

Evolutionary Algorithm for Optimization Regarding the Planning of Topological Facilities in Layout of a Shipyard

W. Azzolini, and F. Azzolini

Abstract—The purpose of this study is to contribute the approach to the problem of optimization regarding the planning of topological facilities in layout of a shipyard, with the objective of finding a robust solution to the problem by improving the solution space search through refining the genetic operators. For this, the computational results of the evolutionary algorithm proposed by Choi with changes made by the authors, being: 1) the use of the Partially-Matched Crossover (PMX) genetic operator; 2) the use of a recursive expression in the topological optimization step in addition to implementing the Biased Random-Key Genetic Algorithm (BRKGA) for the purpose of comparing the results. As a plan of the computational experiments two groups of experiments were performed: 1) with the parameters and variables of the work of Choi, in order to validate the efficiency and effectiveness of the AE proposed in this work and; 2) with the parameters and variables of the work of Choi with Department 03 fixed in the position of the best solution found in the 1st group of experiments (position 11 of the topological Grid). Each group contains 50 experiments with 100 iterations and variation of the number of individuals from 100 to 80,000 individuals. As a result, a better solution characterized by the reduction of material handling costs, of 11,816 presented by Choi, for 11,489 monetary units of cost, found from the changes made by the authors of the original proposal of the evolutionary algorithm and the use of BRKGA.

Index Terms—Evolutionary Algorithm, Facilities Planning, Minimization of Materials Movement Cost, Shipyard, Topological Optimization, Biased Random-Key Genetic Algorithm.

I. INTRODUÇÃO

NA literatura o problema da otimização do custo de movimentação de materiais dentro do escopo de projetos de planejamento das facilidades de *layout* é abordado desde a década de 1.950, sendo que a maioria dos *layouts* de estaleiros foram projetados com base nas experiências adquiridas pelos profissionais do chão de fábrica [1] [2].

Isto implica em uma abordagem muito pequena desse tema nas últimas sete décadas e, dentro das características deste tipo de indústria, há um *gap* com grande potencial para o desenvolvimento de soluções computacionais [2] [3] [4] [5].

Neste contexto, o problema de planejamento das facilidades de *layout* (*Facility Layout Planning* (FLP)) vem sendo estudado e é conhecido por ter um impacto significativo nos custos

O trabalho foi apoiado em parte pela Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq).

W. Azzolini, Junior, é docente da Universidade de São Paulo, São Carlos, SP, BR, wazzolini@sc.usp.br.

F. G. P. Azzolini, é estudante de graduação do curso de Ciências da Computação (UNESP), Bauru, SP, BR, fred_gpa@hotmail.com.

e na produtividade de uma planta de construção naval [1] [2] [3] [4] [5].

Dentre os custos presentes na produção destaca-se o resultante da movimentação de materiais (*Material Handling Cost* (MHC)), o qual pode chegar a até 70% do total na fabricação de navios, dependendo do fluxo de matéria prima e de materiais transportados nos processos intermediários, principalmente devido a grande dimensão e peso dos subconjuntos e conjuntos de um navio [1] [4] [5]. Este tipo de problema é classificado na literatura como NP-difícil [1] [2] [3] [4] [5].

Métodos analíticos são capazes de garantir a otimalidade das soluções. A maior restrição quanto ao uso destes métodos é o tempo computacional que cresce exponencialmente com o tamanho do problema em uma proporção superior aos métodos heurísticos, o que conduz à aplicação de métodos heurísticos como os algoritmos evolutivos (*Evolutionary Algorithms* (EA)) em contrapartida ao tempo computacional, na obtenção de uma boa solução, quando utilizados para a resolução de problemas de *layout* [4].

O presente trabalho apresenta uma adequação do algoritmo evolutivo aplicado no 1º estágio do trabalho de [1] com duas modificações:

- 1) O uso do operador genético *crossover* de mapeamento parcial; e
- 2) O uso de um código recursivo no estágio de otimização topológica, definido como 1º estágio.

Além da implementação de um *Biased Random-Key Genetic Algorithm* (BRKGA) [6] com o propósito de comparação de resultados.

A otimização topológica (1º estágio) é a fase mais importante da proposta dos autores [1] que contempla dois estágios do processo de otimização do planejamento de facilidades de *layout*. Deste modo, é abordado neste trabalho somente o 1º estágio e, portanto, não será mencionado o 2º estágio, sem prejuízo da solução apresentada.

O presente trabalho encontra-se dividido em oito seções: I. Introdução; II. Trabalhos Relacionados; III. Descrição do Problema; IV. Abordagem do Problema; V. Método; VI. Resultados Computacionais; VII. Discussão dos resultados; VIII. Conclusão.

II. TRABALHOS RELACIONADOS

Com o propósito de selecionar trabalhos com este tema de pesquisa os autores deste artigo realizaram uma Revisão Sistemática de Literatura com as *strings* de busca e de exclusão

da Tabela I. A base de dados utilizada foi a ISI Web of Science no período de 2015–2019. Na literatura há um número incipiente de trabalhos de métodos de programação matemática específicos para o problema de atribuição das facilidades de *layout* de processos de manufatura com fluxo discreto e com sistema de Gestão de Materiais que utiliza equipamentos de transporte e de manuseio de grande porte (*Material-handling system* (MHS)) [2] [12].

TABELA I
STRINGS DE BUSCA E DE EXCLUSÃO

Strings de Inclusão	Strings de Exclusão	Artigos
(ALL=Optimal Shipyard Facility Layout Planning) OR (ALL=Shipyard) AND (ALL=Alignment) AND (ALL=Adjacency) AND (ALL=Aspect Ratio) AND (ALL=Fixed Position Layout) AND (ALL=Handling Equipment HE)	(ALL=Loop Layout) NOT (ALL=Multi Floor Layout) NOT (ALL=Open-field Layout) NOT (ALL=Single Row Layout)	[1]

Há um caráter de ineditismo do trabalho de [1] uma vez que em todos os trabalhos relacionados em [12] nenhum aborda a otimização do fluxo de materiais das facilidades do projeto de *layout* quando considerado nas *workshop characteristics* o caso específico de sistemas de manufatura com *fixed position layout* e o resultado de busca da base ISI Web of Science foi de apenas um trabalho selecionado, o trabalho de [1].

A inovação e a contribuição de [1] encontra-se na abordagem da solução do problema de atribuição de facilidades de *layout* para sistemas de manufatura com *fixed position layout* com dois estágios, sendo: o 1º estágio trata da otimização topológica com o propósito de atender as restrições de proximidade e de adjacência entre os Departamentos do estaleiro e o 2º estágio trata da otimização geométrica com o propósito de obter o melhor aproveitamento de área do terreno do estaleiro respeitando as restrições de dimensão.

