

Trabalho de Graduação

Curso de Graduação em Bacharelado em Ciências da Computação

DESENVOLVIMENTO DE SOLUÇÃO PARA O PROBLEMA DE ALOCAÇÃO EM
BERÇOS DE TERMINAIS PORTUÁRIOS: APLICAÇÃO DE ALGORITMOS
GENÉTICOS E OTIMIZAÇÃO POR NUVEM DE PARTÍCULAS

Larissa Cristina Moraes

Profª. Dra. Adriane Beatriz de Souza Serapião

Rio Claro (SP)

2013

Desenvolvimento de Solução para o Problema de Alocação de Navios em Berços de Terminais Portuários: Aplicação de Algoritmos Genéticos e Otimização por Nuvens de Partículas.

Trabalho de Conclusão do Curso, modalidade Trabalho de Graduação, apresentado, no 1º semestre de 2013, à disciplina ES/TG do Curso de Bacharelado em Ciências da Computação, período Integral, do Instituto de Geociências e Ciências Exatas da Universidade Estadual Paulista “Júlio de Mesquita Filho”, campus de Rio Claro, para apreciação segundo as normas estabelecidas pelo Conselho do Curso, em 27.11.2007.

Aluna: Larissa Cristina Moraes
Orientadora: Profa. Dra. Adriane Beatriz de Souza Serapião
DEMAC/IGCE

Rio Claro - SP
2013

519.72 Moraes, Larissa Cristina
M827d Desenvolvimento de solução para o problema de alocação de navios em berços de terminais portuários: aplicação de algoritmos genéticos e otimização por nuvens de partículas / Larissa Cristina Moraes. - Rio Claro, 2013
115 f. : il., figs., tabs.

Trabalho de conclusão de curso (bacharelado - Ciências da Computação) - Universidade Estadual Paulista, Instituto de Geociências e Ciências Exatas

Orientador: Adriane Beatriz de Souza Serapião

1. Programação linear. 2. Transporte marítimo. 3. Nuvem de partículas. I. Título.

Larissa Cristina Moraes

Desenvolvimento de Solução para o Problema de Alocação de Navios em Berços de Terminais Portuários: Aplicação de Algoritmos Genéticos e Otimização por Nuvens de Partículas.

Trabalho de Graduação apresentado ao Instituto de Geociências e Ciências Exatas - Campus de Rio Claro, da Universidade Estadual Paulista Júlio de Mesquita Filho, para obtenção do grau de Bacharel em Ciências da Computação.

Comissão Examinadora:

Adriane Beatriz de Souza Serapião (orientadora)

Daniel Carlos Guimarães Pedronette

Veronica Oliveira de Carvalho

Aprovado em: 03 de Julho de 2013.

DEDICATÓRIA

À Deus e à minha família, por me acompanharem nessa trajetória difícil, me darem força e acreditarem em minha capacidade, dedico-lhes essa conquista com muito amor e gratidão.

AGRADECIMENTOS

À Deus, pela minha vida e pela proteção recebida em todos os passos dessa minha caminhada.

Ao meu anjo da guarda especial por estar sempre me iluminando.

Aos meus pais, Carlos e Ana Paula, que são meus exemplos de vida. Obrigada por me incentivarem e acreditarem em mim, me dando forças em todos os momentos difíceis, me ensinando a caminhar sempre no caminho certo.

Aos meus irmãos, Thaís e Guilherme, por toda a alegria que trazem em minha vida, e também pelo amor e carinho que têm por mim.

À toda a minha família, por acreditarem em mim e torcerem pela minha vitória.

Ao meu namorado, Fernando, pela paciência e força nos momentos difíceis, e também por toda a ajuda que ele me deu nesse trabalho. Obrigada pelo carinho e amor sempre.

À minha orientadora, Adriane, que sempre acreditou no meu potencial e sempre esteve presente na minha trajetória, me ensinando e colaborando pra que esse trabalho fosse concluído.

À todos os meus professores da universidade, por todo o conhecimento que me passaram todos os anos do curso, contribuindo para que eu alcançasse esse título de bacharel.

Aos meus amigos que fizeram com que essa luta fosse mais tranquila e mais feliz.

Enfim, queria agradecer a todos que me ajudaram de alguma maneira para que esse sonho fosse realizado. Esse sonho não é só resultado da minha perseverança, dedicação, paciência e determinação, mas também do apoio de todos vocês. Deixo aqui, meus sinceros votos de agradecimento.

RESUMO

Esse trabalho apresenta a proposta de dois algoritmos de otimização para a solução do Problema de Alocação de Berços (PAB). Devido ao desenvolvimento econômico do país, se fez necessário o melhoramento dos meios de transporte, atingindo principalmente o transporte marítimo. Para isso, é necessário um melhor gerenciamento do sistema portuário, que receberá uma grande quantidade de navios que transportam cargas. Nesse trabalho, o PAB é abordado de forma que os objetivos sejam diminuir os custos e os tempos de movimentação nos portos. Para isso, foram aplicadas duas técnicas computacionais, os algoritmos genéticos e a otimização por nuvem de partículas, visando obter melhores resultados para esse problema. Os resultados obtidos com cada tipo de algoritmo são comparados para concluir qual método é mais eficiente para o sistema portuário.

Palavras-chave: Transporte Marítimo. Sistema Portuário. Problema de alocação de berços. Algoritmos Genéticos. Otimização por Nuvem de Partículas.

ABSTRACT

This work proposes two optimization algorithms for the solution of the Berth Allocation Problem (PAB). Due to the economic development of the country, it became necessary for the improvement of means of transport, which mainly shipping. For this, you need a better system management port, you will receive a lot of ships carrying cargo. In this work the PAB is approached so that the goals are to reduce costs and time handling in ports. For this, we applied two computational techniques, genetic algorithms and optimization for cloud particles, to obtain the best results for this problem. The results obtained with each type of algorithm are compared to conclude which method is more efficient for the port system.

Keywords: Shipping. Port System. Berth Allocation Problem. Genetic Algorithms. Particle Swarm Optimization.

LISTA DE FIGURAS

Figura 2.1: Exemplificação dos tempos de serviço, de atendimento e de espera	24
Figura 2.2: Exemplo de alocação dos navios nos berços	25
Figura 3.1: Pseudocódigo para o Algoritmo Genético	31
Figura 3.2: Cromossomo binário	32
Figura 3.3: Cromossomo por permutação	32
Figura 3.4: Cromossomo por valores	33
Figura 3.5: Cromossomo em árvore.....	33
Figura 3.6: Seleção por roleta	34
Figura 3.7: Taxa de adequação dos cromossomos antes e depois da classificação	35
Figura 3.8: Exemplo de <i>crossover</i> de 1 ponto	36
Figura 3.9: Exemplo de <i>crossover</i> de 2 pontos.....	36
Figura 3.10: Exemplo de cruzamento uniforme	36
Figura 3.11: Exemplo de cruzamento aritmético	37
Figura 3.12: Mutação para codificação binária	37
Figura 3.13: Exemplo do operador de mutação OX	37
Figura 3.14: Exemplo de mutação do tipo troca recíproca.....	38
Figura 3.15: Exemplo de mutação do tipo troca aleatória.....	38
Figura 3.16: Exemplo de mutação do tipo inversão.....	38
Figura 3.17: Exemplo de mutação do tipo inserção	38
Figura 3.18: Exemplo de mutação do tipo embaralhamento	39
Figura 3.19: <i>Crossover</i> para codificação por valores.....	39
Figura 3.20: Mutação para codificação por valores	39
Figura 3.21: Cruzamento para codificação em árvore	40
Figura 3.22: Topologias para o PSO	44
Figura 3.23: Pseudocódigo para o PSO.....	46
Figura 4.1: Elemento da Classe Navios	50
Figura 4.2: Elemento da Classe Berços.....	51
Figura 4.3: Representação de um cromossomo em uma lista de inteiros.	51
Figura 4.4: Fluxograma para o Algoritmo Genético	56
Figura 4.5: Fluxograma para o Algoritmo de Otimização por Nuvem de Partículas	57
Figura 5.1: Comparação dos operadores de <i>crossover</i>	59

Figura 5.2: Comparação dos operadores de mutação.....	59
Figura 5.3: Teste 6. Representação dos atendimentos para 20 navios, 5 berços, 50 indivíduos e 1000 gerações com AG, operador de <i>crossover</i> de 1 ponto e operador de mutação do tipo troca recíproca.....	61
Figura 5.4: Teste 15. Representação dos atendimentos para 20 navios, 10 berços, 50 indivíduos e 1000 gerações com AG, operador de <i>crossover</i> de 1 ponto e operador de mutação do tipo embaralhamento	61
Figura 5.5: Teste 27. Representação dos atendimentos para 30 navios, 5 berços, 100 indivíduos e 1000 gerações com AG, operador de <i>crossover</i> de 1 ponto e operador de mutação do tipo inversão	62
Figura 5.6: Teste 33. Representação dos atendimentos para 30 navios, 10 berços, 50 indivíduos e 1000 gerações com AG, operador de <i>crossover</i> de 1 ponto e operador de mutação do tipo inserção	62
Figura 5.7: Teste 39. Representação dos atendimentos para 30 navios, 20 berços, 20 indivíduos e 1000 gerações com AG, operador de <i>crossover</i> de 1 ponto e operador de mutação do tipo inversão	63
Figura 5.8: Teste 51. Representação dos atendimentos para 50 navios, 5 berços, 50 indivíduos e 1000 gerações com AG, operador de <i>crossover</i> de 1 ponto e operador de mutação do tipo troca recíproca	63
Figura 5.9: Teste 60. Representação dos atendimentos para 50 navios, 10 berços, 50 indivíduos e 1000 gerações com AG, operador de <i>crossover</i> de 1 ponto e operador de mutação do tipo inserção	64
Figura 5.10: Teste 66. Representação dos atendimentos para 50 navios, 20 berços, 50 indivíduos e 1000 gerações com AG, operador de <i>crossover</i> de 1 ponto e operador de mutação do tipo embaralhamento	65
Figura 5.11: Teste 78. Representação dos atendimentos para 50 navios, 30 berços, 50 indivíduos e 1000 gerações com AG, operador de <i>crossover</i> de 1 ponto e operador de mutação do tipo troca aleatória	66
Figura 5.12: Teste 84. Representação dos atendimentos para 100 navios, 5 berços, 20 indivíduos e 1000 gerações com AG, operador de <i>crossover</i> de 1 ponto e operador de mutação do tipo troca recíproca	67
Figura 5.13: Teste 105. Representação dos atendimentos para 100 navios, 10 berços, 20 indivíduos e 1000 gerações com AG, operador de <i>crossover</i> de 1 ponto e operador de mutação do tipo troca recíproca	68
Figura 5.14: Teste 111. Representação dos atendimentos para 100 navios, 20 berços, 20 indivíduos e 1000 gerações com AG, operador de <i>crossover</i> de 1 ponto e operador de mutação do tipo embaralhamento	69
Figura 5.15: Teste 123. Representação dos atendimentos para 100 navios, 30 berços, 50 indivíduos e 1000 gerações com AG, operador de <i>crossover</i> de 1 ponto e operador de mutação do tipo embaralhamento	70
Figura 5.16: Representação dos atendimentos para 20 navios, 5 berços, 100 partículas e 500 iterações com PSO.....	74

Figura 5.17: Representação dos atendimentos para 20 navios, 10 berços, 100 partículas e 1000 iterações com PSO.....	74
Figura 5.18: Representação dos atendimentos para 30 navios, 5 berços, 100 partículas e 100 iterações com PSO.....	75
Figura 5.19: Representação dos atendimentos para 30 navios, 10 berços, 100 partículas e 500 iterações com PSO.....	75
Figura 5.20: Representação dos atendimentos para 30 navios, 20 berços, 100 partículas e 1000 iterações com PSO.....	76
Figura 5.21: Representação dos atendimentos para 50 navios, 5 berços, 100 partículas e 100 iterações com PSO.....	77
Figura 5.22: Representação dos atendimentos para 50 navios, 10 berços, 100 partículas e 500 iterações com PSO.....	78
Figura 5.23: Representação dos atendimentos para 50 navios, 20 berços, 50 partículas e 100 iterações com PSO.....	79
Figura 5.24: Representação dos atendimentos para 50 navios, 30 berços, 100 partículas e 100 iterações com PSO.....	80
Figura 5.25: Representação dos atendimentos para 100 navios, 5 berços, 50 partículas e 1000 iterações com PSO.....	81
Figura 5.26: Representação dos atendimentos para 100 navios, 10 berços, 100 partículas e 1000 iterações com PSO.....	82
Figura 5.27: Representação dos atendimentos para 100 navios, 20 berços, 100 partículas e 1000 iterações com PSO.....	83
Figura 5.28: Representação dos atendimentos para 100 navios, 30 berços, 100 partículas e 100 iterações com PSO.....	84
Figura 5.29: Avaliação do desempenho computacional para 20 navios e 5 berços.....	87
Figura 5.30: Avaliação do desempenho computacional para 20 navios e 10 berços.....	87
Figura 5.31: Avaliação do desempenho computacional para 30 navios e 5 berços.....	87
Figura 5.32: Avaliação do desempenho computacional para 30 navios e 10 berços.....	88
Figura 5.33: Avaliação do desempenho computacional para 30 navios e 20 berços.....	88
Figura 5.34: Avaliação do desempenho computacional para 50 navios e 5 berços.....	88
Figura 5.35: Avaliação do desempenho computacional para 50 navios e 10 berços.....	89
Figura 5.36: Avaliação do desempenho computacional para 50 navios e 20 berços.....	89
Figura 5.37: Avaliação do desempenho computacional para 50 navios e 30 berços.....	89
Figura 5.38: Avaliação do desempenho computacional para 100 navios e 5 berços.....	90
Figura 5.39: Avaliação do desempenho computacional para 100 navios e 10 berços.....	90
Figura 5.40: Avaliação do desempenho computacional para 100 navios e 20 berços.....	90
Figura 5.41: Avaliação do desempenho computacional para 100 navios e 30 berços.....	91

LISTA DE TABELAS

Tabela 2.1 – Resumo do levantamento bibliográfico	22
Tabela 5.1: Melhores resultados obtidos para o algoritmo genético	60
Tabela 5.2: Resultados obtidos para o PSO	73
Tabela A.1: Base de dados de 20 navios	102
Tabela A.2: Base de dados de 30 navios	103
Tabela A.3: Base de dados de 50 navios	104
Tabela A.4: Base de dados de 100 navios	106
Tabela A.5: Base de dados de 5 berços	108
Tabela A.6: Base de dados de 10 berços	108
Tabela A.7: Base de dados de 20 berços	109
Tabela A.8: Base de dados de 30 berços	110
Tabela B: Melhores resultados variando-se o número de gerações e de indivíduos para o algoritmo genético	111

SUMÁRIO

	Página
LISTA DE FIGURAS	5
LISTA DE TABELAS.....	8
SUMÁRIO	9
1. INTRODUÇÃO	11
1.1 Apresentação do problema	11
1.2 Relevância do trabalho	12
1.3 Objetivos do trabalho	14
1.4 Organização do trabalho.....	14
2. PROBLEMA DE ALOCAÇÃO DE BERÇOS.....	16
2.1 Considerações iniciais	16
2.2 Revisão bibliográfica.....	19
2.3 Abordagem utilizada nesse projeto	22
2.4 Parâmetros para resolução do PAB.....	25
2.4.1 Custos (Navio e Porto)	25
2.4.2 Restrições	26
2.4.3 Variáveis de decisão (Quando e Quanto alocar cada navio)	27
3. ALGORITMOS EVOLUTIVOS E DE NUVENS DE PARTÍCULAS.....	28
3.1 Algoritmos Genéticos	28
3.1.1 Histórico	28
3.1.2 Conceitos iniciais	29
3.1.3 Funcionamento dos algoritmos genéticos.....	31
3.1.4 Tipos de codificação de um cromossomo	32
3.1.5 Operadores de seleção.....	34
3.1.6 Operadores de cruzamento e mutação	35
3.1.7 Parâmetros dos algoritmos genéticos	40
3.1.8 Codificação, Operadores e Parâmetros para o PAB.....	41

3.2	Otimização por Nuvens de Partículas	42
3.2.1	Histórico	42
3.2.2	Conceitos básicos	43
3.2.3	PSO discreto	47
3.2.4	PSO aplicado ao problema de alocação de berços	49
4.	MODELAGEM DO PROBLEMA DE ALOCAÇÃO DE BERÇOS PARA OTIMIZAÇÃO COM MÉTODOS EVOLUTIVOS	50
4.1	Implementação computacional.....	50
4.2	Representação dos cromossomos e das partículas.....	51
4.3	Operadores.....	52
4.3.1	Operadores genéticos	52
4.3.2	Operadores para otimização por nuvem de partículas	52
4.4	Funções utilizadas.....	53
4.4.1	Formulação matemática do PAB	53
4.4.2	Funções de Restrição	54
4.5	Funcionamento do Algoritmo Genético.....	55
4.6	Funcionamento do Algoritmo de Otimização por Nuvem de Partículas	56
5.	RESULTADOS	58
5.1	Avaliação do Algoritmo Genético	59
5.2	Avaliação do PSO	72
5.3	Avaliação do desempenho computacional dos algoritmos	86
5.4	Discussão.....	92
6	CONCLUSÃO	95
	REFERÊNCIAS BIBLIOGRÁFICAS	98
	APÊNDICE A	102
	APÊNDICE B	111

1 INTRODUÇÃO

1.1 APRESENTAÇÃO DO PROBLEMA

No setor marítimo consideramos os portos como verdadeiros eixos de atividades, pois são como nós de escoamento de mercadorias e é o espaço de intermediação entre o produtor e o mercado, sendo de extrema importância para esse tipo de transporte. Estes são confrontados com as exigências das transportadoras marítimas, em concorrência com os operadores de carga, que procuram um serviço de transporte mais rápido, mais seguro e mais barato. Portanto, consideramos o gerenciamento dos portos como uma questão a ser mais bem estudada para que esse modal de transporte cresça ainda mais e eleve a economia brasileira.

Os portos brasileiros sofreram uma completa reestruturação através da “Lei de Modernização dos Portos” (Lei nº 8.630/1993), visando torná-los mais ágeis e competitivos frente ao mercado internacional, visto que os custos operacionais são ainda muito superiores aos praticados no exterior. Através desta Lei, praticamente todos os serviços e estruturas até então operados pelo governo foram privatizados através de contratos ou arrendamentos.

Os navios de carga contribuíram para o crescimento do tráfego marítimo já que as principais companhias de logística focalizaram-se no tráfego de contêineres, que permite uma movimentação rápida e eficaz nos maiores portos. Hoje em dia, a produtividade da operação é condição fundamental para o atual sucesso da atividade portuária.

Os navios cresceram de porte e conseqüentemente a quantidade de cargas também, sendo necessário o crescimento do trabalho portuário para o embarque/desembarque das cargas. O uso da força humana passou a ser impossível para movimentação das cargas. Assim, estamos evoluindo para um estágio tecnológico, com uso de equipamentos cada vez mais potentes e velozes. Atualmente, o trabalho portuário é muito mais de inteligência, planejamento e preparo do que de força muscular.

No futuro, para além dos navios maiores, é de esperar que se façam progressos na velocidade, com a construção de navios mais rápidos, já que os operadores tentam reduzir os prazos de entrega. Os portos sofrem já uma pressão considerável, que os obriga a melhorar os tempos de rotação, assim como a qualidade e a fiabilidade das suas atividades, aumentando a eficiência dos serviços portuários.

É claro que o custo é um fator importante para os produtores e os transportadores, mas é também importante para os consumidores, que parecem preocupar-se mais com o preço de compra do que com o país de produção ou os meios de transporte utilizados. Como o consumidor se focaliza no preço, as empresas devem reduzir os seus custos de produção e de transporte para continuarem a ser competitivas.

Existe uma expectativa de melhoria da produtividade do sistema portuário, com investimentos da ordem de 3 bilhões de dólares na sua modernização face a redução de 25% a 30% no custo das máquinas e equipamentos para os portos. Essa redução de custos poderá ser obtida com o investimento em estudos tecnológicos que aperfeiçoem o gerenciamento dos portos.

1.2 RELEVÂNCIA DO TRABALHO

Os últimos 50 anos foram marcados por um crescimento extraordinário do comércio e, particularmente, do comércio internacional. Este crescimento, e a sua natureza global aumentaram as necessidades de transporte. O tráfego marítimo internacional apresenta um crescimento significativo desde os anos 60.

Estudos apontam que no Brasil o transporte marítimo é ao mesmo tempo um reflexo da economia e um fator que impulsiona o desenvolvimento econômico do país. Do ponto de vista econômico, o transporte tem uma tarefa bem específica a desempenhar no processo global de produção, na medida em que dá à mercadoria um acrescido valor econômico, através da aproximação da mercadoria aos centros utilizadores ou consumidores. E ao mesmo tempo a criação de oportunidade de negócio à escala mundial gera a necessidade de deslocação de grandes volumes de matérias primas e a consequente movimentação dos bens transformados e de consumo.

Este modo de transporte é imprescindível para a circulação de mercadorias e pode ser considerado o mecanismo de entrada do Brasil na economia internacional, devido ao grande número de importações e exportações realizado através do mesmo. O

transporte marítimo apresentou crescimento de 130% nos últimos trinta anos e, atualmente, representa 80% do comércio global.

A dinâmica do transporte marítimo potencializa o desenvolvimento econômico, pois permite o transporte de mercadorias entre todas as regiões do Brasil e também para outros países, fomenta a produção agropecuária e industrial assim como a extração de minérios no país e ao mesmo tempo gera empregos e renda à população brasileira.

O sistema marítimo brasileiro está concentrado nos portos do Sul e do Sudeste. Ou seja, estas regiões são responsáveis por grande parte das importações e exportações nacionais de longo curso, havendo uma concentração dos fluxos internacionais de cargas nestas duas macrorregiões do país. O transporte marítimo do Estado de São Paulo, por exemplo, movimenta produtos como grãos, farelos, açúcar e industrializados de diversas regiões do país e também de outros países, como Bolívia e Paraguai, e ao mesmo tempo recebem bens que seguem para os outros estados.

Podemos resumir dizendo que as principais características do transporte marítimo no Brasil são: capacidade de transporte de uma grande quantidade de cargas, sendo essas cargas de grandes tamanhos e de diversos tipos, baixo custo de transporte para grandes distâncias em relação aos outros tipos de transportes, pouca poluição ambiental, transporte lento e necessidade de portos. As primeiras características são vantagens que justificam o porquê da utilização do transporte marítimo no Brasil. Entretanto, as duas últimas características apresentam desvantagens que justificam porque esse tipo de transporte deve ser melhorado.

Também podemos citar a importância do transporte marítimo em outros países. Portugal, por exemplo, apresenta uma diferenciação positiva em relação aos outros países da Europa, devido ao fato de ser um país de uma longa tradição marítima, considerando que sua posição geográfica favorece o país para isso.

O transporte marítimo é essencial à economia europeia e ao comércio internacional, sendo que cerca de 90% do comércio externo da União Europeia e mais de 40% do seu comércio interno é realizado por via marítima. Só por si, estes fatos justificam o fato do transporte marítimo ser uma das soluções mais credíveis para a problemática da mobilidade sustentada e da qualidade ambiental. Portanto, podemos concluir que o transporte marítimo também tem uma importância crucial no desenvolvimento econômico da Europa.

Concluimos que os fatores de qualidade do transporte marítimo apresentam vantagem competitiva à capacidade de deslocar grandes quantidades de mercadorias a

grande distância em relação aos outros meios de transporte. Porém, ao mesmo tempo, apresenta como desvantagens o alto custo e a velocidade reduzida. Portanto, a importância desse modal de transporte justifica a necessidade de maiores investimentos e estudos para a melhoria do mesmo.

1.3 OBJETIVOS DO TRABALHO

Devido à importância do transporte marítimo no cerne econômico mundial e aos problemas encontrados no sistema de gerenciamento dos portos marítimos, este projeto visa desenvolver um sistema computacional que gerencie a atracação de navios no porto de forma otimizada para que os custos e o tempo total de serviço diminuam. Este sistema indicará qual a melhor alocação dos navios nos berços, considerando uma situação ótima quando a permanência de um navio no porto é mínima, a utilização de um berço é máxima e a soma das operações mecanizadas de carga/descarga com os custos obtidos pelos navios e berços também é mínima.

1.4 ORGANIZAÇÃO DO TRABALHO

Este relatório descreve o projeto que foi desenvolvido a partir da motivação apresentada acima. Para isso, foi estudado mais profundamente o problema de alocação de berços, que será apresentado no capítulo 2. Serão mostradas as diferentes abordagens que foram feitas para o problema de alocação de berços, destacando a abordagem que será utilizada nesse projeto. A partir do modelo escolhido, são apresentadas suas características, os objetivos para resolução do problema e a classificação do problema para as áreas de estudo de computação.

A partir do detalhamento do problema foram estudadas as técnicas de computação nas quais se baseiam o software desenvolvido. O estudo dos algoritmos evolutivos e de nuvens de partículas é apresentado no capítulo 3. Está detalhado nesse capítulo a ideia básica de cada algoritmo, o desenvolvimento histórico do mesmo, um algoritmo básico com os passos principais, a forma como são representadas as soluções

para cada tipo de problema, os operadores e parâmetros utilizados em cada algoritmo e possíveis observações.

Após termos a base do desenvolvimento do projeto, algumas modificações foram necessárias para que os algoritmos evolutivos e de nuvens de partículas gerassem bons resultados para o nosso problema. Com isso, temos a modelagem do problema de alocação de berços para aplicação de algoritmos de otimização evolutivos e de nuvens de partículas que se encontra no capítulo 4.

Com o software desenvolvido foram realizados vários testes que serão apresentados no capítulo 5. Inicialmente serão apresentadas as bases de dados que foram utilizadas para os testes. Então, serão apresentados os resultados para cada tipo de algoritmo, com os gráficos das soluções de cada um e também gráficos de comparação entre os resultados de ambos os algoritmos. A partir dos resultados apresentados serão descritas as discussões obtidas no trabalho, que se baseiam em vantagens e desvantagens de cada algoritmo particularmente e também a comparação da eficiência entre os dois algoritmos.

Para finalizar, trazemos então as conclusões obtidas com o trabalho realizado, apresentadas no capítulo 6. Nesse capítulo será concluída a importância desse trabalho, destacando quais as vantagens da realização do mesmo.

2 PROBLEMA DE ALOCAÇÃO DE BERÇOS

Devido à importância do transporte marítimo no cenário econômico mundial, foram estudadas suas características e seus problemas. Apesar de apresentar menor custo para exportação em relação aos outros meios de transporte, este ainda supera o tempo gasto em suas operações que os outros meios. Para melhorar a eficiência deste tipo de transporte, algumas modificações devem ser feitas no sistema portuário. Os problemas existentes no sistema portuário são estudados por diversos pesquisadores no mundo todo. Então, primeiramente serão apresentadas as diferentes abordagens existentes para o problema em questão, e, posteriormente, será detalhado o ponto de vista utilizado como base deste trabalho.

2.1 CONSIDERAÇÕES INICIAIS

Considerando que aproximadamente 80% do comércio global de mercadorias é realizado através do transporte marítimo, e que o mesmo vem apresentando crescimento desde o início da década, podemos prever que a quantidade de contêineres que deve ser descarregada nos terminais portuários está aumentando significativamente.

Devido ao fluxo intenso de carga e descarga de contêineres nos portos, a administração dos mesmos é obrigada a investir muito para acomodar todos os navios que chegam, seja aumentando o número de canais, seja melhorando os canais já existentes, ou comprando equipamentos para carga e descarga de mercadorias. Estas melhorias são realizadas para que o custo desse tipo de transporte e também o tempo de atendimento dos navios nos portos diminuam.

A partir desse cenário surgiu um problema conhecido na literatura como Problema de Alocação de Berços (PAB), em inglês, *Berth Allocation Problem* (BAP), que retrata a busca da melhor logística para que o sistema portuário atenda todos os navios que chegam ao porto ao longo do tempo, determinando o melhor berço a ser alocado pelo mesmo para que o atendimento seja realizado o mais rápido possível, resultando em um custo menor para os navios. As decisões a serem tomadas dizem

respeito à posição (onde?) e ao tempo (quando?) em que o navio deverá atracar (Imai *et al.*, 2001).

A Confederação Nacional de Transportes (CNT) define berço como sendo o “espaço destinado ao navio em um cais no qual ele pode operar com segurança”. Entendemos por alocação de berços nesse contexto quando os berços são atribuídos para os navios que aportam para a manipulação de cargas.

O PAB possui grande quantidade de restrições físicas, técnicas, entre outras. Isso faz com que seja possível modelá-lo de diferentes maneiras. Estas maneiras serão detalhadas abaixo.

Quanto aos aspectos espaciais dos berços, o PAB pode ser modelado como discreto, contínuo ou híbrido (Imai *et al.*, 2005). No caso discreto, o cais é dividido em vários berços e somente um navio é atendido de cada vez em cada berço, independente do seu tamanho. No caso contínuo, não há nenhuma divisão do cais e, dessa forma, os navios podem atracar em qualquer posição. Já no caso híbrido, como no caso discreto, o cais é dividido em berços, mas os navios grandes podem ocupar mais de uma posição, permitindo assim que navios pequenos compartilhem seu berço.

Além disso, se for levado em conta a chegada dos navios, o problema pode ser tratado como estático ou dinâmico (Imai *et al.*, 2001). O caso estático assume que todos os navios já estão no porto para o atendimento, o caso dinâmico permite aos navios chegarem a qualquer momento. Em ambos os casos, busca-se por uma melhor distribuição do espaço minimizando o tempo total de permanência dos navios no porto.

Segundo Nishimura *et al.* (2001), os berços nos grandes portos são utilizados pelos operadores dos navios para alcançar maior produtividade com o processamento de contêineres. Esse objetivo é alcançado com uma rigorosa manipulação de um grande volume de contêineres com uma grande quantidade de navios aportando, porém pode resultar em prejuízos se essas quantidades não forem suficientes, pois os custos do porto serão aumentados para cargas de volumes muito pequenos.

Neste contexto é interessante limitar o número de berços operantes. A minimização do tempo de alocação de berços se torna importante nesse contexto, devido ao fato do tempo de manipulação para um navio específico não ser o mesmo para cada berço.

De acordo com Paquette *et al.* (1982), a taxa de carga e descarga depende dos tipos das cargas, do tipo de embarcação e tamanho, da disponibilidade e tamanho da equipe de estivadores e do grau de mecanização e métodos de manipulação de cargas.

Estes acreditam que os engenheiros que realizam o planejamento dos portos devem obter previsões do número e tempo de distribuição da movimentação de navios para que possam antecipar o número de berços requeridos.

Portanto, as dimensões adequadas dos berços devem ser estipuladas para permitir ativa movimentação dos navios e atracação. Além disso, os espaços para carga e descarga de navios e também para abastecimento e suprimento dos mesmos devem ser estipulados. A profundidade do porto e a abordagem do canal devem ser suficientes para permitirem o completo carregamento de navios para navegarem seguramente. A largura de entrada necessária naturalmente será influenciada pelo tamanho do porto e dos navios que o utilizam. Como regra, esta deve ser aproximadamente igual ao comprimento do maior navio que utiliza tal porto.

O berço e espaço requerido para atracação variam de acordo com os tipos de navios que atracam no porto. Os espaços para os berços e o dimensionamento do porto dependem também dos procedimentos de manipulação de cargas e equipamentos. O comprimento de um berço deve ser igual ao comprimento do navio, adicionado ao espaço entre navios adjacentes e o espaço para as linhas dos navios. A forma e extensão da área do ancoradouro dependem principalmente do número máximo de navios a serem atendidos, do tamanho dos navios, do método de atracação, da necessidade de mobilidade e das condições topográficas do local proposto.

Ainda segundo Paquette *et al.* (1982), quando são definidas as características físicas do terminal portuário devem ser atendida a habilidade para realizar no mínimo três funções principais: carregar/descarregar cargas de navios com eficiência e rapidez; prover adequado sistema, temporário e de longo prazo, de armazenagem para cargas que chegam e partem e, prover conexões, seja ferroviária ou rodoviária, para movimento do frete dentro e fora da área do porto.

Enquanto que a capacidade prática do porto pode ser limitada por alguma dessas funções, a primeira é geralmente o fator de controle. Portanto, a capacidade de operação será essencialmente o produto da taxa de manipulação de cargas e o número e o grau de utilização dos berços.

Dessa forma, os gerentes da maioria dos portos buscam reduzir os custos através de uma eficiente utilização dos recursos incluindo recursos humanos, berços, área destinada aos contêineres, guindastes para movimentação dos contêineres e vários outros equipamentos. E, dentre todos estes recursos, os berços são os mais importantes recursos. Portanto, um planejamento e agendamento dos berços correto melhora a

satisfação dos consumidores e aumenta a movimentação nos portos, conduzindo aos altos rendimentos no porto.

2.2 REVISÃO BIBLIOGRÁFICA

O início dos trabalhos em relação ao PAB aconteceu com Thurman (1989) quando o mesmo apresentou um modelo de otimização para o planejamento de berços para a estação naval Norfolk (EUA). A partir desse modelo, Brown *et al.* (1994) o aprimoraram, apresentando um plano para minimizar os conflitos existentes entre carregamentos de navios diferentes nessa mesma estação naval. Estes mesmos autores propuseram ainda um planejamento de berços para submarinos em Brown *et al.* (1997).

De acordo com Brown *et al.* (1997) algumas mudanças, como requisição de mudanças de serviços, atrasos e chegadas antecipadas dos navios, são inevitáveis e fazem com que o plano de atracação já aprovado passe por algumas modificações. Com isso, é alta a frequência de revisões realizadas no plano e para evitar erros e, conseqüentemente, atrasos e custos por contas das alterações é necessário um planejamento da alocação de berços computadorizado (Brown *et al.*, 1994).

Brown *et al.* (1994) complementa o problema com outras variáveis, como limitações de comprimento do berço, profundidade no berço e força dos cabos utilizados para carga/descarga. Nishimura *et al.* (2001) acrescentam ainda que o navio que está planejando atracar em um berço não deve exceder a profundidade da água do berço.

Para Moon (2000) o problema consistia em determinar o momento e as posições de atracação de cada navio no terminal portuário. Para ele, o planejamento da atracação dos navios atribui embarcações da superfície a um berço antes da sua entrada no porto, ou reatribui navios que já estão no porto a realizar um avanço no treinamento de manutenção e eventos de certificação, os quais desenvolvem prontidão para futuras operações.

Conforme Moon (2000), cada navio necessita de uma quantidade específica de espaço no berço durante um pré-determinado período de tempo para descarregar e carregar os contêineres no porto. Para a realização desse processo existem diversas variáveis a serem consideradas, como a duração e o momento de chegada de cada navio,

o número de contêineres para descarregar e carregar, e a localização de estocagem dos contêineres a serem carregados nos respectivos navios para deixarem o porto.

Imai *et al.* (2001) apresentaram um método baseado na relaxação lagrangiana do problema original do PAB, abordando o problema em sua forma dinâmica e contínua, considerando as dimensões dos berços e dos navios.

Imai *et al.* (2003) alteraram sua abordagem inicial, considerando que existem prioridades no atendimento dos navios. Os autores propuseram também um Algoritmo Genético para gerar as soluções do problema.

Após serem determinadas as restrições que se deseja considerar, devemos notar que quando a taxa de chegadas de navios é alta ou quando chegadas inesperadas ocorrem com muita frequência então poderá ser impossível terminar os serviços para todos os navios no tempo previamente planejado. Dessa forma, alguns navios podem tardar na partida. Neste caso, existem alguns operadores de terminais portuários que possuem diferentes prioridades para diferentes tipos de navios, podendo haver penalizações de custo para os mesmos, segundo Kim e Moon (2003).

Outro fator muito importante é a posição de atracação no porto. Os contêineres que serão carregados para os navios geralmente chegam antes que o navio no porto. Então, se um navio é atracado em uma localização próxima de onde estão armazenados os contêineres, o custo do carregamento dos mesmos para os navios e a ser cobrado pelos caminhões ou outro equipamento de transporte interno do porto será menor, resultando na minimização dos custos.

