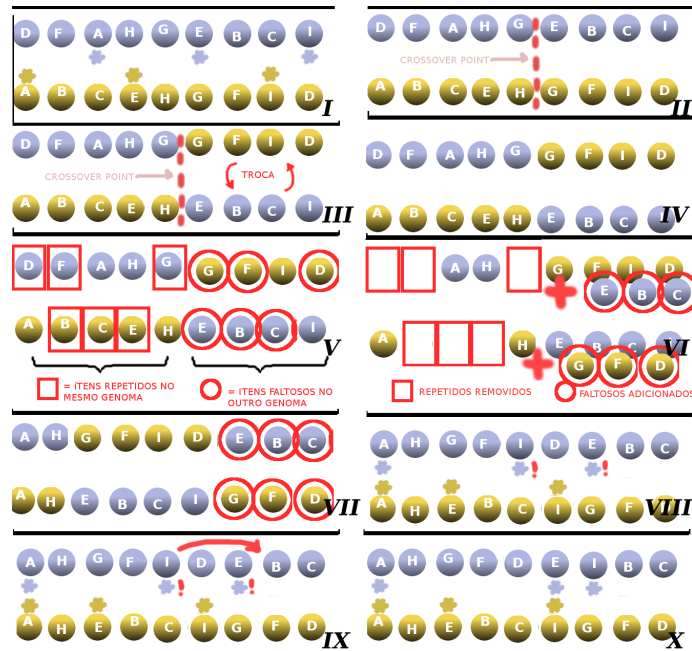


Figura 3.10: Etapas da primeira proposta de metodologia de crossover

posição relativa entre eles está de acordo com a restrição. No recorte "II" é apontado o ponto de crossover onde será feita a troca do cromossomos, que no nosso modelo é escolhido aleatoriamente. Nos recortes 'III' e 'IV' é apresentado as listas já alteradas pela troca de cromossomos. O recorte 'V' apresenta uma análise do resultado demonstrado na parte 'IV', onde são apontados os itens que, com a chegada do novo trecho, estão duplicados na mesma lista (marcados com quadrados vermelhos) e quais os itens que, na troca de itens, ficaram faltando ao DNA original (marcados com círculos vermelhos). No recorte 'VI' é apresentado como esses problemas (itens duplicados/faltosos) é resolvida, os itens que marcados como duplicados são deletados e os itens sinalizados como faltosos são copiados para serem reincorporados à sua lista original. O recorte 'VII' mostra o resultados das operações anteriormente descritas. No recorte 'VIII' é chamado a atenção que, durante o processo de crossover, as restrições em relação as blocos vogais foram desrespeitadas em um dos indivíduos (listas). Para solucionar esse problema no recorte 'IX' o bloco I é movido para a ordem plausível mais próxima. O último arranjo apresenta o resultado final do crossover.

Na Figura 3.11 estão representadas todas as etapas do processo da metodologia

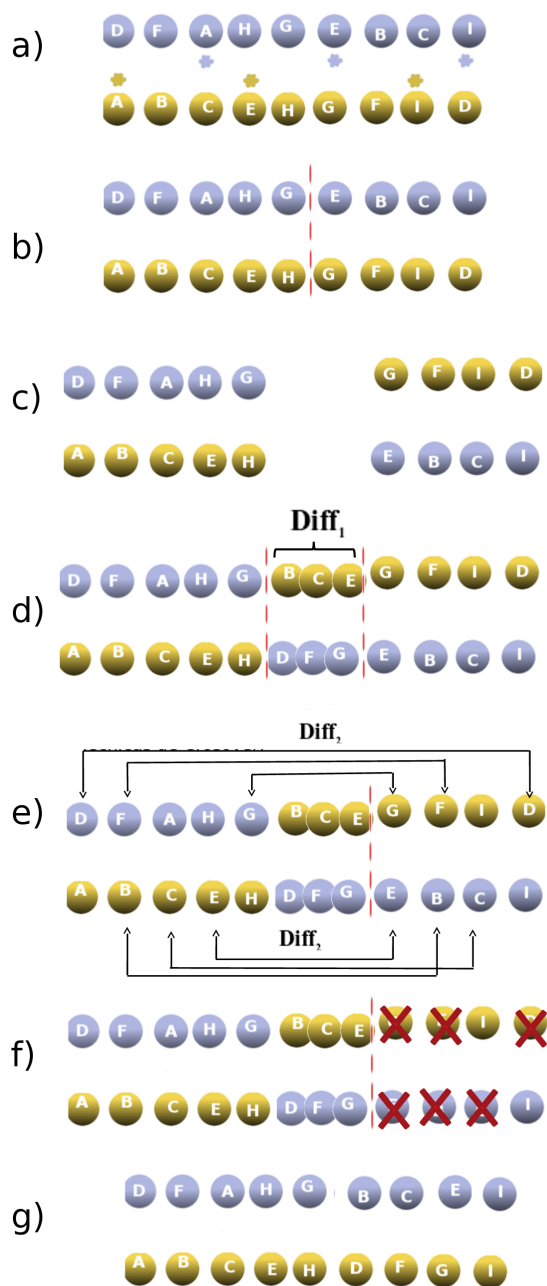


Figura 3.11: Etapas do método do crossover implementado

final (implementada). Vale ressaltar que dentro desse processo existem varias etapas análogas ao exemplo anterior ( etapas "a","b"e "c"são as etapas "I", "II"e "III"). Porém a definição dos conjuntos  $Diff_1$  e  $Diff_2$  permitem simplificar o número de comparações necessárias. O Algoritmo definido a seguir representa o método apresentado, em sequencia será discutido o método na análise do algoritmo.

---

**Algorithm 2** Algoritmo de Crossover
 

---

```

1: procedure CROSSOVER( $Cromossomo_1, Cromossomo_2$ )
2:    $n \leftarrow randomIntegerBetween(1, 69)$ 
3:    $Cromo_1^a, Cromo_1^b \leftarrow split(n, Cromossomo_1)$  ▷ Divide a lista em duas
4:    $Cromo_2^a, Cromo_2^b \leftarrow split(n, Cromossomo_2)$ 
5:    $Diff_1^a \leftarrow \{x \mid x \in Cromo_2^a \wedge (\neg Cromo_1^a)\}$ 
6:    $Diff_2^a \leftarrow \{x \mid x \in Cromo_1^a \wedge (\neg Cromo_2^a)\}$ 
7:    $NovoCromo_1 \leftarrow Cromo_1^a + Diff_1^a$ 
8:    $NovoCromo_2 \leftarrow Cromo_2^a + Diff_2^a$ 
9:    $Diff_1^b \leftarrow \{x \mid x \in Cromo_1^b \wedge (\neg Diff_1^a)\}$ 
10:   $Diff_2^b \leftarrow \{x \mid x \in Cromo_2^b \wedge (\neg Diff_2^a)\}$ 
11:   $NovoCromo_1 \leftarrow NovoCromo_1 + Diff_1^b$ 
12:   $NovoCromo_2 \leftarrow NovoCromo_2 + Diff_2^b$ 
13:  return  $NovoCromo_1 \& NovoCromo_2$  ▷ The gcd is b

```

---

O Algoritmo 2 foi desenvolvido para a implementação da técnica de crossover no problema de edificação naval. Basicamente o algoritmo pode ser interpretado como, a partir de duas ordenações iniciais ( $Cromossomo_1$  e  $Cromossomo_2$ ), são construídas duas novas ordenações ( $NovoCromo_1$  e  $NovoCromo_2$ ) que sejam uma recombinação aleatória das duas primeiras, ponderadas pelo índice " $n$ " de maneira que se garanta que a reordenação se dará apenas após a  $n$ -ésima posição da ordem de blocos.