Nos estaleiros os materiais tipicamente fluem de um departamento diretamente para outro adjacente e alinhado, o que caracteriza a importância das restrições de adjacência, alinhamento, dimensão de área requerida e relação de aspecto como parâmetros de decisão [1] [4] [5].

Os autores deste artigo, como trabalho futuro, pretendem desenvolver um procedimento integrado com dois estágios. A interface entre os estágios deve ser realizada com a aplicação de um procedimento de transferência da localização do centroide dos Departamentos do estaleiro na *Grid* selecionada no 1º estágio para a *Grid* que representa o terreno real do estaleiro (*Grid* do 2º estágio) considerando a dimensão das áreas de cada departamento previamente definidas. A *Grid* do 2º estágio ao ter os centroides posicionados é a referência inicial para a heurística de crescimento estocástico ampliar os limites de área de cada um dos Departamentos a partir da posição inicial do centroide. No 1º estágio o objetivo é a otimização topológica através do algoritmo evolutivo Genético apresentado nas seções III a VII deste trabalho e no 2º estágio o objetivo é a otimização geométrica respeitando as restrições: relação de aspecto, adjacência, alinhamento e de ocupação física destinada a corredores com a fixação de pilares para o traslado de equipamentos de grande capacidade em toneladas de transporte dos materiais orientado pela matriz de fluxo do

sistema de manufatura dentro dos limites do espaço físico do terreno até minimizar a função multi-objetivo.

III. DESCRIÇÃO DO PROBLEMA

O problema de pesquisa abordado neste trabalho resume-se em aplicar duas propostas de algoritmo genético (*Genetic Algorithms* (GA)) com o propósito de comparação de resultados para a otimização do *layout* de uma indústria de construção naval, com o objetivo de otimizar o custo MHC atendendo aos requerimentos de áreas reservadas, assim como o alinhamento e a adjacência dos departamentos do primeiro estágio da proposta de [1].

A Tabela II apresenta os departamentos do estaleiro objeto do estudo [1]. As Tabelas III e IV apresentam o Fluxo de material entre os Departamentos e as Restrições de Adjacência e Alinhamento entre os Departamentos.

TABELA II
DEPARTAMENTOS DO ESTALEIRO [1]

Departamento	Descrição
1	Escritório
2	Armazém Geral
3	Estoque
4	Pré-Tratamento
5	Fabricação
6	Fabricação dos anteparos (<i>Bulkhead</i> (BHD) <i>fabrication</i>)
7	Linha de Painéis
8	Montagem de Componentes
9	Sub-montagem
10	Montagem de Blocos
11	1ª pré-edificação
12	Espaço destinado a estoque de blocos
13	Pré- <i>outfitting</i>
14	Pintura
15	2ª pré-edificação
16	Espaço destinado a estoque 1º cais
17	Espaço destinado a estoque 2º cais

TABELA III
FLUXO DE MATERIAL ENTRE OS DEPARTAMENTOS DO ESTALEIRO [1]

De	para	Quantidade	De	para	Quantidade
3	4	1860 t	8	10	120 t
4	5	1200 t	9	10	1300 t
4	7	660 t	10	11	700 b
5	8	800 t	10	13	300 b
5	9	95 t	11	12	600 b
5	10	48 t	12	14	600 b
7	9	660 t	13	11	300 b
8	7	4 t	14	15	600 b
8	9	670 t	—	—	—

O estaleiro das tabelas II e III foi projetado para uma capacidade de 150.000 t. de aço anual. Para este caso específico [1] propõe um método de otimização do planejamento das facilidades do *layout* de um estaleiro estruturado a partir de um procedimento computacional de dois estágios, sendo

que no primeiro estágio otimiza-se a topologia dos departamentos utilizando um algoritmo evolutivo (Genético) e, no segundo estágio, a geometria das áreas mediante o uso de uma heurística de crescimento estocástico.

O presente trabalho trata somente do 1º estágio, com ênfase na apresentação dos resultados computacionais obtidos através da implementação conjunta de duas melhorias:

- 1) Aplicação do operador genético *crossover* de mapeamento parcial no estágio de otimização topológica (1º estágio);
- 2) Aplicação de um código recursivo no estágio de otimização topológica (1º estágio).

Além da implementação de um *Biased Random-Key Genetic Algorithm* (BRKGA) com o propósito de comparação de resultados.

A Tabela IV apresenta os departamentos que devem cumprir a restrição de adjacência ou alinhamento.

TABELA IV
RESTRICÇÕES ENTRE OS DEPARTAMENTOS [1]

Departamento	Departamento	Adjacência	Alinhamento
3	4	Sim	Sim
4	5	Sim	Não
4	7	Sim	Sim
5	9	Sim	Sim
7	9	Sim	Não
9	10	Sim	Sim
11	12	Sim	Sim

A Fig. 1 representa a área a qual a topologia da *layout* deve ser otimizada. Os Departamentos 15, 16, 17, 18, 19 e 20 encontram-se fixos e os demais Departamentos da Tabela II, a serem posicionados pelo AE, estão sujeitos a restrições de volume de material transportado entre os pares, que consta na matriz de fluxo da Tabela III, assim como de proximidade entre departamentos, indicados na Tabela IV.

A similaridade da Fig. 1 com o terreno físico do estaleiro objeto do estudo [1] deve ser considerada apenas com relação a entrada do terreno, posicionada no lado esquerdo e na parte superior da *Grid* topológica e o final do fluxo de fabricação representado pelo Departamento 15, o qual encontra-se fixo na posição 19 (Fig. 1 e Fig. 3).

entrada				15	16
					17
	18				
	19	20			

Fig. 1. Representação Topológica da Área do Estaleiro [1].

IV. ABORDAGEM DO PROBLEMA

O FLP tem sido tratado pela literatura através da minimização do custo MHC. Argumenta-se que é o fator mais preponderante nos custos operacionais pois engloba de 20 a 50% do custo total operacional e de 15 a 70% do custo total de produtos manufaturados [5]. Sendo assim, o FLP foi modelado matematicamente como um problema de atribuição quadrática (*Quadratic Assignment Problem* (QAP)) de minimização da função custo MHC, definido como o produto entre o fluxo de materiais, $f_{i,j}$; o custo de transporte por distância percorrida e por volume, $c_{i,j}$; e a distância entre os departamentos, $d_{i,j}$ [7]. Função Objetivo (1). A abordagem QAP para resolver o FLP tem sido classificada como um problema matemático do tipo NP-hard, onde os tempos para encontrar a solução ideal utilizando modelos exatos são improcedentes para problemas de *layout* com mais de vinte e cinco departamentos [4].