De acordo com Kim e Moon (2003), algumas localizações de atracação são preferidas em relação às outras por diversos fatores, como contratos de longo período que especificam o uso dos berços com transportadores, o mínimo de espaço requerido para um navio e os diferentes níveis de ondas de acordo com as localizações no cais.

Conforme Guan e Cheung (2004) considera-se que o problema de alocar espaço nos berços para navios em terminais portuários é o problema de alocação de berços propriamente dito. Uma definição mais completa é que o PAB consiste em determinar o plano de alocação dos navios que atracam no porto aos berços, de modo que cada navio seja alocado em um berço em um período de tempo para realizar as atividades de carga e descarga de mercadorias com o intuito de reduzir os custos operacionais.

Devido ao espaço dos berços ser limitado e milhares de contêineres serem manipulados diariamente, uma efetiva alocação de berços torna-se essencial para a eficiência do gerenciamento do fluxo de tráfico de contêineres. Um típico berço de um

terminal portuário pode acomodar múltiplos navios ao mesmo tempo, segundo Guan e Cheung (2004).

Quando não existe espaço disponível no berço, o navio precisa aguardar para atracar. Para simplificar, Guan e Cheung (2004) denominam o tempo de fluxo de um navio como sendo a soma do tempo de espera e o tempo de processamento, ou atendimento, do mesmo.

Cordeau (2005) propuseram duas formulações matemáticas e duas heurísticas baseadas na Busca Tabu para resolver o PAB. Os autores apresentaram os resultados de testes realizados para o porto de Gioia Tauro (Itália).

Cheong *et al.* (2008) apresentam um modelo com a aplicação do método *Multiobjective Evolutionary Algorithm* (MOEA) para resolver o PAB.

No Brasil, Mauri *et al.* (2008a) propuseram uma abordagem baseada na aplicação do *Simulated Annealing* para resolução do caso discreto do PAB. Os autores tratam o problema como um Problema de Roteamento de Veículos com Múltiplas Garagens e Janelas de Tempo (PRVGMJT). Os resultados obtidos superaram os obtidos pelo CPLEX e pela Busca Tabu proposta por Cordeau *et al.* (2005). E mais tarde Mauri *et al.* (2008a) sugeriram uma metaheurística híbrida que utiliza o Algoritmo de Treinamento Populacional em conjunto com um modelo de Programação Linear por meio da técnica de Geração de Colunas. Os resultados obtidos superaram os apresentados em Mauri *et al.* (2008a).

Ainda no Brasil, Silva *et al.* (2008b) propuseram uma abordagem baseada na implementação de um método para resolução de problemas de otimização chamado de aplicação de algoritmos genéticos para a resolução do PAB. A implementação desse método foi baseada nos estudos apresentados por Imai *et al.* (1994, 2001, 2003, 2005) e Nishimura *et al.* (2001).

Mais tarde Giallombardo *et al.* (2010) apresentam um modelo de programação quadrática, um de programação linear, uma Busca Tabu e uma técnica de programação matemática para resolver o PAB utilizando instâncias baseadas em dados reais. E, por fim, Oliveira *et al.* (2010a) utilizaram os métodos *Clustering Search* com *Simulated Annealing* para resolver o PAB.

Um resumo do levantamento bibliográfico pode ser encontrado na Tabela 1.1, contendo o nome dos autores, o ano em que publicaram seus trabalhos, o método de resolução para o problema e o modelo do PAB utilizado.

Tabela 2.1 – Resumo do levantamento bibliográfico

Pesquisador	Ano	Método Utilizado	Modelo do PAB
Thurman	1989	Programação linear inteira	Discreto e estático
Brown et al.	1994, 1997	Programação matemática	Discreto e dinâmico
Moon	2000	Programação linear inteira	Contínuo e dinâmico
Nishimura et al.	2001	Algoritmo genético	Discreto e estático
Imai et al.	2003	Relaxação Lagrangiana	Contínuo e dinâmico
Kim e Moon	2003	Simulated Annealing	Contínuo e dinâmico
Guan e Cheung	2004	Simulated Annealing	Contínuo e dinâmico
Cordeau et al.	2005	Heurísticas baseadas na Busca Tabu	Discreto e dinâmico
Cheong et al.	2008	Método Multiobjective Evolutionary Algorithm	Discreto e estático
Mauri et al.	2008	Simulated Annealing e Algoritmo de Treinamento Populacional com a técnica de Geração de Colunas	Discreto e dinâmico
Silva et al.	2008	Algoritmos genéticos	Discreto e estático
Giallombardo et al.	2010	Programação quadrática, Programação linear, Busca Tabu e Programação matemática	Discreto e dinâmico
Oliveira et al.	2010	Clustering Search com Simulated Annealing	Discreto, contínuo e dinâmico

2.3 ABORDAGEM UTILIZADA NESTE TRABALHO

Seguindo a ideia de Imai *et al.* (2001, 2003), neste trabalho é proposta uma ferramenta alternativa para a solução do PAB, diferente das demais abordagens encontradas com relação às variáveis relacionadas aos custos de operação, tentando criar um modelo que aproxime mais da atividade praticada nos portos.

Além de ser diferentes das outras abordagens levantadas por essa pesquisa, o modelo apresentado também será diferente do apresentado por Nishimura *et al.* (2001). Nesse modelo, são realizadas divisões do problema em subproblemas, de forma que cada subproblema é resolvido individualmente pelo algoritmo genético e a solução do primeiro serve de base para o segundo, e assim por diante. Esse processo se repete até que todos os subproblemas sejam resolvidos. Nesse caso, a solução final pode ser influenciada pelas soluções intermediárias, resultando em um resultado não favorável ao problema. Neste trabalho isso não ocorre, ou seja, considera-se um único problema com todos os navios que necessitam atracar no porto.

Outra característica diferente do trabalho de Nishimura *et al.* (2001) é a forma como serão caracterizados os cromossomos. Neste trabalho o cromossomo é formado pela sequência de atracação dos navios nos portos através da análise da melhor posição de atracação para cada navio individual e separadamente. Na pesquisa dos autores citados acima, os cromossomos são criados de forma aleatória e a análise do *fitness*

somente é feita posteriormente. Neste trabalho, entretanto, consideram-se alguns parâmetros relevantes para a solução do PAB, sendo proposto um método de resolução de fácil implementação e que possa proporcionar bons resultados, tais como os utilizados por Silva *et al.* (2008b).

O modelo matemático e a função objetivo utilizados nesse projeto são as mesmas apresentadas no trabalho de Silva (2008). Porém, a implementação do algoritmo genético não seguiu os mesmos passos, havendo diferenças dos operadores genéticos. Inicialmente, a representação do cromossomo não é equivalente, já que no algoritmo aqui apresentado cada posição do vetor cromossomo indica qual berço será ocupado pelo navio indicado pelo índice do vetor, enquanto que no trabalho de Silva (2008), os cromossomos indicam uma sequência de navios com as posições ocupadas por eles. Os operadores de cruzamento do trabalho de Silva (2008) foram criados pelos pesquisadores, sendo diferenciado dos operadores aqui aplicados. Nesse trabalho, o cruzamento ocorre realizando uma média entre as posições ocupadas por cada navio, e a mutação é realizada de forma aleatória, mas não foi especificada pelos autores.

Além disso, após a implementação seguindo o modelo proposto para o algoritmo genético foi feita a implementação para o algoritmo de otimização por nuvem de partículas. Para a mesma as partículas do algoritmo serão representadas da mesma maneira que são representados os cromossomos, e têm o mesmo significado. Quanto às funções objetivo e as funções de restrição, estas serão as mesmas, pois o segundo algoritmo baseia-se na mesma abordagem para o PAB e, portanto, tem os mesmos objetivos que o primeiro. Já os operadores serão diferentes, baseados nos conceitos específicos de cada um dos algoritmos.

Baseado na abordagem de Guan e Cheung (2004), iremos considerar o tempo de serviço de um navio no berço como sendo a soma do tempo de espera com o tempo de atendimento do navio no berço. Estes tempos estão exemplificados para o navio 3 na Figura 2.1.

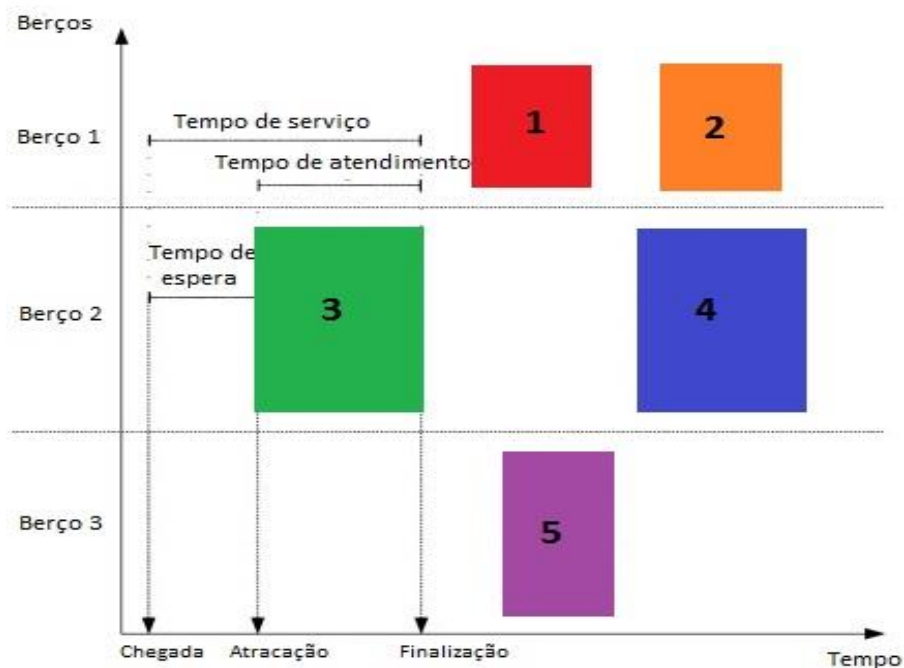


Figura 2.1 – Exemplificação dos tempos de serviço, de atendimento e de espera.

Outra característica que deve ser destacada para o projeto são as escolhas feitas para os aspectos espaciais dos berços e para os momentos de chegada dos navios. Para a abordagem proposta será considerado o modelo que contém o espaço de alocação discreto e o tempo de chegada dos navios estático. Dizemos que o espaço de alocação é discreto porque tem-se um número finito de berços com as dimensões definidas, que são comprimento e profundidade, assim como os navios que têm comprimento e calado definidos, e também consideramos que cada berço atenderá somente um navio de cada vez. E o tempo de chegada dos navios é estático porque o tempo de chegada de cada navio é determinado no início do problema, ou seja, não chegarão novos navios ao longo da resolução do problema.

Podemos exemplificar uma resolução para o nosso problema na Figura 2.2. Nela podemos observar que teremos 5 berços para alocar 20 navios. Temos então uma linha do tempo que determina qual navio vai alocar qual berço para cada período de tempo. Nesse gráfico ainda não temos a representação do tempo de espera, mas temos a representação do tempo de atendimento dos navios nos berços. Podemos notar que o berço 4, por exemplo, está o tempo todo alocado, com os navios 1, 7, 17, 8, 13 e 14, respectivamente. Ao tempo que o berço 1 só está sendo alocado pelos navios 11, 18 e 9, havendo períodos que o berço está vazio. Essa solução ainda não é ótima, porque alguns berços estão menos ocupados que os outros, mas podemos notar que as alocações nos

berços são finalizadas quase que ao mesmo tempo, destacando um aspecto positivo dessa alocação.

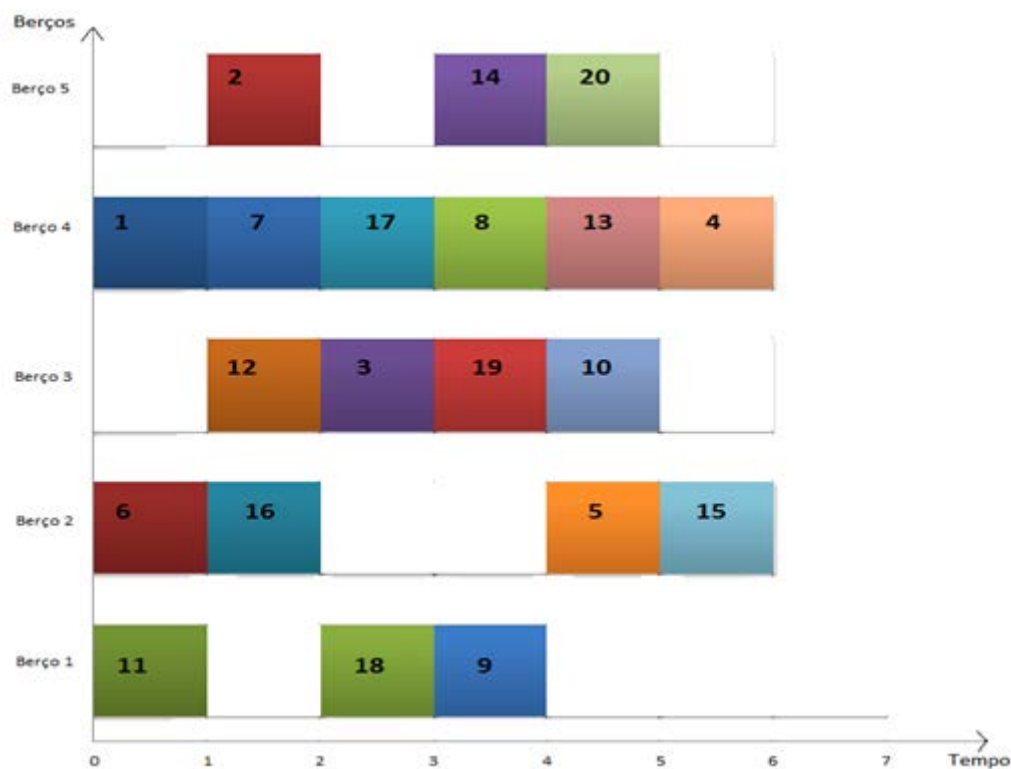


Figura 2.2 – Exemplo de alocação dos navios nos berços.

2.4 PARÂMETROS PARA RESOLUÇÃO DO PAB

Considerando o fato de queremos aproximar a solução do problema presente ao problema real encontrado nos portos, serão determinados alguns parâmetros que retratam o funcionamento portuário quanto ao plano de atracação.

2.4.1 CUSTOS (NAVIO E PORTO)

Podemos citar diversos custos decorrentes das atividades portuárias. Para esse projeto iremos considerar as tarifas de atracação e movimentação de cargas e também o custo de um navio parado. Essas taxas foram consideradas as mais importantes e de maior impacto na formação do custo operacional portuário.

A tarifa de atracação é cobrada por metro de cais ocupado pelo navio e por período. É utilizada a denominação de período para representar a quantidade de horas corridas desde o momento de atracação do navio no berço até o momento de liberação do mesmo. Exemplificando, se consideramos um período como sendo 3 horas e um navio necessita 4 horas para realizar suas atividades no berço, então esse mesmo navio utilizará dois períodos do berço.

A tarifa de movimentação de cargas é cobrada em função da quantidade de carga movimentada no berço, podendo ser cobrada tanto por toneladas movimentadas quanto por contêineres, sendo necessário definir esse critério de cobrança no momento de negociação da atividade do porto.

Estas duas tarifas são consideradas as tarifas principais cobradas pela utilização da infraestrutura portuária, havendo ainda as tarifas cobradas pela utilização da infraestrutura terrestre e por serviços gerais.

Além disso, existe o custo de um navio parado, que é custo obtido pelo navio enquanto o mesmo está parado no porto, seja esperando para atracar um berço, quanto enquanto está alocado em um berço realizando as movimentações. Portanto, esse custo é cobrado a partir do período que um navio passa em qualquer uma das instalações presentes no porto.

2.4.2 RESTRIÇÕES

Existem diversas restrições que influenciam no processo decisório de atracação de certo navio, destacando-se a restrição temporal, a de calado, a de comprimento dos equipamentos e a de agendamento dos navios.

A restrição temporal está ligada ao momento de chegada de um navio no porto e também ao momento de liberação dos berços. O momento de liberação de um berço é quando um navio deixa um berço, liberando-o para atender outro navio.

A restrição de calado avalia se a profundidade do berço é igual ou superior ao calado do navio para que ocorra a atracação sem problemas.

A restrição de comprimento deve verificar se o comprimento do berço é igual ou superior ao comprimento do navio, para permitir que o navio atraque no mesmo.

Quanto aos equipamentos, é necessário verificar se os berços que já obedeceram às restrições anteriores possuem os equipamentos necessários para a operação de carga e descarga de mercadorias. Nesse projeto serão considerados que todos os berços possuem os equipamentos necessários para alocação de qualquer tipo de navio.

A restrição de agendamento considerada nesse projeto visa garantir o escalonamento dos navios nos berços de forma que ocorra a menor data de liberação dos navios ou o menor custo de alocação. Em alguns casos, como o transporte de produtos perecíveis, é preferível que o atendimento dos navios ocorra no menor período de tempo possível para que a mercadoria não perca a validade. Já em outros casos, como transporte de produtos não perecíveis, como minérios, é preferível que o atendimento dos navios ocorra com o menor custo, para não encarecer a mercadoria.

2.4.3 VARIÁVEIS DE DECISÃO (QUANDO E QUANTO ALOCAR CADA NAVIO)

A partir de uma lista com os navios que irão atracar no porto em um período de tempo, a administração do porto deve realizar um plano de atracação, ou seja, criar um planejamento para os navios de modo a garantir que todos os navios sejam atendidos no menor período de tempo e menor custo possíveis. Esse planejamento deve obedecer todas as restrições para que seja feito o escalonamento.

O plano de atracação deve ser capaz de alocar os navios aos berços da melhor maneira possível, ou seja, determinar em ‘qual’ berço deverá atracar cada navio, bem como ‘o momento’ de atracação, analisando se houve tempo de espera por parte de cada navio, contabilizando custos a fim de otimizar o sistema operacional portuário para, no final do escalonamento, determinar um conjunto de berços atendendo a um conjunto de navios atracados em diferentes momentos, com capacidades, tarifas e produtividade diferenciadas.

3 ALGORITMOS EVOLUTIVOS E DE NUVENS DE PARTÍCULAS

Para a resolução do problema de alocação de berços foram implementadas duas soluções baseadas em dois tipos de algoritmos de otimização bioinspirados. Os dois tipos de algoritmos utilizados podem ser considerados algoritmos evolutivos, ou seja, as soluções iniciais encontradas no algoritmo evoluem ao longo do tempo. Cada tipo de algoritmo tem suas especificidades, porém, pode-se notar a mesma linha de raciocínio inicial em ambos, que é criar uma população inicial aleatoriamente com as possíveis soluções para o problema e aprimorar essas soluções em cada iteração do algoritmo, privilegiando sempre as melhores soluções e tentando encontrar a melhor solução para o problema.

A maneira como ocorre cada iteração será específica para cada tipo de algoritmo e será detalhada nas seções abaixo.

3.1 ALGORITMOS GENÉTICOS

3.1.1 HISTÓRICO

Os Algoritmos Genéticos (AG) foram concebidos por Holland (*Michigan University*) nos anos 60, e pertencem a uma classe de procedimentos de busca adaptativa inspirada nos princípios evolutivos das populações genéticas da natureza. Sua denominação originou-se da analogia entre a representação de uma estrutura complexa através de um vetor de componentes e a ideia da estrutura genética de um cromossomo.

Tanto no contexto natural quanto o dos Algoritmos Genéticos existem dois conceitos fundamentais, que são: a *evolução*, que pode ser conceituada como a geração espontânea e contínua de estruturas funcionais inovativas (BEDAU e PACKARD, 1992), e a *adaptação*, que designa qualquer processo através do qual uma estrutura é

progressivamente modificada para obter melhor desempenho no seu ambiente (HOLLAND, 1992).

Durante o curso da evolução, as populações naturais evoluem conforme os princípios de seleção natural e sobrevivência da adaptação. Ou seja, indivíduos que se adaptam melhor ao ambiente possuem maiores chances de sobreviver e reproduzir, enquanto os indivíduos de menor ajuste tendem a ser eliminados. Isto significa que os genes dos indivíduos altamente aptos influenciarão um número crescente de indivíduos a cada geração, e sua espécie evolui para se tornar cada vez mais adaptada ao seu ambiente.

Um AG simula este processo, tomando uma população inicial de indivíduos e aplicando operadores genéticos artificiais em cada geração. Em condições de otimização, cada indivíduo da população é codificado em uma *string* ou cromossomo, o qual representa uma solução possível para um determinado problema, enquanto a adaptação dos indivíduos é avaliada através de uma função de aptidão.

Basicamente, aos indivíduos altamente aptos (melhores soluções) são dadas maiores oportunidades de reproduzirem-se trocando partes de informação genética, em um procedimento de acasalamento denominado cruzamento (*crossover*); o operador de mutação é utilizado para alterar alguns genes nos cromossomos, e causar diversidade na população. A descendência ou nova população pode substituir toda a população atual ou substituir apenas os indivíduos de menor ajuste. Este ciclo de avaliação, seleção e geração, é repetido até que uma solução satisfatória seja encontrada.

O trabalho inicial que serviu de base para os estudos realizados de algoritmos genéticos está presente em Genetic Algorithms in Search, Optimization and Machine Learning (GOLDBERG, 1989).

3.1.2 CONCEITOS INICIAIS

Inicialmente, serão apresentados alguns conceitos sobre genética para que o entendimento sobre os algoritmos genéticos seja completo. Os cromossomos são cadeias de DNA e servem como modelo para todo o organismo, estando presentes em conjuntos em cada célula do organismo. Este é constituído de genes, que são blocos de DNA. Cada gene codifica uma determinada proteína. Quando dois genes codificam um mesmo tipo de característica do organismo eles podem ser chamados de alelos. Cada

gene possui sua posição própria dentro do cromossomo, e essa posição é chamada local. Um conjunto completo de material genético com todos os cromossomos é chamado de genoma. Um conjunto partículas de genes de um genoma é chamado de genótipo. O fenótipo é um genótipo desenvolvido que apresenta as características físicas e mentais dos organismos (CARVALHO, 2013).

A reprodução ocorre com a recombinação ou cruzamento. Os genes dos pais são combinados para formar um novo cromossomo. Os descendentes da reprodução podem sofrer mutação. Mutação é quando os elementos do DNA são ligeiramente modificados. Essas mudanças ocorrem principalmente por erros na cópia dos genes dos pais. A adaptação de um organismo é medida pelo sucesso do organismo durante a sua sobrevivência.

Agora, os conceitos biológicos citados acima serão transformados em conceitos para aplicação dos algoritmos. Portanto, cada conceito sobre genético deverá ser um novo conceito computacional, que possa ser aplicado em um software. Esses conceitos computacionais estão descritos abaixo.

Entendemos por espaço de soluções o espaço de todas as soluções possíveis para o problema que estamos resolvendo. Cada solução pode ser reconhecida pelo seu valor ou adequação para o problema. Com os Algoritmos Genéticos nós procuramos pela melhor solução, que consideramos como a solução que apresenta um valor extremo, mínimo ou máximo, no espaço de soluções.

Problemas NP (não polinomiais) são aqueles pertencentes a uma classe de problemas que não podem ser resolvidos de modo tradicional. A solução para esses problemas não é encontrada com algoritmos rápidos, mas uma vez encontrada é fácil de ser verificada, portanto chamamos esses problemas de NP-completos. Considerando o fato de existirem muitos problemas com saídas complicadas, existe a classe de problemas chamada de problemas NP-difíceis (FELIPE *et al.*, 2007).

Uma característica desses problemas é que um algoritmo simples pode parecer bom, apresentando soluções úteis rapidamente. Porém, geralmente esses algoritmos produzem muitas soluções possíveis e testar todas as soluções é um processo muito lento. Portanto, não existe nenhum algoritmo que funcione de maneira ótima para a resolução desse tipo de problemas. Um método alternativo que vem sendo utilizado com frequência são os algoritmos genéticos.

3.1.3 FUNCIONAMENTO DOS ALGORITMOS GENÉTICOS

Os algoritmos genéticos são inspirados na teoria da evolução de Darwin. Portanto, um problema que é resolvido por algoritmos genéticos apresenta uma solução que foi evoluída, ou seja, que passou por um processo evolucionário.

O algoritmo genético tem início com um conjunto de soluções (indivíduos), representadas por cromossomos, chamado de população. As soluções que vão formar as novas gerações de soluções são selecionadas de acordo com a sua adequação (*fitness*). As novas gerações são formadas pelo cruzamento de indivíduos da população, chamados de pais. Quanto melhores são as soluções (indivíduos), mais chances de reprodução terão. Nos cruzamentos podem ocorrer modificações genéticas, que melhoram ou pioram os indivíduos gerados.

Esse processo é repetido até que alguma condição é satisfeita (por exemplo, o número de populações ou o aperfeiçoamento da melhor solução). Um pseudocódigo para a representação do AG pode ser visto na Figura 3.1.

```
1: Gere uma população aleatória de  $n$  cromossomos (soluções para o problema)
2: Avalie a adequação  $f(x)$  de cada cromossomo  $x$  da população
3: Crie uma nova população repetindo os passos seguintes até que a nova população esteja completa:
    3.1: Selecione de acordo com sua adequação (melhor adequação, mais chances de ser selecionado)
        dois cromossomos para serem os pais;
    3.2: Com a probabilidade de cruzamento cruze os pais para formar a nova geração. Se não realizar
        cruzamento, a nova geração será uma cópia exata dos pais.
    3.3: Com a probabilidade de mutação, altere os cromossoms da nova geração nos locus (posição nos
        cromossomos).
    3.4: Coloque a nova descendência na nova população
    3.5: Utilize a nova população gerada para a próxima rodada do algoritmo
4: Se a condição final foi atingida, pare, e retorne a melhor solução da população atual
5: Vá para o passo 2
```

Figura 3.1 – Pseudocódigo para o Algoritmo Genético.

3.1.4 TIPOS DE CODIFICAÇÃO DE UM CROMOSSOMO

A forma como um cromossomo é codificado está diretamente ligada ao tipo de problema que está sendo estudado. Este capítulo tratará dos tipos de codificação existentes e já utilizados para os algoritmos genéticos.

A codificação binária é a forma mais comum devido à sua simplicidade. Nela, cada cromossomo é uma série de bits, 0 ou 1. A Figura 3.2 apresenta um exemplo de cromossomo binário, onde cada bit corresponde a um gene.

Cromossomo 1:	1101100100110110
Cromossomo 2:	1101111000011110

Figura 3.2 – Cromossomo binário.

Um exemplo de problema que utiliza a codificação binária é o problema da mochila. Nele é dada uma lista de objetos com preços e tamanhos e o valor da capacidade da mochila. Com isso, temos que escolher os objetos de forma a maximizar o valor das coisas que cabem dentro da mochila sem ultrapassar sua capacidade. Nesse exemplo, cada bit representa se o objeto correspondente está ou não na mochila.

A codificação por permutação pode ser utilizada em problemas que envolvem ordenação. Nesse tipo de codificação, cada cromossomo é uma série de números que representa uma posição em uma sequência. A Figura 3.3 apresenta um exemplo de cromossomo por permutação, onde cada gene é representado por um número inteiro.

Cromossomo 1:	1 3 2 4 7 6 8 5 10 9
Cromossomo 2:	2 5 3 9 6 1 7 8 4 10

Figura 3.3 – Cromossomo por permutação.

Um exemplo de problema que utiliza a codificação por permutação é o problema do caixeiro viajante. Nele são dadas cidades e as distâncias entre elas, e o caixeiro viajante tem que visitar todas elas sem andar mais do que o necessário. Com isso, temos que achar uma solução que contenha a sequência de cidades em que as viagens devem ser feitas para que a distância percorrida seja a mínima possível. Nesse exemplo, os cromossomos descrevem a ordem das cidades que o caixeiro visitará, onde cada gene do cromossomo representa uma cidade.

A codificação por valores é usada em problemas que contêm valores mais complexos, como números reais. Nesse tipo de codificação, cada cromossomo é uma sequência de valores, que pode ser qualquer valor relacionado com o problema, como

números reais ou caracteres. A Figura 3.4 apresenta um exemplo de cromossomo por valores.

Cromossomo 1:	2.3456 0.1234 1.4567 3.7890
Cromossomo 2:	A B C D E F G H I J K L M N
Cromossomo 3:	(direita), (esquerda), (frente), (atrás)

Figura 3.4 – Cromossomo por valores.

Um exemplo de problema que utiliza a codificação por valores é o cálculo de pesos para uma rede neural. Nele, é dada uma rede neural com arquitetura definida e temos que encontrar os pesos entre os neurônios da rede para obter a resposta desejada da rede. Nesse exemplo, os cromossomos indicam quais os pesos da rede neural com valores reais.

A codificação em árvore é usada principalmente para desenvolver programas ou expressões para programação genética. A linguagem LISP, por exemplo, utiliza muito esse tipo de codificação. Nela, cada cromossomo é uma árvore de objetos, como funções ou comandos de uma linguagem de programação. A Figura 3.5 apresenta um exemplo de cromossomo em árvore.

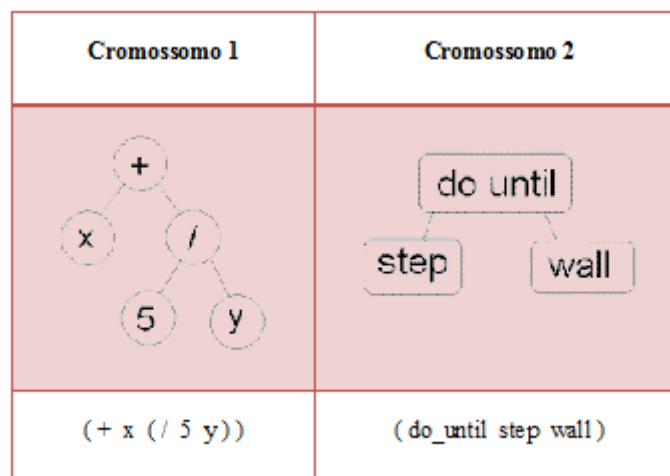


Figura 3.5 – Cromossomo em árvore.

Um exemplo de problema que utiliza a codificação em árvore é quando queremos encontrar uma função que aproxime dois valores. São dados os valores de entrada e saída e temos que encontrar uma função que forneça a melhor saída para todas as entradas. Os cromossomos são funções representadas em uma árvore.

3.1.5 OPERADORES DE SELEÇÃO

A operação de seleção acontece para que se obtenham os pais que participarão do cruzamento para gerar os filhos. Existem vários tipos de seleção, como seleção por roleta, seleção Boltzman, seleção por campeonato, seleção por classificação, seleção por estado estacionário, entre outros.

Na seleção por roleta, os pais são selecionados de acordo com sua adequação. Nesse caso, os cromossomos da população são colocados em uma roleta na secção da roleta proporcional ao valor de adequação de cada cromossomo. Quanto maior for o valor, mais larga é a seção e maior é a chance de seleção. A Figura 3.6 mostra que o cromossomo 1 tem maior chance de ser selecionado, seguido pelo cromossomo 2, e assim por diante.

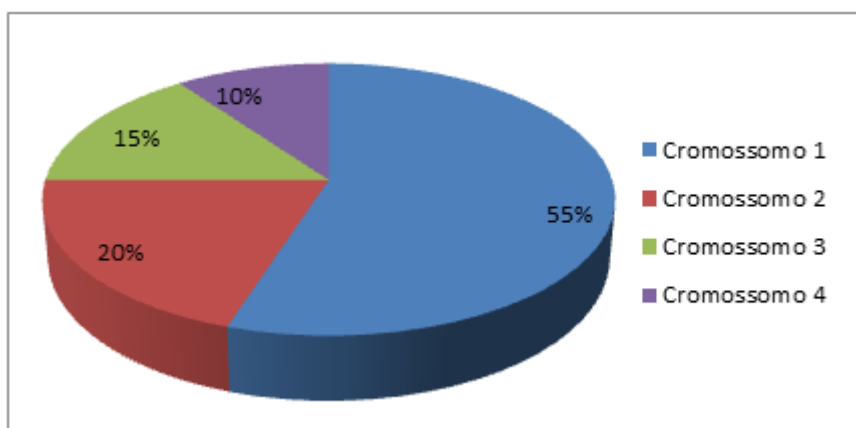


Figura 3.6 – Seleção por roleta.

No torneio estocástico escolhe-se os indivíduos mais adaptados, associando a cada indivíduo uma probabilidade de sobrevivência de acordo com sua adequação. Nele são escolhidos k indivíduos da população, sendo que os melhores são selecionados e os piores morrem, repetindo esse processo N vezes, onde N é o tamanho da população.

Na seleção por classificação ou *ranking* uniforme, primeiro é feita uma classificação da população, e então é dado um valor de adequação para cada cromossomo de acordo com sua classificação. O pior cromossomo terá adequação igual a 1, o segundo pior igual a 2, e o melhor terá adequação igual a N , sendo N o tamanho da população. Dessa forma, todos os cromossomos terão chance de serem selecionados. A Figura 3.7 mostra as taxas de adequação antes e depois da classificação.

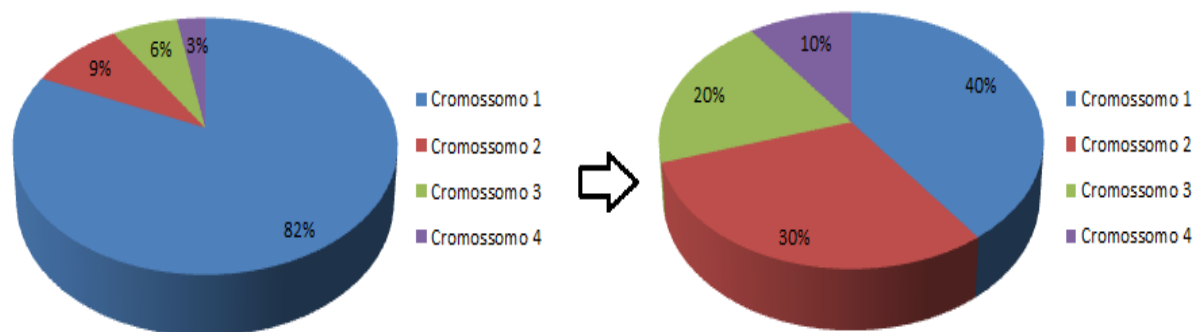


Figura 3.7 – Taxa de adequação dos cromossomos antes e depois da classificação.

Na seleção por estado estacionário não é um método para seleção dos pais, e a ideia principal é que a nova população deve ter muitos cromossomos que sobreviverão para a próxima geração. Nesse tipo de seleção, em cada nova geração alguns cromossomos com alta adequação são selecionados para criar a descendência, e depois alguns cromossomos com baixa adequação são removidos e novos descendentes são colocados em seus lugares.

Na seleção por truncamento, com base em um valor de limiar T entre 0 e 1, a seleção é feita aleatoriamente entre os T melhores indivíduos. Como exemplo, se $T = 0,4$, então a seleção é feita entre os 40% melhores indivíduos e os outros 60% são descartados.

No método elitismo levamos em consideração que quando criamos uma nova população com cruzamento e mutação, tem-se o risco de perder os melhores cromossomos. Nesse método então, copiamos os melhores cromossomos para a nova população.

3.1.6 OPERADORES DE CRUZAMENTO E MUTAÇÃO

Os operadores de cruzamento e mutação são básicos para os algoritmos genéticos, e o desempenho do algoritmo depende deles. O tipo e a implementação a ser utilizada para esses operadores vai depender do tipo de codificação escolhido para o problema. Esses operadores serão detalhados abaixo.

Os operadores de cruzamento para codificação binária são: cruzamento de 1 ponto, cruzamento de 2 pontos, cruzamento uniforme e o cruzamento aritmético.

No cruzamento de 1 ponto, um ponto de cruzamento é escolhido e a série binária desde o começo do cromossomo até esse ponto é copiada do primeiro pai para o filho e desse ponto para o final do cromossomo é copiada do segundo pai para o filho. Podemos ver um exemplo na Figura 3.8.