A discussão do algoritmo é a seguinte: Seja escolhido um valor aleatório entre "1" e "69" (número total de blocos menos um), as listas de blocos  $Cromossomo_1$  e  $Cromossomo_2$  são divididas de maneira que os  $n$  primeiro indivíduos de  $Cromossomo_1$  são reagrupados em uma nova lista denominada  $Cromo_1^a$  ( Análogo para  $Cromossomo_2$

e  $Cromo_2^a$ ). De forma semelhante os "70 - n" blocos seguintes são reagrupados na lista  $Cromo_1^b$  ( $Cromo_2^b$ ). Nesse processo não se altera a ordem relativa dos blocos nas listas (isso vale para todos os processos do algoritmo). A partir daí é feita uma comparação entre essas listas. Todos os elementos que fazem parte da  $Cromo_2^a$ , mas não estiverem em  $Cromo_1^a$ , são adicionados à uma lista chamada  $Dif_1^a$ ; Ao contrário, todos os elementos que fazem parte da  $Cromo_1^a$ , mas não estiverem em  $Cromo_2^a$ , são reunidos na lista  $Dif_2^a$ . De maneira análoga, se um bloco está na lista  $Cromo_1^b$  ou na lista  $Cromo_2^b$ , porém não está em  $Dif_1^a$  e  $Dif_2^a$ , respectivamente, é adicionado a uma lista para completar o conjunto dos blocos da ordenação "1" ou "2" (essas listas são denominadas " $Dif_1^b$ " e " $Dif_2^b$ "). Por fim, a composição ordenada  $Cromo_1^a$ ,  $Dif_1^a$  e  $Dif_1^b$  forma o resultado  $NovoCromossomo_1$  e  $Cromo_2^a$ ,  $Dif_2^a$  e  $Dif_2^b$  forma o resultado  $NovoCromossomo_2$ .

### 3.6 Mutação

O processo de mutação é muito importante porque ela permite obter respostas que não poderiam ser construídas a partir de recombinações do espaço inicial, ampliando assim o poder de busca do método. Enquanto a seleção destrói as respostas menos "aptas", e ao fazer isso ele reduz a diversidade da associação genética (genetic pool), o operador de mutação é responsável por manter a variabilidade entre as gerações, evitando assim problemas como a convergência prematura.

Normalmente operadores de mutação mudam o valor de um único alelo, uma vez que a ideia da mutação é uma mudança mínima que gera a variação genética, mas, no nosso problema, isso não faz sentido, uma vez que ordenação é uma medida relativa. Alterar a posição de um bloco na ordenação significa alterar a posição de, no mínimo, um outro bloco. Assim, dentro do conceito proposto de cromossomo como transcrição da ordenação numa lista, uma mutação deve ser vista com uma alteração aleatória na ordem dos blocos. Ao mesmo tempo é necessário garantir que a "ordem mutante" não deixe de ser uma ordenação factível (I).

Foi então necessário propor um operador de mutação, que busca "movimen-

tar um bloco na "fila de blocos" (alteração mínima) de forma a fazer o mínimo de verificações possíveis.

Seja um cromossomo uma lista indexada de  $N$  blocos ordenados. Se escolhermos uma posição na ordenação aleatoriamente teremos um bloco correspondente a essa posição. Se escolhermos outra posição, de forma aleatória, para levar o bloco inicial a essas posições, existe a possibilidade de isso acarretar numa ordenação que desrespeita as regras de precedência. Por outro lado, é possível provar que, dado um problema de ordenação com regras de precedência, caso exista mais de uma solução não trivial, existe sempre um bloco em uma posição  $(p)$ , para o qual se pode definir uma região  $(\delta)$  de todas as posições que podem receber o bloco em  $p$  sem desrespeitar as regras de precedência. É possível provar também que essa região é única e que fica nos arredores de  $p$ . Ou seja, se encontramos uma maneira de identificar essa região para um bloco qualquer podemos trocar sua posição dentro dessa região, e assim garantir a funcionalidade do operador de mutação. As regras de precedência podem ser escritas como se cada bloco tivesse uma lista de blocos que devem ser edificados anteriormente e posteriormente a ele. Se partimos do bloco sorteado podemos realizar uma busca para encontrar o primeiro bloco, sucessor ao sorteado, que está na lista de blocos posteriores na ordem de edificação. Por outro lado se começarmos uma busca para trás podemos encontrar o "primeiro" bloco antecessor que está na lista para ser edificado antes. Estes dois blocos determinam os limites da região  $\delta$ .

Assim, o processo de mutação se dá da seguinte forma, escolhe-se uma posição de bloco  $p$ , delimita-se  $\delta(p)$ , se  $\delta(p)$  existir, sorteamos uma nova posição  $p_n$  e o bloco que estava em  $p$  é movido para  $p_n$ . Caso contrário, escolhesse uma nova posição aleatoriamente e repete-se o processo.

### 3.7 Convergência

Um parâmetro muito importante dentro do A.G. é o método de parada. Os A.G.s são métodos de busca heurística, isso implica que, a priori o método desconhece se em um determinado momento qualquer ele está próximo ou distante de alcançar o

"ótimo"do problema. Virtualmente seria possível deixar o A.G. rodando infinitamente e o processo sempre buscaria soluções melhores, embora não seja possível garantir que dar ao método mais tempo leva, necessariamente a melhores resultados. Porém é certo que quando se implementa um método como esse é necessário estabelecer um critério de parada (ou convergência).

Nesse trabalho, foram realizadas muitas rodadas de testes exploratórias para se determinar as potencialidades e limitações do método, sendo que muitas vezes o autor já tinha um conhecimento prévio, ou ao menos uma hipótese sobre o resultado esperado. Em tais situações o próprio resultado esperado foi estabelecido como critério de parada. As ordenações de referência foram utilizadas varias vezes para estimar o resultado esperado e definir o critério de parada.

Outra critério que foi definido com default do sistema foi um limite máximo de 200 gerações. Esse tipo de gatilho é importante, principalmente na parte de debug e profile dos algoritmos, porque evita uma execução indefinida do algoritmo. Em um caso específico esse limite foi estendido para 800 gerações, mas isso será discutido no capítulo de resultados.



Por fim, em casos onde não se conhece a priori a ordem da resposta um critério de convergência precisa ser adotado. O critério era de que caso o resultado do individuo mais hábil (melhor fitness) se mantivesse constante por vinte gerações o algoritmo era interrompido. Neste caso o número vinte foi arbitrário, chegou-se a este número percebendo que números maiores não levavam a resultado melhores, por outro lado números menores resultaram em um interrompimento quando havia ainda diversas soluções diferentes dentro da ultima geração.