Pelo motivo exposto, a literatura destaca técnicas heurísticas e meta-heurísticas que se aproximam razoavelmente da solução exata nos problemas de *layout* com maior número de departamentos [8]. Em 2017, os autores [1] apresentaram uma abordagem de otimização de *layout* de estaleiros inspirada nas metodologias aplicadas na arquitetura naval para o *design* do arranjo do convés de navios. Essa metodologia foi estendida e adaptada pelos autores supracitados para projetar o *layout* de um estaleiro de tamanho médio.

A topologia é uma restrição que permite determinar a relação lógica entre os departamentos que dão forma física a um *layout*: especificar a adjacência, não adjacência ou proximidade de um departamento com outro, ou de um departamento com o perímetro do local [9] [10].

Neste trabalho, propõe-se estudar o problema do arranjo topológico de um *layout* a partir da proposta [1], utilizando técnicas de exploração do domínio específicas para problemas combinatórios, no caso, ampliando a eficiência e a eficácia da proposta original com o uso do *crossover* de ciclo e o *crossover* PMX. Destaca-se que o *crossover* PMX, foi aplicado para minimizar a distância total necessária de movimentação entre máquinas no processo de corte e de conformação de chapas de um estaleiro com capacidade para 2000 toneladas a cada turno de 8 horas (*Multi-row machine layout*) [11].

Contudo, o resultado apresentado na sessão VI demonstra que o AG com as melhorias propostas realizou uma exploração abrangente do domínio do espaço solução, encontrando uma topologia possível para o *layout* do problema real utilizado como objeto do estudo. O mesmo comportamento se observa nos resultados do BRKGA. Além da adjacência, considera-se neste problema um tipo especial de restrição: a restrição de alinhamento entre os departamentos, como demonstrado na Fig. 2.

O problema matemático de otimização, de acordo com o modelo apresentado em [1], é descrito constituindo-se das definições da Tabela V, das equações 1 a 7 e das restrições.

As letras superiores tp e tv indicam "parâmetro de topologia" e "variável de topologia". Para a representação da topologia os autores utilizaram uma grade topológica (Fig. 1 e Fig. 3) de 20 genes, na qual cada gene indica uma posição sequencial. O preenchimento é inicializado na parte inferior esquerda da matriz. A Fig. 3 descreve graficamente esta grade.

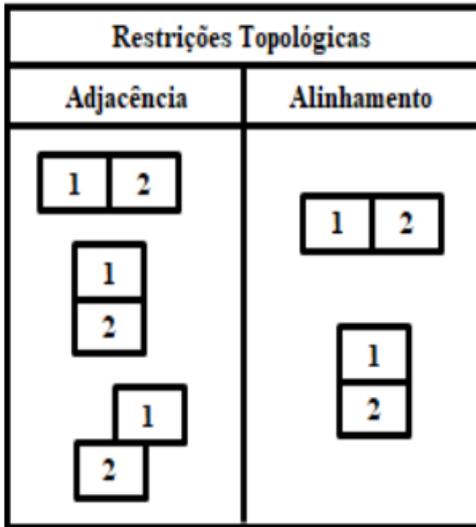


Fig. 2. Definição das Restrições Topológicas de Alinhamento e Adjacência.

Note-se que os departamentos 19, 20, 18, 17, 15 e 16 estão indicados na Fig. 3 com posição fixa, razão pela qual são atribuídos ao conjunto dos genes fixos: x_1^{TV} , x_2^{TV} , x_6^{TV} , x_{15}^{TV} , x_{17}^{TV} e x_{20}^{TV} , respectivamente. Os departamentos não fixos, representados pela variável x_g^{TV} , a qual indica o departamento atribuído ao g -ésimo gene, são então atribuídos aleatoriamente aos genes restantes quando a população inicial é gerada.

TABELA V
DEFINIÇÕES MATEMÁTICAS [1]

Conjuntos	
D:	Conjunto dos departamentos 'd' (ver Tabela II)
G:	Genes do cromossomo 'g'
F:	Conjunto dos genes fixos no cromossomo 'f' (ver Tabela II)
Parâmetros	
f_{ij}^{pp} :	Fluxo de materiais entre os departamentos (dados na Tabela III)
c_{ij}^{pp} :	Custo de transporte por distância percorrida e por unidade de volume entre os departamentos i e j
cl_{ij}^{pp} :	1 se os departamentos i e j tiverem restrição de proximidade, caso contrário 0 (ver Tabela IV)
p_f^{pp} :	Departamento predeterminado inicialmente atribuído ao gene f
N:	Número de departamentos
Variáveis	
x_g^{TV} :	Departamento atribuído ao gene g
d_{ij}^{TV} :	Distância euclidiana entre os departamentos i e j
cx_d^{TV} :	Centroid x do departamento d
cy_d^{TV} :	Centroid y do departamento d
cl_{ij}^{TV} :	1 se os departamentos i e j satisfazem a restrição de proximidade, caso contrário, 0.

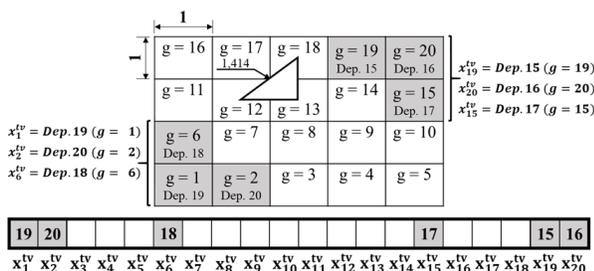


Fig. 3. Cromossomo para Otimização Topológica [1].

A restrição de proximidade entre os departamentos é definida por meio da variável binária $cl_{i,j}^{TV}$, definida a partir do cálculo da distância euclidiana entre o centroide dos departamentos da grade, considerando: caso for menor ou igual a $\sqrt{2}$, a restrição é satisfeita, de acordo com a expressão (8).