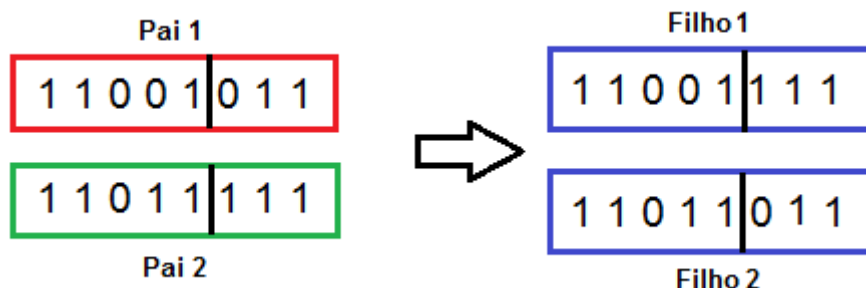


Figura 3.8 – Exemplo de *crossover* de 1 ponto.

No cruzamento de 2 pontos, são definidos dois pontos de cruzamento, e a série binária é copiada desde o início do cromossomo até o primeiro ponto de cruzamento do primeiro pai, do primeiro ponto de cruzamento até o segundo ponto do segundo pai, e do segundo ponto de cruzamento até o final do primeiro pai para gerar o filho. Podemos ver um exemplo na Figura 3.9.

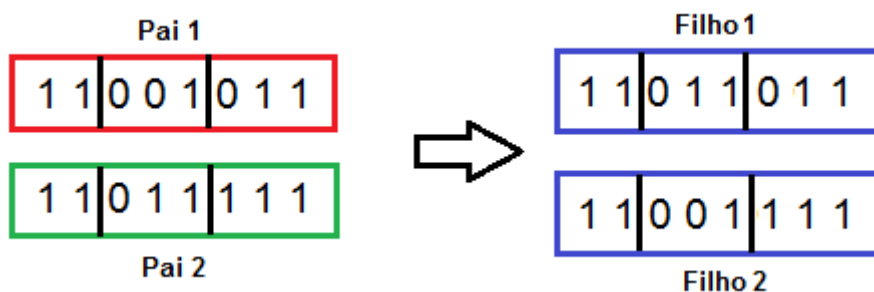


Figura 3.9 – Exemplo de *crossover* de 2 pontos.

No cruzamento uniforme, os bits são copiados aleatoriamente do primeiro ou segundo pai, como podemos ver no exemplo da Figura 3.10.

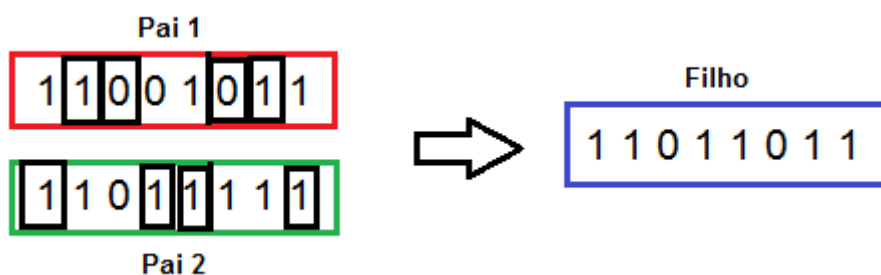


Figura 3.10 – Exemplo de cruzamento uniforme.

No cruzamento aritmético, é realizada uma operação aritmética para obter a nova geração. Podemos ver no exemplo da Figura 3.11 o filho gerado a partir da operação AND realizada com os pais.

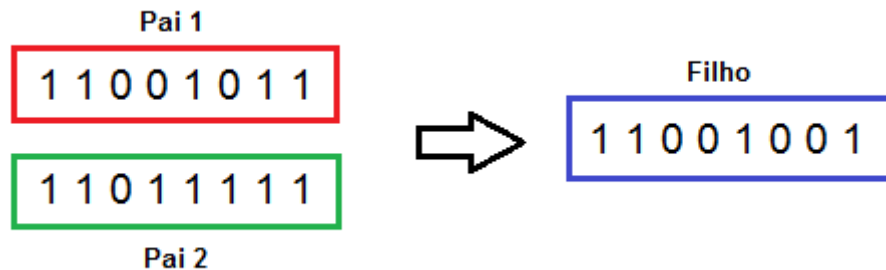


Figura 3.11 – Exemplo de cruzamento aritmético.

A operação de mutação para a codificação binária é chamada inversão de bit. Nessa operação, determinados bits são invertidos, passando de 0 para 1 ou o oposto. Podemos ver um exemplo desse caso na Figura 3.12.



Figura 3.12 – Mutação para codificação binária.

Os operadores de cruzamento para a codificação por permutação são: *Order Crossover* (OX), *Partially Mapped Crossover* (PMX), *Cycle Crossover* (CX), *Order Based Crossover* (OBX), *Position Based Crossover* (PBX) e recombinação de adjacências. Esses operadores são mais complexos, pois dependem de várias operações intermediárias para serem finalizados. No operador OX, por exemplo, são feitos dois pontos de cortes e os filhos herdam parte de um pai e parte do outro pai, de acordo com a visita dos pais. O exemplo da Figura 3.13 mostra como ficaria o cruzamento pelo operador OX.

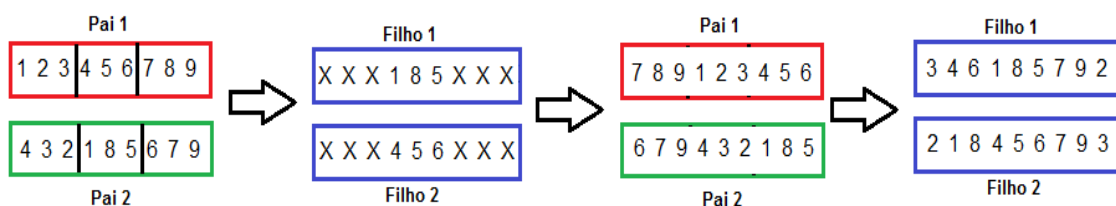


Figura 3.13 – Exemplo do operador de mutação OX.

Alguns dos operadores de mutação existentes para a codificação por permutação são: troca recíproca, troca aleatória, inversão, inserção e embaralhamento.

Na troca recíproca, são sorteados dois pontos, e são invertidos os valores nesses dois pontos, como no exemplo da Figura 3.14.



Figura 3.14 – Exemplo de mutação do tipo troca recíproca.

No troca aleatória, é sorteada uma posição e o valor dessa posição é trocado, como no exemplo da Figura 3.15.



Figura 3.15 – Exemplo de mutação do tipo troca aleatória.

Na inversão, é selecionado um segmento do cromossomo, e os valores desse segmento são colocados na ordem inversa, como no exemplo da Figura 3.16.



Figura 3.16 – Exemplo de mutação do tipo inversão.

Na inserção, são sorteados dois pontos e o valor do segundo ponto é colocado na frente do primeiro, como no exemplo da Figura 3.17.



Figura 3.17 – Exemplo de mutação do tipo inserção.

No embaralhamento, é selecionado um segmento do cromossomo e seus valores são trocados aleatoriamente, como no exemplo da Figura 3.18.



Figura 3.18 – Exemplo de mutação do tipo embaralhamento.

As operações de cruzamento existentes para a codificação por valores são as mesmas utilizadas para a codificação binária. E quando os valores são números reais, existe ainda o cruzamento de média. Nele, os valores dos filhos são a média dos valores dos pais, como no exemplo da Figura 3.19.

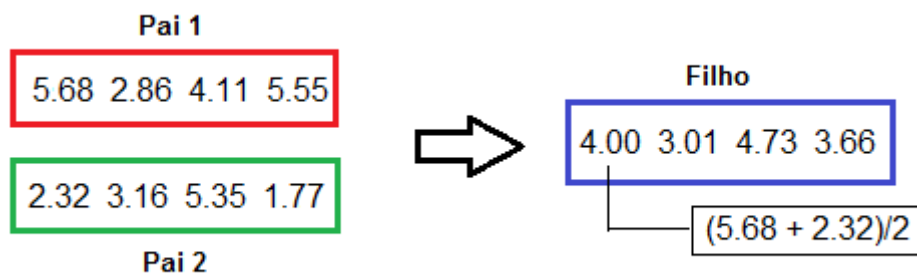


Figura 3.19 – *Crossover* para codificação por valores.

A operação de mutação para a codificação de valores pode ser feita com a mutação aleatória, trocando um valor por outro valor qualquer. Quando os valores são reais, então se pode optar pela mutação CREEP, que é realizada com a adição ou subtração de um valor calculado a alguns valores selecionados do cromossomo. Um exemplo da mutação CREEP pode ser visto na Figura 3.20.



Figura 3.20 – Mutação para codificação por valores.

A operação de cruzamento para a codificação em árvore é feita de forma que a partir de um ponto de cruzamento escolhido em ambos os pais, são copiadas as partes de baixo desse ponto e trocada para produzir o filho, como no exemplo da Figura 3.21.

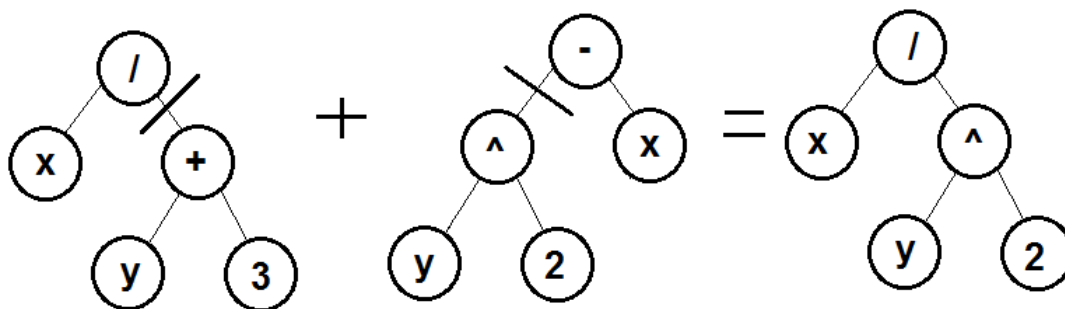


Figura 3.21 – Cruzamento para codificação em árvore.

A operação de mutação para a codificação em árvore envolve a mudança do operador. Então, nos nós escolhidos, são alterados os operadores.

3.1.8 PARÂMETROS DOS ALGORITMOS GENÉTICOS

Existem dois parâmetros básicos que influenciam nos resultados dos algoritmos genéticos. São eles a taxa de cruzamento e a taxa de mutação.

A taxa de cruzamento indica com qual frequência o cruzamento é realizado. Se não houver cruzamento, a descendência é uma cópia exata dos cromossomos de ambos os pais. Se a taxa de cruzamento é de 100%, então toda a descendência é produzida por cruzamento. Se a taxa é 0%, então não existe cruzamento e toda a descendência é formada pela cópia exata dos cromossomos da população antiga. Quando um cruzamento ocorre espera-se que a nova geração contenha as partes boas dos cromossomos antigos, e então sejam produzidos cromossomos melhores. Porém, é sempre melhor deixar que uma parte da antiga população também sobreviva para a próxima geração.

A taxa de mutação indica com qual frequência ocorre a mutação dos cromossomos. Se não houver mutação, então os cromossomos da nova geração são obtidos somente com o cruzamento. Se ocorrer mutação, uma ou mais partes do cromossomo é alterada. Se a taxa é de 100% então todos os cromossomos são alterados, se é de 0%, nenhum cromossomo é alterado. A mutação geralmente evita que o

algoritmo caia num extremo local. Porém, ela não deve ocorrer com muita frequência porque senão o algoritmo se tornará de fato uma busca aleatória.

Existem ainda outros parâmetros que são importantes para o algoritmo genético. Um deles é o tamanho da população, que indica quantos cromossomos existem na população em uma geração. Se houver muitos cromossomos, então existirão poucas possibilidades de cruzamentos e somente uma parte pequena do espaço de soluções será explorada. Entretanto, se houver muitos cromossomos, então o algoritmo se tornará muito lento.

3.1.9 CODIFICAÇÃO, OPERADORES E PARÂMETROS PARA O PAB

Devido ao fato do Problema de Alocação de Berços ser um problema combinatorial, que podemos assimilar ao Problema do Caixeiro Viajante, iremos codificar os cromossomos através da codificação por permutação. Nesse caso, os valores dos cromossomos serão correspondentes aos valores dos berços que cada navio vai ocupar, considerando que a posição do berço no cromossomo indica qual o número do navio.

Para a operação de seleção utilizaremos o operador de seleção por roleta. E o critério de substituição escolhido para cada iteração foi eliminar os dois piores cromossomos e adicionar os dois filhos gerados à população.

Para a operação de cruzamento não será possível utilizar os operadores de cruzamento usuais para a codificação por permutação, pois no nosso problema consideramos que poderá haver a repetição de berços. Então, iremos utilizar dois operadores similares aos operadores para codificação binária, que são o cruzamento de um ponto e o cruzamento de dois pontos. Porém, este tipo de cruzamento pode levar a geração de indivíduos inválidos para o problema, o que deve ser contornado.

Para a operação de mutação iremos utilizar os mesmos operadores que usualmente são utilizadas com a codificação por permutação. Com relação à taxa de *crossover* e mutação, usaremos os valores de 90% e 10%, respectivamente.

E, finalmente, descreveremos a função de *fitness* no capítulo seguinte.

3.2 OTIMIZAÇÃO POR NUVEM DE PARTÍCULAS

3.2.1 HISTÓRICO

Os métodos de otimização e busca estocástica baseados nos princípios e modelos da evolução biológica natural têm recebido crescente interesse nas últimas décadas, devido principalmente a sua versatilidade para a resolução de problemas complexos, nas áreas de otimização e aprendizado de máquina. O desenvolvimento de modelos computacionais, inspirados nos mecanismos evolutivos, caracteriza-se pela configuração de algoritmos de otimização robustos e sistemas adaptativos.

As técnicas de computação evolucionária (programação evolucionária (FOGEL, 1994), algoritmos genéticos (GOLDBERG, 1989), estratégias evolucionárias (KOZA, 1992) e programação genética (RECHEMBERG, 1994)) são motivadas pela evolução da natureza. Uma população de indivíduos, os quais representam as possíveis soluções para o problema, são manipulados, de acordo com as regras de sobrevivência, que melhor se adaptam aos operadores genéticos, tais como mutação, cruzamento e reprodução. A melhor solução evolui através das gerações.

Otimização por Nuvem de Partículas (ONP) ou *Particle Swarm Optimization* (PSO) é uma técnica de computação evolucionária desenvolvida por James Kennedy, um psicólogo social, e por Russell Eberhart, um engenheiro elétrico, em 1995, inspirada na simulação de um sistema social simplificado (KENNEDY e EBERHARD, 1995). A intenção original era simular graficamente o comportamento de um bando de pássaros em voo com seu movimento localmente aleatório, mas globalmente determinado.

O modelo proposto continha quatro regras, que são: *separação*, para evitar que cada pássaro colida com outro, *alinhamento*, para fazer com que cada pássaro siga a mesma direção de seus vizinhos, *coesão*, para que cada pássaro siga a mesma posição de seus vizinhos e *desvio*, para que cada pássaro desvie de obstáculos a frente.

Computacionalmente, os Algoritmos de Nuvem ou Enxame de Partículas são uma abstração desse processo natural, onde a procura pela posição mais apta é a busca de uma solução ‘ótima’ para um problema, sendo o conjunto de possíveis posições das partículas o espaço de busca do problema, e cada posição ocupada por uma partícula uma possível solução para o problema.

Diferentemente dos algoritmos genéticos, onde os cromossomos evoluem através de operadores genéticos, no PSO as partículas evoluem através da cooperação e competição entre si por diversas gerações.

O comportamento de cada partícula é baseado na sua experiência anterior e na experiência das outras partículas com os quais ele se relaciona. Similarmente aos algoritmos genéticos, o conjunto das partículas tende a preservar aquelas posições que determinam uma maior aptidão e a descartar as posições de menor aptidão.

O trabalho inicial que serviu de base para os estudos realizados de algoritmos de otimização por nuvem de partículas está presente no livro *Particle Swarm Optimization* (KENNEDY e EBERHARD, 1995).

3.2.2 CONCEITOS BÁSICOS

O PSO clássico é muito similar a algumas técnicas da computação evolucionária, como os algoritmos genéticos (GA), onde o sistema é inicializado com uma população de soluções aleatórias, chamadas de nuvens de partículas. Entretanto, difere dessas outras técnicas populacionais, pois nenhum operador inspirado pelos procedimentos de DNA é aplicado na população para obter uma nova geração de indivíduos. Ao invés disso, cada partícula (individualmente) ajusta seu voo de acordo com sua própria experiência de voo e na de seus companheiros.

Cada partícula é tratada como um ponto em um espaço D-dimensional. A i -ésima partícula é representada como:

$$X_i = (X_{i1}, X_{i2}, \dots, X_{iD}). \quad (1)$$

A melhor posição prévia (a posição que dá o melhor valor de aptidão) da i -ésima partícula é registrada e representada como:

$$P_i = (P_{i1}, P_{i2}, \dots, P_{iD}). \quad (2)$$

O índice da melhor partícula entre todas as partículas na população é representado pelo símbolo “g”.

A maior parte dos algoritmos de PSO empregam dois princípios sócio-métricos. O primeiro, denominado de *gbest* (g = global), conecta conceitualmente todos os

membros de uma população entre si. Como consequência, o comportamento de cada partícula é influenciado pelo comportamento de todas as outras partículas. O segundo, denominado de *lbest* ($l = \text{local}$), cria uma vizinhança para cada indivíduo composta por ele próprio e seus k -vizinhos mais próximos (ROSENDO, 2010).

No PSO existem duas topologias básicas para representar as nuvens de partículas usadas na literatura, que são: a topologia em anel (vizinhança de 3) e a topologia em estrela (vizinhança global). Estas estão representadas na Figura 3.22.

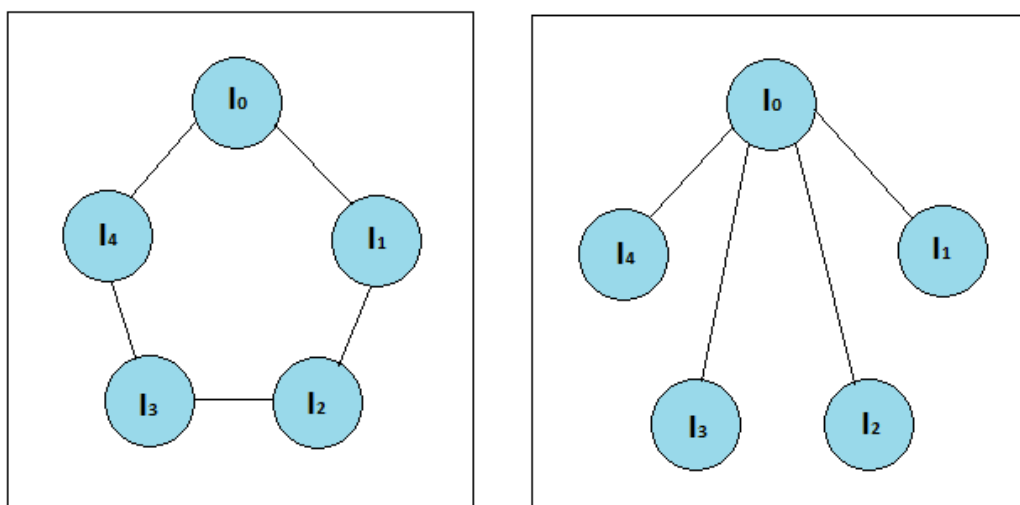


Figura 3.22 – Topologias para o PSO.

A taxa da mudança de posição (velocidade) para partícula i é representada como:

$$V_i = (V_{i1}, V_{i2}, \dots, V_{iD}). \quad (3)$$

A movimentação de cada partícula é fundamentada em alguns procedimentos determinísticos, baseando-se em três parâmetros: *fator de sociabilidade* que determina a atração das partículas para a melhor posição descoberta por qualquer elemento da nuvem, *fator de individualidade* que determina a atração da partícula com sua melhor posição já descoberta e *velocidade máxima* que delimita o movimento, uma vez que esse é direcional e determinado (ROSENDO, 2010). O algoritmo combina estes parâmetros com um número gerado aleatoriamente para determinar o próximo local da partícula.

As partículas são manipuladas de acordo com as seguintes equações:

$$V_{id} = \omega * V_{id} + c_1 * rand() * (P_{id} - X_{id}) + c_2 * rand() * (P_{gd} - X_{id}) \quad (4)$$

$$X_{id} = X_{id} + V_{id} \quad (5)$$

onde:

c_1 e c_2 - são duas constantes positivas que correspondem as componentes cognitivas e sociais;

$rand()$ - é uma função aleatória no intervalo $[0,1]$,

ω - é o peso de inércia,

V_{id} - é a atual velocidade da partícula i ,

P_{id} - é a melhor posição da partícula i ($pbest$),

X_{id} - é a atual posição da partícula i e,

P_{gd} - é a melhor posição global entre todas as partículas ($gbest$).

A Equação (4) é usada para calcular a nova velocidade da partícula de acordo com sua velocidade anterior e as distâncias entre sua posição atual, sua melhor posição e a melhor posição do grupo. Então a partícula “voa” para uma nova posição de acordo com equação (5). O desempenho de cada partícula está medido de acordo com uma função de aptidão pré-definida que é relacionada ao problema a ser resolvido.

O peso de inércia ω é empregado para controlar o impacto da velocidade anterior na velocidade atual, assim influenciando as habilidades de exploração global e local das partículas. Um peso de inércia maior facilita exploração global (procurando novas áreas), enquanto um peso de inércia menor tende a facilitar exploração local para refinar a área de procura atual. A seleção satisfatória do peso de inércia ω pode prover um equilíbrio entre habilidades de exploração global e local, e assim pode requerer menos repetições, em média, para encontrar o valor ótimo.

Cada partícula mantém rastro de suas coordenadas, no espaço do problema que é associado com a melhor solução, sendo na verdade, o quanto a partícula deslocou-se. Este valor é chamado $pbest$. Outro valor que é determinado pelas partículas é o melhor valor obtido por qualquer partícula vizinha. Este local é chamado $lbest$. Quando uma partícula levar toda a população, ou seja, seus vizinhos, o melhor valor é chamado $gbest$. O número de nuvens em um espaço é claramente conhecido como um fator na probabilidade de achar o ótimo, pois quanto maior o número de partículas em um determinado espaço, mais alta será a probabilidade de achar o ótimo. Porém,

reciprocamente, um número maior de partículas resultará no aumento de pontos individuais que serão testados, aumentando assim o tempo de computação.

Um pseudocódigo para a representação do algoritmo PSO pode ser visto na Figura 3.23.

Passo 0 (Definição das variáveis):
 P o tamanho da população do PSO.
 PSO[i] a posição da i-ésima partícula da população do PSO, que representa uma solução candidata para o problema.
 fitness[i] o custo da função da i-ésima partícula.
 V[i] a velocidade da partícula.
 Gbest um índice para a melhor posição global.
 Pbest[i] a posição de melhor localização da i-ésima partícula.
 Pbest_fitness[i] o melhor fitness local visitado pela i-ésima partícula.

Passo 1 (Inicialização): Para cada partícula i na população:
 Passo 1.1: Inicialize o PSO[i] randomicamente.
 Passo 1.2: Inicialize V[i] randomicamente.
 Passo 1.3: Avalie o fitness[i].
 Passo 1.4: Inicialize Gbest.
 Passo 1.5: Inicialize Pbest[i] com uma cópia de PSO[i] $\forall i \leq P$.

Passo 2: Repita até que um critério de parada seja satisfeito
 Passo 2.1: Encontre o Gbest tal que fitness[Gbest] = min fitness[i] $i \leq P$.
 Passo 2.2: Para cada partícula i:
 Pbest[i] = PSO[i] se fitness[i] < Pbest_fitness[i] $\forall i \leq P$.
 Passo 2.3: Para cada partícula i:
 Atualize V[i] e PSO[i] de acordo com as equações 1 e 2.
 Passo 2.4: Avalie o fitness[i] $\forall i \leq P$

Figura 3.23 – Pseudocódigo para o PSO.

No algoritmo, a nuvem de partículas é lançada inicialmente dentro do espaço de busca, tendo cada partícula as seguintes características:

- Uma posição e uma velocidade;
- Conhecimento de sua posição e o valor da função objetivo para esta posição;
- Conhecimento sobre seus vizinhos (vizinhança): a melhor posição encontrada e o valor da sua função objetivo;
- Armazenamento de sua melhor posição encontrada

Em cada espaço de tempo, o comportamento de uma partícula é determinado dentre três possíveis escolhas:

- Seguir seu próprio caminho;
- Seguir para sua melhor posição encontrada;
- Seguir para a melhor posição encontrada por algum de seus vizinhos.

A condição de parada é ultrapassar o limite de número de iterações pré-definido ou quando não houver mais melhorias (estagnação).

O PSO pode ser aplicado para: determinação de pesos e arquitetura de redes neurais artificiais, problemas de otimização de funções numéricas em geral e problemas do mundo real, como análise de tremor humano, otimização de moinhos, otimização de misturas de ingredientes, controle de potência e tensão reativa, estimação de carregamento de baterias, e composição de música, dentre inúmeros outros.

3.2.3 PSO DISCRETO

Para aplicarmos o PSO discreto são necessárias algumas modificações no PSO clássico. Mesmo com essas modificações, a ideia é preservar a estrutura do algoritmo original. As mudanças foram feitas em relação aos vetores de posição atual e melhor posição, transformando-os em valores discretos, e o vetor velocidade, substituindo-o por probabilidades de deslocamentos, fazendo com que um valor discreto modifique uma determinada posição de uma partícula. Além disso, os operadores utilizados no espaço de busca e nas soluções encontradas também foram modificados, já que o espaço de busca não será mais contínuo.

As partículas no PSO discreto são representadas por arestas/nós e cada um destes elementos corresponde a uma dimensão, representadas em um vetor posição. A mudança de posição consiste numa troca de posições no vetor posição da partícula. Estas mudanças recebem o nome de *swap* (ALOISE *et al.*, 2006).

Os operadores discretos, ou tipos de *swaps*, são:

- **Velocidade:** No espaço discreto a velocidade não é mais somente um número. Ela deve ser um operador que, quando aplicado a uma posição, dê outra posição. Ou seja, a velocidade é definida como sendo uma lista de transposições. Ela pode ser equivalente se aplicarmos v_1 ou v_2 e obtivermos o mesmo resultado. Uma velocidade nula é uma lista de velocidades sem transposições.
- **Movimentação:** Dado P como sendo uma posição e v uma velocidade, logo, P' será uma nova posição onde $P' = P + v$, aplicando-se todas as transposições de v à posição P . Por exemplo, se temos $P = [2,3,4,5,1]$ e $v = \{(1,2),(2,4)\}$, então resolveremos da seguinte forma:
 - 1- Aplique a transposição (1,2), trocando o conteúdo da primeira célula em P pelo conteúdo da segunda, obtendo $P_1 = [3,5,3,2,1]$
 - 2- Aplique a transposição (2,4), obtendo $P_2 = [3,5,4,2,1]$
 - 3- Obtemos $P' = P_2 = [3,5,4,2,1]$.
- **Obtenção da Velocidade:** Supondo duas posições A e B , teremos a velocidade v subtraindo uma posição da outra, ou seja, $v = A - B$. Essa diferença equivale a uma lista de transposições. Podemos escrever como $A = B + v$. Por exemplo, se temos $A = [1,2,3,4,5]$ e $B = [2,3,1,5,4]$, então resolveremos da seguinte forma:
 - 1- $A(1) = B(3) = 1$, logo a primeira transposição será (1,3) e teremos $B' = B + S(1,3)$, resultando em $B' = [1,3,2,5,4]$
 - 2- $A(2) = B'(3) = 2$, logo a segunda transposição será (2,3) e teremos $B'' = B' + S(2,3)$, resultando em $B'' = [1,2,3,5,4]$
 - 3- O terceiro operador será então (4,5) e obteremos a lista de transposições (v) como sendo igual a $A - B = S(1,3), S(2,3), S(4,5)$.
- **Adição de Velocidades:** A adição entre duas velocidades consiste na concatenação de listas de transposições resultando uma nova lista.
- **Multiplicação:** Podemos fazer a multiplicação da velocidade por um coeficiente C , que apresenta quatro valores diferentes:
 - (i) $C = 0$, teremos $C*v = \text{nulo}$;
 - (ii) C pertence $[0,1]$, truncamos v por $C*|v|$;
 - (iii) $C > 1$, aplicamos v , C vezes mais a parte decimal de $C*|v|$;
 - (iv) $C < 0$, não é definido pra este tipo de implementação.

3.2.4 PSO APLICADO AO PROBLEMA DE ALOCAÇÃO DE BERÇOS

Podemos considerar o PAB como sendo um problema de otimização combinatorial, então teremos que modificar o algoritmo de PSO para aplicarmos neste trabalho. As características das nuvens presentes nos algoritmos não serão modificadas e o processo de otimização será similar ao apresentado no PSO clássico.

Toda partícula terá sua posição atual $x_i(t)$ e sua melhor posição pessoal $ypbest(t)$. A vizinhança é global, de forma que no tempo t toda partícula sabe a melhor posição global $ygbest(t-1)$ comparada com a iteração anterior $t - 1$. As melhores posições pessoais e globais são dadas da mesma maneira que no PSO clássico. No entanto, a velocidade será dada de outra maneira, específica para o nosso problema.

Baseado na geração aleatória da alocação dos berços para os navios, a nova posição de um navio é determinada através de diferentes passos baseados nas atuais posições. Esses passos são:

- *Swap* local – Troca aleatória de dois berços dentro de uma mesma partícula;
- *Swap* com melhor pessoal – Troca aleatória de dois berços entre a partícula atual e a sua melhor posição pessoal;
- *Swap* com melhor global – Troca aleatória de dois berços entre a partícula atual e a melhor posição global;
- Troca aleatória – Com uma probabilidade dada, troca a posição de um navio aleatoriamente.

O processo iterativo acaba quando um número definido de iterações é atingido ou uma solução definida previamente é obtida.

Essa implementação foi baseada no trabalho de Brodersen *et al.* (2010).

4 MODELAGEM DO PROBLEMA DE ALOCAÇÃO DE BERÇOS PARA OTIMIZAÇÃO COM MÉTODOS EVOLUTIVOS

Após estudo aprofundado do problema de alocação de berços e das técnicas de otimização, algoritmos genéticos e otimização por nuvens de partículas, foi feita a modelagem do problema para que ele pudesse ser resolvido com a implementação do software baseado nessas técnicas. As técnicas apresentam algumas funções diferentes, que serão explicitadas, mas possuem uma base similar.

4.1 IMPLEMENTAÇÃO COMPUTACIONAL

O software criado foi implementado utilizando a linguagem Java, sendo desenvolvido no Eclipse 4.2 e executado no sistema operacional Windows Seven. Esse software contém a implementação dos dois algoritmos, AG e PSO.

Para ambos os algoritmos temos algumas implementações similares. Uma delas é que no início do programa temos uma função que lê a informação de dois arquivos de entrada em formato de planilha e carrega essas informações para elementos das Classes Navios e Berços. Ao final desse processo temos uma lista de elementos da Classe Navios e outra lista de elementos da Classe Berços. Tais elementos estão demonstrados nas Figuras 4.1 e 4.2.

<i>Navios:</i>
- Comprimento (metros);
- Calado (metros);
- Quantidade de carga (contêiner);
- Custo do navio parado (reais);
- Horário de chegada (hora);

Figura 4.1 – Elemento da Classe Navios.

Berços:
- Comprimento (metros);
- Profundidade (metros);
- Produtividade (contêiner/hora);
- Tarifa de atracação (reais);
- Tarifa de movimentação (reais);
- Horário de liberação (hora);

Figura 4.2 – Elemento da Classe Berços.

Além disso, em ambos os algoritmos teremos uma lista de possibilidades que é também criada automaticamente no início do programa, contendo as possibilidades de berços a serem ocupados por cada navio. Dessa forma podemos efetuar as operações e conferir no mesmo momento se o cromossomo ou a partícula será válida, evitando gerar soluções impróprias.

Abaixo serão apresentados como foram representados os cromossomos e as partículas para o AG e o PSO, respectivamente, tais como os operadores e as funções que são utilizados para cada um dos algoritmos.

4.2 REPRESENTAÇÃO DOS CROMOSSOMOS E DAS PARTÍCULAS

Tanto no AG quanto no PSO temos que representar a combinação que gera a alocação dos berços pelos navios, resultando nos cromossomos ou nas partículas. Para isso foram utilizadas listas de inteiros, onde os índices representam o número dos navios e os valores representam o número dos berços. A Figura 4.3 representa um exemplo de cromossomo, sendo a mesma para uma partícula.

Cromossomo = [1,3,2,5,4,6,2,5,4,3,1,6]

Figura 4.3 – Representação de um cromossomo em uma lista de inteiros.

No cromossomo ou partícula da Figura 4.3 está representado que o navio 1 aloca o berço 1, o navio 2 o berço 3, o navio 3 o berço 2, o navio 4 o berço 5, e assim sucessivamente. Nota-se que poderá haver repetição de berços, ou seja, mais de um navio pode alocar o mesmo berço. Para tratar esse problema, será criada uma lista de atendimentos, aonde o navio que chegar primeiro tem o direito de alocar o berço

primeiro. Dois navios alocados no mesmo berço significa que eles serão atendidos em tempos diferentes. Assim, deve-se também manter uma lista auxiliar contendo o tempo do início do atendimento de cada navio no berço.

4.3 OPERADORES

4.3.1 OPERADORES GENÉTICOS

Para a implementação do PAB, os operadores genéticos utilizados são próprios para cromossomos por permutação. Isso se deve ao fato de haver repetição de berços nos cromossomos.

Assim, para o PAB foram utilizados os seguintes operadores:

- seleção pelo método da roleta;
- reprodução (*crossover*) pelo método de um ponto e pelo método de dois pontos;
- mutação pelo método da troca recíproca, da troca aleatória, da inserção, do embaralhamento e da inversão.

Uma descrição detalhada destes operadores pode ser encontrada em Sivanandam & Deepa (2008).

4.3.2 OPERADORES PARA OTIMIZAÇÃO POR NUVEM DE PARTÍCULAS

Considerando que no PSO não ocorre a reprodução, foram criados somente operadores que podemos considerar como sendo operadores de mutação, de três tipos.

Portanto, temos os seguintes tipos de mutação:

- *Swap* do tipo 1 – troca de duas posições aleatórias da mesma partícula;
- *Swap* do tipo 2 – troca de uma posição da partícula atual com a mesma posição da melhor combinação encontrada para essa partícula;
- *Swap* do tipo 3 – troca de uma posição da partícula atual com a mesma posição da melhor partícula global.

4.4 FUNÇÕES UTILIZADAS

A função objetivo descrita abaixo foi utilizada tanto para a implementação do AG quanto para a do PSO. É ela que coordena os movimentos de cada iteração, avaliando as soluções encontradas. Esta função é a mesma encontrada em Silva (2008). Porém, a representação dos cromossomos e os operadores de *crossover* e mutação utilizados para o algoritmo genético não serão os mesmos dos apresentados no referido trabalho.

4.4.1 FORMULAÇÃO MATEMÁTICA DO PAB

A função objetivo utilizada se baseia na função que calcula o gasto obtido por uma possível solução (indivíduo) com as operações de alocação dos berços pelos navios. Os algoritmos desenvolvidos têm por objetivo encontrar o indivíduo que produz a solução com o menor custo.

A função de custo usada em ambos os algoritmos para o PAB é representada pela equação de minimização abaixo:

$$\text{Min } CA = \sum_{i \in N} \sum_{j \in B} \{[(E_{ij} + A_{ij}) \cdot CP_i] + (C_i \cdot TMov_j) + [(L_i \cdot TAtrac_j) \cdot P]\} \cdot x_{ij} \quad (6)$$

onde:

- $N = \{n_1, n_2, \dots, n_i\}$: representa o conjunto de navios;
- $B = \{b_1, b_2, \dots, b_j\}$: representa o conjunto de berços;
- CA : função objetivo do custo de alocação que será minimizada;
- E_{ij} : tempo de espera do navio i no berço j ;
- A_{ij} : tempo de atendimento do navio i no berço j ;
- CP_i : custo do navio i parado;
- L_i : comprimento do navio i ;
- $TAtrac_j$: tarifa de atracação cobrada pelo berço j ;
- C_i : carga do navio i ;
- $TMov_j$: tarifa de movimentação cobrada pelo berço j ;
- P : número de períodos executados;
- x_{ij} : variável binária (0 ou 1) que assume 1 caso o navio i tenha sido atendido no berço j e assume 0 caso contrário.