# Capítulo 4

## Análise dos Resultados

Alguns resultados já foram discutidos no capítulo anterior, mas esse capítulo discute a aplicabilidade da técnica em problemas do mundo real. A maioria dos estaleiros trabalha com um único guindaste na sua linha de edificação, a busca da melhor ordenação de tarefas quando só existe um ponto de atendimento no sistema se diferencia do caso geral porque, supondo o tempo das atividades independentes, não é possível diminuir o tempo total de trabalho. Nesse sentido, existem outras métricas relevantes e normalmente aplicadas para definir a "melhor ordem". Busca-se minimizar o atraso, minimizar o tempo de espera, minimizar custos ou algum outro tipo de otimização baseada em pesos para as tarefas [33].

Talvez nesse ponto seja importante lembrar as hipóteses básicas do algoritmo genético que estamos implementando. Todas as construções são factíveis, ou seja, respeitam as restrições de precedência entre os tipos de bloco em cada região da construção. Em todo caso é adicionada uma função de penalidade (a função de fitness), que impede que apareçam construções descontínuas. Quando não há informação mais precisa, é suposto que o tempo de edificação de todos os blocos é contante e igual a uma unidade de tempo.



## 4.1 Minimização da ocupação de um dique

Melhorar a taxa de utilização do espaço é uma aplicação de interesse dentro da construção de navios [34]. Por exemplo, num dique seco pequeno onde estão sendo edificados dois navios simultaneamente, a sequência de construção ocorre de maneira que exista um navio mais avançado (que foi iniciado primeiro e se encontra mais perto das comportas) e um nos estágios iniciais da construção. A medida que a construção do navio mais recente avança, ela demandará uma região do dique cada vez maior, até que chegará um momento quando não será mais possível continuar com as duas embarcações naquele espaço. Nesse momento é necessário que as comportas sejam abertas e então o primeiro navio é retirado e é finalizado fora do dique e a segunda passa para onde estava o local do primeiro e uma nova edificação é iniciada. Num caso como esse, não se deseja minimizar o tempo, mas sim otimizar a ocupação do dique.

Esse problema pode ser entendido como a busca pela ordenação que seja longitudinalmente mais compacta possível, ou seja, que, a cada momento da edificação, ocupe o menor espaço do dique possível. Para podermos aplicar o algoritmo genético num problema como esse devemos definir uma função de mérito que capture o feito da ocupação do dique. A seguir é proposta a equação 4.1 aonde " $O(C)$ " é a ocupação de uma ordenação definida pelo cromossomo " $C$ ".



$$\sum_{i=1}^N T(b_i) * s(C, b_i) = O(C) \quad (4.1)$$

A Equação 4.1 é a função de fitness proposta para minimizar a ocupação do dique, onde " $T(b)$ " é o tempo necessário para edificar o bloco " $b$ " (contando a penalidade) e " $s(C, b)$ " é o comprimento total edificado até aquele bloco (inclusive). Ou seja, a função ocupação é um produto do comprimento edificado pelo tempo da edificação.

Se aplicarmos essa função às sequências de edificação de referência (resultados sem algoritmo genético) obteremos os primeiros resultados apresentados na Tabela 4.1. É possível perceber que a ordenação de grandes blocos se destaca como

tendo menor *score* entre as três ordenações selecionadas para teste. Esses resultados são coerentes com esperado uma vez que um dos principais motivos para adotar a construção em grandes blocos é justamente para reduzir espaço ocupado durante a edificação. Também podemos observar na Tabela 4.1 que o resultado obtido utilizando A.G. (9300391,39) praticamente coincide com o resultado anterior o que serve como validação do método.

Tabela 4.1: Resultados para Função de Ocupação

Index	Edificação	Score = $O(C)$	
1	Grandes Blocos	8793.99	
2	Camadas	12173.3	
3	Pirâmide	9736.09	
4	Resultado A.G.	9039.39	

A convergência para esse resultado foi particularmente difícil, para se obter o resultado próximo ao esperado o limite de número de gerações (normalmente 200 gerações) precisou ser alterado, a convergência do algoritmo genético se deu por critério de variabilidade genética, ou seja, o A.G. interrompeu os ciclos de gerações quando todos os indivíduos da população convergiram para a mesma genética. O resultado dessa otimização é apresentado em mapa de cor na Fig. 4.1. Como é possível perceber o resultado é bem próximo de uma construção em grandes blocos, embora o algoritmo tenha convergido antes de encontrar o resultado ótimo. Isso pode ser explicado pelas características do espaço de resposta que privilegiam resultados horizontalizados, como já foi discutido nas seções anteriores desse trabalho.

#### 4.1.1 **Integração** da construção ao modelo

Como já discutido nas seções anteriores, não está dentro do escopo desse trabalho aplicar a otimização de todas as tarefas que compõem a construção da embarcação. É de especial interesse o estudo da ordem de edificação porque os blocos são as maiores unidades construtivas, e por isso é necessário planejar a construção para que haja pouco, ou nem um, estoque de blocos. Porém, muitas das vezes, a ordem de fabricação dos blocos é determinada em função da capacidade construtiva do

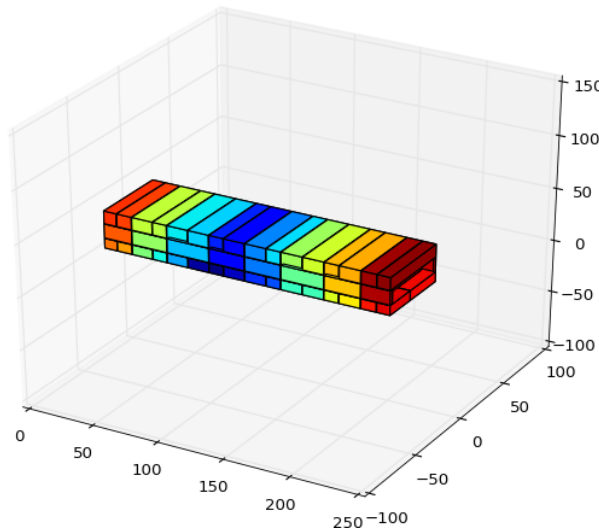


Figura 4.1: Resultado da otimização de ocupação.

estaleiro, e para poder analisar este efeito dentro do método é necessário propor um meio de levar em conta a correlação entre os tempos de fabricação dos blocos e a ordem de edificação.

Não foi encontrada muita informação sobre o efeito da ordem de edificação no tempo total da construção do navio. Normalmente o problema é tratado por simulação numérica: Sendo dado uma ordem (ou conjunto de ordens) de edificação definida, como tornar mais eficiente as fases anteriores como a pré edificação, montagem, e solda. Essa abordagem é válida, mas nem sempre a ordem de edificação é imposta, e a simulação de todo o processo de fabricação, mesmo sendo uma ferramenta muito poderosa, requer um tempo de processamento que não é compatível com um algoritmo de otimização por busca (em termos de uma função de fitness).