Modelo (função objetivo e restrições):

$$\text{Minimizar} \left\{ \sum_{i \in D} \sum_{j \in D} f_{ij}^{pp} \cdot c_{ij}^{pp} \cdot d_{ij}^{TV} \right\} \quad (1)$$

s.a:

$$cl_{i,j}^{TV} \geq cl_{i,j}^{pp} \quad ; \quad i, j \in D \quad (2)$$

$$cl_{i,j}^{TV} \in \{0, 1\} \quad ; \quad i, j \in D \quad (3)$$

$$x_f^{TV} = p_f^{pp} \quad ; \quad f \in F \quad (4)$$

$$x_i^{TV} \neq x_j^{TV} \quad ; \quad i, j \in G \quad (5)$$

$$x_g^{TV} \in \{1, 2, \dots, N\} \quad ; \quad g \in G \quad (6)$$

$$d_{ij}^{TV} = \sqrt{(cx_i^{TV} - cx_j^{TV})^2 + (cy_i^{TV} - cy_j^{TV})^2} \quad ; \quad i, j \in D \quad (7)$$

$$(cl_{i,j}^{TV} | cl_{i,j}^{pp} = 1) \quad \text{se} \quad d_{ij}^{TV} \leq \sqrt{2} \quad ; \quad \text{senão} \quad cl_{i,j}^{TV} = 0 \quad (8)$$

V. MÉTODO

Codificaram-se dois algoritmos evolutivos com a finalidade de replicar o problema detalhado nas seções I e III. Para tanto utilizou-se o *GA_framework* para MATLAB [12] e o *software MATLAB* Copyright©2018.

Dois fatores principais foram levados em consideração ao fazer essa escolha:

- 1) Praticidade. Uma plataforma robusta, completa e genérica de AG permite economizar tempo e esforço durante o desenvolvimento do código, que podem ser aplicados em outros aspectos do trabalho;
- 2) Customização. Por permitir modificações extensas e profundas em seu funcionamento a plataforma não se torna um impedimento para um desenvolvimento livre do código;

- 1º algoritmo evolutivo (AG)

Para cada teste do plano de experimentos definido, o AG gera uma população de N indivíduos aleatoriamente. Para cada resultado o algoritmo avalia o *fitness* de cada indivíduo, selecionando os 50% melhores mediante torneio por duplas, onde somente o melhor sobrevive.

Depois, inicia a combinação deles mediante os operadores de *crossover* e mutação, utilizando *crossover* do tipo cíclico, do tipo PMX e mutação randômica binária.

O critério de parada do algoritmo genético é o momento no qual, em alguma geração, seja atingido um delta de 1% entre os valores da função objetivo com relação a geração anterior. A Fig. 4 é o fluxograma do AE utilizado no processo heurístico deste trabalho.

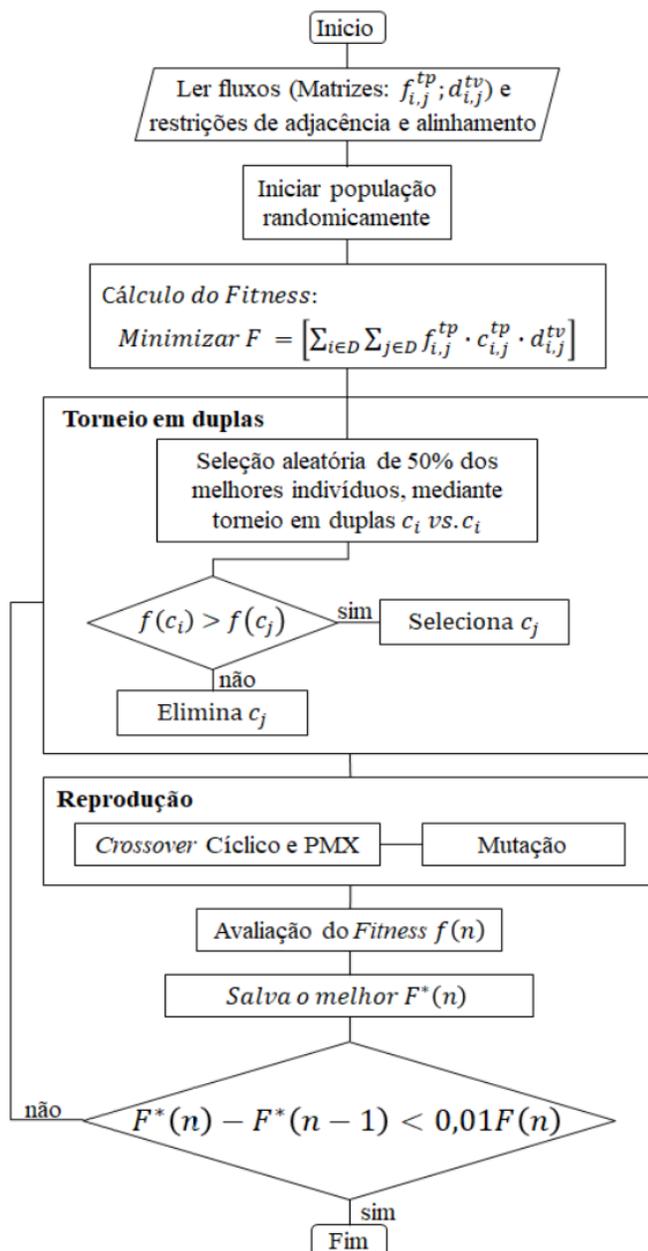


Fig. 4. Fluxograma do Algoritmo Genético Proposto.

Contudo, deve ser destacado, que um Algoritmo Genético opera através de mecanismos de seleção natural com ênfase na sobrevivência do indivíduo mais apto. Como consequência não garante a obtenção da melhor solução possível, mas permite obter boas soluções para problemas NP-hard, considerados complexos e NP-completos [13].

A Fig. 5 apresenta o pseudocódigo correspondente à função de geração de indivíduos utilizada no AE.

Os operadores de *crossover* utilizados para a criação das novas gerações, como exposto na Fig. 4, é descrito.

- *Crossover* cíclico [14].

- 1) Cada gene vem de um dos pais junto com sua posição:
 - a) Realizar um ciclo de alelos de P1.
 - b) Inicie com o primeiro gene de P1.
 - c) Verifique o gene na mesma posição em P2.
 - d) Vá para a posição com o mesmo gene em P1.
 - e) Adicione esse gene ao ciclo.
 - f) Repetir o passo b até d até o primeiro gene de P1.
- 2) Colocar os genes do ciclo no primeiro filho nas posições que eles têm no primeiro pai.
- 3) Realizar o próximo ciclo do segundo pai.

As Fig. 6 e Fig. 7 representam os passos do operador.

- *Crossover* PMX. Procedimento para os pais P1 e P2 [14].

- 1) Selecionar um segmento aleatório e copiar de P1.
- 2) A partir do primeiro ponto de cruzamento procurar elementos nesse segmento de P2 que não foram copiados.
- 3) Para cada um desses i , procurar no filho e verificar qual elemento j foi copiado em seu lugar de P1.
- 4) Colocar i na posição ocupada por j em P2, j encontra-se no filho.
- 5) Se o lugar ocupado por j em P2 já tiver sido preenchido no filho k , colocar i na posição ocupada por k em P2.