A função objetivo mede para cada conjunto berço/navio o valor gasto para o navio ficar parado durante o tempo de espera e o tempo de atendimento somado ao valor gasto para descarregar a carga do navio no berço somado ainda ao valor gasto para que o navio atraque no berço.

Os algoritmos genéticos são utilizados para problemas de maximização. Por se tratar de um problema de minimização, a função de aptidão (*fitness*) para o PAB é representada pela equação abaixo:

$$Fitness = \frac{1}{1+minCA} \quad (7)$$

A modificação realizada para o algoritmo genético será mantida para o algoritmo de otimização por nuvem de partículas, já que os resultados de ambos os algoritmos serão comparados posteriormente.

Essa função representa o potencial de cada solução para o problema do PAB. A partir dessa função tem-se que quanto maior o valor de *fitness*, melhor é a solução, pois o custo obtido para as alocações dos berços pelos navios é pequeno.

4.4.2 FUNÇÕES DE RESTRIÇÃO

As funções de restrição são aquelas que devem ser obedecidas para que o navio aloque o berço compatível com ele. Elas são representadas pelas equações abaixo:

$$\sum_{i \in N} (L_j - L_i) \cdot x_{ij} \geq 0 \quad (8)$$

$$\sum_{i \in N} (D_j - D_i) \cdot x_{ij} \geq 0 \quad (9)$$

$$Mcheg_i + \sum_{i \in N} A_{ij} \cdot x_{ij} - Mlib_j \geq 0 \quad (10)$$

onde:

- $Mcheg_i$: momento de chegada do navio i ;
- $Mlib_j$: momento de liberação do berço j ;
- L_j : comprimento do navio j ;
- D_i : calado do navio i ;
- D_j : profundidade do berço j .

A Equação (8) expressa que o comprimento do navio deve ser menor ou igual ao comprimento do berço. A Equação (9) expressa que o calado do navio deve ser menor ou igual à profundidade do berço. A Equação (10) expressa que o momento de chegada do navio somado ao tempo de atendimento do navio no berço deve ser maior ou igual ao momento de liberação do berço.

4.5 FUNCIONAMENTO DO ALGORITMO GENÉTICO

No algoritmo genético implementado, seguem-se os seguintes passos: inicialmente são gerados aleatoriamente os indivíduos que formarão a população inicial; com a população inicial, são selecionados dois indivíduos que participarão da reprodução; é realizada a reprodução obtendo-se dois filhos. Estes filhos podem sofrer mutação genética. Após a geração dos filhos, é realizada uma avaliação para validar os indivíduos, ou seja, somente indivíduos válidos poderão entrar na população, e a partir de então, os dois piores indivíduos são retirados da população e os dois filhos são inseridos na população da próxima geração.

Os passos acima são repetidos até um determinado número máximo de gerações (iterações). Ao chegar ao número máximo de gerações (critério de parada do algoritmo), a melhor solução para o PAB está contida no indivíduo com o melhor *fitness* entre todos os indivíduos da população final. Com o melhor indivíduo da população é recuperado o custo total e o tempo final de alocação obtido com tal indivíduo. A Figura 4.4 apresenta um fluxograma para o AG.

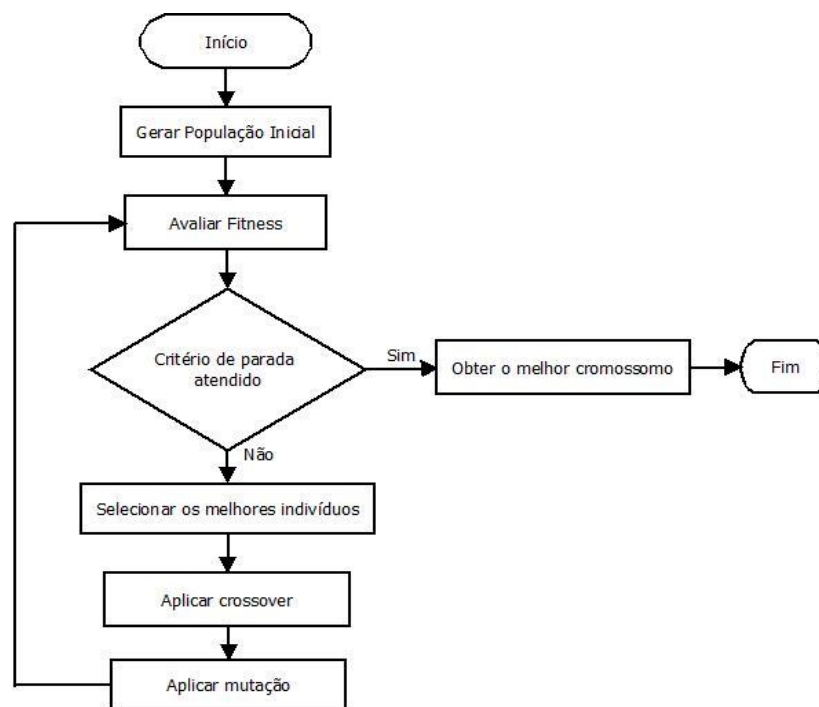


Figura 4.4 – Fluxograma para o Algoritmo Genético.

Para os testes realizados, considera-se uma taxa de cruzamento e uma taxa de mutação e os operadores obedecem a essa taxa a cada iteração. Portanto, as operações genéticas ocorrem se, e somente se, um valor aleatório sorteado for menor que o valor da taxa estabelecida para cada operador. Para os nossos testes foram utilizadas as taxas de cruzamento e de mutação sendo 90% e 10%, respectivamente. Além disso, pode ser que ocorra mutação sem ter ocorrido reprodução. Nesse caso, os dois pais selecionados passarão por mutação gerando dois filhos. Dessa forma, pode-se encontrar a melhor solução para o problema sem que seja necessário um número muito grande de iterações.

4.6 FUNCIONAMENTO DO ALGORITMO DE OTIMIZAÇÃO POR NUVEM DE PARTÍCULAS

No algoritmo de otimização por nuvem de partículas implementado, seguem-se os seguintes passos: inicialmente são gerados aleatoriamente os indivíduos que formarão a população inicial; com toda a população inicial são realizados *swaps* de três tipos que modificarão todos os indivíduos da população. Devido ao fato das operações de *swap* ocorrer de forma totalmente aleatória, para cada troca a ser realizada, é feita

uma avaliação prévia para confirmar que essa troca gerará indivíduos válidos. Portanto, ao final de todas as operações de *swap*, só resultarão indivíduos válidos.

A partir dos novos indivíduos é encontrada a melhor partícula global e também atualiza-se a lista de melhores partículas locais. Comparando-se com o AG, podemos dizer que após cada geração todos os indivíduos estarão modificados, mantendo-se sempre o melhor para continuar nas próximas iterações, sem que ocorra a eliminação de nenhum dos indivíduos.

Os passos acima são repetidos até um determinado número máximo de gerações (iteraões). Ao chegar ao número máximo de gerações (critério de parada do algoritmo), a melhor solução para o PAB está contida no indivíduo com o melhor *fitness* entre todos os indivíduos da população final. Com o melhor indivíduo da população é recuperado o custo total e o tempo final de alocação obtido com tal indivíduo. A Figura 4.5 apresenta um fluxograma para o PSO.

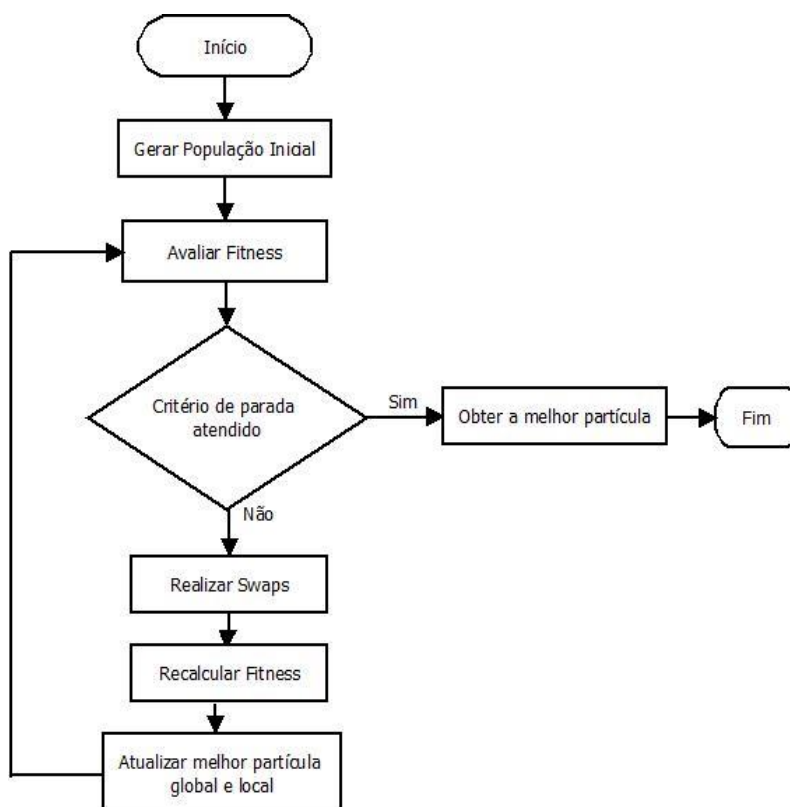


Figura 4.5 – Fluxograma para o Algoritmo de Otimização por Nuvem de Partículas.

5 RESULTADOS

Após a implementação dos dois algoritmos foi realizada uma série de testes para avaliação de ambos em alguns aspectos e, após análise dos resultados chegar-se à conclusão de qual deles é melhor algoritmo para a solução do problema aqui tratado. Inicialmente foram realizados testes para o algoritmo genético a fim de encontrar qual a combinação de operadores de *crossover* e mutação que resultam nos melhores indivíduos. Posteriormente foram realizados testes com os dois algoritmos, AG e PSO, obtendo-se os melhores resultados de ambos e comparando-os a fim de se descobrir qual é o algoritmo que se comporta melhor para a maioria dos casos.

Após a geração dos melhores resultados para cada algoritmo, as alocações dos navios nos berços foram representadas graficamente para que fosse possível avaliar qual dos algoritmos gera as melhores soluções, onde os berços são melhores ocupados, resultando no atendimento dos navios nos berços no menor período de tempo possível.

Em todos os testes são utilizadas as mesmas bases de dados, contendo os dados para 20, 30, 50 e 100 navios e para 5, 10, 20 e 30 berços. Esses dados podem ser encontrados no Apêndice A, nas Tabelas A.1 até A.8, respectivamente. E ainda, os dois parâmetros que são alterados para cada teste são o número de indivíduos da população e o número de gerações. Foram realizados testes com 20, 50 e 100 indivíduos e com 100, 500 e 1000 iterações.

Todos os resultados, discussões e conclusões obtidas com os testes realizados são apresentados nesse capítulo. Inicialmente, na seção 5.1, será apresentada a avaliação do algoritmo genético, seguido da avaliação do algoritmo de otimização por nuvem de partículas, na seção 5.2. Em seguida, a avaliação do custo computacional será apresentada na seção 5.3, e então as discussões acerca dos resultados obtidos serão apresentadas na seção 5.4.

5.1 AVALIAÇÃO DO ALGORITMO GENÉTICO

Primeiramente são detalhados os testes realizados com os algoritmos genéticos. A Tabela B encontrada no Apêndice B representa os resultados para cada combinação de navios e berços, com as variações dos dois parâmetros mencionados, o número de indivíduos e o número de gerações. Foram avaliados 126 testes com diferentes configurações. Considerando esta tabela temos a conclusão de quais são os melhores operadores para o algoritmo genético. Os melhores resultados para cada combinação de navios e berços encontrados estão representados nos testes descritos na Tabela 5.1 referenciados na Tabela B.

A partir desses resultados foram avaliados quais os operadores que geram os melhores resultados. O resultado dessa avaliação está representado pelas Figuras 5.1 e 5.2, para o operador de *crossover* e mutação, respectivamente.

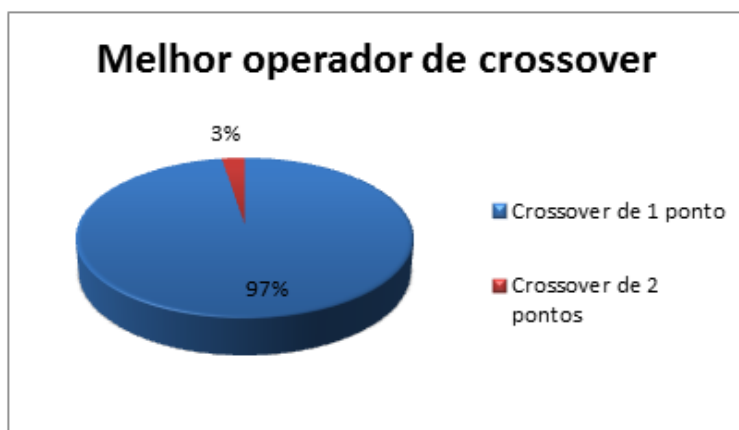


Figura 5.1 – Comparação dos operadores de *crossover*.

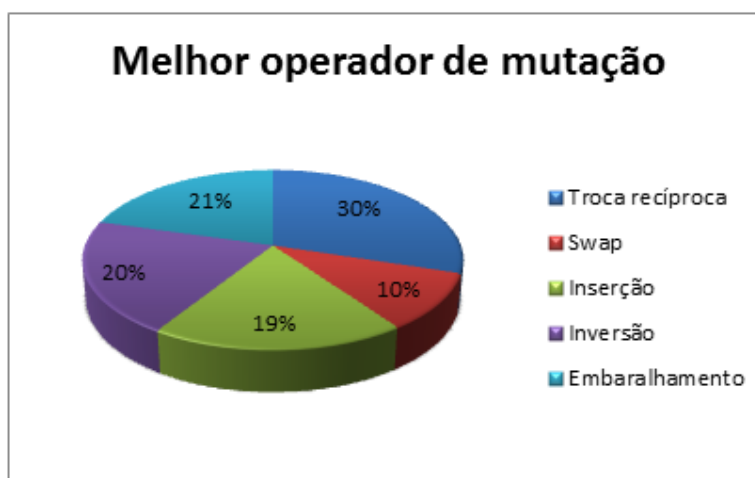


Figura 5.2 – Comparação dos operadores de mutação.

Tabela 5.1 – Melhores resultados obtidos para o algoritmo genético.

Teste	Navios	Berços	Cromossomos	Iterações	Solução	Custo Total (reais)	Tempo
6	20	5	50	1000	[4, 3, 3, 2, 2, 2, 4, 4, 2, 1, 1, 5, 4, 4, 2, 2, 4, 1, 5, 3]	104.367,91	21
15	20	10	50	1000	[2, 5, 3, 4, 4, 4, 6, 8, 2, 3, 1, 9, 10, 8, 6, 10, 2, 5, 7, 9]	103.686,21	10
27	30	5	100	1000	[2, 2, 1, 1, 2, 4, 2, 2, 4, 2, 4, 5, 3, 5, 2, 2, 4, 2, 4, 2, 2, 3, 1, 1, 3, 5, 4, 4, 4, 4]	173.964,58	39
33	30	10	50	1000	[8, 6, 7, 3, 4, 2, 4, 2, 10, 4, 8, 1, 3, 9, 6, 8, 2, 10, 6, 2, 10, 5, 1, 7, 7, 5, 8, 4, 6, 10]	171.837,13	17
39	30	20	20	1000	[18, 6, 6, 5, 12, 6, 18, 15, 9, 6, 18, 6, 15, 15, 9, 9, 12, 12, 9, 9, 12, 12, 18, 12, 15, 15, 15, 18, 18, 18]	155.775,72	25
51	50	5	50	1000	[4, 5, 2, 2, 4, 4, 4, 4, 5, 1, 2, 2, 5, 3, 2, 2, 4, 2, 4, 2, 4, 3, 3, 3, 1, 4, 2, 4, 4, 2, 4, 2, 1, 1, 1, 1, 5, 1, 4, 4, 4, 2, 2, 5, 5, 3, 5, 5, 3, 3]	293.729,13	50
60	50	10	50	1000	[7, 3, 4, 8, 6, 2, 2, 10, 2, 10, 10, 2, 3, 7, 10, 3, 8, 6, 8, 8, 6, 1, 1, 5, 9, 6, 10, 4, 4, 6, 8, 4, 9, 7, 5, 1, 1, 1, 2, 2, 2, 8, 8, 9, 7, 7, 5, 5, 9, 7]	289.661,88	26
66	50	20	20	1000	[7, 20, 18, 12, 6, 6, 12, 12, 12, 3, 11, 17, 20, 5, 9, 6, 12, 6, 6, 12, 9, 6, 11, 15, 9, 9, 6, 6, 12, 15, 6, 9, 20, 9, 9, 9, 15, 15, 12, 12, 12, 15, 15, 15, 18, 18, 18, 18, 18, 20]	267.055,69	30
78	50	30	50	1000	[17, 8, 18, 17, 28, 16, 16, 25, 28, 7, 15, 13, 11, 2, 27, 6, 19, 19, 4, 4, 7, 26, 5, 14, 8, 18, 24, 15, 15, 13, 29, 16, 23, 28, 20, 29, 23, 23, 28, 30, 28, 30, 22, 26, 26, 26, 29, 29, 23, 23]	294.363,50	25
84	100	5	20	1000	[2, 2, 2, 2, 1, 1, 1, 2, 4, 4, 2, 2, 2, 4, 1, 2, 3, 2, 5, 1, 1, 3, 2, 4, 4, 3, 4, 4, 2, 3, 1, 3, 1, 3, 5, 1, 4, 1, 2, 1, 2, 2, 4, 2, 2, 4, 4, 5, 5, 1, 3, 3, 4, 1, 4, 2, 4, 4, 4, 2, 4, 1, 3, 3, 3, 5, 4, 5, 4, 1, 1, 3, 4, 2, 2, 4, 2, 4, 4, 2, 2, 2, 4, 2, 5, 1, 5, 1, 3, 5, 5, 2, 5, 3, 5, 3, 3, 5, 3]	586.993,81	90
105	100	10	50	1000	[6, 5, 8, 1, 1, 7, 1, 6, 1, 2, 4, 2, 9, 10, 4, 6, 10, 10, 8, 3, 2, 8, 2, 8, 6, 2, 4, 4, 2, 4, 5, 3, 3, 1, 3, 1, 10, 8, 4, 2, 2, 1, 5, 6, 8, 2, 1, 7, 5, 1, 3, 1, 8, 2, 4, 6, 6, 10, 8, 5, 2, 4, 9, 3, 3, 5, 9, 1, 5, 3, 9, 7, 3, 6, 2, 8, 8, 2, 6, 10, 6, 4, 4, 10, 8, 1, 9, 5, 7, 5, 3, 5, 4, 7, 7, 9, 5, 7, 7, 9]	566.035,38	42
111	100	20	20	1000	[10, 9, 5, 9, 18, 3, 6, 15, 12, 6, 6, 15, 9, 14, 18, 9, 6, 9, 6, 9, 9, 6, 18, 20, 11, 9, 3, 6, 15, 12, 12, 6, 18, 15, 14, 3, 6, 6, 15, 3, 9, 15, 15, 9, 3, 3, 15, 5, 9, 9, 9, 14, 6, 12, 6, 15, 14, 12, 12, 18, 12, 9, 12, 6, 11, 9, 18, 15, 12, 17, 9, 9, 18, 9, 15, 9, 18, 18, 12, 14, 18, 15, 12, 15, 13, 15, 20, 17, 20, 18, 20, 18, 17, 16, 20, 20, 18, 18, 18, 18]	538.357,88	62
123	100	30	50	1000	[16, 23, 20, 29, 14, 27, 23, 27, 9, 9, 21, 28, 13, 13, 8, 8, 30, 20, 22, 10, 16, 1, 4, 28, 14, 4, 20, 9, 25, 17, 26, 23, 28, 11, 14, 24, 24, 12, 12, 25, 10, 25, 20, 7, 22, 8, 7, 8, 8, 26, 27, 11, 30, 15, 28, 18, 24, 19, 21, 16, 16, 19, 11, 15, 11, 16, 20, 23, 24, 14, 26, 17, 29, 30, 24, 19, 24, 21, 25, 27, 22, 21, 27, 24, 27, 21, 29, 29, 23, 26, 26, 22, 28, 23, 23, 29, 29, 28, 26, 25]	576.426,19	29

A Tabela 5.1 demonstra que, como era esperado, o custo total diminui com o aumento de berços. O único caso incongruente é quando se tem 30 berços, mas esse resultado acontece devido ao valor das taxas aplicadas aos navios pelos berços, que é mais alta do que os valores com 5, 10 e 20 berços. Portanto, em relação aos custos totais, o algoritmo segue os resultados previstos.

A partir dos melhores resultados do AG, apresentados na Tabela 5.1, foram representadas graficamente as alocações dos navios nos berços em cada caso, apresentados nas Figuras de 5.3 à 5.15.

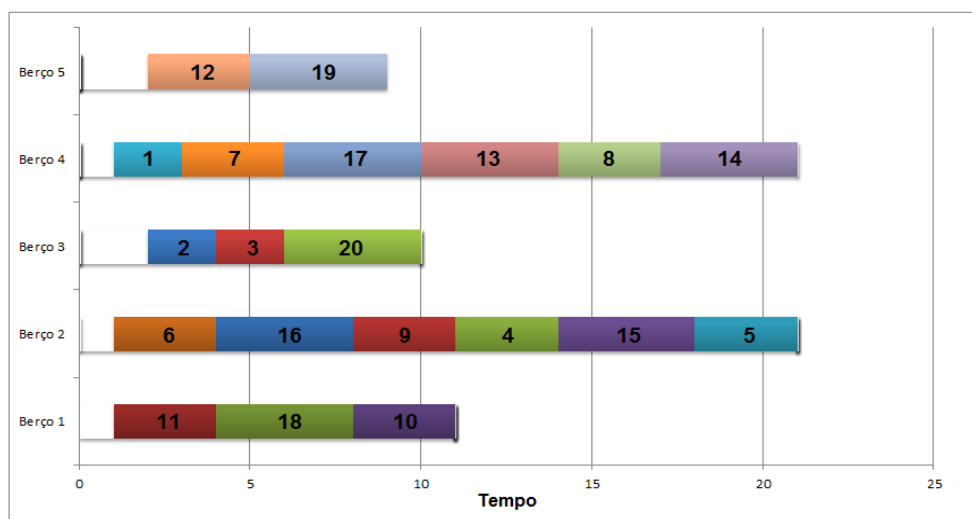


Figura 5.3 – Teste 6. Representação dos atendimentos para 20 navios, 5 berços, 50 indivíduos e 1000 gerações com AG, operador de *crossover* de 1 ponto e operador de mutação do tipo troca recíproca.

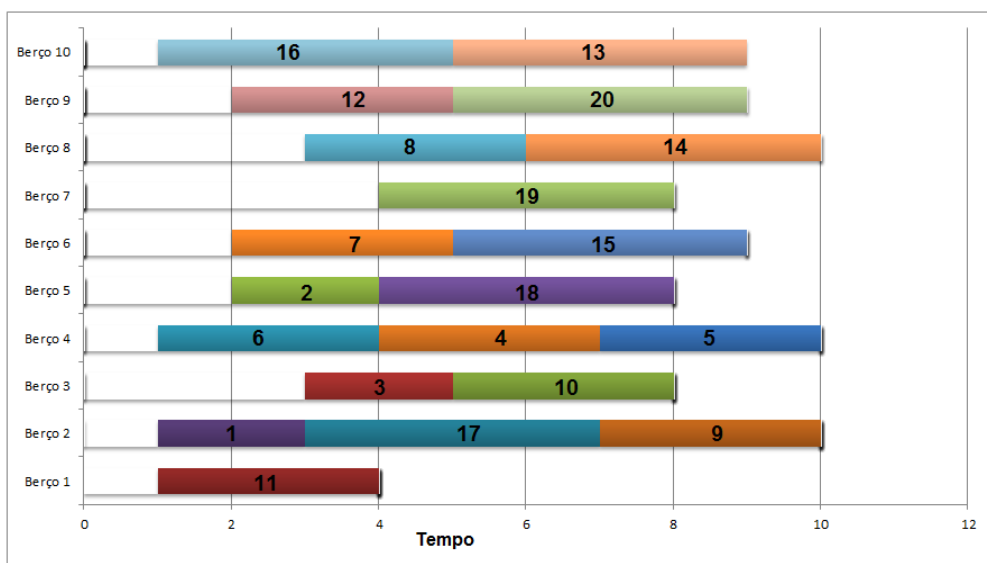


Figura 5.4 – Teste 15. Representação dos atendimentos para 20 navios, 10 berços, 50 indivíduos e 1000 gerações com AG, operador de *crossover* de 1 ponto e operador de mutação do tipo embaralhamento.

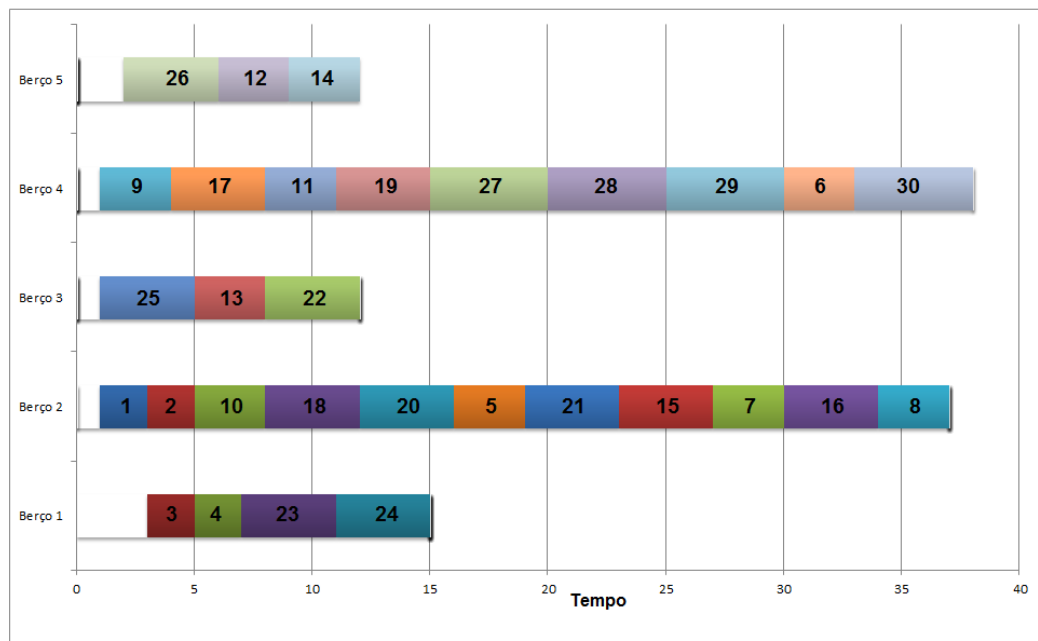


Figura 5.5 – Teste 27. Representação dos atendimentos para 30 navios, 5 berços, 100 indivíduos e 1000 gerações com AG, operador de *crossover* de 1 ponto e operador de mutação do tipo inversão.

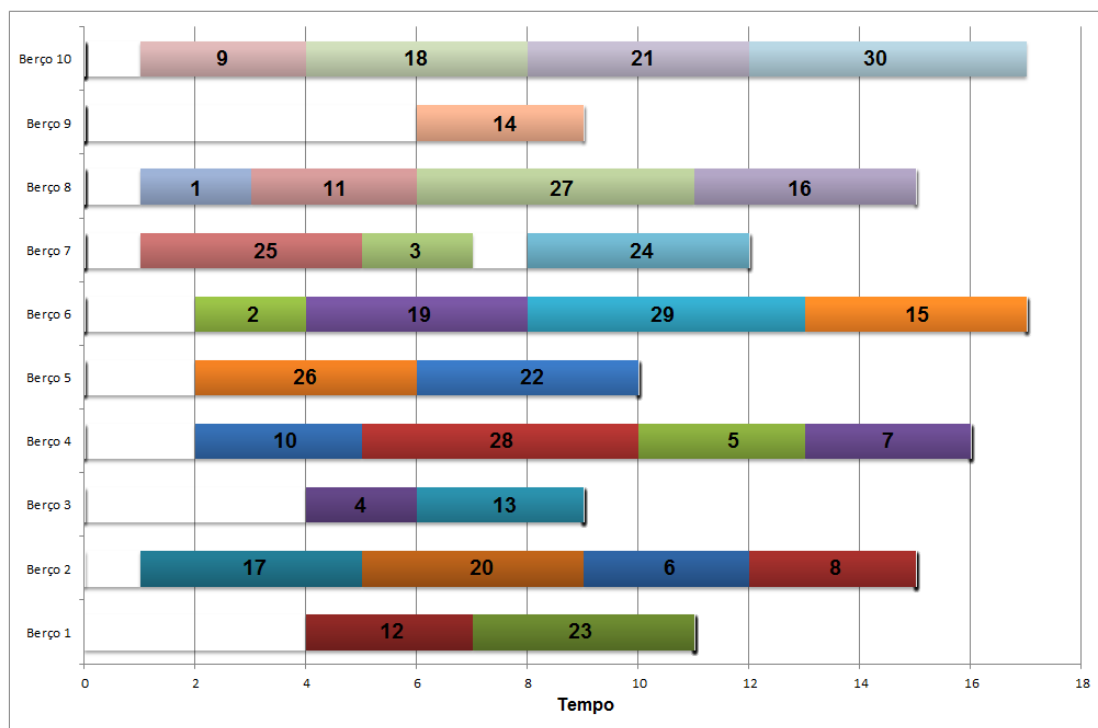


Figura 5.6 – Teste 33. Representação dos atendimentos para 30 navios, 10 berços, 50 indivíduos e 1000 gerações com AG, operador de *crossover* de 1 ponto e operador de mutação do tipo inserção.

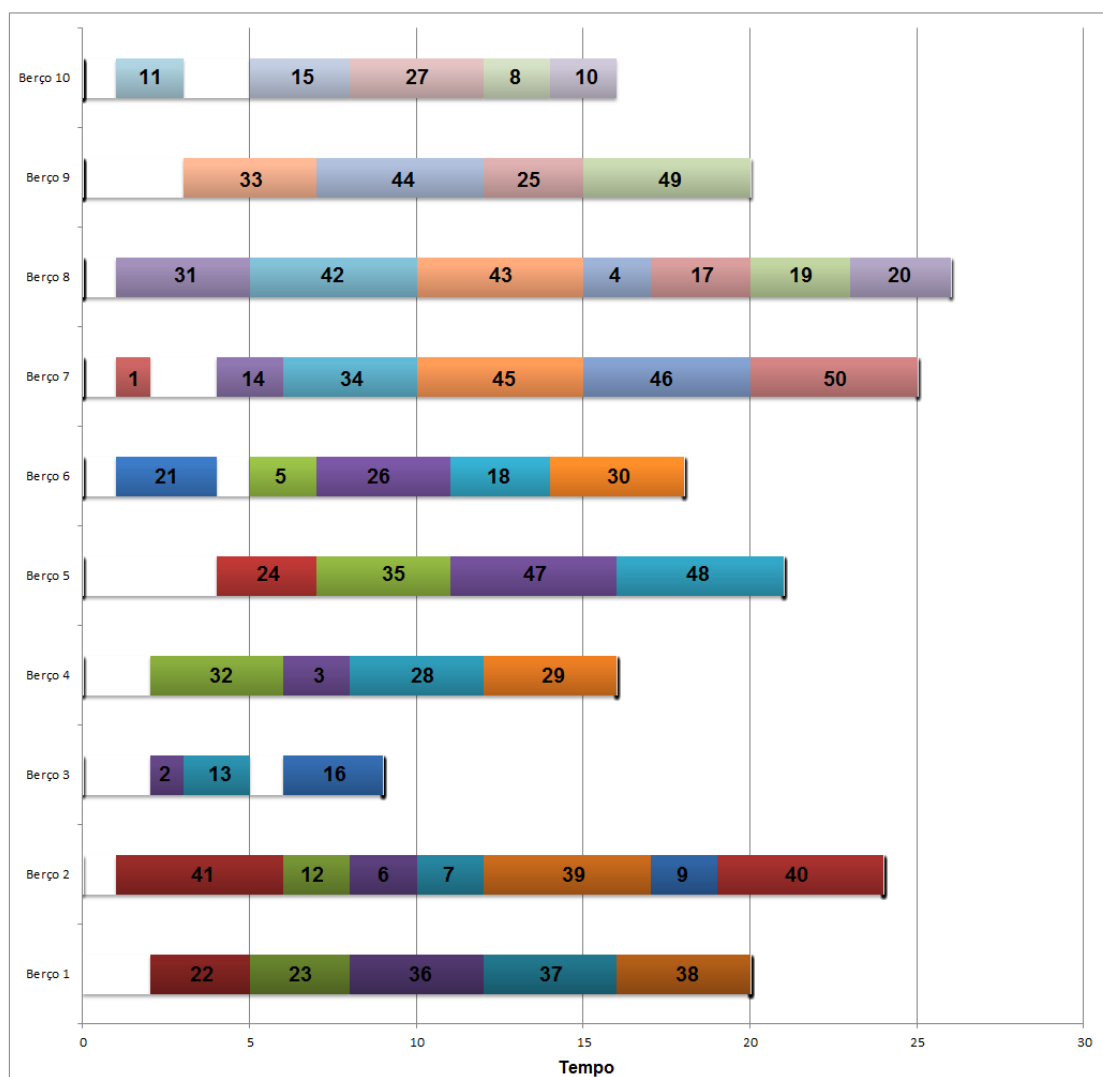


Figura 5.9 – Teste 60. Representação dos atendimentos para 50 navios, 10 berços, 50 indivíduos e 1000 gerações com AG, operador de *crossover* de 1 ponto e operador de mutação do tipo inserção.