Para de alguma forma relacionar o tempo total de edificação com a ordenação dos blocos será proposta uma matriz de correlação. A ideia é de que se a ordem de edificação entre os blocos afeta o tempo de edificação, então isso pode ser, na hipótese mais simples, quantificado por um fator.

Para exemplificar essa proposta podemos supor um caso simples. Digamos que durante a parte de planejamento de construção se chegou à conclusão de que a construção de blocos do mesmo tipo permite o planejamento de uma produção mais

eficiente e por isso reduz o tempo de construção do bloco.

A interpretação de um caso como esse pode ser de que, numa situação normal todos os blocos teriam aproximadamente o mesmo tempo de edificação e de espera para a disponibilização do bloco (por exemplo, uma unidade de tempo para cada). Porém, devido a um ganho de eficiência, relacionado a repetição, quando é realizada uma edificação de um bloco do mesmo tipo do bloco anterior o tempo total de edificação desse novo bloco é menor, estando disponível para edificação assim que necessário.

Nesse tipo de problema, quando o tempo total de edificação de cada bloco pode ser relacionado ao do último construído/edificado, o conceito de matriz de correlação pode ser muito útil. A matriz de correlação é simplesmente uma matriz de fatores de correlação de tempo onde cada linha ou coluna se refere um tipo de bloco. Se, no exemplo anterior, for suposto que tempo total de edificação cai a metade do tempo médio, quando são edificados blocos do mesmo tipo em sequencia. Teremos a seguinte matriz:

$$\mathbf{M}_{\text{cor}} = \begin{matrix} & \begin{matrix} fundo & cofferdam & costado & conves \end{matrix} \\ \begin{pmatrix} 1 & 2 & 2 & 2 \\ 2 & 1 & 2 & 2 \\ 2 & 2 & 1 & 2 \\ 2 & 2 & 2 & 1 \end{pmatrix} & \begin{matrix} fundo \\ cofferdam \\ costado \\ conves \end{matrix} \end{matrix} \quad (4.2)$$

A leitura da Matriz 4.2 deve ser: Se o último bloco a ser edificado foi do tipo  $i$  e próximo a ser edificado é do tipo  $j$ , o tempo de edificação desse ( $t_b$ ) é o tempo médio de edificação desse tipo de bloco  $\bar{t}_b$  multiplicado pelo fator  $m_{ij}$ . Essa relação pode ser transcrita como

$$t_b = \bar{t}_b * m_{ij} \quad (4.3)$$

Assim se o ultimo bloco que foi edificado era do tipo costado (linha 3) e o novo

é do tipo convés (coluna 4) o tempo total de edificação do bloco de convés é uma multiplicação do tempo de edificação para esse bloco (nas nossas implementações uma u.t.) pelo fato  $m_{34}$  da matriz (que no exemplo acima é "2", porque blocos de tempos diferentes não se beneficiam). Ou seja o tempo de edificação nesse caso seria de duas unidade de tempo.



Aqui foi introduzido também um conceito de contagem de tempo diferente do apresentado até então. Inicialmente, nesse trabalho, o tempo de edificação era a soma dos tempos e movimentação do guindaste, de posicionamento do bloco e da soldagem inicial, até que a movimentação para o próximo bloco pudesse começar. Dessa maneira o problema tratado até aqui é o de que, por hipótese, todos os blocos estão disponíveis a priori, e assim o trabalho se resume a união desses blocos dentro do plano de edificação. O interessante da introdução da matriz de correlação nesse trabalho é que ela permite um primeiro passo, embora muito simplificado, na direção de relaxar essa hipótese num modelo acoplado a otimização por algoritmo genético. Se assumirmos que os blocos são construídos ao mesmo tempo em que a embarcação é edificada pode-se pensar no tempo total de edificação como o tempo de edificação, da maneira anteriormente discutida, acrescido do tempo de espera pela disponibilidade do próximo bloco. Dessa maneira, é claro, o processo de edificação segue inicialmente as mesmas etapas já discutidas (movimentação, posicionamento e fixação), porém não necessariamente o próximo bloco estará disponível no momento em que a edificação acabar.

A introdução do fator de correlação explicita que o tempo de construção de um determinado bloco é fortemente influenciado pela disponibilidade dos agentes e recursos que serão empenhados na fabricação dele bloco e a ocupação desses meios depende dos blocos (ou tipo de blocos) que foram edificados anteriormente, sobre tudo do ultimo bloco a ser edificado. A Fig. 4.2 ilustra esse fenômeno. É possível ver dois cronogramas de construção e edificação de dois blocos de tipos distintos, cada um com seus processos representados em uma cor, que são realizadas em 9 etapas, sendo a última a edificação dos blocos. Neles é possível perceber que cada bloco tem o mesmo tempo de edificação e mesmo tempo total de construção, porém alterando a ordem relativa entre eles o tempo total de construção e edificação do

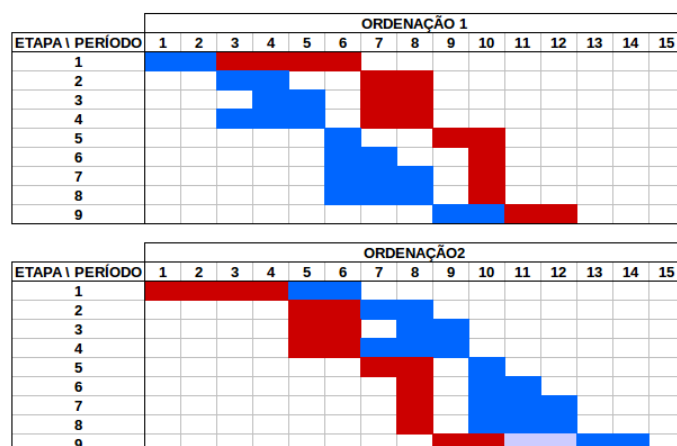


Figura 4.2: Ilustração de dois cronogramas de construção e edificação de dois blocos, apresentando o efeito da ordenação no tempo total de edificação

conjunto é alterado. Na figura também pode-se observar o tempo total de edificação (do segundo bloco): Na "ordenação 1" esse tempo vai do período 11 até o período 12, que é o próprio período de edificação, ou seja, é o fator de correlação nesse caso é 1. O mesmo raciocínio pode ser aplicado a "ordenação 2", o tempo total de edificação vai do período 11 ao 14, o que levará a um fator de correlação 2.

É claro que a pré edificação é composta por um grande número de processos e se levarmos em consideração todas as oficinas, diversos blocos são constantemente fabricados ao mesmo tempo (algumas vezes até de mais de uma embarcação) e todos eles influenciam mutuamente o tempo de construção uns dos outros. É claro, também, que a disponibilidade dos meios de produção e recursos não são os únicos fatores que influenciam no tempo de construção, mas também há a curva de aprendizado, a qualidade dimensional, o tamanho dos "buffers", quantidade de retrabalho, etc. Existe um sem número de fatores que podem ser ponderados e que tem influência no tempo total de edificação de cada bloco. Independente de tudo isso a análise feita adiante ainda tem o valor de um experimente exploratório, que facilmente pode ser expandido em complexidade e generalidade, e mesmo sendo uma simples "aproximação linear" permite chegar a resultados interessantes que serão apresentados a seguir.