Tendo trabalhado com os elementos do segmento de *crossover*, a parte restante da descendência pode ser preenchida de P2. O segundo filho é criado de maneira análoga, de acordo com a Fig. 8.

- *Mutaçao*:

Swapping: Este tipo de mutação seleciona dois genes aleatoriamente e troca suas posições. Este operador preserva a maioria das informações de adjacência [14], de acordo com a Fig. 9.

- 2º algoritmo evolutivo (BRKGA)

O BRKGA implementado pelos autores deste trabalho cria cromossomos que respeitam a aleatoriedade do algoritmo sem deixar de se manterem fieis às restrições de posição e adjacência impostas pelo problema.

Portanto a solução encontrada considera os genes que representam os departamentos com posição fixa atribuindo o valor -1 e para os genes que representam os departamentos sem restrições atribui valores aleatórios entre 0 e 1. Esse valor ao ser lido pelo decoder indica se a posição do departamento é fixa ou aleatória. A partir do vetor de posição, o algoritmo irá, de forma recursiva, adicionar todos os departamentos que possuem restrições de adjacência de modo que não desrespeitem nenhuma delas. Após esse processo, por fim, os departamentos sem nenhum tipo de restrição serão adicionados e receberão rótulos aleatórios. Conhecendo, então, qual será a posição final

dos departamentos após a aplicação do decoder, é importante que os departamentos com restrição de adjacência recebam rótulos condizentes com suas posições. Os rótulos destes genes são calculados a partir dos rótulos de genes próximos, de modo a mantê-los exatamente naquelas posições. Esses rótulos são adicionados ao cromossomo em seus respectivos genes e então um indivíduo estará criado.

Dados: número de departamentos N , lista de departamentos com posição fixa e suas respectivas posições **fixedPos**, lista de departamentos com restrições de posição **dep_constraints**

- 1- Criar vetor **solution** de tamanho N ;
- 2- Se houverem departamentos com posição fixa;
- 3- Inserir-los no vetor **solution** nas posições exigidas;
- 4- Se houverem departamentos com restrições de posição;
- 5- Armazenar o estado atual de **solution**;
- 6- Para cada item em **dep_constraints**
- 7- Verificar se algum desses departamentos já estão inseridos;
- 8- Se sim
- 9- Inserir o departamento que ainda não possui posição. Sua posição deve ser escolhida aleatoriamente de um espaço amostral composto apenas por posições que não desrespeitem as restrições obrigatórias; Caso não seja possível, restaurar o estado de **solution** armazenado no passo 6 e voltar ao passo 7;
- 10- Se não
- 11- Inserir o primeiro departamento de modo totalmente aleatório;
- 12- Inserir o departamento que ainda não possui posição. Sua posição deve ser escolhida aleatoriamente de um espaço amostral composto apenas por posições que não desrespeitem as restrições obrigatórias; Caso não seja possível, restaurar o estado de **solution** armazenado no passo 6 e voltar ao passo 7;
- 13- Inserir os demais departamentos de modo totalmente aleatório nas posições vazias;

Fig. 5. Pseudocódigo da Função de Criação de Indivíduos [16].

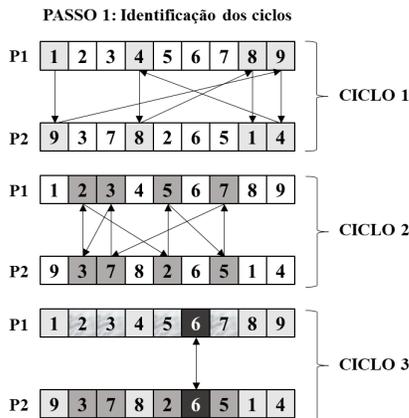


Fig. 6. Crossover cíclico, Passo 1 [14].

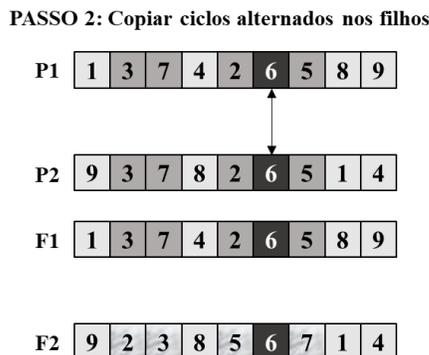


Fig. 7. Crossover cíclico, Passo 2 [14].

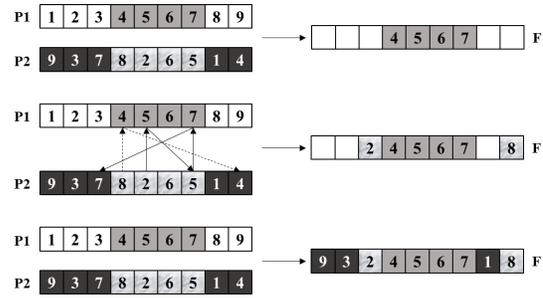


Fig. 8. Crossover PMX [14].



Fig. 9. Mutação swapping [14].

Repetindo esse processo N vezes, teremos a população inicial. Dessa população serão selecionados os $X\%$ melhores indivíduos ($0 < X <= 25$) para fazerem parte da Elite, um grupo de indivíduos que automaticamente fará parte da próxima população até a convergência para o ótimo global.

VI. RESULTADOS COMPUTACIONAIS

A Tabela VI mostra os resultados do 1º grupo de 50 testes computacionais do primeiro algoritmo evolutivo (GA) para quarenta e três indivíduos, mantendo o departamento 03 sem posição fixa.

A Tabela VII mostra os resultados do 2º grupo de 50 testes computacionais do primeiro algoritmo evolutivo (GA) para quarenta e três indivíduos, mantendo o departamento 03 fixo na posição indicada na melhor solução do 1º grupo ($posição: x_{11}^{IV} = Dep. 03 (g = 11)$). Ambos os grupos com 100 iterações. Os índices de mutação e crossover indicados nas tabelas VI e VII foram obtidos através dos experimentos computacionais realizados pelos autores deste trabalho.

Na Tabela VI, a indicação dos resultados dos experimentos 10 e 33, com 1.400 e 7.000 indivíduos respectivamente, observa-se a melhor solução com custo de 11.489 unidades monetárias, custo inferior a 11.816 encontrado anteriormente em [1].

Na Tabela VII a melhor solução é encontrada a partir do experimento 13 e se mantém do experimento 32 ao experimento 50, ou seja, a partir de 8.000 indivíduos a melhor solução é encontrada repetidamente. O experimento ** da Tabela VI é o experimento 50 da Tabela VII.

Em [1] é mencionado pelos autores, no 1º estágio, o número de iterações da ordem de 3.740 para a obtenção da melhor solução encontrada.