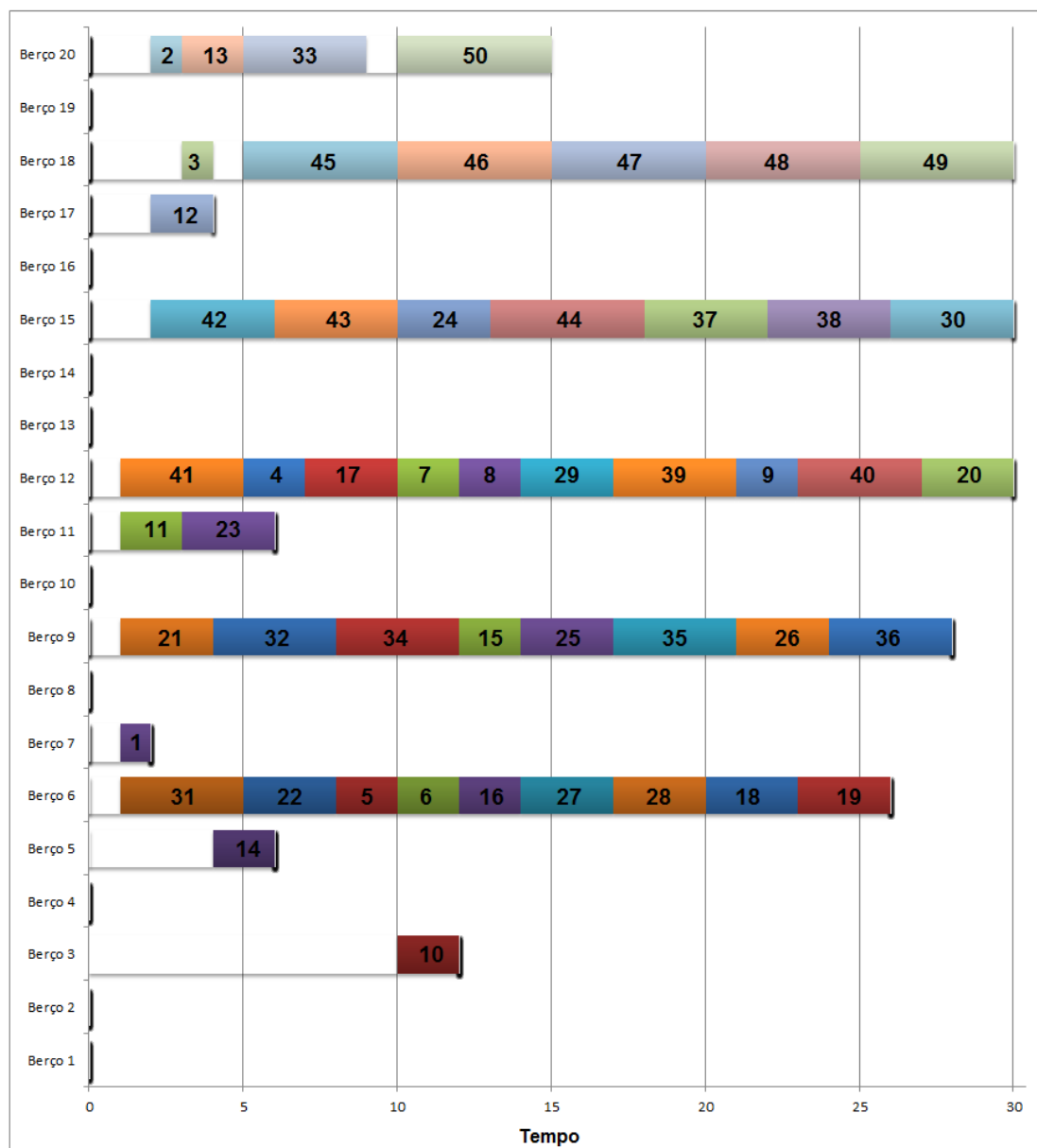


Figura 5.10 – Teste 66. Representação dos atendimentos para 50 navios, 20 berços, 50 indivíduos e 1000 gerações com AG, operador de *crossover* de 1 ponto e operador de mutação do tipo embaralhamento.

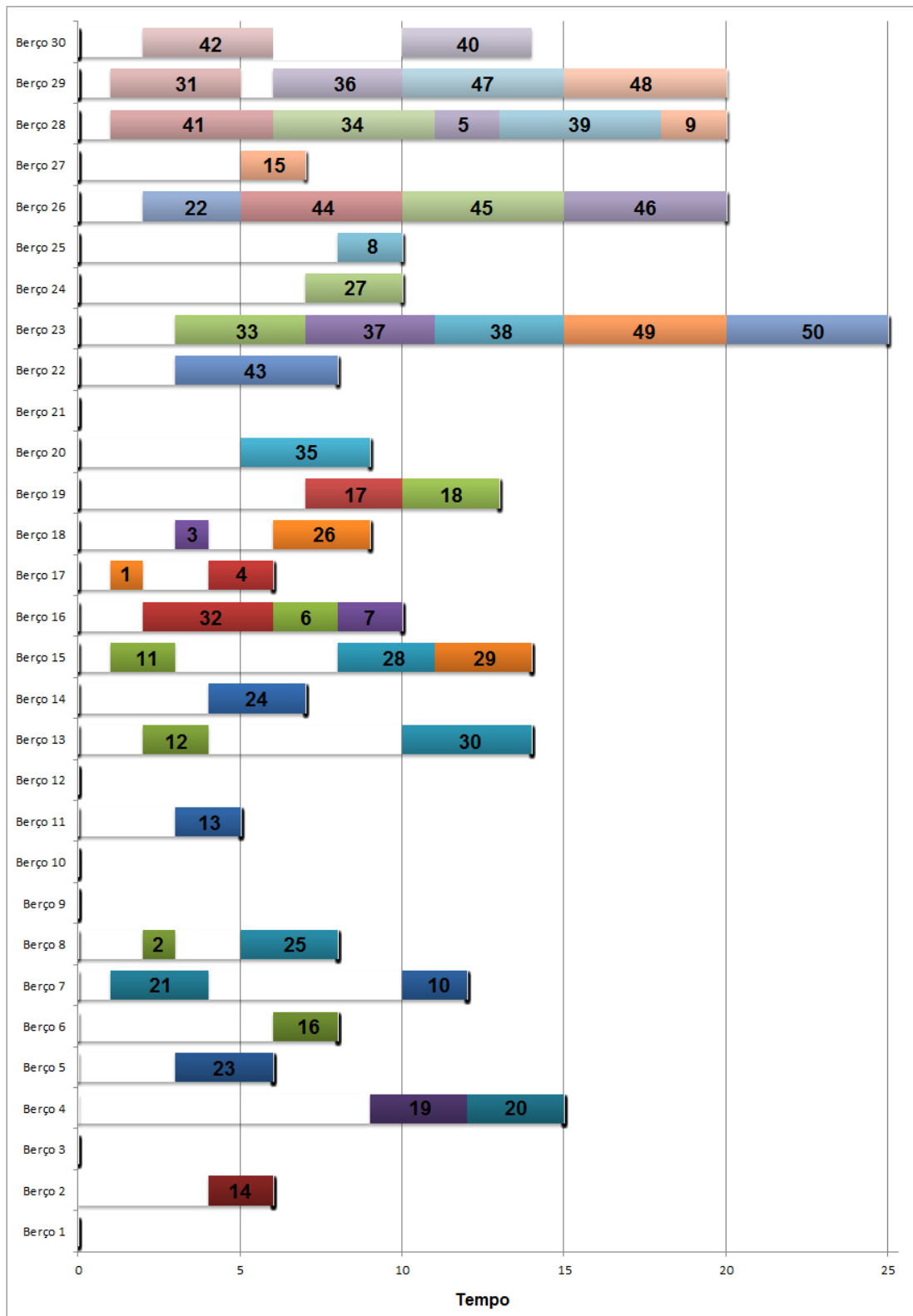


Figura 5.11 – Teste 78. Representação dos atendimentos para 50 navios, 30 berços, 50 indivíduos e 1000 gerações com AG, operador de *crossover* de 1 ponto e operador de mutação do tipo troca aleatória.

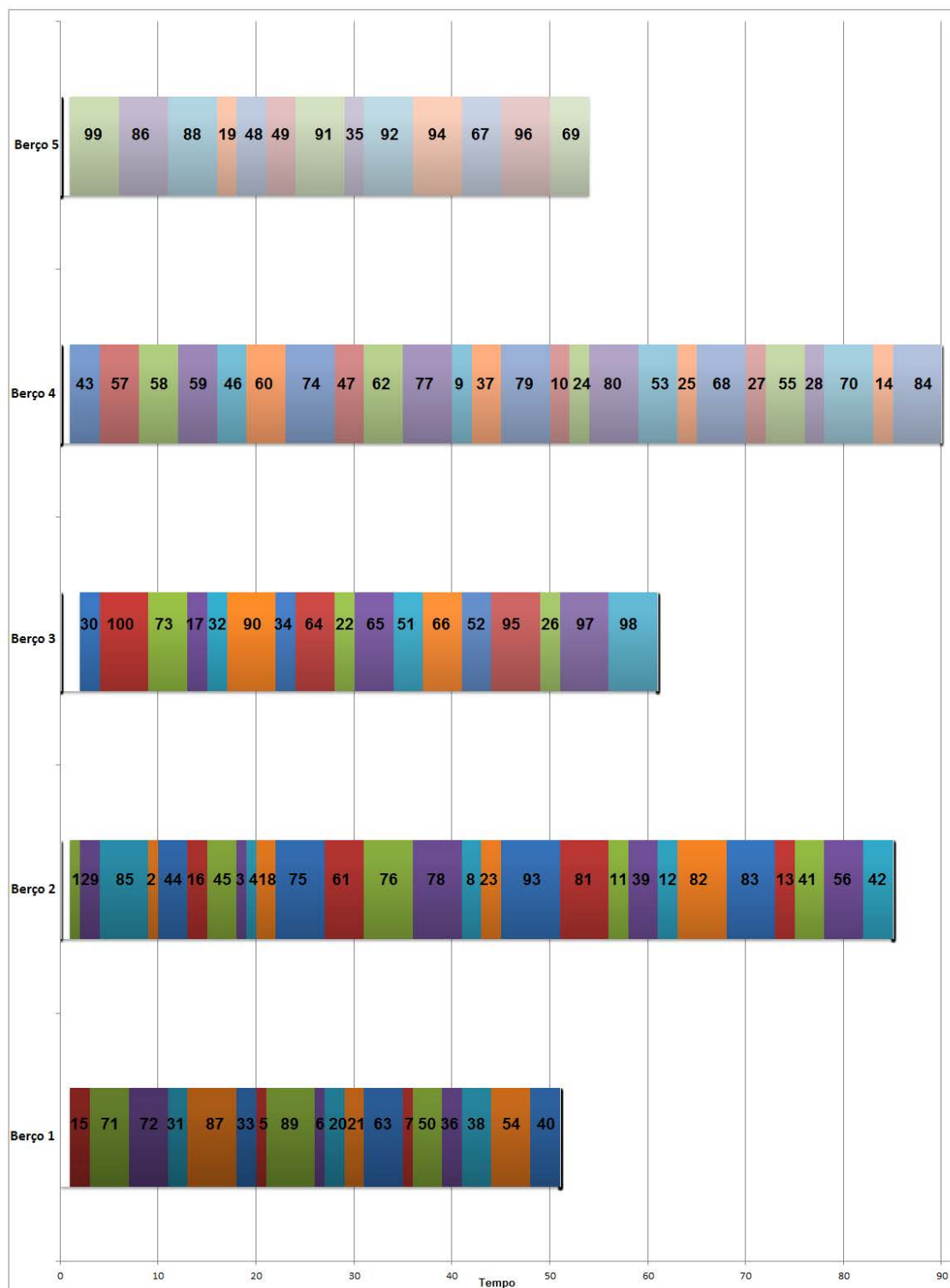


Figura 5.12 – Teste 84. Representação dos atendimentos para 100 navios, 5 berços, 20 indivíduos e 1000 gerações com AG, operador de *crossover* de 1 ponto e operador de mutação do tipo troca recíproca.

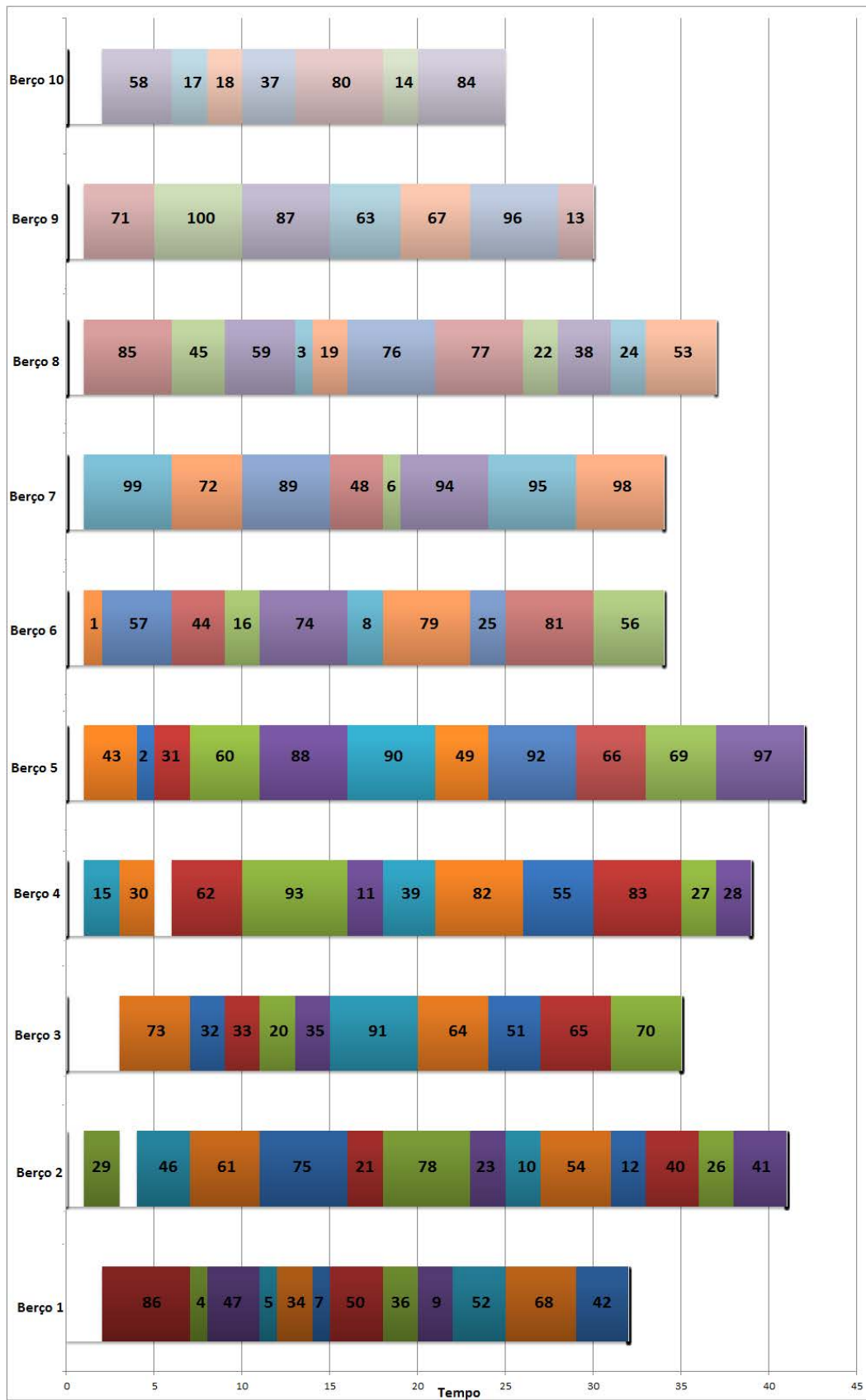


Figura 5.13 – Teste 105. Representação dos atendimentos para 100 navios, 10 berços, 20 indivíduos e 1000 gerações com AG, operador de *crossover* de 1 ponto e operador de mutação do tipo troca recíproca.

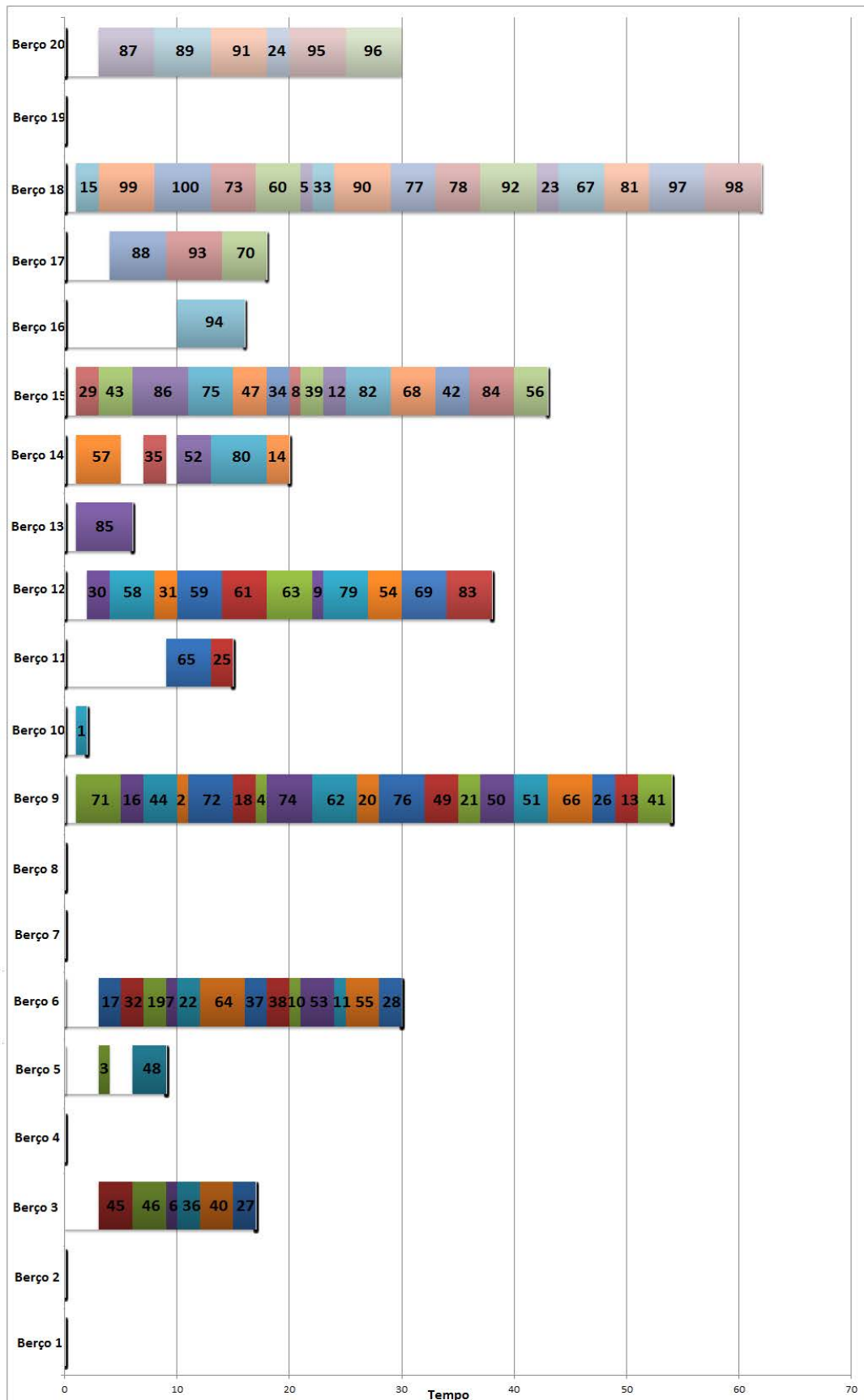


Figura 5.14 – Teste 111. Representação dos atendimentos para 100 navios, 20 berços, 20 indivíduos e 1000 gerações com AG, operador de *crossover* de 1 ponto e operador de mutação do tipo embaralhamento.

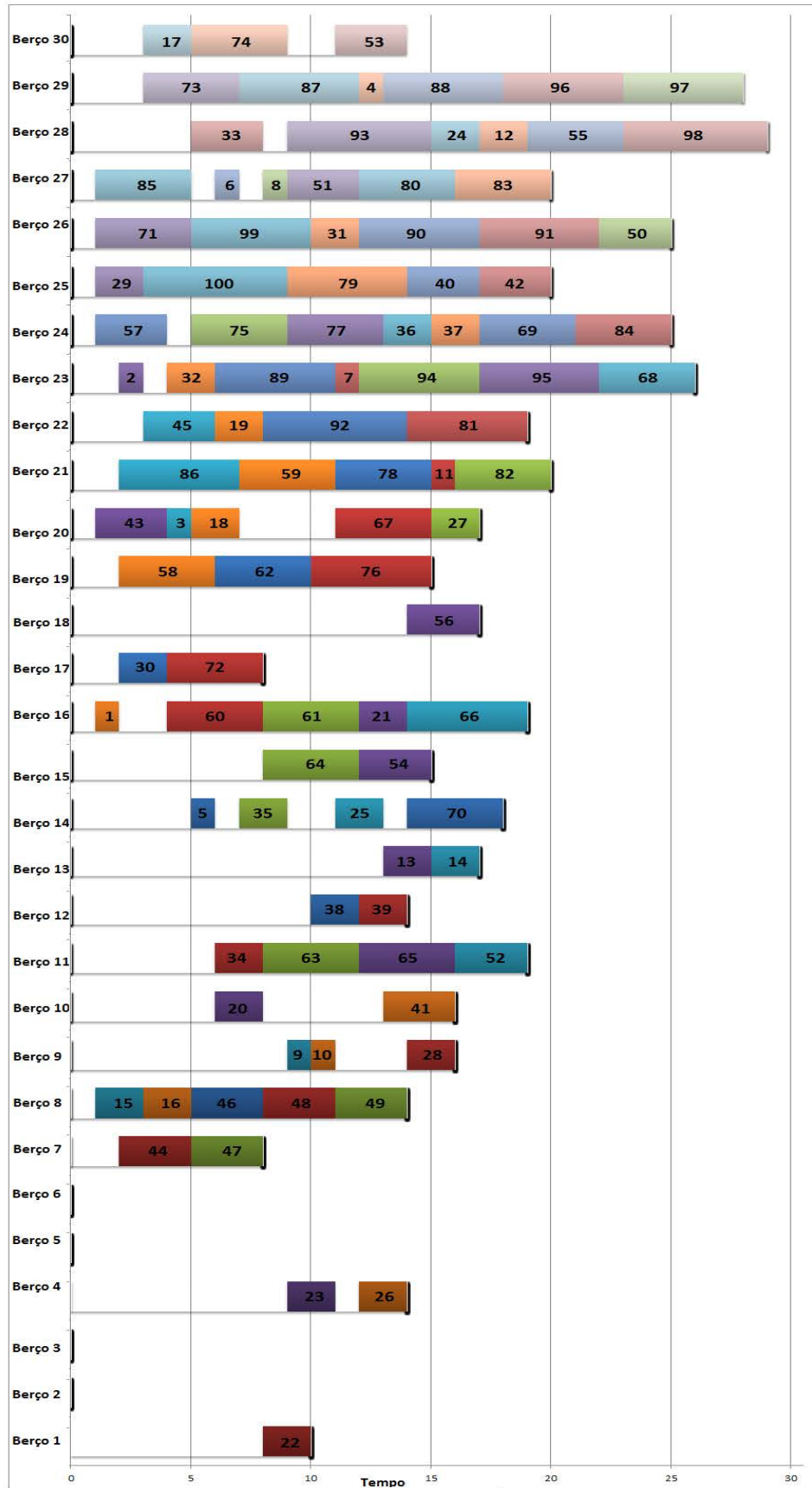


Figura 5.15 – Teste 123. Representação dos atendimentos para 100 navios, 30 berços, 50 indivíduos e 1000 gerações com AG, operador de *crossover* de 1 ponto e operador de mutação do tipo embaralhamento.

Nos gráficos que simbolizam o atendimento de 20 navios do AG, observamos que a distribuição é uniforme tanto para 5 quanto para 10 berços. Em ambos os casos, todos os berços são ocupados. A diferença apresentada é que com 5 berços, alguns berços são mais ocupados do que outros, como o berço 4 sendo ocupado por 6 navios enquanto que o berço 5 é ocupado somente por 2 navios.

Nesse caso, com 20 navios, o melhor resultado do AG acontece quando existem 10 berços, sendo que o tempo diminui 50% quando aumenta a quantidade de berços.

Nos gráficos que simbolizam o atendimento de 30 navios do AG, observamos que a distribuição está otimizada tanto para 5 quanto para 10 berços, sendo que todos os berços são ocupados. Já com 20 berços, a distribuição está incoerente, já que somente 30% dos berços estão sendo ocupados. Ainda, nota-se que com 20 berços, o tempo é maior do que com 10 berços.

Nesse caso, com 30 navios, o melhor resultado do AG acontece quando existem 10 berços, sendo que o tempo diminui 50% em relação ao caso com 5 berços e ainda é menor em relação ao caso com 20 berços.

Nos gráficos de atendimento que simbolizam o 50 navios do AG, observamos que a distribuição é uniforme tanto para 5 quanto para 10 berços, sendo que todos os berços são ocupados. Nos casos de 20 e 30 berços, a distribuição está incoerente, já que muitos berços ficam vazios, não ocorrendo a otimização do tempo total. E ainda, com 30 berços a ocupação é melhor do que com 20 berços, sendo que a quantidade de berços vazios em cada caso é de 20% e 45%, respectivamente.

Nesse caso, com 50 navios, o melhor resultado do AG acontece quando existem 30 berços, sendo competitivo com o caso com 10 berços. Com 10 berços, o tempo total diminui em relação ao tempo total com 5 berços. E então, o valor aumenta com 20 berços e cai novamente com 30 berços. Portanto, essas duas configurações apresentam bons resultados.

Nos gráficos que simbolizam o atendimento de 100 navios do AG, observamos que a ocupação está coerente tanto para 5 quanto para 10 berços, sendo que todos os berços estão ocupados. No caso de 20 berços, a distribuição dos navios nos berços não ocorre de forma uniforme, já que 30% dos berços estão vazios. E ainda, no caso de 30 berços, a distribuição não está totalmente uniforme, mas somente 14% dos berços estão vazios.

Nesse caso, com 100 navios, o melhor resultado do AG acontece quando existem 30 berços. O tempo total diminui consideravelmente nessa situação. A configuração mais competitiva seria a de 10 berços, onde o tempo total é aproximadamente 40% maior do que o tempo total com 30 berços, mas ainda é 50% menor do que o tempo total com 5 berços.

Portanto, para os casos de 20 ou 30 navios, a ocupação está uniformizada com 5 e 10 berços, sendo que o caso com 10 berços é o mais otimizado, apresentando o menor tempo final. No caso de 30 navios e 20 berços, muitos berços não são ocupados, não havendo otimização do tempo. Para os casos de 50 ou 100 navios, a ocupação ocorre de forma mais uniformizada com 5 e 10 berços, embora o tempo total seja maior considerando-se a quantidade de berços. Nos casos com 20 e 30 berços, muitos berços ficam vazios, sendo que com 30 berços os resultados são melhores, havendo uma melhor distribuição do que com 20 berços, e a consequente otimização do tempo total.

5.2 AVALIAÇÃO DO PSO

Aqui são detalhados os testes realizados com o algoritmo de otimização por nuvem de partículas. A Tabela 5.2 abaixo representa os resultados encontrados para cada combinação de navios e berços, com os números de partículas e iterações realizadas para chegar-se aos resultados.

A partir da Tabela 5.2 podemos confirmar o pressuposto de que o custo total diminui com o aumento de berços. O único caso que difere do esperado é quando se tem 30 berços, mas esse resultado acontece devido ao valor das taxas de movimentação e taxa de atracação dos berços, que é mais alta do que os valores com 5, 10 e 20 berços. Portanto, para os custos totais pode-se dizer que o algoritmo segue claramente o proposto.

A partir dos resultados do PSO apresentados na Tabela 5.2, foram representadas graficamente as alocações dos navios nos berços em cada caso, apresentadas nas Figuras de 5.16 à 5.28.

Tabela 5.2 – Resultados obtidos para o PSO.

Navios	Berços	Partículas	Iterações	Solução	Custo Total (reais)	Tempo
20	5	100	500	[4, 5, 1, 2, 2, 3, 2, 3, 2, 1, 5, 5, 2, 2, 4, 1, 4, 3, 3, 5]	106.134,09	22
20	10	100	1000	[4, 9, 9, 8, 4, 2, 8, 7, 7, 9, 9, 5, 8, 4, 2, 8, 6, 7, 7, 9]	105.088,50	17
30	5	100	100	[4, 2, 1, 5, 4, 2, 1, 2, 4, 5, 4, 5, 1, 3, 4, 2, 1, 4, 4, 4, 2, 5, 5, 1, 3, 5, 3, 4, 4, 4]	176.729,39	43
30	10	100	500	[6, 8, 9, 1, 6, 4, 10, 2, 8, 10, 8, 1, 5, 5, 6, 10, 4, 6, 4, 1, 2, 9, 9, 3, 4, 8, 8, 10, 7, 8]	175.858,70	24
30	20	100	1000	[9, 8, 12, 8, 15, 12, 19, 16, 16, 14, 19, 12, 6, 12, 6, 18, 6, 14, 12, 20, 15, 18, 12, 12, 17, 18, 18, 18, 18, 20]	167.045,52	25
50	5	100	100	[3, 1, 2, 3, 3, 2, 4, 5, 1, 5, 4, 4, 1, 1, 2, 4, 1, 2, 5, 5, 2, 5, 2, 3, 1, 4, 4, 4, 2, 4, 2, 1, 4, 3, 3, 5, 3, 4, 2, 2, 2, 4, 4, 3, 2, 4, 5, 3, 5, 3]	300.801,60	52
50	10	100	500	[5, 9, 6, 4, 10, 7, 8, 6, 8, 4, 6, 6, 5, 5, 1, 7, 7, 4, 6, 9, 1, 5, 9, 7, 4, 8, 2, 2, 8, 4, 8, 2, 10, 3, 10, 3, 5, 2, 8, 6, 4, 8, 4, 3, 10, 9, 5, 5, 9, 9]	295.470,90	27
50	20	50	100	[9, 10, 18, 14, 18, 17, 15, 18, 12, 17, 10, 17, 14, 18, 18, 13, 15, 20, 18, 11, 17, 12, 15, 15, 15, 20, 18, 17, 15, 19, 16, 14, 9, 17, 20, 12, 18, 12, 18, 20, 15, 18, 15, 15, 17, 17, 18, 17, 18, 18]	287.584,30	44
50	30	100	100	[26, 4, 26, 29, 17, 22, 15, 16, 26, 20, 27, 19, 21, 18, 6, 21, 29, 22, 17, 17, 29, 8, 27, 26, 26, 21, 18, 25, 22, 23, 28, 19, 22, 17, 29, 16, 23, 20, 30, 25, 27, 25, 24, 29, 27, 23, 26, 29, 23, 29]	303.734,50	28
100	5	50	1000	[3, 5, 5, 5, 5, 4, 1, 5, 4, 2, 4, 4, 4, 2, 2, 2, 1, 5, 1, 5, 4, 5, 3, 2, 3, 2, 4, 1, 5, 4, 5, 2, 1, 2, 1, 1, 3, 5, 1, 2, 3, 2, 4, 3, 3, 1, 2, 5, 5, 5, 4, 3, 5, 1, 1, 2, 5, 4, 2, 1, 4, 1, 3, 3, 5, 4, 5, 4, 1, 3, 3, 1, 3, 4, 2, 3, 4, 3, 2, 2, 2, 2, 2, 1, 3, 3, 5, 3, 3, 3, 4, 5, 5, 3, 5, 4, 5, 3, 3]	601.725,30	91
100	10	100	1000	[8, 10, 4, 2, 9, 1, 7, 8, 8, 9, 5, 4, 4, 2, 4, 4, 1, 10, 6, 1, 2, 4, 10, 9, 9, 10, 2, 7, 4, 10, 1, 1, 1, 4, 8, 2, 2, 8, 8, 2, 10, 7, 1, 9, 2, 8, 4, 5, 4, 3, 7, 9, 6, 4, 9, 1, 1, 5, 2, 9, 4, 10, 3, 2, 6, 9, 3, 5, 4, 5, 7, 10, 8, 4, 1, 2, 2, 10, 6, 1, 10, 6, 10, 10, 8, 3, 7, 10, 6, 7, 9, 9, 9, 7, 10, 9, 8, 9, 7, 9]	583.854,20	57
100	20	100	1000	[18, 20, 20, 20, 19, 9, 18, 19, 8, 18, 19, 14, 6, 18, 7, 16, 9, 14, 11, 9, 9, 13, 6, 12, 19, 1, 14, 12, 17, 9, 13, 8, 17, 1, 14, 18, 11, 18, 20, 18, 13, 19, 14, 5, 6, 15, 18, 9, 12, 9, 13, 11, 18, 14, 11, 8, 12, 12, 16, 12, 17, 12, 12, 12, 15, 15, 12, 15, 9, 8, 14, 12, 15, 19, 15, 13, 15, 15, 20, 18, 12, 12, 15, 18, 12, 12, 19, 18, 20, 20, 15, 20, 18, 20, 19, 18, 20, 18, 18, 19]	577.099,44	56
100	30	100	100	[22, 12, 8, 12, 26, 20, 3, 27, 25, 19, 28, 25, 8, 17, 20, 24, 10, 21, 27, 22, 16, 30, 15, 23, 16, 1, 26, 27, 13, 7, 23, 18, 29, 25, 23, 11, 18, 30, 3, 16, 20, 12, 19, 24, 30, 17, 24, 18, 30, 23, 7, 18, 9, 15, 8, 25, 22, 18, 19, 29, 22, 29, 14, 23, 23, 21, 13, 24, 14, 21, 21, 17, 29, 19, 16, 16, 18, 24, 22, 28, 20, 24, 22, 30, 24, 20, 20, 19, 26, 29, 24, 23, 24, 23, 27, 29, 26, 30, 26, 29]	595.645,00	35

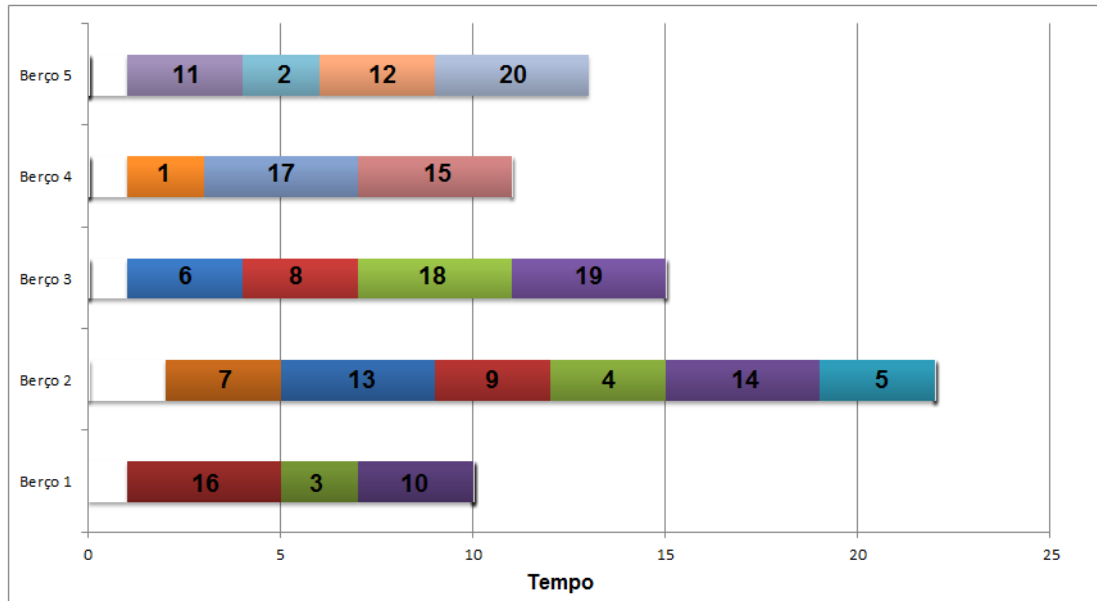


Figura 5.16 – Representação dos atendimentos para 20 navios, 5 berços, 100 partículas e 500 iterações com PSO.

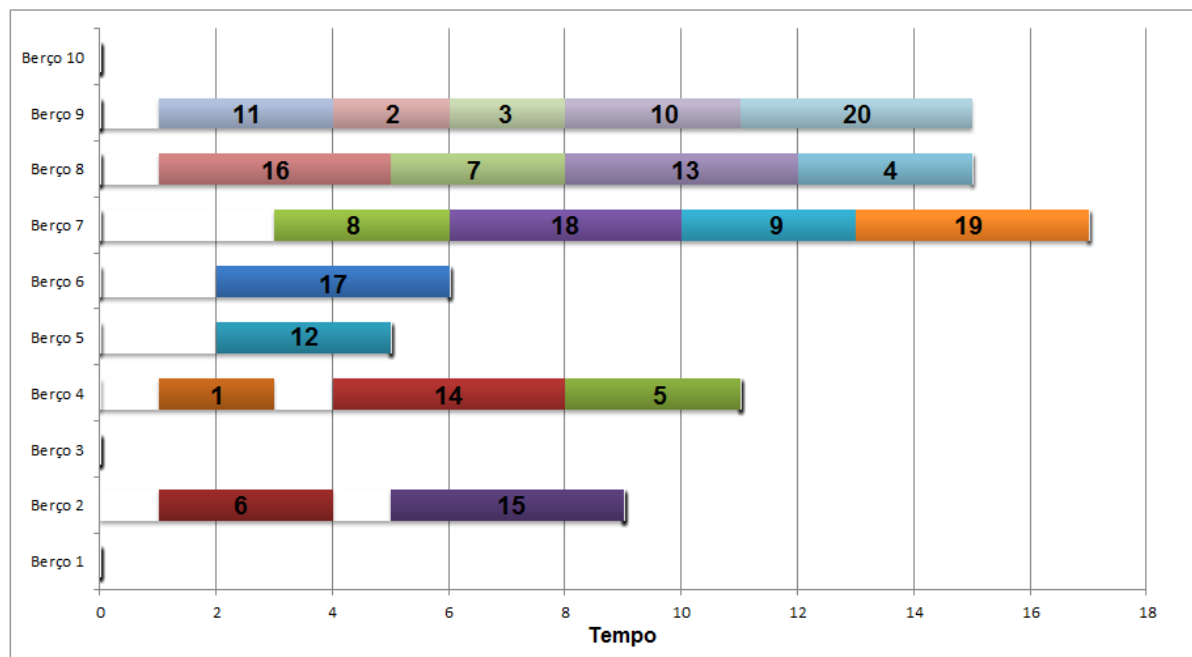


Figura 5.17 – Representação dos atendimentos para 20 navios, 10 berços, 100 partículas e 1000 iterações com PSO.