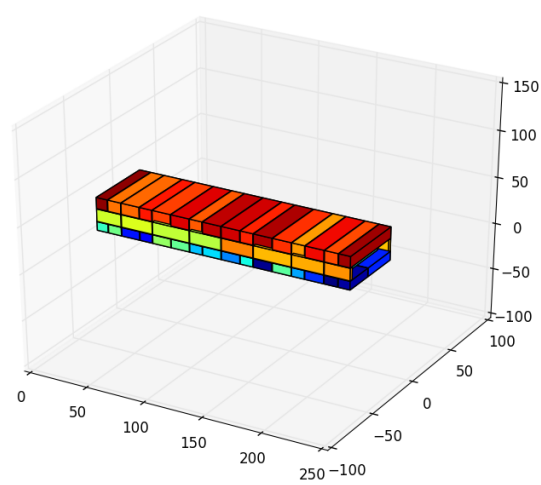


Figura 4.3: Resultado da otimização quando blocos de tipos iguais edificados em sequência são favorecidos (resultado obtido com função de penalização ainda constante).

4.2 Otimização para o caso de blocos similares

A Fig.4.3 mostra o resultado do A.G. com a implementação da Matriz. 4.2. Pode-se observar que o método convergiu para uma organização razoavelmente similar à construção em camadas. A execução com esse parâmetros obteve uma rápida convergência, convergindo para uma equivalente a ótima em menos de sessenta gerações, isso deve esta relacionado as características do método (que já foram discutidas na seção anterior). A Tab. 4.2 apresenta do A.G. e das as ordenações de referência (foi arbitrado as ordenações partindo do centro da embarcação).Esta organização obteve o mesmo score que a construção em camadas.

Tabela 4.2: Resultados para diferentes ordenações quando blocos de tipos iguais edificados em sequência são favorecidos

Index	Edificação	Score = Tempo
1	Grandes Blocos	96.0
2	Camadas	70.0
3	Pirâmide	87.0
4	Resultado A.G.	70.0

### 4.3 Otimização para o caso de blocos de regiões diferentes

De maneira análoga ao discutido anteriormente podemos pensar no caso contrário. A primeira parte da Fig.4.2 esquematiza um caso aonde de tipos diferentes de blocos, justamente por exigirem cargas de trabalho diferentes, em diferentes etapas do processo construtivo, interagem de maneira positiva para reduzir o tempo total de construção e edificação. Se adicionarmos o fato de que blocos diferentes podem ter etapas mutuamente exclusivas, ou seja, certas etapas do processo de fabricação de cada um é exclusivo a ele quando comparado com outro, é possível conceber um estaleiro que encontre vantagens na alternância dos tipos de blocos edificados. Assim vamos tratar o caso onde blocos de tipos diferentes interagem de maneira positiva (fator de correlação igual a um).

Para esse caso a matriz dos fatores de correlação é:

$$\mathbf{M}_{\text{cor}} = \begin{pmatrix} 2 & 1 & 1 & 1 \\ 1 & 2 & 1 & 1 \\ 1 & 1 & 2 & 1 \\ 1 & 1 & 1 & 2 \end{pmatrix} \begin{matrix} fundo \\ cofferdam \\ costado \\ conves \end{matrix} \quad (4.4)$$

A Fig. 4.4 mostra o resultado dessa nova otimização. Nesse caso o resultado da otimização superou consideravelmente o das edificações de referência, como é possível ver na tabela 4.5. Quando analisamos com mais cuidado esse resultado percebemos que o resultado do A.G. é uma configuração aonde existe uma "alternância quase que constante" entre os tipos de blocos na ordenação. Algo como, um lado do costado, depois um bloco de fundo, depois um de cofferdam e depois o outro do costado, etc.

Ao analisarmos esse resultado é possível notar que existe outro fator em comum em relação a todas as edificações de referência. Existem certos blocos que são sempre edificados juntos em todas as sequências base. Esses conjuntos de blocos são os blocos do mesmo tipo dentro de um grande anel, ou seja, todas as sequências edificam



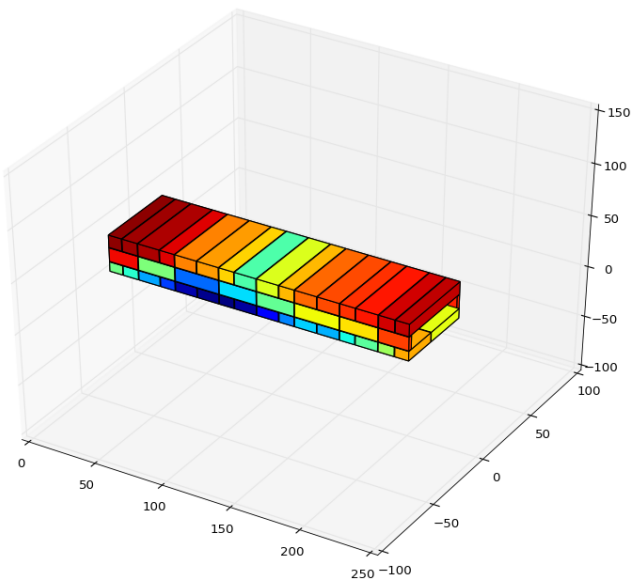


Figura 4.4: Resultado da otimização quando blocos de tipos diferentes edificados em sequência são favorecidos (resultado obtido com função de penalização não constate).

os dois lados do costado em sequência, os quatro blocos do fundo em sequência e os dois do topo em sequência.

Tabela 4.3: Resultados para diferentes ordenações quando blocos de tipos diferentes edificados em sequência são favorecidos

Index	Edificação	Score = Tempo
1	Grandes Blocos	110.0
2	Camadas	136.0
3	Pirâmide	119.0
4	Resultado A.G.	91.0



Essa é uma observação interessante uma vez que, caso isso fosse considerado como uma característica relevante para qualquer edificação ser considerada uma "edificação factível"(um dos objetivos perseguidos nesse trabalho) poderia ser proposto um outro modelo para tratar esses conjuntos como um único bloco e isso representaria uma significativa simplificação do modelo discutido nesse trabalho.