Neste trabalho, os algoritmos evolutivos foram executados para todos os experimentos computacionais com o número de 100 iterações.

Contudo, com o objetivo de comparação, com um número de 7.000 indivíduos, um dos procedimentos computacionais propostos neste artigo foi executado com 100 (Tabela VIII) e 4.000 (Tabela IX) iterações (primeiro algoritmo evolutivo). Os resultados obtidos, de acordo com as Tabelas VIII e IX

mostram que, com o uso da solução computacional apresentada neste trabalho, o resultado com 100 iterações é melhor do que o resultado com 4.000 iterações, além do aumento significativo do tempo computacional, que com 4.000 iterações é igual a 41:21:20 horas (Tabela IX).

TABELA VI
1º GRUPO DE EXPERIMENTOS. PARÂMETROS SIMILARES AOS DO TRABALHO [1] (DEP. 03 SEM POSIÇÃO FIXA)

E.	Ind.	Custo	Mut.	Cross.	Tempo (s)	Tempo (h:m:s)
1	500	12.261	0,078	0,743	286,61	00:04:46
2	600	12.084	0,078	0,703	313,11	00:05:13
3	700	11.851	0,081	0,746	377,37	00:06:17
4	800	11.782	0,090	0,833	450,11	00:07:30
5	900	11.563	0,090	0,833	511,07	00:08:31
6	1.000	11.914	0,073	0,696	534,34	00:08:54
7	1.100	11.314	0,081	0,721	626,45	00:10:26
8	1.200	11.109	0,090	0,833	595,05	00:09:55
9	1.300	12.084	0,072	0,696	743,14	00:12:23
10	1.400	11.489	0,077	0,689	753,15	00:12:33
11	1.500	11.109	0,074	0,685	812,70	00:13:32
12	1.600	11.074	0,072	0,685	858,60	00:14:18
13	1.700	10.944	0,084	0,789	949,90	00:15:49
14	1.800	11.782	0,079	0,763	927,99	00:15:28
15	1.900	11.119	0,090	0,833	986,28	00:16:26
16	2.000	11.923	0,078	0,762	1.117,6	00:18:37
17	2.500	11.604	0,087	0,797	1.250,1	00:20:50
18	3.000	11.674	0,090	0,833	1.577,3	00:26:17
19	3.500	11.544	0,090	0,833	1.687,8	00:28:07
20	4.000	11.192	0,090	0,833	2.109,2	00:35:09
21	4.500	11.780	0,079	0,706	2.464,3	00:41:04
22	5.000	11.544	0,087	0,782	2.462,3	00:41:02
23	5.500	11.544	0,087	0,803	2.877,8	00:47:57
24	100	12.644	0,072	0,691	63,620	00:01:03
25	500	12.757	0,081	0,764	258,67	00:04:18
26	1.000	11.604	0,077	0,719	550,09	00:09:10
27	1.500	11.660	0,072	0,689	808,37	00:13:28
28	2.000	11.660	0,088	0,800	1.105,3	00:18:25
29	2.500	11.544	0,080	0,747	1.350,3	00:22:30
30	3.000	11.780	0,076	0,686	1.552,4	00:25:52
31	5.000	11.445	0,090	0,833	2.682,2	00:44:42
32	6.000	11.563	0,069	0,630	3.011,9	00:50:12
33	7.000	11.489	0,072	0,695	3.383,6	00:56:23
34	8.000	11.544	0,079	0,713	4.027,2	01:07:07
35	9.000	11.109	0,071	0,697	4.745,7	01:19:05
36	10.000	11.250	0,077	0,755	5.402,0	01:30:20
37	15.000	10.944	0,090	0,833	8.255,8	02:17:35
38	20.000	11.074	0,079	0,740	10.833	03:00:33
39	25.000	11.192	0,090	0,833	14.332	03:58:52
40	30.000	10.944	0,080	0,721	15.808	04:23:28
41	35.000	11.109	0,079	0,702	21.136	05:52:16
42	40.000	11.109	0,077	0,695	23.831	06:37:11
43	45.000	11.074	0,075	0,741	24.318	06:45:18
44	50.000	10.944	0,088	0,808	30.482	08:28:02
45	55.000	11.074	0,084	0,773	35.595	09:53:15
46	60.000	11.109	0,076	0,728	38.365	10:39:25
47	65.000	11.074	0,079	0,713	36.672	10:11:12
48	70.000	11.109	0,086	0,806	42.805	11:53:25
49	75.000	10.944	0,076	0,683	44.003	12:13:23
50	80.000	10.944	0,072	0,704	46.705	12:58:25
*	80.000	11.489	0,073	0,709	38.481	10:41:21
				Total	447.350	124:15:5
					5 dias	04:15:54

Deve ser destacado, contudo, que a solução computacional proposta, para ambos os grupos de experimentos adotados neste trabalho, obtém a melhor solução a partir do número de 7.000 indivíduos com 100 iterações, e um tempo computacional de menos de 1 hora de execução para os dados do estaleiro objeto do estudo. Um comportamento similar

se observa nos resultados do BRKGA (segundo algoritmo evolutivo) de acordo com os resultados da Figura 10.

TABELA VII
2º GRUPO DE EXPERIMENTOS. PARÂMETROS SIMILARES AOS DO TRABALHO [1] POSIÇÃO DEP. 03: ($x_{II}^{IV} = Dep. 03 (G = 11)$)