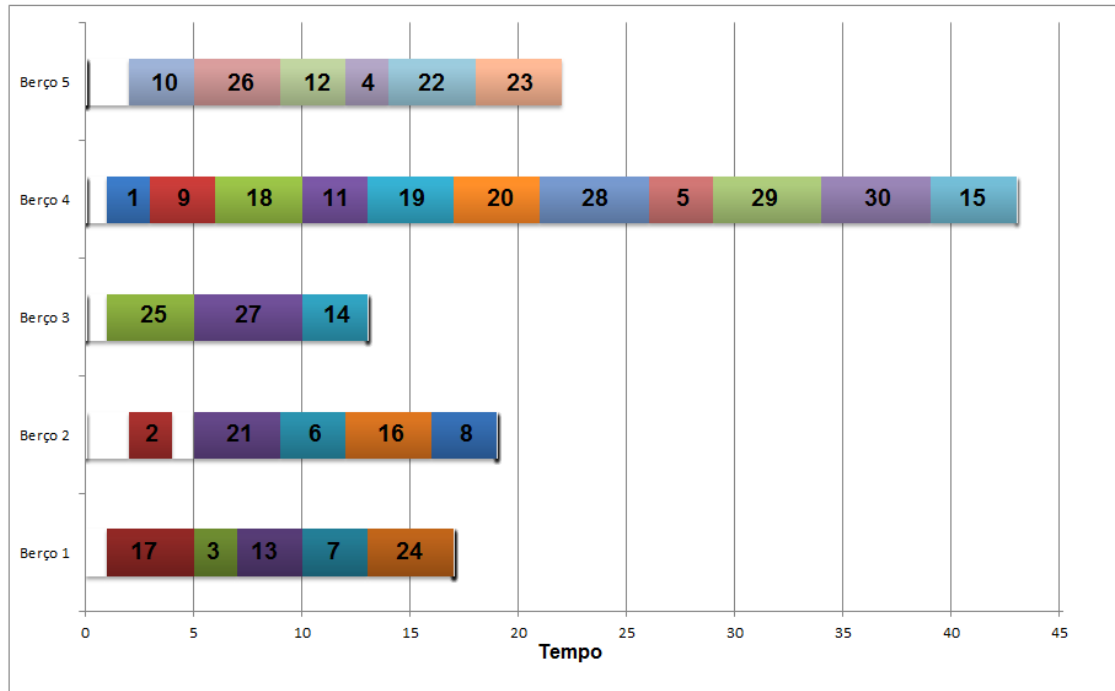


Figura 5.18 – Representação dos atendimentos para 30 navios, 5 berços, 100 partículas e 100 iterações com PSO.

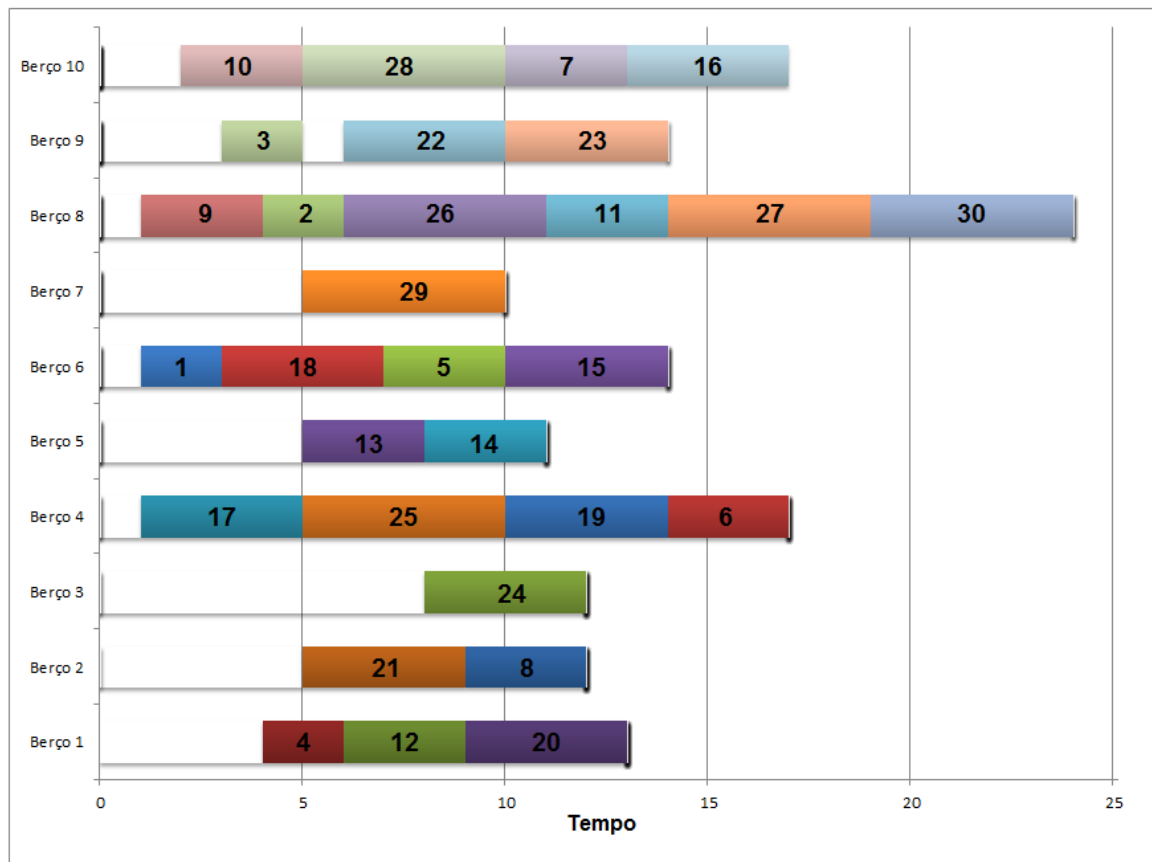


Figura 5.19 – Representação dos atendimentos para 30 navios, 10 berços, 100 partículas e 500 iterações com PSO.

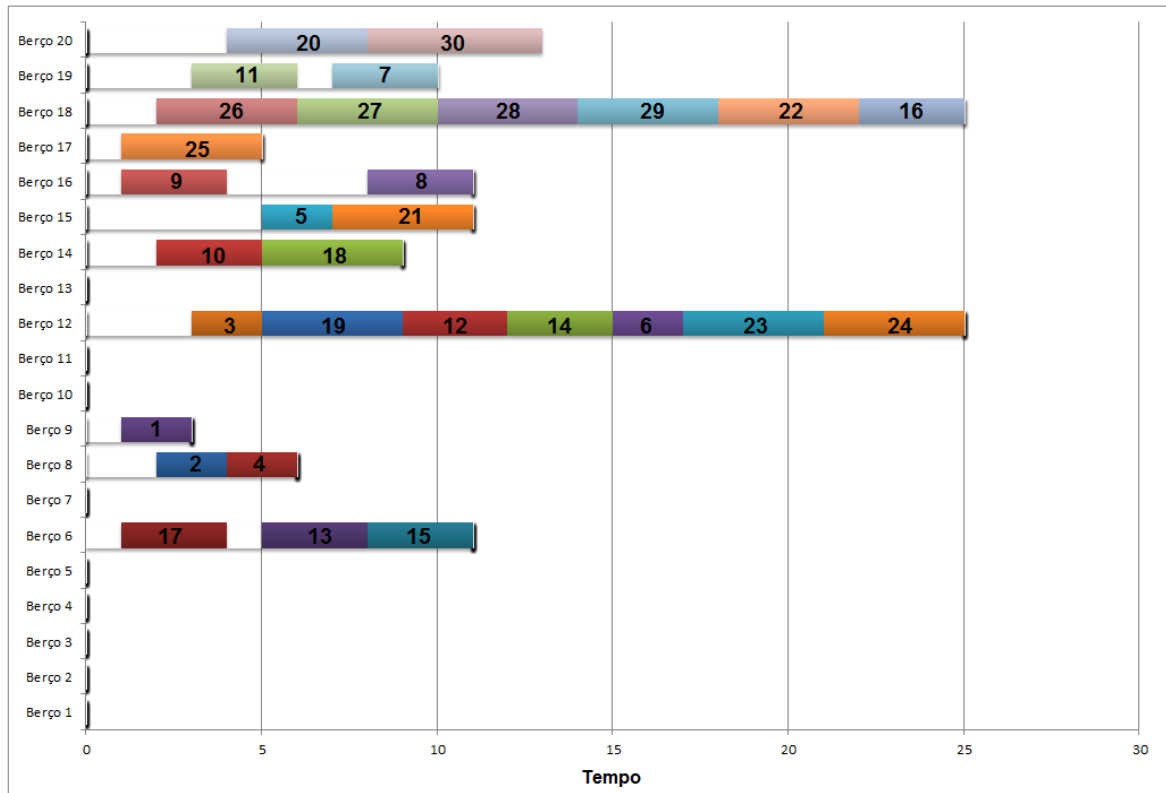


Figura 5.20 – Representação dos atendimentos para 30 navios, 20 berços, 100 partículas e 1000 iterações com PSO.

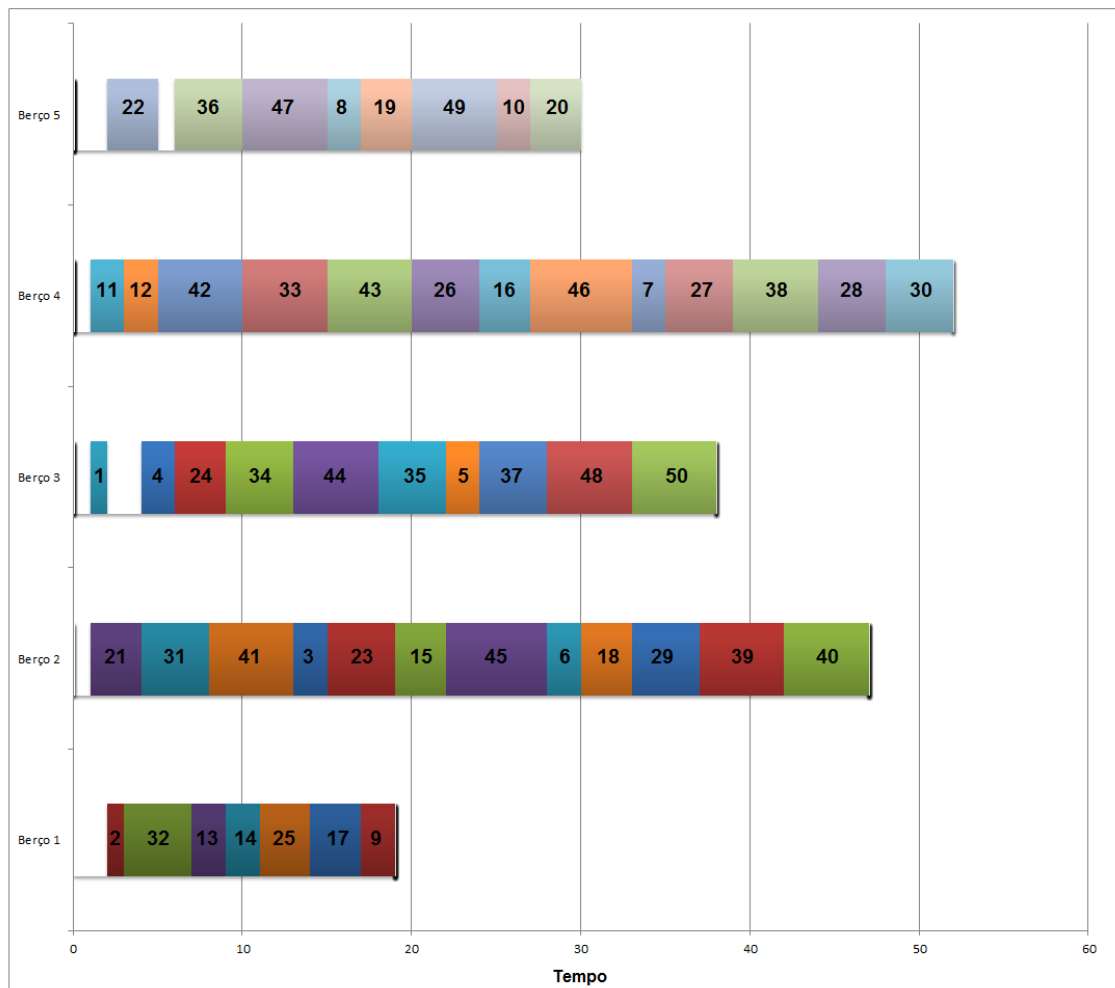


Figura 5.21 – Representação dos atendimentos para 50 navios, 5 berços, 100 partículas e 100 iterações com PSO.

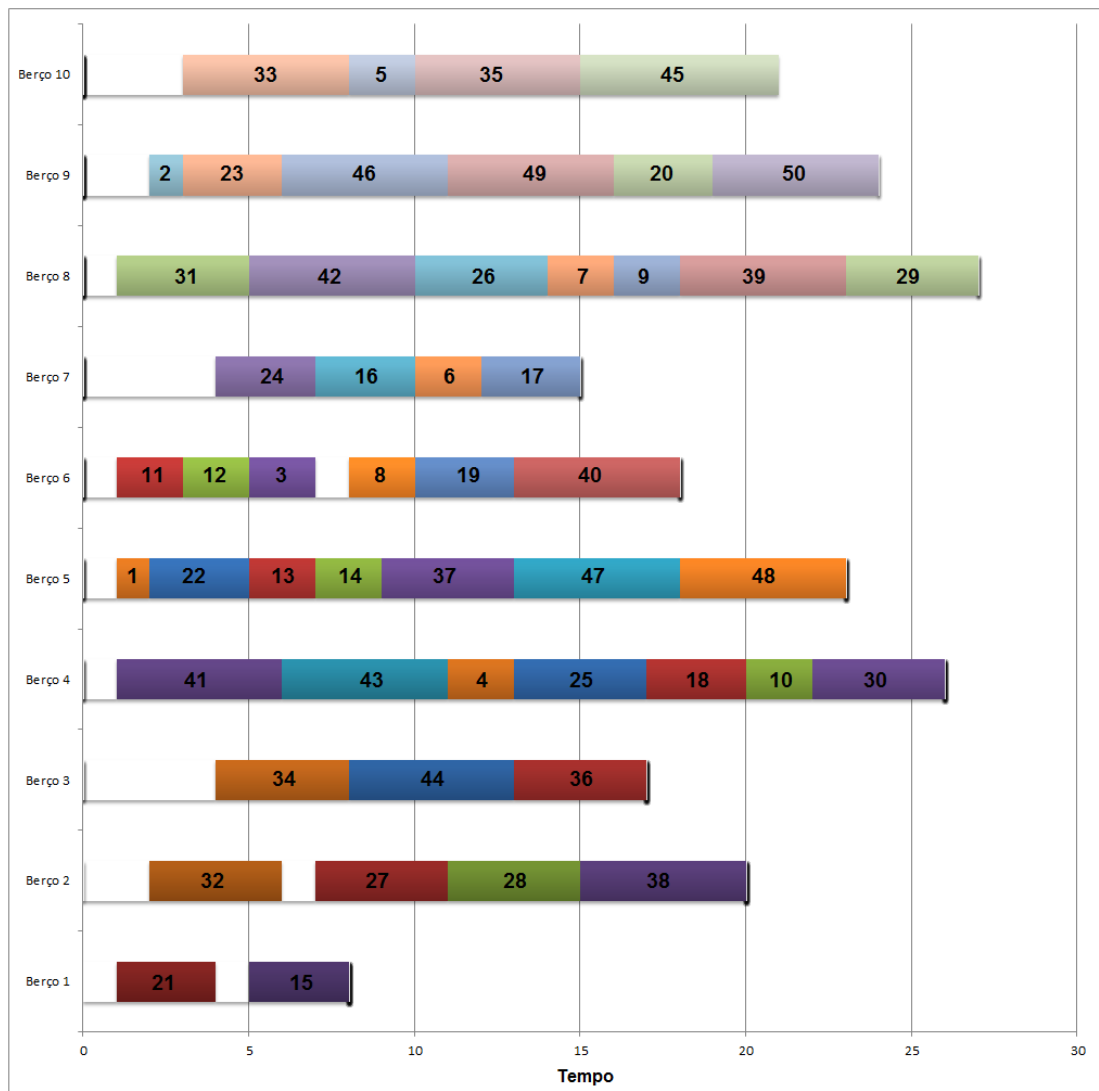


Figura 5.22 – Representação dos atendimentos para 50 navios, 10 berços, 100 partículas e 500 iterações com PSO.

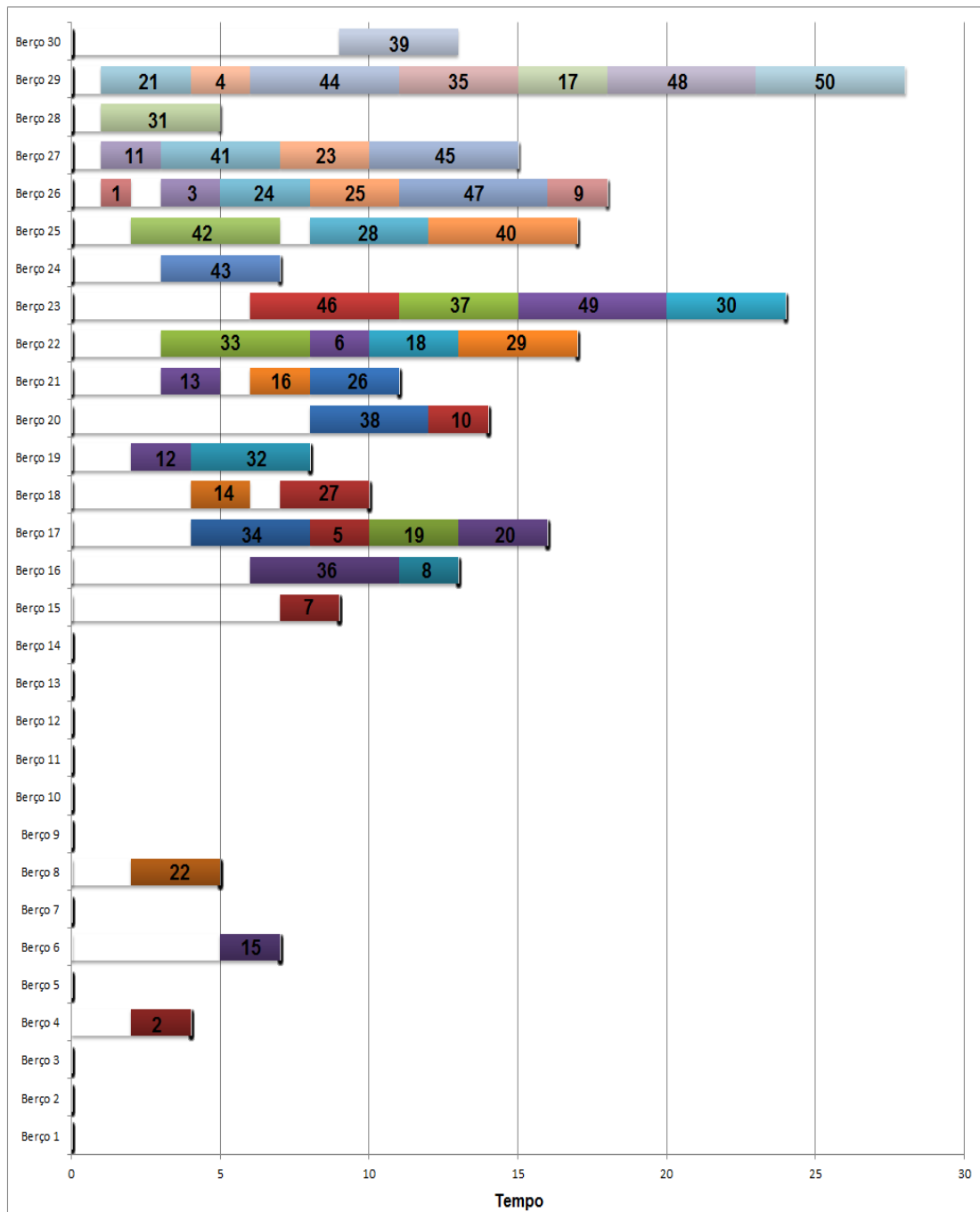


Figura 5.24 – Representação dos atendimentos para 50 navios, 30 berços, 100 partículas e 100 iterações com PSO.

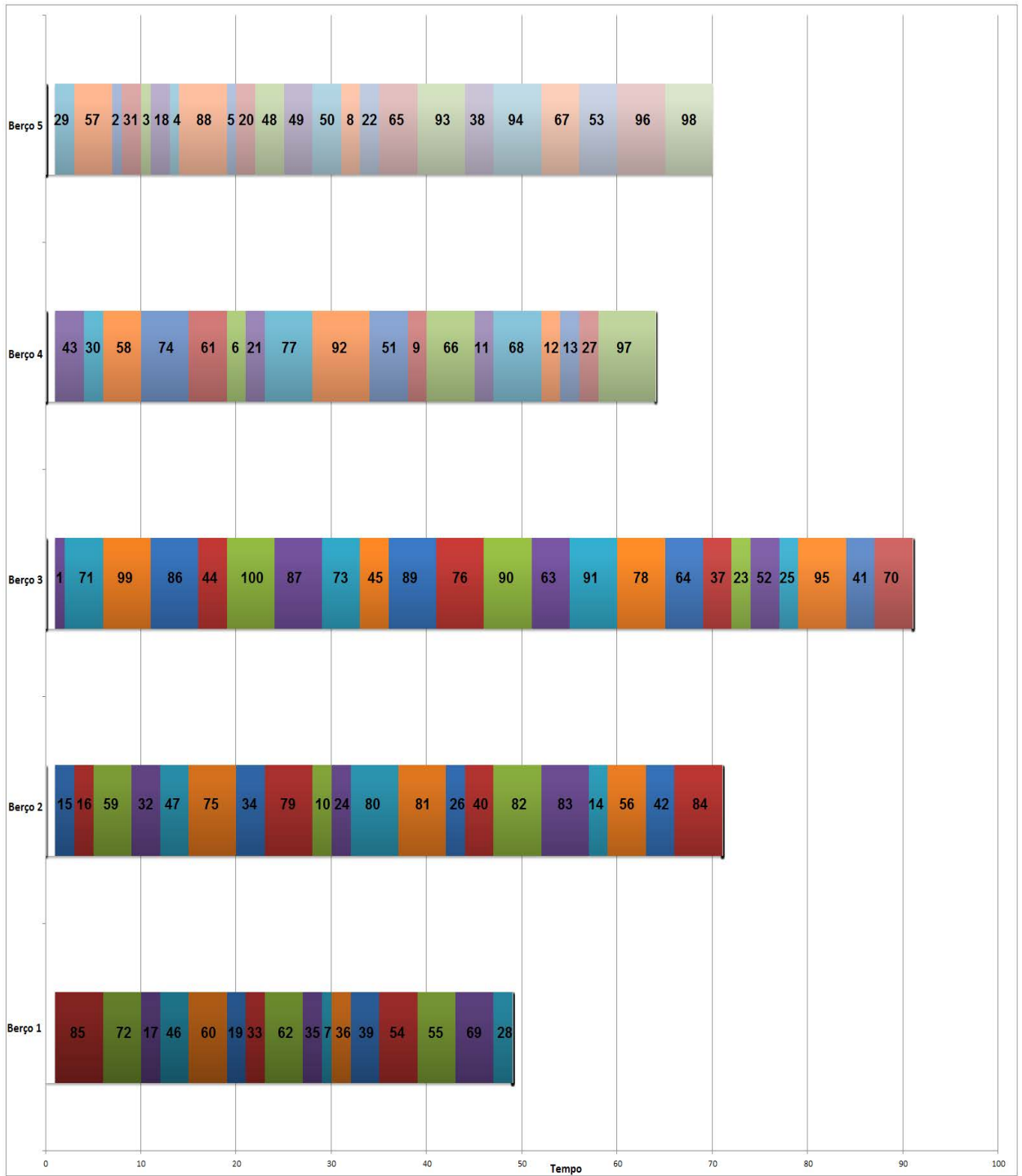


Figura 5.25 – Representação dos atendimentos para 100 navios, 5 berços, 50 partículas e 1000 iterações com PSO.

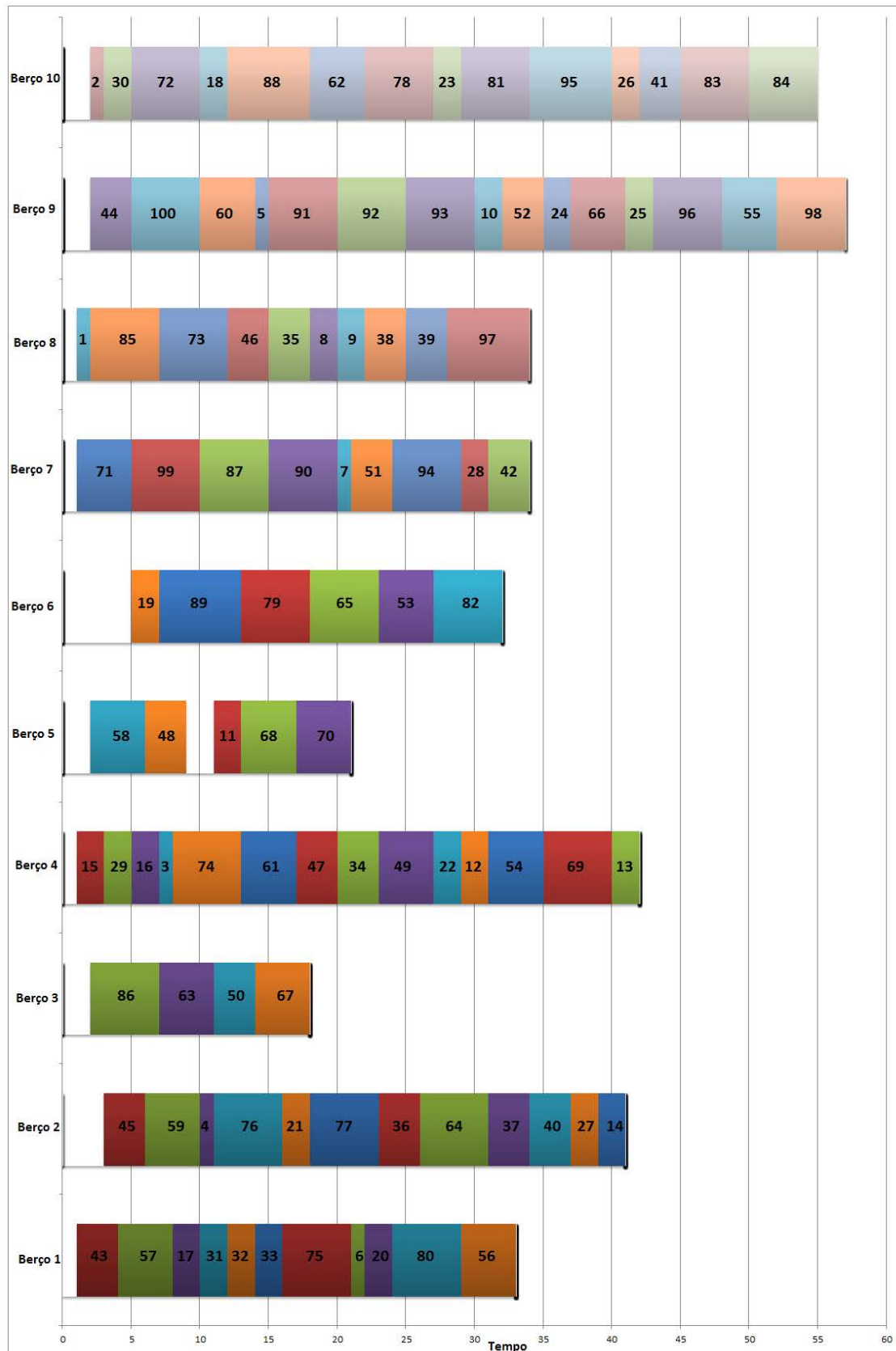


Figura 5.26 – Representação dos atendimentos para 100 navios, 10 berços, 100 partículas e 1000 iterações com PSO.

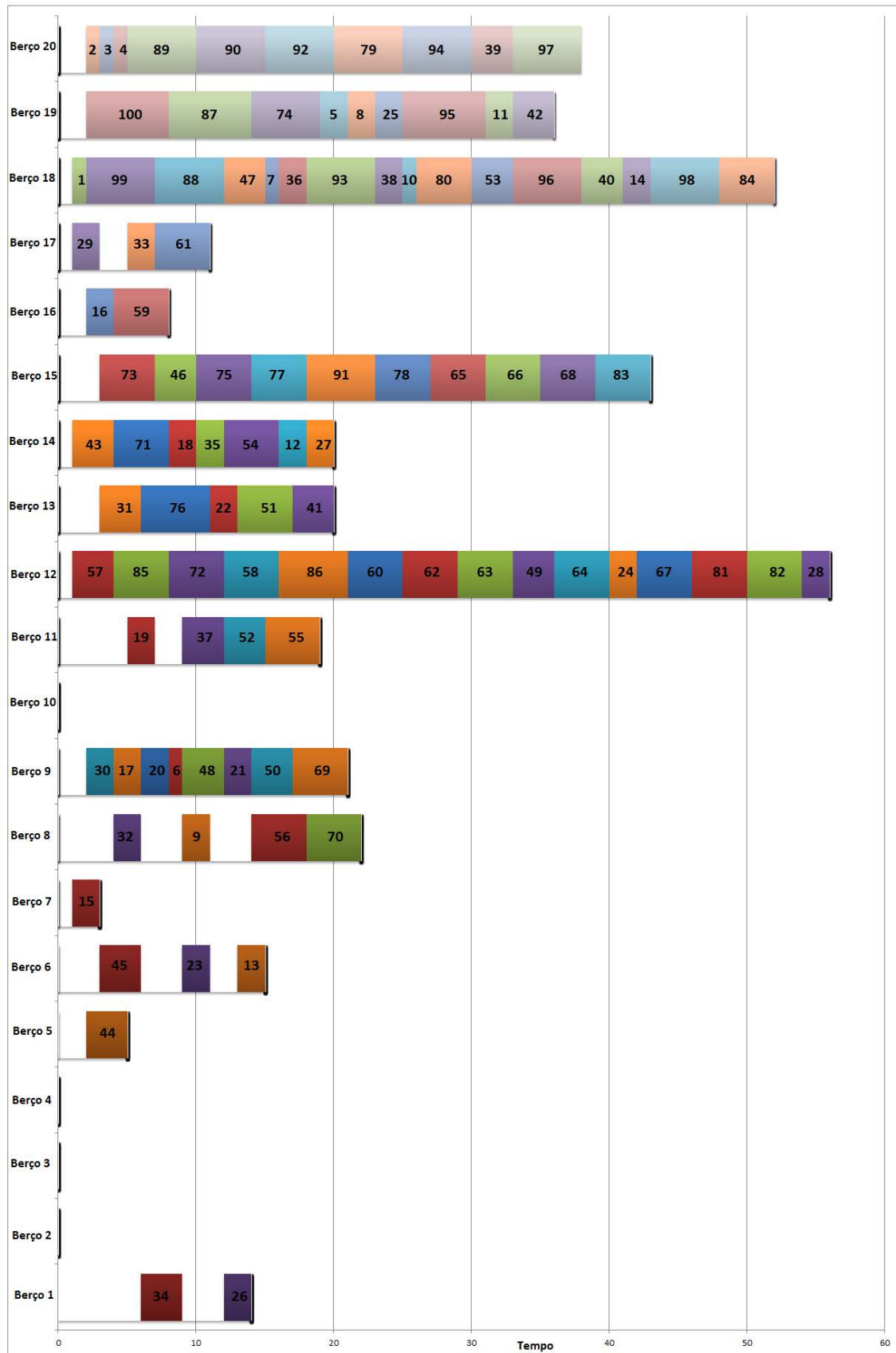


Figura 5.27 – Representação dos atendimentos para 100 navios, 20 berços, 100 partículas e 1000 iterações com PSO.

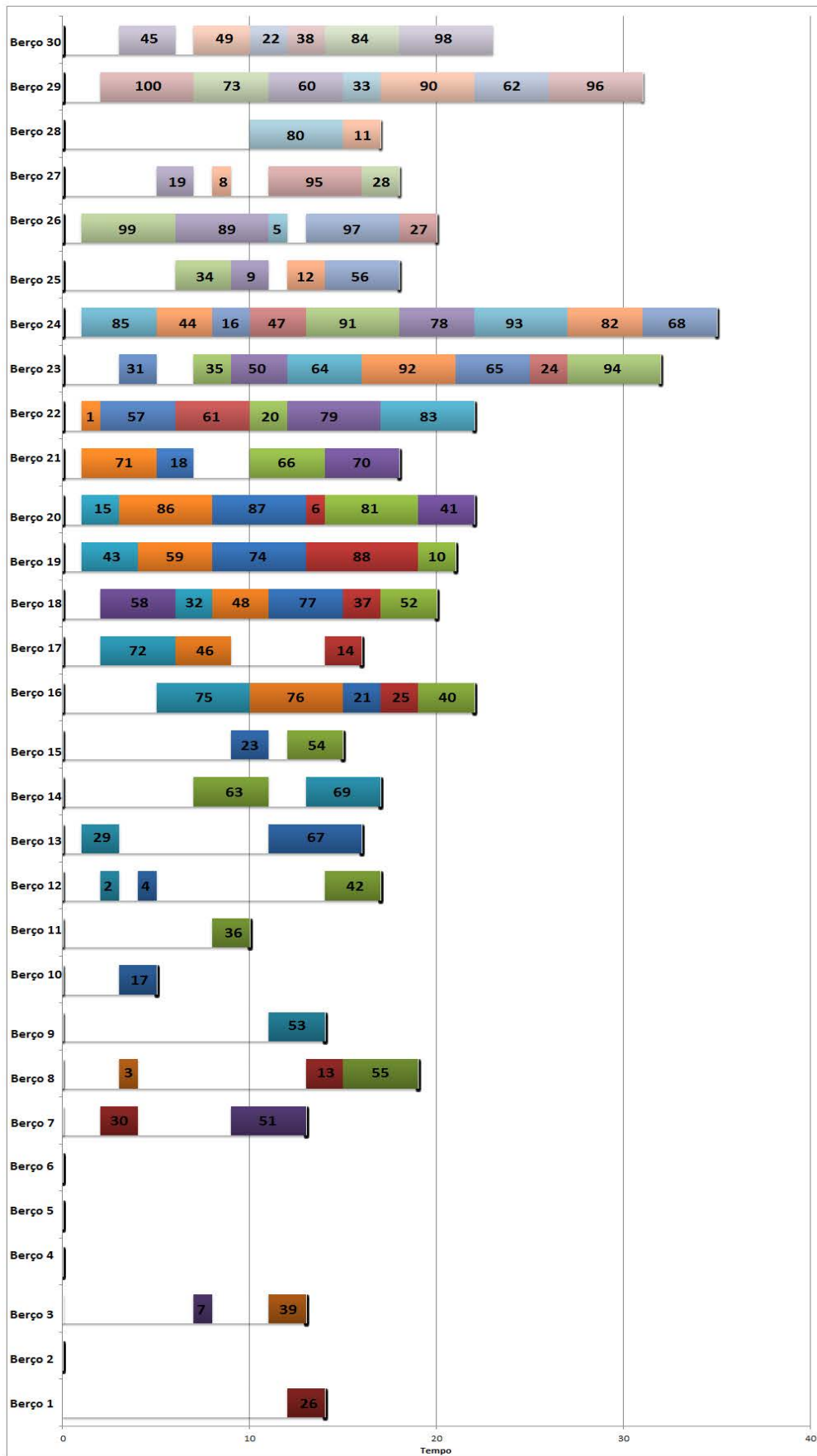


Figura 5.28 – Representação dos atendimentos para 100 navios, 30 berços, 100 partículas e 100 iterações com PSO.