Apesar de ser possível escrever um novo modelo, decidiu-se tratar o problema de maneira diferente. Uma vez que o objetivo desse trabalho é, ao menos em parte, explorar as potencialidades do método de algoritmo genético foi adicionada uma

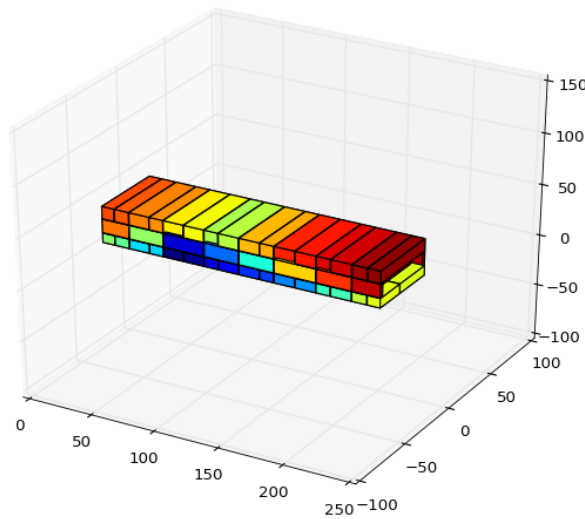


Figura 4.5: Resultado da quando blocos de tipos diferentes edificadas em sequência são favorecidos, resultado obtido com reagrupamento dos blocos por anéis gigantes.


alteração dentro do nosso modelo. Um novo operador foi adicionado e será executado depois da geração dos cromossomos, do crossover e da mutação que reordena os blocos, juntando todos os blocos de cada um dos grupos, porém mantendo a ordem relativa entre eles. A Fig.4.5 demonstra os resultados positivos que essa técnica teve (ou seja ouve convergência) apesar da otimização ter se tornado mais pesada.

Os resultados obtidos mais uma vez não foram parecidos com nenhuma edificação de referência, apesar de o método ter convergido para uma resposta equivalente a edificação em Grandes Blocos (o tempo de edificação desse novo resultado também foi cinquenta e cinco unidades de tempo, como é possível ver na Tab. 4.4 ). A resposta parece uma combinação entre a construção em grandes blocos e em pirâmide, isso pode ser explicado pela tendência do método de encontrar respostas horizontalizadas com maior facilidade.

#### 4.4 Otimização para o caso de blocos com relações arbitrárias da ordenação com o tempo de edificação

Já tratamos os casos onde a produção de blocos similares e blocos diferentes se relacionam em relação ao tempo de produção, agora busca-se generalizar esse

Tabela 4.4: Resultados para blocos de tipos (A.G. com reagrupamento dos blocos por anéis gigantes)

Index	Edificação	Score = Tempo	
1	Grandes Blocos	110.0	
2	Camadas	136.0	
3	Pirâmide	119.0	
4	Resultado A.G.	110.0	

conceito. Se assumirmos que a ordem de edificação influencia no tempo total de construção do navio e que isso pode ser explicado pela influencia da construção de um bloco em outro é razoável que qualquer bloco influencie o seguinte de alguma maneira que a priori é desconhecida. Dessa ideia surgiu a matriz de tempo de correlação. A matriz de correlação de tempo representaria essa influencia através de uma matriz de fatores onde as linhas são os blocos antecessores e as colunas os blocos sucessores.

$$\mathbf{M}_{\text{cor}} = \begin{pmatrix} \textit{fundo} & \textit{cofferdam} & \textit{costado} & \textit{conves} \\ 1.8358 & 1.4328 & 1.0283 & 1.0939 \\ 1.4454 & 1.7215 & 1.0021 & 1.7623 \\ 1.6516 & 1.7885 & 1.4495 & 1.4954 \\ 1.7638 & 1.2551 & 1.8474 & 1.1344 \end{pmatrix} \begin{matrix} \textit{fundo} \\ \textit{cofferdam} \\ \textit{costado} \\ \textit{conves} \end{matrix} \quad (4.5)$$

Baseado nesse conceito podemos tratar casos mais gerais. Para exemplificar as possibilidades geramos a Matriz 4.5, uma matriz de correlação geradas com fatores aleatórios entre um e dois.

A partir da Matriz 4.5, executamos o A.G. nos mesmos parâmetros que na segunda parte da Seção 4.3. Obtem-se os resultados apresentados na Fig.4.6 e na Tab. 4.5. Esses resultados são interessantes porque demonstram o potencial do método de aplicação em casos gerais.

Se imaginarmos um estaleiro com uma serie histórica dos dados da produção, seria possível regredir a correlação entre tempos de construção/edificação entre blo-

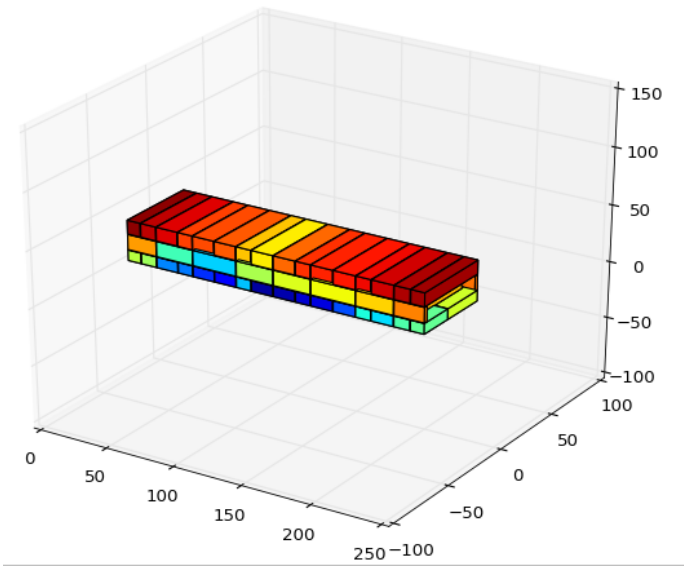


Figura 4.6: Resultado da otimização quando blocos tem relações arbitrárias, resultado obtido com reagrupamento dos blocos por anéis gigantes.

Tabela 4.5: Resultados para diferentes ordenações quando blocos de tipos diferentes edificados em sequência são favorecidos

Index	Edificação	Score
1	Grandes Blocos	96.70
2	Camadas	99.20
3	Pirâmide	96.55
4	Resultado A.G.	91.51



cos. Nesse trabalho foi proposta uma correlação entre blocos de diferentes regiões, porém essas relações podem ser de diversos tipos e o método acomoda esse tipo de alteração. Em um caso como esse seria possível facilmente aplicar um A.G. e descobrir a melhor sequência de edificação,ou ao menos uma hipótese inicial, para suportar a busca através de um métodos mais rebuscados.

## Capítulo 5

### Conclusões e Propostas



O algoritmo genético provou-se uma ferramenta útil para o planejamento das cronograma de edificação dos blocos de uma embarcação. Os resultados finais indicam que esse tipo modelo pode ser aplicado para problemas reais da construção naval.

O modelo desenvolvido demonstra o potencial do A.G., tanto como técnica para busca da melhor ordenação dos blocos para a edificação obtendo resultados melhores que as edificações amplamente empregadas na literatura, como uma ferramenta poderosa afim de gerar rapidamente resultados que suportem uma análise sobre a capacidade produtiva do estaleiro.

Apesar de não ser o foco desse trabalho uma boa parte do tempo de desenvolvimento do código foi empregado em tornar a execução mais do algoritmo mais eficiente. Todos os códigos foram implementado em Python 2.7 e executados em notebook com um processador Intel Core i3, com 3.8 Gb RAM, e o tempo médio de convergência era de 1:00 minuto para os casos mais simples chegando á 8:00 minutos para os casos mais difíceis.