E.	Ind.	Custo	Mut.	Cross.	Tempo (s)	Tempo (h:m:s)
1	500	12.323	0,077	0,753	222,47	00:03:00
2	600	12.226	0,077	0,699	267,86	00:04:00
3	700	11.489	0,085	0,784	330,85	00:05:00
4	800	11.816	0,078	0,740	369,01	00:06:09
5	900	11.816	0,072	0,701	428,61	00:07:08
6	1.000	11.927	0,075	0,742	483,56	00:08:03
7	1.100	11.816	0,072	0,695	508,41	00:08:28
8	1.200	11.685	0,077	0,689	538,06	00:08:58
9	1.300	11.685	0,077	0,733	601,08	00:10:01
10	1.400	11.862	0,077	0,755	618,77	00:10:18
11	1.500	11.489	0,075	0,746	760,08	00:12:40
12	1.600	11.685	0,090	0,833	753,09	00:12:33
13	1.700	11.489	0,081	0,774	806,23	00:13:26
14	1.800	11.862	0,073	0,727	817,45	00:13:37
15	1.900	11.975	0,068	0,632	860,43	00:14:20
16	2.000	11.816	0,090	0,833	948,41	00:15:48
17	2.500	11.780	0,068	0,651	1.106	00:18:26
18	3.000	11.489	0,073	0,710	1.460	00:24:20
19	3.500	11.685	0,076	0,745	1.682	00:28:02
20	4.000	11.489	0,079	0,713	1.777	00:29:37
21	4.500	11.685	0,084	0,799	2.182	00:36:22
22	5.000	11.489	0,090	0,833	2.435	00:40:35
23	5.500	11.685	0,072	0,699	2.432	00:40:32
24	100	11.927	0,089	0,830	60,89	00:01:00
25	500	12.037	0,073	0,716	251,70	00:04:11
26	1.000	11.489	0,079	0,709	497,10	00:08:17
27	1.500	11.975	0,083	0,773	668,00	00:11:08
28	2.000	11.840	0,085	0,785	937,95	00:15:38
29	2.500	11.816	0,064	0,623	1.286	00:21:26
30	3.000	11.816	0,075	0,742	1.451	00:24:11
31	5.000	11.685	0,090	0,833	2.407	00:40:07
32	6.000	11.489	0,090	0,833	2.863	00:47:43
33	7.000	11.489	0,090	0,833	3.256	00:54:16
34	8.000	11.489	0,088	0,797	4.086	01:08:06
35	9.000	11.685	0,076	0,692	4.177	01:09:37
36	10.000	11.489	0,079	0,720	5.052	01:24:12
37	15.000	11.489	0,072	0,708	7.213	02:00:13
38	20.000	11.489	0,085	0,786	10.721	02:58:41
39	25.000	11.489	0,076	0,695	12.705	03:31:45
40	30.000	11.489	0,090	0,833	15.851	04:24:11
41	35.000	11.489	0,085	0,786	19.090	05:18:10
42	40.000	11.489	0,074	0,737	20.405	05:40:05
43	45.000	11.489	0,072	0,698	22.113	06:08:33
44	50.000	11.489	0,079	0,719	26.094	07:14:54
45	55.000	11.489	0,090	0,833	28.715	07:58:35
46	60.000	11.489	0,087	0,795	28.874	08:01:14
47	65.000	11.489	0,090	0,833	32.559	09:02:39
48	70.000	11.489	0,076	0,750	33.353	09:15:53
49	75.000	11.489	0,086	0,807	37.644	10:27:24
50	80.000	11.489	0,073	0,709	38.481	10:41:21
-	-	-	-	-	-	-
				Total	383.200	106:26:4
					4 dias	10:26:47

TABELA VIII
EXPERIMENTO COM 100 ITERAÇÕES (EXPERIMENTO 33 DA TABELA VI)

E.	Ind.	Custo	Mut.	Cross.	Tempo (s)	Tempo (h:m:s)
33	7.000	11.489	0,07	0,6955	3.383,6	00:56:23
-	-	-	-	Total	3.383,6	00:56:23
-	-	-	-		0 dia	00:56:23

TABELA IX
EXPERIMENTO COM 4.000 ITERAÇÕES (EXPERIMENTO 33 DA TABELA VI)

E.	Ind.	Custo	Mut.	Cross.	Tempo (s)	Tempo (h:m:s)
33	7.000	11.544	0,08	0,8034	148.880,0	41:21:20
Total					148.880,0	41:21:20
1 dia						17:21:20

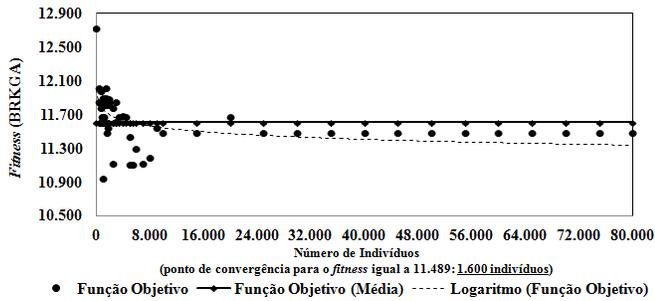


Fig. 10. Variabilidade do *Fitness* como uma função Número de indivíduos. Dep. 03 com posição fixa (dados BRKGA).

VII. DISCUSSÃO

O tempo de execução do procedimento proposto (primeiro algoritmo evolutivo), de acordo com as Tabelas VI e VII, aumenta com maior intensidade com o número de indivíduos maior do que 7.000.

Considerando que a partir deste valor a melhor solução já é obtida, em termos de tempo de execução para os parâmetros e variáveis do estaleiro objeto do estudo, é razoável considerar que a duração do processo é inferior a 1 hora.

Com relação a dispersão do valor *fitness* em função do espectro do índice de mutação e *crossover*, Fig. 11, Fig. 12, Fig. 14 e Fig. 15, pode ser considerado que ambos os AE atendem aos padrões considerados aceitáveis para um algoritmo evolutivo, podendo ser aprimorados com o uso de outras linguagens de programação computacional, uma vez que o *software* MATLAB tem como propósito dar suporte como ferramenta computacional à projetos de pesquisa. Deve ser destacado que o valor ótimo é representado pelo menor custo de movimentação de materiais no estaleiro considerando o uso da matriz de fluxo. As Fig. 11 até 18 apresentam os resultados do primeiro algoritmo evolutivo.

A Fig. 11, mostra o espectro do índice de mutação com variação de 0,0689 a 0,0900 para o 1º Grupo de experimentos.

A Fig. 12, mostra o espectro do índice de *crossover* com variação de 0,6303 a 0,8333 para o 1º Grupo de experimentos.

A Fig. 13, mostra o resultado da função objetivo (*fitness*) com espectro de 10.944,00 a 12.757,00 unidades monetárias para o 1º Grupo de experimentos.

É importante destacar que as Fig. 13 e Fig. 16 indicam que a melhor solução para cada grupo de experimentos com base no custo de movimentação de materiais (*fitness*) é de 11.489 unidades monetárias.

O que, não necessariamente, para cada um dos experimentos com este resultado (*fitness*), a posição de cada um dos departamentos na grade deve coincidir com o resultado

apresentado na Fig. 18 referente ao experimento 18, 33, 34 e 50. Resultado considerado como a melhor configuração da grade neste trabalho.

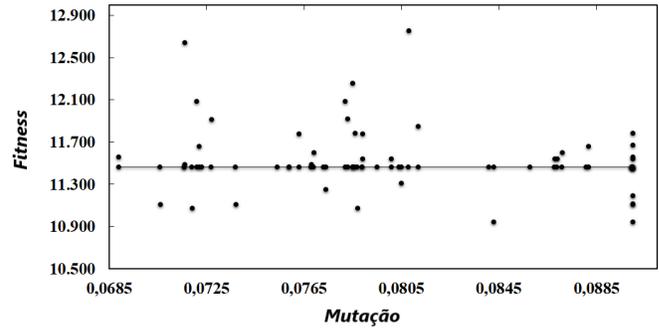


Fig. 11. Variabilidade do *Fitness* como uma Função da Mutação. Dep. 03 sem posição fixa (dados Tabela VI).