Nos gráficos que simbolizam o atendimento de 20 navios do PSO, observamos que a distribuição é quase uniforme tanto para 5 quanto para 10 berços. Há uma pequena diferença quando existem 10 berços disponíveis, sendo que alguns deles atendem somente um navio, enquanto que outros atendem 5 vezes mais navios.

Nesse caso, com 20 navios, o melhor resultado do PSO acontece quando existem 10 berços, embora o tempo total tenha diminuído somente 20% com 10 berços.

Nos gráficos que simbolizam o atendimento de 30 navios do PSO, observamos que com 5 e 10 berços a distribuição é melhor do que com 20 berços. Com 20 berços, quase 50% deles fica vazio, de forma que o tempo total não seja otimizado. Ainda, nota-se que com 10 berços, a distribuição é melhor do que com 5 berços, ficando cada berço com praticamente a mesma quantidade de navios. No caso de 5 berços, o berço 4 está atendendo 11 navios, enquanto que o berço 3 está atendendo somente 3 navios.

Nesse caso, com 30 navios, o melhor resultado do PSO acontece quando existem 10 berços, sendo que o tempo total é menor tanto em relação ao caso com 5 berços quanto em relação ao caso com 20 berços.

Nos gráficos que simbolizam o atendimento de 50 navios do PSO, observamos que com 5 e 10 berços a distribuição acontece de forma que todos os berços são ocupados, enquanto que com 20 e 30 berços, a distribuição está incoerente. Nos dois últimos casos, muitos berços ficam vazios, de forma que o tempo total não seja otimizado. E ainda, com 30 berços, a ocupação é melhor do que com 20 berços, sendo que a quantidade de berços vazios em cada caso é de 36% e 40%, respectivamente.

Nesse caso, com 50 navios, o melhor resultado do PSO acontece quando existem 10 berços, sendo muito próximo ao caso com 30 berços. Com 10 berços, o tempo total diminui em 50% em relação ao tempo total com 5 berços. E então, o valor aumenta com 20 berços e cai novamente com 30 berços. Portanto, essas duas configurações apresentam bons resultados.

Nos gráficos que simbolizam o atendimento de 100 navios do PSO, observamos que com 5 e 10 berços a distribuição está otimizada, enquanto que com 20 e 30 berços, a distribuição está incoerente. Nos dois últimos casos, muitos berços ficam vazios, de forma que o tempo total não seja otimizado. Ainda, proporcionalmente, mais berços estão vazios no caso de 20 berços do que no caso de 30 berços.

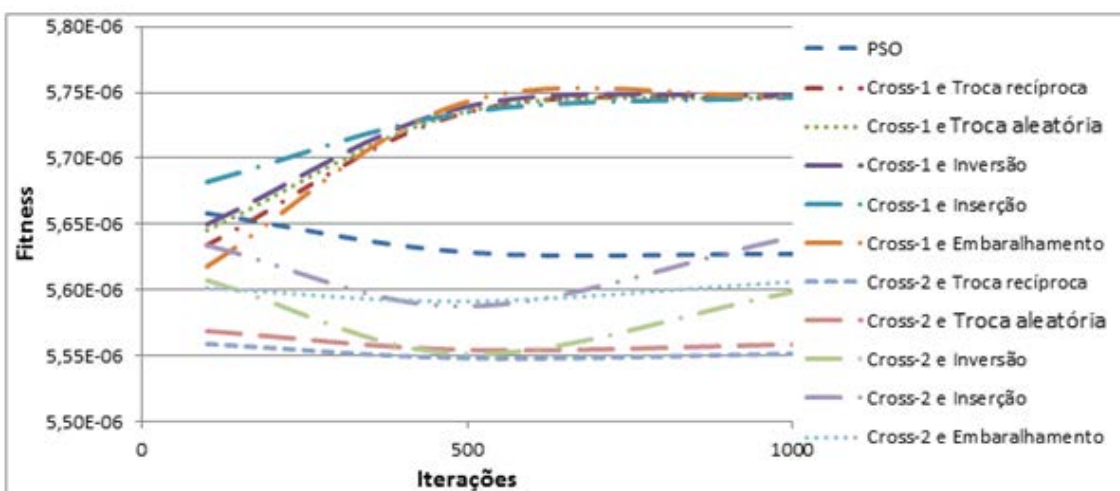
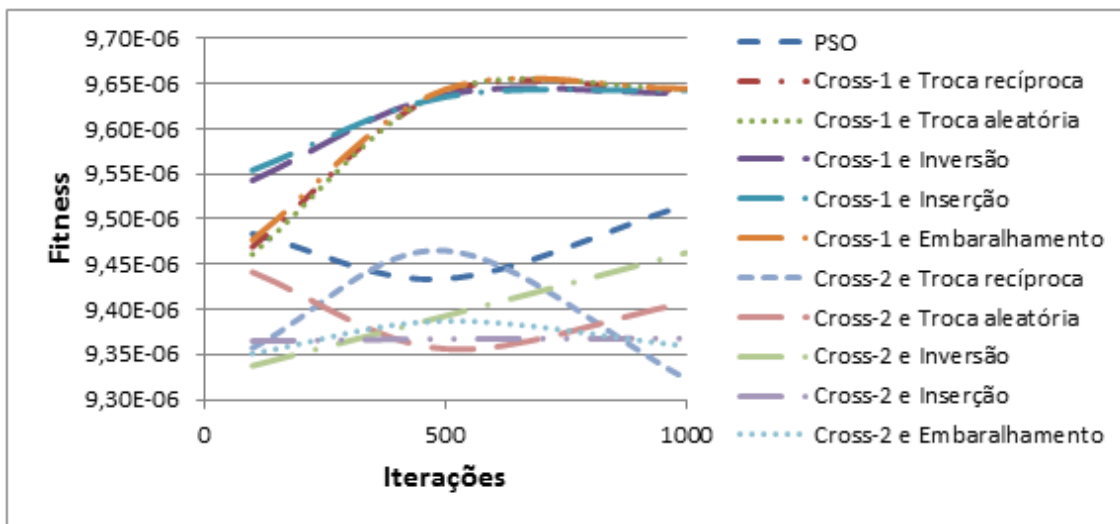
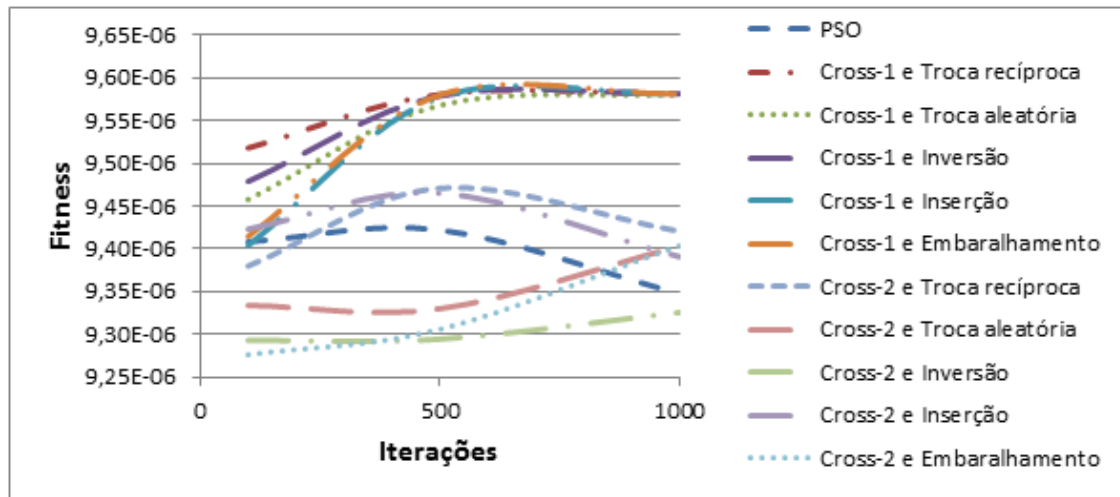
Nesse caso, com 100 navios, o melhor resultado do PSO acontece quando existem 30 berços. O tempo total diminui consideravelmente nessa situação. A

configuração mais competitiva seria a de 20 berços. E ainda, nos casos de 10, 20 e 30 berços, o tempo total diminui em torno de 50% em relação ao tempo total do caso com 5 berços, indicando que a maior quantidade de berços contribui para a distribuição mais uniforme dos navios.

Portanto, para os casos de 20 ou 30 navios, a distribuição dos navios nos berços ocorre de forma mais otimizada com 5 e 10 berços, sendo que o caso com 10 berços é o mais otimizado, apresentando o menor tempo final. No caso de 30 navios e 20 berços, muitos berços permanecem vazios, não havendo otimização do tempo. Para os casos de 50 ou 100 navios, a ocupação ocorre de forma mais uniformizada com 5 e 10 berços, embora o tempo total seja maior considerando-se a quantidade de berços. Enquanto que nos casos com 20 e 30 berços, muitos berços ficam vazios. Ainda, no caso de 50 navios, os resultados são melhores com 10 berços, enquanto que no caso de 100 navios, os resultados são melhores com 30 berços.

5.3 AVALIAÇÃO DO DESEMPENHO COMPUTACIONAL DOS ALGORITMOS

Com os resultados apresentados acima foram avaliados os dois algoritmos a fim de obter qual é o melhor algoritmo para a maioria dos casos. Para isso, além da avaliação dos tempos de alocação e dos custos, foi feita uma avaliação do custo computacional de cada algoritmo, a fim de saber qual algoritmo converge mais rapidamente para o melhor resultado. Ou seja, a avaliação do desempenho computacional consiste em avaliar a convergência do fitness do algoritmo em relação ao número de iterações. Essas avaliações são baseadas nos valores de *fitness* obtidos para cada combinação de operadores de *crossover* e mutação para o AG com os valores de *fitness* obtidos para o PSO para determinado número de iterações. Os gráficos abaixo, representados pelas Figuras de 5.29 à 5.41, apresentam a avaliação do custo computacional.



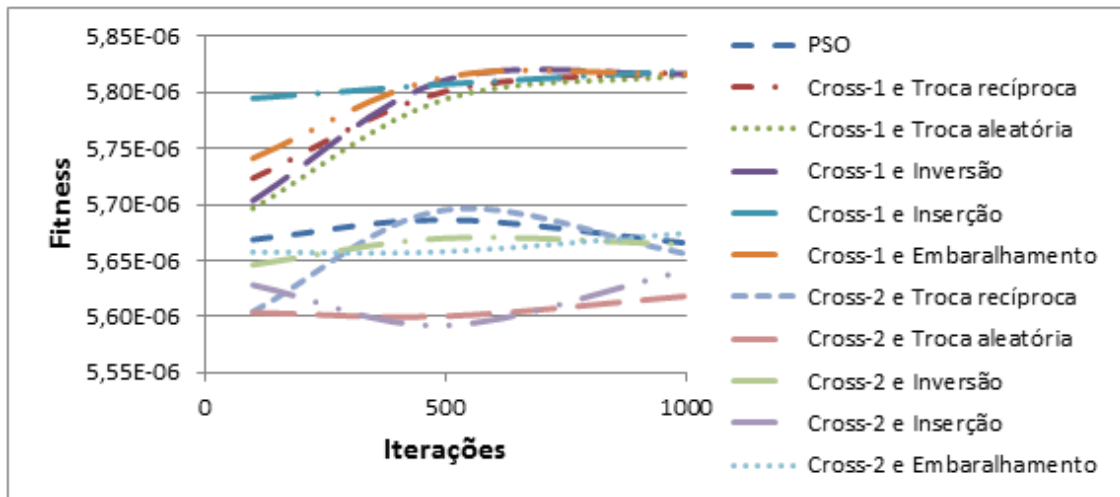


Figura 5.32 – Avaliação do desempenho computacional para 30 navios e 10 berços.

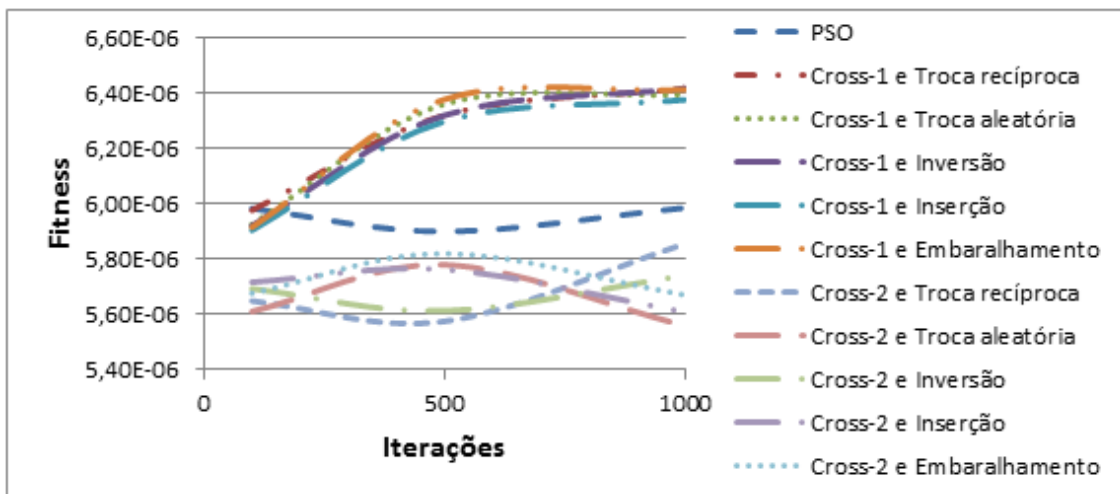


Figura 5.33 – Avaliação do desempenho computacional para 30 navios e 20 berços.

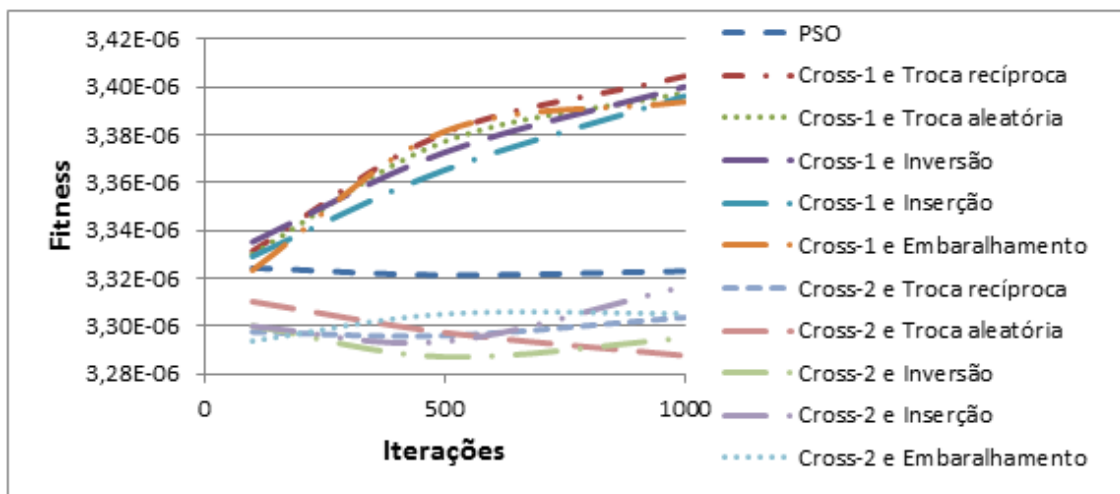


Figura 5.34 – Avaliação do desempenho computacional para 50 navios e 5 berços.

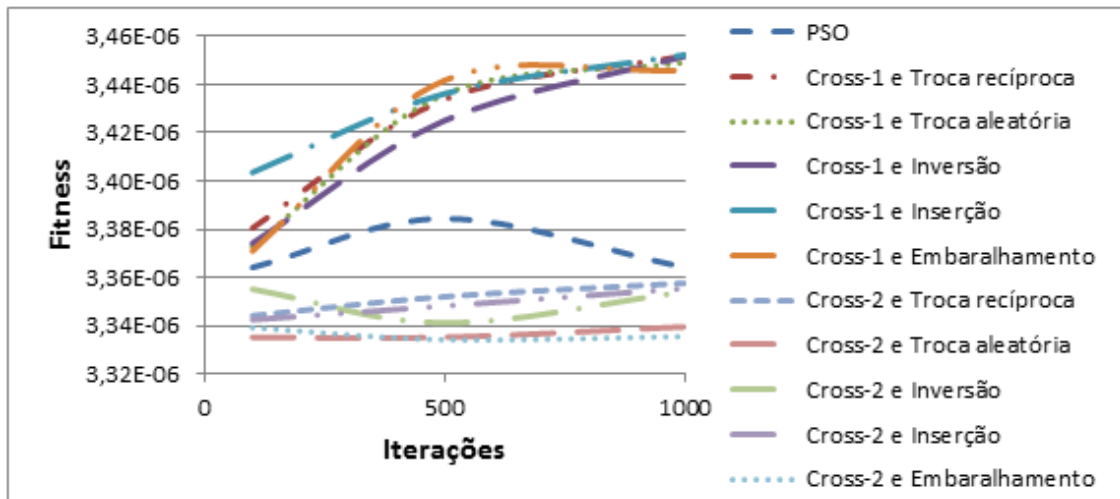


Figura 5.35 – Avaliação do desempenho computacional para 50 navios e 10 berços.

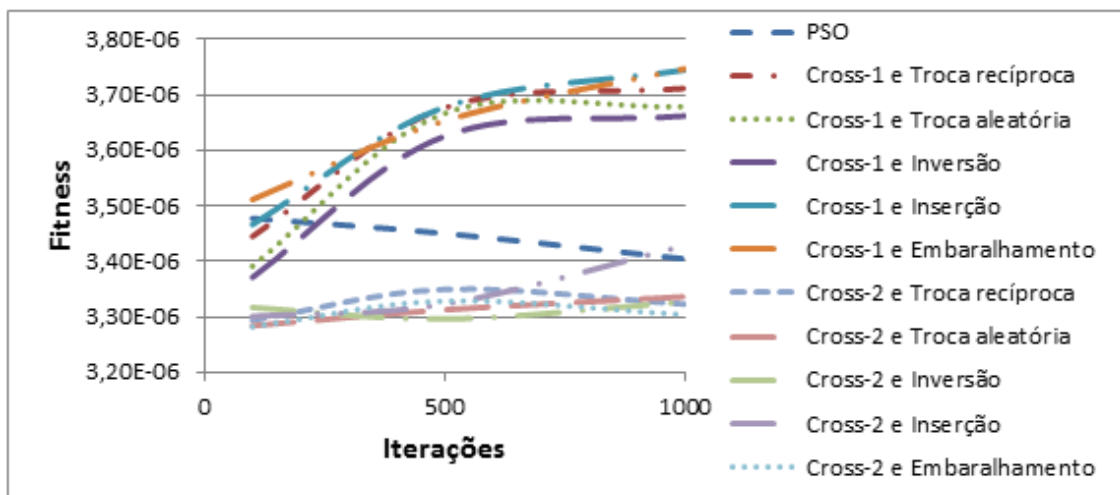


Figura 5.36 – Avaliação do desempenho computacional para 50 navios e 20 berços.

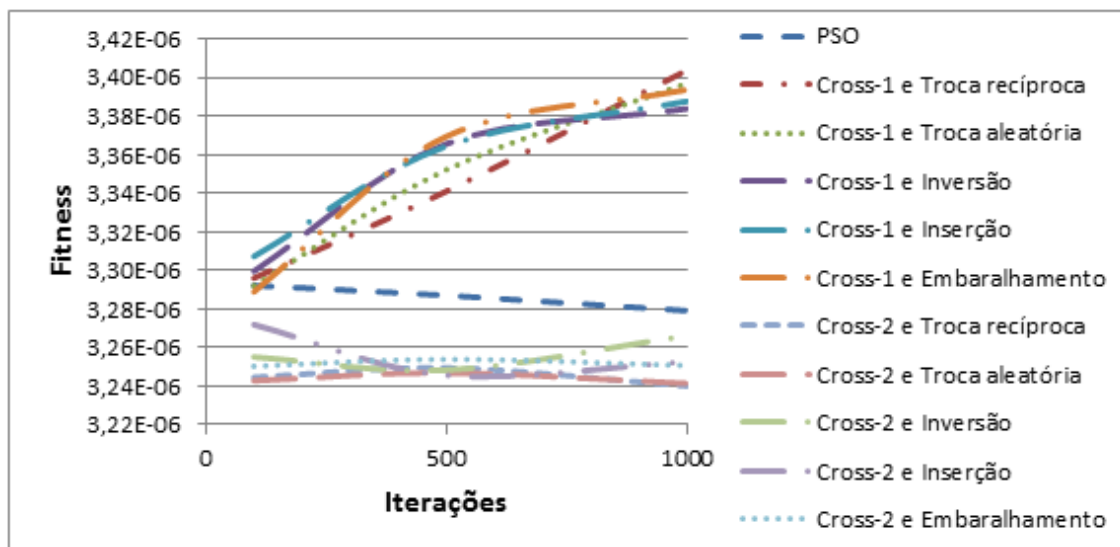


Figura 5.37 – Avaliação do desempenho computacional para 50 navios e 30 berços.

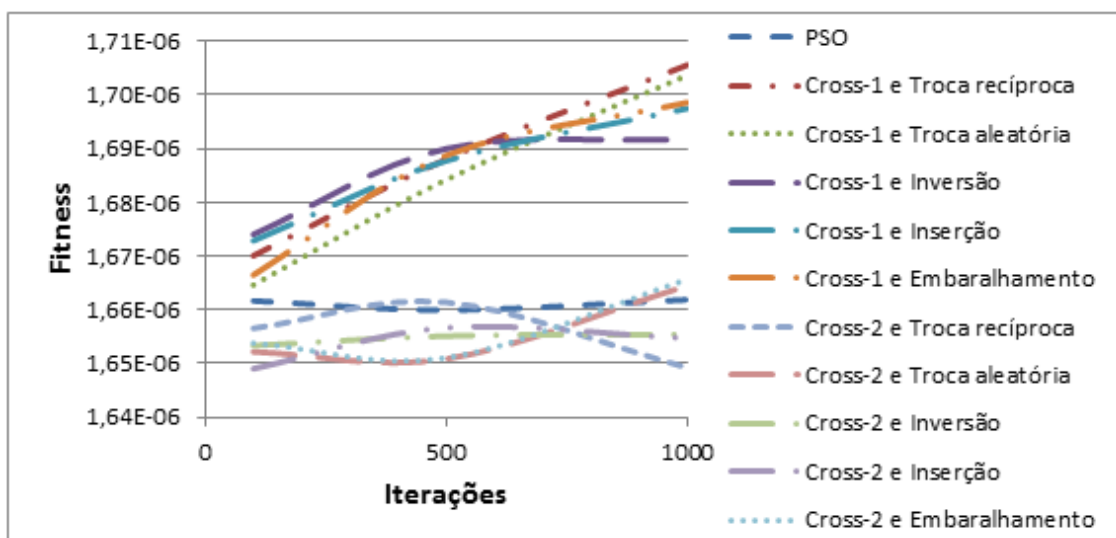


Figura 5.38 – Avaliação do desempenho computacional para 100 navios e 5 berços.

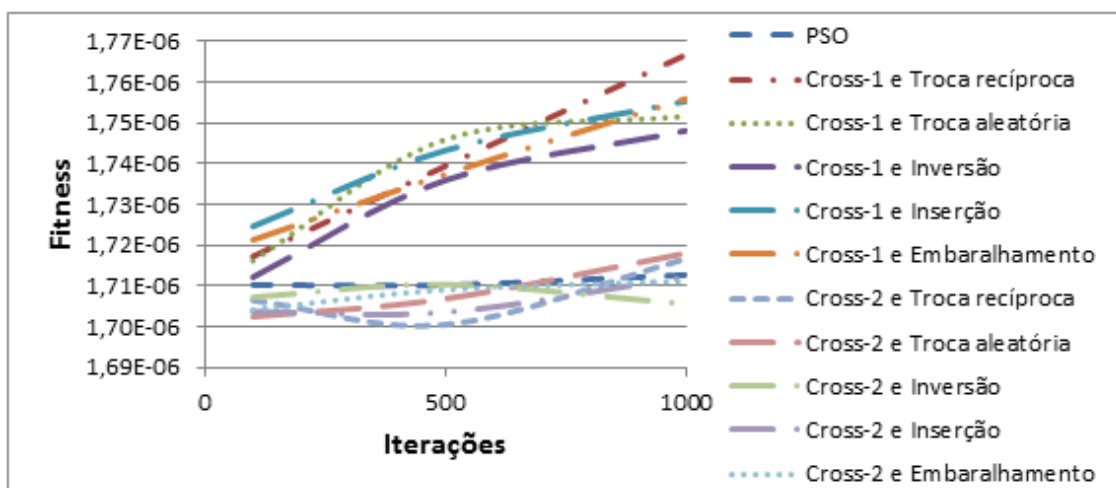


Figura 5.39 – Avaliação do desempenho computacional para 100 navios e 10 berços.

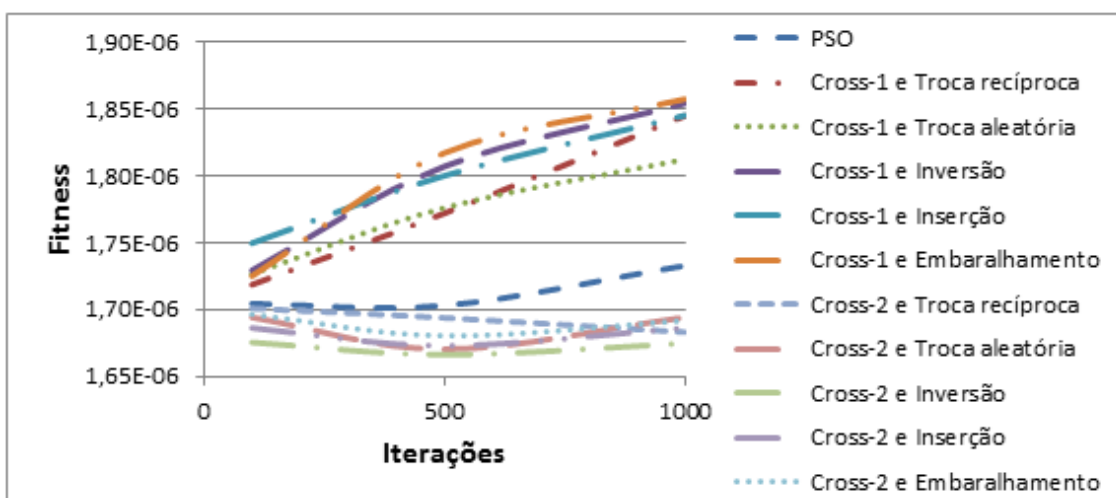


Figura 5.40 – Avaliação do desempenho computacional para 100 navios e 20 berços.

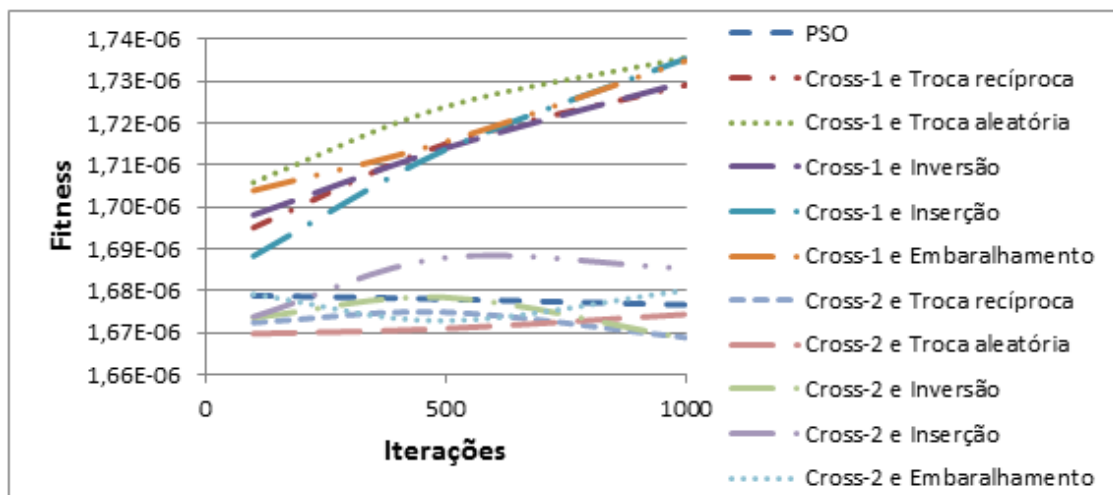


Figura 5.41 – Avaliação do desempenho computacional para 100 navios e 30 berços.

O PSO apresenta diferentes comportamentos, mas na maioria das vezes, o algoritmo converge (estabiliza) mais rapidamente que os AGs para obter os melhores resultados. Com um número maior de iterações, os resultados vão se afastando dos valores ótimos. Assim, dentro dos limites deste trabalho, para obter-se melhores resultados, usando o PSO é necessário utilizar poucos passos de iteração.

No AG com o operador de *crossover* de 2 pontos, na maioria das vezes, os resultados convergem rapidamente para um ótimo local, e depois disso não evoluem mais ou evoluem muito pouco. Entretanto, no AG com o operador de *crossover* de 1 ponto os testes convergem mais devagar para obter os melhores resultados. Ou seja, os resultados evoluem consideravelmente conforme mais iterações acontecem, chegando a valores mais altos e, portanto, parecem mais adequados à tarefa em questão.

Comparando-se o PSO com o AG, podemos observar que o desempenho do PSO está mais próximo ao comportamento apresentado pelo AG com operador de *crossover* de 2 pontos, sobressaindo-se em termos de melhores resultados na maior parte das vezes em comparação com o *crossover* de 2 pontos. Utilizando-se este tipo de operador de cruzamento não foi possível detectar que a sua associação com algum tipo de operador de mutação pudesse ter algum desempenho mais relevante nos cenários avaliados, pois os melhores desempenhos em cada configuração avaliada foram obtidos com diferentes operadores de mutação.

Considerando-se todas as configurações, os testes que convergem mais rápido para os melhores resultados são com *crossover* de 1 ponto e troca recíproca na maioria das avaliações. Pode-se notar, todavia, que para configurações com baixo número de

navios não há diferença significativa no AG com *crossover* de 1 ponto quando se varia o tipo de operador de mutação. Pode-se notar maior variação nos resultados quando se utiliza diferentes operadores de mutação no cenário com 100 navios.

Entretanto, como o foco deste trabalho é a minimização do custo e do tempo, esta análise deve ainda ser explorada em relação à função objetivo, e não apenas com a função de *fitness*, como será discutido a seguir.

5.4 DISCUSSÃO

Inspecionando as Tabelas 5.1 e 5.2, em relação ao custo total, o AG apresenta melhor otimização do custo do que o PSO em todos os testes realizados, destacando-se a configuração de *crossover* de 1 ponto e mutação do tipo troca recíproca para o algoritmo. Porém, comparando-se o PSO com o algoritmo genético configurado com o *crossover* de 2 pontos, então o PSO teria destaque em relação ao AG, devido aos resultados inferiores gerados por essa configuração.

Podemos citar o teste com 100 navios e 20 berços. O melhor resultado com o AG gera um custo total de R\$ 538.357,88, enquanto que o melhor resultado com o PSO gera um custo total de R\$ 577.099,44. Portanto, a diferença do custo total para os dois algoritmos é muito grande, destacando-se o AG para a otimização do custo, configurado com o operador de *crossover* de 1 ponto e o operador de mutação do tipo embaralhamento.

Já em relação ao tempo total, o AG apresenta melhor otimização do tempo do que o PSO na maioria dos casos, porém não em todos os testes. Em alguns deles, os dois algoritmos apresentam resultados similares, enquanto que em outros, o AG apresenta grande vantagem em relação ao PSO. Portanto, generalizando, pode-se considerar o AG melhor do que o PSO para a otimização do tempo também.

Podemos citar o teste com 30 navios e 20 berços. O tempo final para o melhor resultado do AG composto por *crossover* de 1 ponto e mutação do tipo inversão é o mesmo que o tempo final para o melhor resultado do PSO, sendo de 25 horas. Já com o teste com 100 navios e 10 berços, o melhor resultado com o AG apresenta como tempo

final 42 horas usando *crossover* de 1 ponto e mutação do tipo troca recíproca, enquanto que o melhor resultado com o PSO apresenta como tempo final 57 horas. Nesse caso, a diferença de tempo entre dois algoritmos é grande, destacando-se o AG para a otimização do tempo. E ainda, não existe nenhum caso em que o PSO apresenta menor tempo final do que o AG.

Por fim, avaliando-se a otimização das duas variáveis, custo total e tempo final, pode-se concluir que o AG apresenta melhor desempenho do que o PSO, e, particularmente, o AG com *crossover* de 1 ponto e mutação do tipo troca recíproca é o caso onde se obtém os melhores resultados. No AG, além de obtermos menores custos, os tempos finais de atendimento são menores, além de ser possível observar o melhor aproveitamento de dos berços do porto. Essa conclusão é feita em termos de custo total e tempo final, sem levar em consideração o desempenho computacional dos dois algoritmos.

Ainda, podemos citar algumas deficiências encontradas nas soluções dos dois algoritmos. Observando-se os testes com 50 navios e 20 berços, 50 navios e 30 berços, 100 navio e 20 berços e também 100 navios e 30 berços, nota-se que quando se tem um número grande de berços disponíveis, existe a baixa ocupação dos berços. Ou seja, sempre existem espaços de tempo onde grande parte dos berços não está sendo ocupada por nenhum navio.

Por conta dessa deficiência, o custo total de ocupação não diminuiu como era esperado nos testes com 30 berços, embora o tempo total tenha diminuído. Ou seja, nessa situação podemos dizer que somente o tempo total foi otimizado. E o oposto ocorre nos testes com 20 berços, onde o custo total foi minimizado, enquanto que o tempo total de ocupação aumentou. Ou seja, nessa situação podemos dizer que somente o custo total foi otimizado.

Nota-se que com mais berços, como é a situação do teste com 30 berços, os algoritmos se comportam de forma que mais berços são ocupados, diminuindo o tempo total, sem que seja optado por berços mais baratos, não havendo a minimização dos custos. Já na situação onde existem menos berços, como acontece no teste com 20 berços, os algoritmos se comportam de forma que menos berços são ocupados, sendo optada pela maior ocupação dos berços mais baratos, diminuindo o custo total e ao mesmo tempo aumentando o tempo total.

Conforme observado, ambos os algoritmos apresentam resultados incongruentes. Porém, apesar da grande quantidade de testes realizada, não foi possível chegar a uma conclusão de qual o motivo que gera esse efeito nos algoritmos. Portanto, para trabalhos futuros, outros tipos de avaliações poderiam ser feitas na implementação adotada para cada algoritmo para que ambos gerem melhores resultados para todas as configurações existentes do BAP. Dessa forma, os resultados esperados para os dois algoritmos seriam sempre a otimização do tempo e também do custo conforme o aumento da quantidade de berços existentes.

6 CONCLUSÃO

No trabalho desenvolvido estudou-se a aplicação de algoritmos genéticos e também de algoritmos de otimização por nuvem de partículas, duas técnicas conhecidas e utilizadas em problemas de otimização, para solucionar um conhecido problema computacional, o problema de alocação de berços. Considera-se uma solução ótima para o PAB quando a permanência de um navio no porto é mínima e a utilização de um berço é máxima, tornando a alocação mais barata para o navio e lucrativa para o porto.