Durante o desenvolvimento da metodologia foram feitas **muitas considerações e simplificações.** Existe, é claro, muito espaço para melhorias e a perfeiçoaementos no método, sobretudo em relação a função fitness, ou seja, qual a melhor maneira de avaliar a sequência de edificação e a representação da sequência através de um



cromossomo. Ainda vale ressaltar que, na visão do autor, a edificação é um caso de aplicação e que o A.G., como foi proposto, poderia facilmente ser expandido para as etapas anteriores da construção naval.

Todo o modelo foi desenvolvido através da criação de um extenso código computacional, bastando um clique para a execução de todo o algoritmo. Nesse trabalho são abordadas várias aplicações e a diversas vezes adições são propostas dentro da metodologia, isso é reflexo da construção modular do código o que permite que alterações sejam feitas com agilidade e simplicidade. Se for objetivado realizar alguma mudança significativas no tipo de análise como: alterações nas características do estaleiro, adicionar múltiplos guindaste trabalhando em paralelo ou até um buffer de blocos antes da edificação, etc. Para maioria das alterações bastaria alterar a função de mérito (função de fitness) e executar o programa para obter o resultado.

## 5.1 Propostas

Nesse trabalho foi proposta uma metodologia para aplicar algoritmo genético a varias casos do problema de edificação. Como aprimoramento da metodologia aqui apresentada poder-se-ia propor a criação de um modelo mais robusto do problema de edificação.

Adicionar as etapas de pré-edificação ao modelo seria uma extensão interessante. Isso poderia ser feito simplesmente "quebrando" cada nó, que representa um bloco, no modelo em uma serie de nós das etapas de construção anteriores. Esse representaria um grande avanço porque à medida que o modelo cresce para englobar mais etapas da construção, mais independente ele se torna de fatores que muitas vezes são desconhecidos sobre o sistema (como é caso dos fatores de correlação).

Um avanço para o método seria buscar novas representações genéticas para o problema de ordenação da construção. Dentro do problema de edificação utilizar a ordenação como cromossomo é interessante porque ela está diretamente ligada a natureza do problema, porém isso gera alguns problemas. Os operadores de crossover e mutação fazem alterações ao cromossomo, mas como a ordenação tem varias

restrições, toda vez que ordenação é alterada é necessário garantir que a nova ordenação está de acordo com todas as restrições, aumentando a complexidade dos operadores. Outro problema é que é simples gerar um conjunto de regras, que sejam validas durante todo processo de edificação, e que garantam uma "construção continuada". Esse problema também pode ser exemplificado como "Como combinar uma edificação de grandes blocos partindo da popa e edificação em grandes blocos partindo da proa em uma nova ordenação, que mantenha características em comum com as ordenações parentais e que seja continuada".

Uma proposta para de solução para esse problema seria abandonar a ordenação como representação cromossômica. Uma ordenação é mediada por todas as restrições do problema, por isso, não é possível julgar nenhum trecho da ordenação isolado do todo. Nesse sentido, olhar para ordenação é ter uma visão global do problema, porque o valor de cada alelo faz referência ao problema como todo. Uma outra abordagem seria, dado um estado inicial, qual a progressão que minha ordenação deveria tomar, ou seja, podemos pensar cada alelo como um indicador do próximo bloco (ou etapa) da construção em relação ao estado anterior. Nessa solução se perde a visão do todo, mas cada alelo indica a direção de transição entre duas etapas.

Existem várias maneiras de se implementar um método nesse molde, a que talvez seja mais intuitiva é de entender cada alelo como um vetor. Assim, a leitura de um cromossomo seria: dado um estado inicial, qual dos blocos que, poderiam ser edificados em seguida, melhor se adéqua a direção do vetor; escolhido um bloco temo-se um novo conjunto de possíveis blocos para edificação e assim seguimos ao próximo vetor, caso não seja possível definir qual bloco escolher um sorteio seria realizado.

Uma outra maneira de tratar esse problema seria imaginar os cromossomos como um conjunto de indicadores que, a partir de um "estado" da edificação, apontam para qual estado deve-se seguir. Por exemplo, como podemos ver na Fig. 5.1, foi construído um grafo para uma edificação de seis blocos, onde os blocos superiores se apoiam sobre os inferiores, ou seja, existe uma restrição de precedência entre eles. Na rede, é possível construir todas as edificações possíveis partindo do primeiro

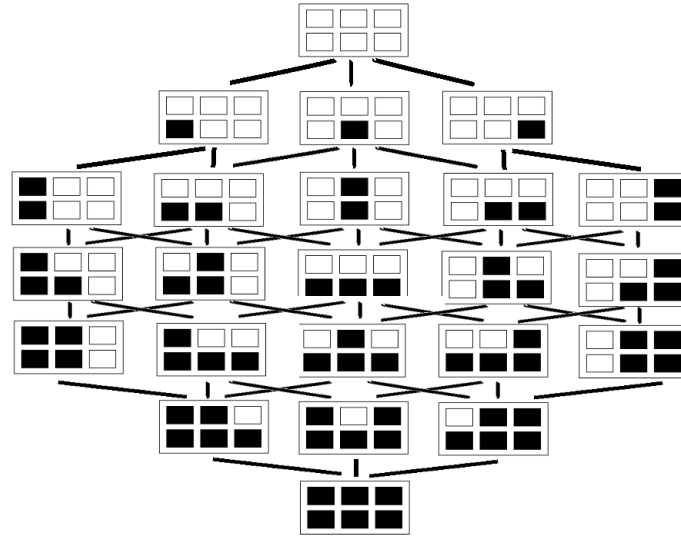


Figura 5.1: Diagrama de possibilidade para uma edificação simplificada.

estado e seguindo para os próximos através dos links. Assim, se estabelecermos uma probabilidade igual para cada link que sai de cada estado, podemos gerar uma ordenação qualquer através de uma caminhada aleatória entre os estados através dos links. Para poder implementar essa proposta, em cada estado poderia ser feito um sorteio de um indicador (entre 0 e 1) e a partir desse resultado seria decidido qual caminho seguir. Assim, o conjunto desses indicadores seria um possível cromossomo.

Existem desafios nesse tipo de propostas, porque uma vez que se abandona a ordenação como representação do espaço de resposta, a maneira de avaliar a edificação se torna mais complexa. No caso de um cromossomo de vetores como avaliar qual bloco está sendo indicado em cada estágio é um desafio. No caso do caminho aleatório, a construção da rede de estados, que para um caso real pode ser muito grande, é um desafio. Por outro lado, uma vez que se adota um caminho como esse os operadores de mutação e crossover se tornam triviais.

Por fim, na visão do autor a principal contribuição desse trabalho é a análise da metodologia proposta. Esta se deu em grande parte na tentativa de estudar situações onde as edificações conhecidas eram, intuitivamente, soluções ótimas e, com esses resultados como referência, visualizar a capacidade do método de alcançar essas respostas. Recomendo fortemente que esse tipo de análise seja feita sempre que um novo algoritmo genético para edificação seja proposto, por que permite um



grande enriquecimento do entendimento das propriedades do A.G. proposto. Apesar de não terem sido discutido, esse tipo de abordagem permite também calibrar os parâmetros do algoritmo como taxa de mutação, taxa de crossover, tamanho da população.