Considerando a importância do transporte marítimo no cerne econômico mundial, nota-se necessário um planejamento do sistema de alocação de berços portuários, para que os portos atendam à grande quantidade de navios que chegam com mercadorias para serem carregadas ou descarregadas, visando diminuir os custos dos navios e também diminuir o tempo total de atendimento dos navios nos berços.

Esse gerenciamento poderia ser feito manualmente, e seria simples para o caso de 20 navios e 5 berços, por exemplo. Porém, quando se tem uma quantidade grande de navios e também de berços, o planejamento manual se torna impossível. Foram implementados dois tipos de algoritmos de otimização, para que esse software realize esse trabalho de planejamento automaticamente.

Após a implementação foi realizada uma série de testes para concluir qual dos dois algoritmos se comporta melhor para a maioria dos casos. Com esses testes foram feitos vários tipos de avaliação, seja no âmbito de custos e tempos para os navios, como também no âmbito de custo computacional. Os diversos casos avaliados, com diferentes tamanhos de base de dados, permitiram verificar que os algoritmos genéticos conseguiram realizar um bom aproveitamento dos berços fazendo que seus tempos de ocupação fossem o maior possível. Dessa forma, acomoda-se a maior quantidade de navios possível, otimizando o tempo total de atendimento a todos os navios e, conseqüentemente, reduzindo os custos operacionais.

A partir dessa análise pode-se concluir que os algoritmos genéticos apresentam melhores resultados do que os algoritmos de otimização por nuvem de partículas para a maioria dos casos, apresentando os melhores resultados com a configuração para os operadores de *crossover* e mutação como sendo de 1 ponto e troca recíproca, respectivamente.

O trabalho realizado alcançou os objetivos propostos, porém ainda poderão existir trabalhos futuros que complementem o mesmo. Para estes trabalhos poderão ser implementados outros tipos de algoritmos de otimização, para que sejam feitas mais comparações de eficiência dos algoritmos, chegando-se a uma conclusão mais segura. Alguns exemplos de algoritmos de otimização que poderiam ser modelados para a utilização para resolução do PAB são: Algoritmos de Estimação de Distribuição, *Branch and Bound* e Algoritmo Guloso.

Algumas alterações nos dois tipos de algoritmos implementados também poderão ser feitas para que os mesmos gerem melhores resultados.

Em relação aos algoritmos genéticos, uma modificação importante seria a realização de testes após cada iteração, comparando-se sempre os resultados obtidos durante toda a execução, para que os indivíduos caminhassem sempre para a solução ótima, sem que eles se estabilizassem em um máximo local ou seguissem para um mínimo. Ou seja, deveria ser memorizado qual foi o melhor indivíduo, global e local, em cada iteração, para que todos os indivíduos caminhassem em direção a esse. Essa modificação se assemelha ao processo existente nos algoritmos de otimização por nuvem de partículas.

Em relação aos algoritmos de otimização por nuvem de partículas, uma modificação que poderia ser realizada é em relação aos operadores utilizados para que se encontrem novas partículas. Além da operação de *swap*, que ocorre de forma totalmente aleatória, novos operadores poderiam trazer resultados diferentes e melhores para o algoritmo.

E também, devido à grande variedade de abordagens existentes para o problema de alocação de berços, poderiam ser feitas modificações nas equações do problema para atender a outra abordagem. Em relação à função de *fitness*, por exemplo, poderiam entrar outras variáveis que priorizassem e aperfeiçoassem outros fatores existentes nas operações do sistema portuário, além do custo total e do tempo final. Ainda, em relação às restrições, outras variáveis, como máquinas existentes em cada berço e apropriadas para o descarregamento da carga trazida pelos navios, poderiam ser introduzidas para indicar a operação de alocação dos berços correta.

Ainda, poderia ser realizado um estudo com uma melhor avaliação dos resultados encontrados nesse trabalho, de forma que sejam melhoradas algumas deficiências nele encontradas. Esses trabalhos poderiam trazer pequenas modificações nos algoritmos implementados, para que eles funcionassem melhor para o BAP,

gerando bons resultados independente das variações dos parâmetros utilizadas. Ou seja, estudos futuros deveriam trazer um algoritmo onde o tempo e o custo fossem otimizados ao mesmo tempo, independentemente do número de navios ou de berços, dos números de indivíduos ou iterações.

Essas melhorias poderiam ser realizadas com seguidas modificações e testes. Por exemplo, alguns parâmetros (ou constantes) utilizados nos dois algoritmos poderiam ser modificados e testados para se descobrir o valor que gera os melhores resultados. Outra modificação seria a realização de mais testes após cada iteração, sendo avaliados os novos resultados, comparando-os com os resultados antigos e também com os novos resultados dessa iteração.

Por fim, conclui-se que com esse projeto foi possível agregar novos conhecimentos de diferentes áreas da computação, principalmente na área de otimização, podendo ser útil para trabalhos futuros, tais como trabalhos onde se queira otimizar outros tipos de funções relacionadas com o BAP e também trabalhos que estudem outros tipos de problemas computacionais que podem ser resolvidos com algoritmos de otimização.

REFERÊNCIAS BIBLIOGRÁFICAS

ALOISE, D. J.; OLIVEIRA, M. C. S.; SILVA, T. L. **Otimização Discreta por Nuvem de Partículas Aplicada ao Problema do Caixeiro Viajante**, Universidade Federal do Rio Grande do Norte. 2006.

ARRUDA, C.M.; JÚNIOR, E.F.N.; MAGALHÃES, P.S.B. **Método dos indicadores de desempenho proposto pela ANTAQ: Uma aplicação ao terminal portuário do Pecém**. Rio de Janeiro, Brasil. 2008.

BRODERSEN, O.B.; MEIER, L.; SCHUMANN, M. **Optimizing the Berth Allocation Problem using a Genetic Algorithm and Particle Swarm Optimization**. In: MKWI, 2010.

BROWN, G.G.; LAWPHONGPANICH, S.; THURMAN, K.P. **Optimizing ship berthing**. Naval Research Logistics, 41, 1-15. 1994.

BROWN, G.G.; CORMICAN, K.J.; LAWPHONGPANICH, S.; WIDDIS, D.B. **Optimizing submarine berthing with a persistence incentive**. Naval Research Logistics, 44, 301-318. 1997.

BUHRKAL, K.; ZUGLIAN, S.; ROPKE, S.; LARSEN, J.; LUSBY, R. **Models for the discrete berth allocation problem: A computational comparison**. Transportation Research Part E: Logistics and Transportation Review, Volume 47, Issue 4, Pages 461-473, July 2011.

CARVALHO, A.C.L.F. **Algoritmos Genéticos**. São Carlos. Disponível em: < <http://www2.icmc.usp.br/~andre/research/genetic/>>. Acesso em 06/06/2013.

CHEONG, C.Y.; TAN, K.C.; LIU, D.K.; LIN, C.J. **Multi-objective and prioritized berth allocation in container ports**. Annals of Operations Research, 180(1), 63-103. 2008.

COLE, S.; VILLA, A. **Intermodalidade no transporte de carga: Portos e Hinterland, transporte marítimo, incluindo o de curta distância**. Abril, 2006.

CORDEAU, J.F.; LAPORTE, G.; LEGATO, P.; MOCCIA, P. **Models and tabu search heuristics for the berth allocation problem**. Transportation Science, 39, 526 -538. 2005.

DOLCETA. **A importância do transporte marítimo**. Portugal, 2011. Disponível em: < <http://www.dolceta.eu/portugal/Mod6/A-importancia-do-transporte.html>>. Acesso em 06/06/2013.

Felipe e Daniel, **Problema de Sequenciamento de Navios**. In ACMO 2007.

GIALLOMBARDO, G.; MOCCIA, L.; SALANI, M.; VACCA, I. **Modeling and solving the tactical berth allocation problem**. Transportation Research Part B, 44, 232-245. 2010.

GOLDBERG, D.E. **Genetic Algorithms in Search. Optimization and Machine Learning**, Addison-Wesley, 1989.

GOLIAS, M. M.; BOILE, M.; THEOFANIS, S. **Berth scheduling by customer service differentiation: A multi-objective approach**. In: *Transportation Research, Part E-45*, p. 878-892. 2009a.

GOLIAS, M.M.; SAHARIDIS, G.K.; BOILE, M.; THEOFANIS, S.; IERAPETRITOU M.G. **The berth allocation problem: Optimizing vessel arrival time**. USA, 2009b.

ICHIHARA, J.A. **Algoritmos Genéticos**. Santa Catarina, [1998]. Disponível em: <<http://www.eps.ufsc.br/teses98/ichihara/cap3.html>>. Acesso em 06/06/2013.

IMAI, A.; NISHIMURA, E.; PAPADIMITRIOU, S. **The dynamic berth allocation problem for a container port**. *Transportation Research Part B*, 35, 401-417. 2001.

IMAI, A.; NISHIMURA, E.; PAPADIMITRIOU, S. **Berth allocation with service priority**. *Transportation Research Part B*, 37, 437-457. 2003.

IMAI A.; SUN X.; NISHIMURA E.; PAPADIMITRIOU S. **Berth Allocation in a Container Port: Using Continuous Location Space Approach**. *Transportation Research Part B*, Vol. 39, pp. 199–221. 2005.

JERONYMO, D.C.; BORGES, Y.C.; COELHO, L.S. **Algoritmo genético e otimização por enxame de partículas aplicados ao planejamento de trajetória de um manipulador robótico**, Curitiba, Paraná, Brasil. 2009.

JUNIOR, N.F.P. **A importância do transporte marítimo para o desenvolvimento econômico brasileiro**. Pós-Graduação em Geografia da FCT/UNESP, Campus de Presidente Prudente, São Paulo. 2011.

KAPPEL, R.F. **Portos Brasileiros – novo desafio para a sociedade**. Disponível em: <http://www.sbpcnet.org.br/livro/57ra/programas/CONF_SIMP/textos/raimundokappel.htm>. Acesso em 06/06/2013.

KENNEDY, J.; EBERHART, R. C. **Particle Swarm Optimization**, 1995.

KRČUM, M.; GUDELJ, A.; VLAHINIĆ, S. **Genetic Algorithm for Solving Berth and Quay Cranes Assignment Problems**. In: *Second International Conference on Ports and Waterways, POWA*, 2007.

LIANG, C.; HUANG, Y.; YANG, Y. **A Quay Crane Dynamic Scheduling Problem by Hybrid Evolutionary Algorithm for Berth Allocation Planning**. In: *Computers & Industrial Engineering*, vol. 56, p.1021-1028, 2008.

LINDEN, R. **Algoritmos Genéticos, uma importante ferramenta da inteligência computacional**, 428 p., Ed. Brasport, 2008.

MAURI, G.R. **Novas abordagens para representação e obtenção de limitantes e**

soluções para alguns problemas de otimização combinatória. Tese (Doutorado em Computação Aplicada), Instituto Nacional de Pesquisas Espaciais (INPE). 2008.

MAURI, G.R.; OLIVEIRA, A.C.M.; LORENA, L.A.N. **Heurística baseada no simulated annealing aplicada ao problema de alocação de berços.** GEPROS - Gestão da Produção, Operações e Sistemas, 1, 113-127. 2008a.

MAURI, G.R.; OLIVEIRA, A.C.M.; LORENA, L.A.N. **A hybrid column generation approach for the berth allocation problem.** Lecture Notes in Computer Science, 4972, 110-122. 2008b.

NISHIMURA, E.; IMAI, A.; PAPADIMITRIOU, S. **Berth allocation planning in the public berth system by genetic algorithms.** European Journal of Operational Research, v. 131, p. 282-292. 2001.

OBTIKO, M. **Introduction to genetic algorithms.** Dresden, 1998. Disponível em: < <http://www.obitko.com/tutorials/genetic-algorithms/>>. Acesso em 06/06/2013.

OLIVEIRA, R.M.; MAURI, G.R.; LORENA, L.A.N. **Clustering Search aplicado ao problema de alocação de berços,** Bento Golçalves, RS, 2010a.

OLIVEIRA, G.M. **Transporte Marítimo de Contêiner.** Rio de Janeiro, 2010b.

PAQUETTE, R.J.; ASHFORD, N.J.; WRIGHT, P.H. **Transportation engineering: planning and design.** 2. ed. John Wiley & Sons, Inc., 1982.

ROSENDO, M. **Um Algoritmo de Otimização por Nuvem de Partículas para Resolução de Problemas Combinatórios,** Universidade Federal do Paraná, Curitiba. 2010.

ROVINA, E. **Seqüenciamento de Embarque de Granéis Sólidos por Meio de Terminais Portuários utilizando Algoritmos Genéticos,** 112 f. Tese (Pós graduação em Métodos Numéricos de Engenharia) – Universidade Federal do Paraná, Curitiba. 2008.

SERAPIÃO, A.B.S. **Fundamentos de Otimização por Inteligência de Enxames: Uma Visão Geral.** Controle & Automação (Impresso), v. 20(3), p. 271-304, 2009.

SILVA, V.M.D.; COELHO, A.S.; MAYERLE, S.F. **Proposta Heurística de Resolução do Problema de Alocação de Navios em Berços Usando Algoritmos Genéticos,** Rio de Janeiro, 2008a.

SILVA, V.M.D.; COELHO, A.S.; MAYERLE, S.F.; ABREU, L.F. **Heurística para a Resolução do Problema de Alocação de Navios em Berços Usando Algoritmos Genéticos.** In: Revista de Engenharia de Instalações no Mar, n.1 p.1-15, 2008b.

SILVA, V. M. D. **Um modelo heurístico para alocação de navios em berços,** 111 f. Tese (Mestrado em Engenharia) – Centro Tecnológico, Universidade Federal de Santa Catarina, Florianópolis. 2008.

SILVA, V.M.D.; COELHO, A.S. **Revisão sobre o problema de alocação de berços**, 2007, Rio de Janeiro.

SIVANANDAM, S.N.; DEEPA, S.N. **Introduction to Genetic Algorithms**, Springer, p.1-453, 2008.

THURMAN, K.P. **Optimal ship berthing plans**. Dissertation (Masters of Science in Operations Research), Naval Postgraduate School - USA. 1989.

APÊNDICE A

Tabela A.1 – Base de dados de 20 navios.

ID	Comprimento (m)	Quantidade de carga (ton)	Calado (m)	Custo parado (reais/hora)	Hora de chegada
1	150	80.000	6	8.000	1
2	155	85.000	6.5	8.300	2
3	160	90.000	7	8.500	3
4	165	95.000	8	8.700	4
5	170	100.000	8.5	8.900	5
6	180	105.000	9	9.100	1
7	190	110.000	10	9.400	2
8	200	115.000	11	9.600	3
9	210	120.000	12	9.800	4
10	220	125.000	13	10.000	5
11	230	130.000	14	10.200	1
12	240	135.000	14.5	10.300	2
13	245	140.000	15	10.500	3
14	250	145.000	16	10.700	4
15	260	150.000	16	10.900	5
16	275	155.000	16.5	11.100	1
17	270	160.000	17	11.400	2
18	280	165.000	17.5	11.600	3
19	290	170.000	18	11.800	4
20	300	175.000	18	12.000	5

Tabela A.2 – Base de dados de 30 navios.

ID	Comprimento (m)	Quantidade de carga (ton)	Calado (m)	Custo parado (reais/hora)	Hora de chegada
1	150	80.000	6	7.200	1
2	160	80.000	6	7.400	2
3	165	85.000	6.5	7.600	3
4	170	90.000	7	7.800	4
5	175	95.000	7.5	8.000	5
6	180	100.000	8	8.100	6
7	185	105.000	8	8.300	7
8	190	110.000	8.5	8.500	8
9	195	115.000	9	8.700	1
10	200	120.000	9.5	8.900	2
11	205	120.000	10	9.200	3
12	210	125.000	10	9.400	4
13	215	130.000	10.5	9.600	5
14	220	135.000	11	9.800	6
15	225	140.000	11.5	10.000	7
16	230	140.000	12	10.100	8
17	235	145.000	12	10.300	1
18	240	150.000	12.5	10.500	2
19	245	155.000	13	10.700	3
20	250	160.000	13.5	10.900	4
21	255	160.000	14	11.200	5
22	260	165.000	1.5	11.400	6
23	265	170.000	15	11.600	7
24	270	175.000	15.5	11.800	8
25	275	180.000	16	12.000	1
26	280	180.000	16	12.100	2
27	285	185.000	16.5	12.300	3
28	290	190.000	17	12.700	4
29	295	195.000	17.5	12.900	5
30	300	200.000	18	13.000	6

Tabela A.3 – Base de dados de 50 navios.

ID	Comprimento (m)	Quantidade de carga (ton)	Calado (m)	Custo parado (reais/hora)	Hora de chegada
1	150	40.000	5	6.800	1
2	152	45.000	5	7.000	2
3	155	50.000	5.5	7.100	3
4	157	55.000	6	7.300	4
5	160	60.000	6	7.500	5
6	163	60.000	6.5	7.700	6
7	165	65.000	6.5	7.900	7
8	170	70.000	7	8.200	8
9	175	75.000	7.5	8.400	9
10	177	75.000	7.5	8.600	10
11	180	80.000	8	8.800	1
12	182	80.000	8	9.000	2
13	185	85.000	8.5	9.100	3
14	190	90.000	8.5	9.300	4
15	194	95.000	9	9.500	5
16	198	100.000	9.5	9.700	6
17	200	105.000	10	9.900	7
18	202	110.000	10	8.000	8
19	205	115.000	10.5	8.200	9
20	210	120.000	10.5	8.400	10
21	215	120.000	11	8.600	1
22	217	125.000	11.5	8.800	2
23	220	125.000	12	9.100	3
24	223	130.000	12	9.300	4
25	225	135.000	12.5	9.500	5
26	230	140.000	12.5	9.700	6
27	232	140.000	13	9.900	7
28	236	145.000	13.5	10.000	8
29	240	150.000	14	10.200	9
30	242	155.000	14	10.400	10
31	245	160.000	14.5	10.600	1
32	249	160.000	14.5	10.800	2
33	251	165.000	15	11.100	3
34	255	170.000	15.5	11.300	4
35	257	175.000	16	11.500	5
36	260	175.000	16	11.700	6
37	263	180.000	16.5	11.900	7
38	268	180.000	16.5	12.000	8
39	270	185.000	17	12.200	9
40	272	190.000	17.5	12.400	10
41	275	195.000	17.5	12.600	1
42	279	195.000	18	12.800	2

43	280	200.000	18	13.000	3
44	285	205.000	18.5	13.100	4
45	288	205.000	18.5	13.300	5
46	290	210.000	19	13.500	6
47	292	215.000	19.5	13.700	7
48	295	215.000	19.5	13.900	8
49	297	220.000	20	14.000	9
50	300	220.000	20	14.200	10

Tabela A.4 – Base de dados de 100 navios.

ID	Comprimento (m)	Quantidade de carga (ton)	Calado (m)	Custo parado (reais/hora)	Hora de chegada
1	150	35.000	5	6.100	1
2	151	36.000	5.1	6.150	2
3	153	38.000	5.2	6.300	3
4	154	39.000	5.4	6.420	4
5	156	41.000	5.5	6.540	5
6	157	42.000	5.6	6.670	6
7	158	44.000	5.8	6.700	7
8	160	46.000	5.9	6.820	8
9	162	47.000	6	6.910	9
10	163	49.000	6.2	7.000	10
11	165	50.000	6.3	7.150	11
12	166	52.000	6.4	7.200	12
13	168	53.000	6.6	7.320	13
14	169	55.000	6.7	7.450	14
15	171	56.000	6.8	7.560	1
16	172	58.000	6.9	7.620	2
17	174	59.000	7	7.800	3
18	175	61.000	7.1	7.910	4
19	176	63.000	7.2	8.000	5
20	177	64.000	7.5	8.140	6
21	179	66.000	7.6	8.250	7
22	180	67.000	7.8	8.330	8
23	182	69.000	7.9	8.410	9
24	183	70.000	8	8.490	10
25	184	72.000	8.2	8.540	11
26	186	73.000	8.3	8.670	12
27	187	75.000	8.4	8.790	13
28	189	76.000	8.6	8.820	14
29	192	78.000	8.7	8.910	1
30	193	79.000	8.9	8.990	2
31	195	81.000	9	9.130	3
32	196	83.000	9.1	9.250	4
33	197	85.000	9.2	9.310	5
34	198	87.000	9.4	9.460	6
35	200	88.000	9.5	9.520	7
36	201	90.000	9.7	9.670	8
37	202	93.000	9.9	9.780	9
38	204	95.000	10	9.810	10
39	205	99.000	10.1	9.930	11
40	207	101.000	10.2	10.000	12
41	209	104.000	10.4	10.250	13
42	210	106.000	10.5	10.370	14

43	212	109.000	10.6	10.420	1
44	213	112.000	10.7	10.630	2
45	215	115.000	10.9	10.710	3
46	216	118.000	11	10.860	4
47	217	120.000	11.2	10.970	5
48	219	123.000	11.3	11.010	6
49	221	126.000	11.4	11.090	7
50	222	130.000	11.6	11.120	8
51	223	133.000	11.8	11.250	9
52	225	135.000	11.9	11.300	10
53	226	138.000	12	11.370	11
54	228	141.000	12.1	11.490	12
55	230	144.000	12.2	11.530	13
56	231	147.000	12.4	11.640	14
57	233	149.000	12.5	11.780	1
58	234	152.000	12.6	11.850	2
59	236	155.000	12.8	11.910	3
60	237	157.000	13	12.000	4
61	239	158.000	13.2	12.070	5
62	241	160.000	13.4	12.120	6
63	242	162.000	13.6	12.190	7
64	244	164.000	13.8	12.220	8
65	245	167.000	13.9	12.300	9
66	246	169.000	14	12.380	10
67	248	170.000	14.1	12.440	11
68	249	171.000	14.3	12.500	12
69	250	174.000	14.5	12.580	13
70	252	175.000	14.7	12.650	14
71	254	177.000	14.9	12.700	1
72	255	178.000	15	12.850	2
73	256	179.000	15.2	12.880	3
74	257	182.000	15.4	12.900	4
75	258	184.000	15.6	12.960	5
76	260	186.000	15.8	13.000	6
77	261	187.000	15.9	13.050	7
78	263	188.000	16	13.100	8
79	264	189.000	16.1	13.150	9
80	265	191.000	16.2	13.220	10
81	267	192.000	16.3	13.400	11
82	268	194.000	16.5	13.450	12
83	270	195.000	16.7	13.500	13
84	272	197.000	16.9	13.640	14
85	273	200.000	17	13.750	1
86	274	201.000	17.1	13.870	2
87	276	203.000	17.3	13.980	3

88	279	204.000	17.6	14.000	4
89	280	206.000	17.9	14.050	5
90	281	207.000	18	14.150	6
91	284	209.000	18.1	14.200	7
92	286	211.000	18.3	14.300	8
93	288	213.000	18.5	14.350	9
94	289	215.000	18.6	14.400	10
95	291	216.000	18.8	14.500	11
96	293	218.000	19	14.550	12
97	294	220.000	19.2	14.600	13
98	296	222.000	19.4	14.750	14
99	298	224.000	19.8	14.880	1
100	300	225.000	20	15.000	2

Tabela A.5 – Base de dados de 5 berços.

ID	Comprimento (m)	Profundidade (m)	Tarifa de atracação (reais/m)	Tarifa de movimentação (reais/contêiner)	Produtividade
1	280	20	7	2	45
2	290	21	6	3	40
3	300	22	7	2	45
4	310	23	6	3	40
5	320	24	7	2	45

Tabela A.6 – Base de dados de 10 berços.

ID	Comprimento (m)	Profundidade (m)	Tarifa de atracação (reais/m)	Tarifa de movimentação (reais/contêiner)	Produtividade
1	275	18	8	2	45
2	280	18	7	3	40
3	285	19	8	2	45
4	290	20	7	3	40
5	295	20	8	2	45
6	300	21	7	3	40
7	305	22	8	2	45
8	310	22	7	3	40
9	315	23	8	2	45
10	320	24	7	3	40

Tabela A.7 – Base de dados de 20 berços.

ID	Comprimento (m)	Profundidade (m)	Tarifa de atracação (reais/m)	Tarifa de movimentação (reais/contêiner)	Produtividade
1	200	12	8	2	40
2	210	13	7	3	45
3	220	14	6	4	50
4	230	14	8	2	40
5	240	15	7	3	45
6	245	16	6	4	50
7	250	16	8	2	40
8	255	17	7	3	45
9	260	18	6	4	50
10	265	18	8	2	40
11	270	19	7	3	45
12	275	20	6	4	50
13	280	20	8	2	40
14	280	21	7	3	45
15	285	22	6	4	50
16	290	23	8	2	40
17	295	24	7	3	45
18	300	24	6	4	50
19	310	25	8	2	40
20	320	26	7	3	45

Tabela A.8 – Base de dados de 30 berços.

ID	Comprimento (m)	Profundidade (m)	Tarifa de atracação (reais/m)	Tarifa de movimentação (reais/contêiner)	Produtividade
1	200	10	6	4	40
2	205	10	7	3	45
3	210	11	8	2	50
4	215	12	6	4	40
5	220	12	7	3	45
6	220	13	8	2	50
7	225	14	6	4	40
8	230	14	7	3	45
9	235	15	8	2	50
10	240	16	6	4	40
11	245	16	7	3	45
12	250	17	8	2	50
13	255	18	6	4	40
14	260	18	7	3	45
15	260	19	8	2	50
16	265	20	6	4	40
17	270	20	7	3	45
18	275	21	8	2	50
19	280	22	6	4	40
20	285	22	7	3	45
21	290	23	8	2	50
22	295	24	6	4	40
23	300	24	7	3	45
24	300	25	8	2	50
25	305	25	6	4	40
26	310	26	7	3	45
27	315	26	8	2	50
28	320	27	6	4	40
29	320	28	7	3	45
30	325	28	8	2	50

APÊNDICE B

Tabela B – Melhores resultados variando-se o número de gerações e de indivíduos para o algoritmo genético.

Teste	Navios	Berços	Indivíduos	Gerações	Crossover	Mutação	Custo Total	Tempo
1	20	5	20	100	1 ponto	Troca recíproca	R\$ 105.065,50	23
2	20	5	20	500	1 ponto	Troca recíproca	R\$ 104.377,30	23
3	20	5	20	1000	1 ponto	Inversão	R\$ 104.369,00	23
4	20	5	50	100	1 ponto	Inversão	R\$ 105.499,30	30
5	20	5	50	500	1 ponto	Embaralhamento	R\$ 104.385,80	21
6	20	5	50	1000	1 ponto	Troca recíproca	R\$ 104.367,91	21
7	20	5	100	100	1 ponto	Troca recíproca	R\$ 105.949,00	33
8	20	5	100	500	1 ponto	Inserção	R\$ 104.415,10	25
9	20	5	100	1000	1 ponto	Embaralhamento	R\$ 104.374,60	22
10	20	10	20	100	1 ponto	Inserção	R\$ 104.666,80	16
11	20	10	20	500	1 ponto	Troca recíproca	R\$ 103.754,59	12
12	20	10	20	1000	1 ponto	Inserção	R\$ 103.731,09	12
13	20	10	50	100	1 ponto	Inserção	R\$ 105.348,51	19
14	20	10	50	500	1 ponto	Inversão	R\$ 103.793,01	13
15	20	10	50	1000	1 ponto	Embaralhamento	R\$ 103.686,21	10
16	20	10	100	100	1 ponto	Embaralhamento	R\$ 105.811,70	20
17	20	10	100	500	1 ponto	Embaralhamento	R\$ 103.696,91	20
18	20	10	100	1000	1 ponto	Troca aleatória	R\$ 103.690,60	13
19	30	5	20	100	1 ponto	Inserção	R\$ 175.989,80	32
20	30	5	20	500	1 ponto	Inserção	R\$ 174.964,50	35
21	30	5	20	1000	1 ponto	Troca recíproca	R\$ 173.991,81	39
22	30	5	50	100	1 ponto	Inversão	R\$ 176.992,11	41
23	30	5	50	500	1 ponto	Embaralhamento	R\$ 174.121,69	41
24	30	5	50	1000	1 ponto	Troca recíproca	R\$ 173.985,02	39
25	30	5	100	100	2 pontos	Inserção	R\$ 177.492,89	43
26	30	5	100	500	1 ponto	Troca recíproca	R\$ 174.365,52	41
27	30	5	100	1000	1 ponto	Inversão	R\$ 173.964,58	39
28	30	10	20	100	1 ponto	Inserção	R\$ 172.571,00	25
29	30	10	20	500	1 ponto	Troca recíproca	R\$ 172.389,50	21
30	30	10	20	1000	1 ponto	Troca recíproca	R\$ 171.870,81	17
31	30	10	50	100	1 ponto	Embaralhamento	R\$ 174.178,59	26
32	30	10	50	500	1 ponto	Inversão	R\$ 172.088,81	27
33	30	10	50	1000	1 ponto	Inserção	R\$ 171.837,13	18
34	30	10	100	100	1 ponto	Troca recíproca	R\$ 175.941,69	33
35	30	10	100	500	1 ponto	Embaralhamento	R\$ 172.005,83	27
36	30	10	100	1000	1 ponto	Inserção	R\$ 171.969,70	20
37	30	20	20	100	1 ponto	Troca recíproca	R\$ 167.292,11	26
38	30	20	20	500	1 ponto	Embaralhamento	R\$ 158.545,61	26
39	30	20	20	1000	1 ponto	Inversão	R\$ 155.775,72	25
40	30	20	50	100	1 ponto	Inversão	R\$ 168.877,31	26
41	30	20	50	500	1 ponto	Embaralhamento	R\$ 156.773,41	28
42	30	20	50	1000	1 ponto	Troca recíproca	R\$ 155.784,69	25

43	30	20	100	100	1 ponto	Troca aleatória	R\$ 168.765,19	28
44	30	20	100	500	1 ponto	Troca recíproca	R\$ 158.189,20	44
45	30	20	100	1000	1 ponto	Inversão	R\$ 156.248,22	35
46	50	5	20	100	1 ponto	Inversão	R\$ 299.820,97	53
47	50	5	20	500	1 ponto	Troca recíproca	R\$ 296.081,63	51
48	50	5	20	1000	1 ponto	Troca recíproca	R\$ 295.209,25	46
49	50	5	50	100	1 ponto	Inserção	R\$ 301.042,38	56
50	50	5	50	500	1 ponto	Troca recíproca	R\$ 295.793,41	45
51	50	5	50	1000	1 ponto	Troca recíproca	R\$ 293.729,13	50
52	50	5	100	100	1 ponto	Inversão	R\$ 302.068,00	58
53	50	5	100	500	1 ponto	Inversão	R\$ 296.495,78	52
54	50	5	100	1000	1 ponto	Troca aleatória	R\$ 294.326,00	48
55	50	10	20	100	1 ponto	Inserção	R\$ 293.809,81	45
56	50	10	20	500	1 ponto	Troca recíproca	R\$ 291.591,09	27
57	50	10	20	1000	1 ponto	Troca recíproca	R\$ 290.727,47	26
58	50	10	50	100	2 pontos	Inversão	R\$ 298.051,59	37
59	50	10	50	500	1 ponto	Embaralhamento	R\$ 290.556,78	31
60	50	10	50	1000	1 ponto	Inserção	R\$ 289.661,88	26
61	50	10	100	100	1 ponto	Inversão	R\$ 296.374,97	46
62	50	10	100	500	1 ponto	Inserção	R\$ 291.021,19	38
63	50	10	100	1000	1 ponto	Inversão	R\$ 289.716,31	34
64	50	20	20	100	1 ponto	Embaralhamento	R\$ 284.814,38	34
65	50	20	20	500	1 ponto	Troca recíproca	R\$ 272.001,00	41
66	50	20	20	1000	1 ponto	Inserção	R\$ 267.055,69	30
67	50	20	50	100	1 ponto	Troca recíproca	R\$ 290.300,19	50
68	50	20	50	500	1 ponto	Troca aleatória	R\$ 272.745,41	39
69	50	20	50	1000	1 ponto	Embaralhamento	R\$ 266.901,63	50
70	50	20	100	100	1 ponto	Embaralhamento	R\$ 291.289,97	40
71	50	20	100	500	1 ponto	Embaralhamento	R\$ 277.839,31	48
72	50	20	100	1000	1 ponto	Troca recíproca	R\$ 271.873,41	48
73	50	30	20	100	1 ponto	Inversão	R\$ 303.054,84	23
74	50	30	20	500	1 ponto	Troca recíproca	R\$ 299.334,91	20
75	50	30	20	1000	1 ponto	Troca recíproca	R\$ 296.133,59	25
76	50	30	50	100	1 ponto	Troca recíproca	R\$ 303.631,63	38
77	50	30	50	500	1 ponto	Embaralhamento	R\$ 296.766,56	26
78	50	30	50	1000	1 ponto	Troca aleatória	R\$ 294.363,50	25
79	50	30	100	100	1 ponto	Troca aleatória	R\$ 304.102,88	33
80	50	30	100	500	1 ponto	Embaralhamento	R\$ 297.649,66	29
81	50	30	100	1000	1 ponto	Troca recíproca	R\$ 293.779,47	28
82	100	5	20	100	1 ponto	Inversão	R\$ 597.357,56	89
83	100	5	20	500	1 ponto	Embaralhamento	R\$ 592.154,44	75
84	100	5	20	1000	1 ponto	Troca recíproca	R\$ 586.993,81	90
85	100	5	50	100	1 ponto	Inserção	R\$ 597.783,31	97
86	100	5	50	500	1 ponto	Inversão	R\$ 591.717,25	86
87	100	5	50	1000	1 ponto	Troca recíproca	R\$ 586.318,19	95
88	100	5	100	100	1 ponto	Inversão	R\$ 602.759,88	106
89	100	5	100	500	1 ponto	Troca recíproca	R\$ 592.707,38	94
90	100	5	100	1000	1 ponto	Troca aleatória	R\$ 586.974,19	92
100	100	10	20	100	1 ponto	Inserção	R\$ 579.831,06	57
101	100	10	20	500	1 ponto	Inserção	R\$ 575.188,81	53

102	100	10	20	1000	1 ponto	Troca recíproca	R\$ 569.550,56	49
103	100	10	50	100	1 ponto	Embaralhamento	R\$ 580.946,56	72
104	100	10	50	500	1 ponto	Troca aleatória	R\$ 572.764,81	52
105	100	10	50	1000	1 ponto	Troca recíproca	R\$ 566.035,38	42
106	100	10	100	100	1 ponto	Inversão	R\$ 584.044,31	74
107	100	10	100	500	1 ponto	Troca aleatória	R\$ 574.165,63	55
108	100	10	100	1000	1 ponto	Embaralhamento	R\$ 569.502,13	56
109	100	20	20	100	1 ponto	Inserção	R\$ 572.519,75	56
110	100	20	20	500	1 ponto	Inserção	R\$ 566.779,19	48
111	100	20	20	1000	1 ponto	Embaralhamento	R\$ 538.357,88	62
112	100	20	50	100	1 ponto	Inserção	R\$ 571.547,94	59
113	100	20	50	500	1 ponto	Inserção	R\$ 555.525,06	64
114	100	20	50	1000	1 ponto	Inversão	R\$ 539.296,63	64
115	100	20	100	100	1 ponto	Inversão	R\$ 578.874,56	63
116	100	20	100	500	1 ponto	Inversão	R\$ 559.467,81	66
117	100	20	100	1000	1 ponto	Troca recíproca	R\$ 543.722,00	67
118	100	30	20	100	1 ponto	Embaralhamento	R\$ 586.884,13	45
119	100	30	20	500	1 ponto	Troca recíproca	R\$ 583.086,75	33
120	100	30	20	1000	1 ponto	Troca recíproca	R\$ 578.354,38	45
121	100	30	50	100	1 ponto	Inversão	R\$ 592.616,50	39
122	100	30	50	500	1 ponto	Troca aleatória	R\$ 580.068,81	38
123	100	30	50	1000	1 ponto	Embaralhamento	R\$ 576.426,19	29
124	100	30	100	100	2 pontos	Troca aleatória	R\$ 599.526,38	41
125	100	30	100	500	1 ponto	Embaralhamento	R\$ 586.860,38	33
126	100	30	100	1000	1 ponto	Troca aleatória	R\$ 576.179,00	42