Há razão para acreditar também que exista espaço para trabalhos que visem otimizar os parâmetros do A.G. usando as edificações de referência. Outra abordagem interessante seria adicionar as ordenações de referencia na população inicial e testar se estas agem como um catalisador da convergência.

# Referências

- [1] Taggart, Robert, *Ship design and construction*, Society of Naval Architects & Marine Engineers, 1980.
- [2] Evans, J. Harvey, *Basic design concepts*, Naval Engineers Journal, v.71, n.4, p.671-678, 1959.
- [3] Storch, Richard Lee, Colin P. Hammon, and Howard M. Bunch, *Ship production*, 2nd edition, 1988.
- [4] Lima, Eriksom Teixeira, Mario Cordeiro de Carvalho Junior, and Luciano Otávio Marques de Velasco, *Removendo obstáculos às exportações brasileiras*, Revista do BNDES, 1998.
- [5] Goularti Filho, Alcides, *História econômica da construção naval no Brasil: formação de aglomerado e performance inovativa*, Revista de Economia, 2011.
- [6] de Jesus, Claudiana Guedes, and Robson Dias da Silva, *Trabalhadores a ver navios: reflexões sobre o mercado de trabalho na indústria naval na Região Metropolitana do Rio de Janeiro*, Cadernos Metrópole, ed.19, p.225-248, 2017.
- [7] Lamb, Thomas, *Group technology in shipbuilding.*, Journal of ship production, ed.4-1, p.30-50, 1988.
- [8] Caprace, Jean-David, et al. *Discrete event production simulation in shipyard workshops*, Proceeding of the 21th COPINAVAL, 2011.

- 
- [9] Chabane, Hamid, *Design of a small shipyard facility layout optimised for production and repair.*, Proceedings of Symposium International: Qualite et Maintenance au Service de l'Entreprise, 2004.
- [10] Pasin, Jorge Antonio Bozoti, *Indústria Naval do Brasil: panorama, desafios e perspectivas.*, Revista do BNDES, 2002.
- [11] Tokola, Henri, Esko Niemi, and Heikki Remes., *Block erection sequencing in shipbuilding with general lifting and joining times.*, Journal of Ship Production and Design, 2013.
- [12] Caprace, Jean-David, et al., *Discrete event production simulation and optimisation of ship block erection process.*, 10th International Conference on Computer Applications and Information Technology in the Maritime Industries, 2011.
- [13] Tokola, H. A., Assis, L. F., Freire, R. M., & Niemi, E., *OPTIMIZATION OF THE WELDING IN THE ERECTION SCHEDULING OF A SUEZMAX TANKER SHIP.*, 2013.
- [14] Koehn, Philipp, *Combining genetic algorithms and neural networks: The encoding problem.* 1994.
- [15] Okumoto, Yasuhisa., *Optimization of block erection using a genetic algorithm.*, Journal of ship production, v. 18, n. 2, p. 116-119, 2002.
- [16] Melanie, Mitchell. *An introduction to genetic algorithms.*, Cambridge,Massachusetts London, England, 1999.
- [17] Holland, John H. *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence.*, MIT press, 1999.
- [18] Haupt, Randy L., and Sue Ellen Haupt, *Practical genetic algorithms*, John Wiley e Sons, 2004.
- [19] Sastry, Kumara, David E. Goldberg, and Graham Kendall *Genetic algorithms*, Search methodologies. Springer US, p.93-117, 2014.

- 
- [20] Ghaheri, Ali, et al., *The applications of genetic algorithms in medicine*. Oman medical journal p.406 2015
- [21] Zhang, Stephen X., and Vladan Babovic *A Real Options Approach to the Design and Architecture of Water Supply Systems.*, 2012
- [22] Buurman, Joost, Stephen Zhang, and Vladan Babovic. *Reducing risk through real options in systems design: the case of architecting a maritime domain protection system*. Risk Analysis, 2009
- [23] Kornilakis, H., *Crew Pairing Optimization with Genetic Algorithms.*, 2002
- [24] Ghaheri, Ali, et al., *Optimized Shipyard Steel Plate Cutting*. EPPM, Singapore 2011
- [25] Werner, Frank, *Genetic algorithms for shop scheduling problems: a survey*. Preprint, p.31, 2011
- Deb, Kalyanmoy, et al. "A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II." *International Conference on Parallel Problem Solving From Nature*. Springer, Berlin, Heidelberg, 2000.
- [26] Deb, Kalyanmoy. *A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II*. *International Conference on Parallel Problem Solving From Nature*, Springer, 2000.
- [27] Davis, Lawrence. *Job shop scheduling with genetic algorithms.* *Proceedings of an international conference on genetic algorithms and their applications*. Vol. 140, 1985.
- [28] Davis, Lawrence, *Handbook of genetic algorithms*. 1991.
- [29] Todd, David, and Pratyush Sen, *Multiple criteria scheduling using genetic algorithms in a shipyard environment*. *Proceedings of the 9th international conference on computer applications in shipbuilding*, 1997
- [30] Brown, Alan, and Juan Salcedo, *Multiple-objective optimization in naval ship design*. *Naval Engineers Journal*, 2003

- 
- [31] Caprace, J-D., Frédéric Bair, and Philippe Rigo, *Scantling multi-objective optimisation of a LNG carrier*. Marine Structures, 2014.
- [32] Rigo, Philippe, Jérôme Matagne, and Jean-David Caprace *Least construction cost of FSO offshore structures and LNG gas carriers*. The fifteenth International Offshore and Polar Engineering Conference, 2005.
- [33] Lenstra, Jan Karel, AHG Rinnooy Kan, and Peter Brucker. *Complexity of machine scheduling problems*. Annals of discrete mathematics 1977.
- [34] Shang, Zhengyang, et al. *Spatial Scheduling Optimization Algorithm for Block Assembly in Shipbuilding*. Mathematical Problems in Engineering, 2017.
- [35] Baptista, Marcos Thadeu Palmeira. *Simulação de Processos de Construção Naval para Análise de Estratégias Alternativas de Edificação*. Dissertação Universidade Federal do Rio de Janeiro, 2013.
- [36] Sims, Karl. *Evolving virtual creatures*. Proceedings of the 21st annual conference on Computer graphics and interactive techniques, 1994.
- [37] Mittal, Mittu. *Comparison between BBO and Genetic Algorithm*. International Journal of Science, Engineering and Technology Research, v2.2, pp-284, 2013
- [38] Winston, Patrick H. *Lecture 13: Learning: Genetic Algorithms* Notas de aula, 2010, <http://ocw.polytechnic.edu.na/courses/electrical-engineering-and-computer-science/6-034-artificial-intelligence-fall-2010/lecture-videos/lecture-13-learning-genetic-algorithms/>
- [39] SCB Economic Intelligence Center, *Stepping on board Asia's shipbuilding industry*. Bangkok Post, 2014, <http://www.bangkokpost.com/print/429822/>