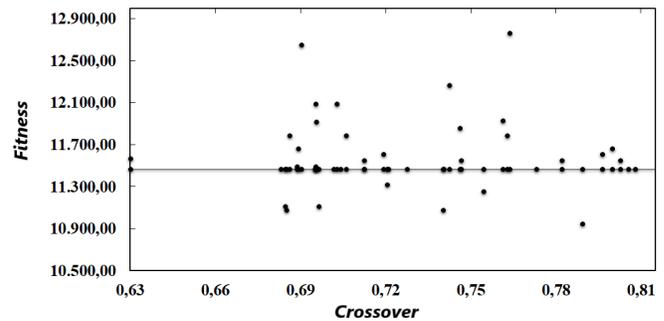


Fig. 12. Variabilidade do *Fitness* como uma Função do *Crossover*. Dep. 03 sem posição fixa (dados Tabela VI).

A Fig. 14, mostra o espectro do índice de mutação com variação de 0,0643 a 0,0900 para o 2º Grupo de experimentos.

A Fig. 15, mostra o espectro do índice de *crossover* com variação de 0,6235 a 0,8333 para o 2º Grupo de experimentos.

A Fig. 16, mostra o resultado da função objetivo (*fitness*) com espectro de 11.489,00 e 12.323,00 unidades monetárias para o 2º Grupo de experimentos.

A Fig. 17 mostra o resultado da ocupação da grade topológica com o custo de movimentação de materiais igual a 11.816 unidades monetárias, similar ao resultado obtido através de [1].

A Fig. 18 mostra o resultado da ocupação da grade topológica com o custo de movimentação de materiais igual a 11.489 unidades monetárias, resultado obtido neste trabalho.

Considerando que diferentes configurações da posição dos departamentos na grade representa mais de uma alternativa de projeto de facilidades de *layout*, cabe ao tomador de decisão avaliar cada uma das possibilidades ou cenários de mesmo valor *fitness*, de modo a decidir qual configuração é oportunamente mais adequada.

Neste caso a decisão ou escolha da melhor configuração deve considerar além do *fitness*, o melhor fluxo e a melhor disposição dos departamentos.

Isto requer uma análise precisa dos resultados obtidos dos experimentos indicados na Tabela V (primeiro algoritmo

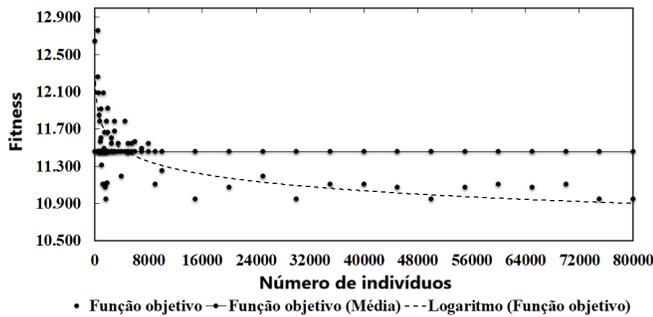


Fig. 13. Variabilidade do *Fitness* como uma função Número de indivíduos. Dep. 03 sem posição fixa (dados Tabela VI).

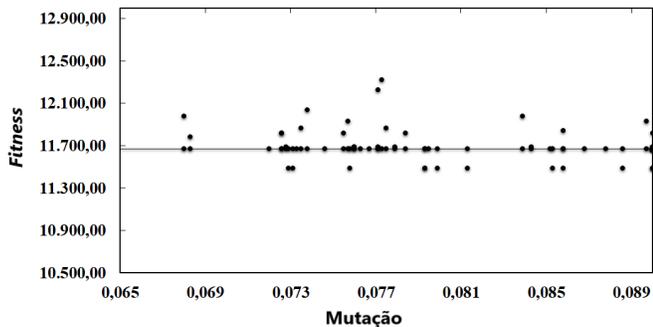


Fig. 14. Variabilidade do *Fitness* como uma Função da Mutação. Dep. 03 com posição fixa (dados Tabela VII).

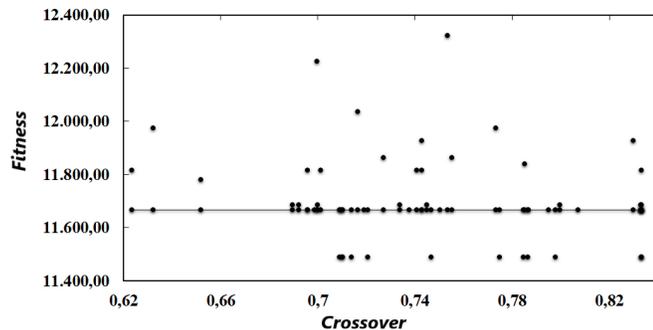


Fig. 15. Variabilidade do *Fitness* como uma Função do *Crossover*. Dep. 03 com posição fixa (dados Tabela VII).

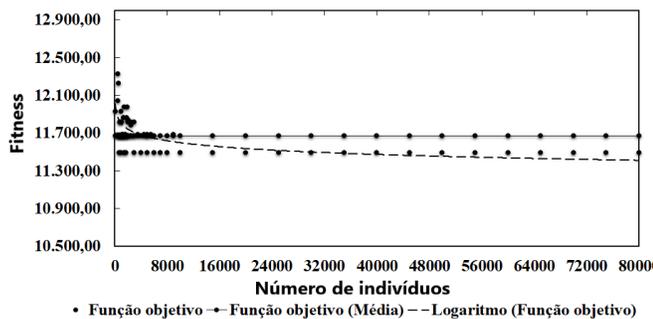


Fig. 16. Variabilidade do *Fitness* como uma função Número de indivíduos. Dep. 03 com posição fixa (dados Tabela VII).

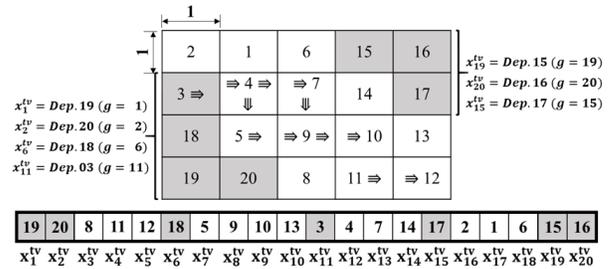


Fig. 17. Resultado do experimento 16 e 30 do 2º grupo. Custo igual a 11.816.

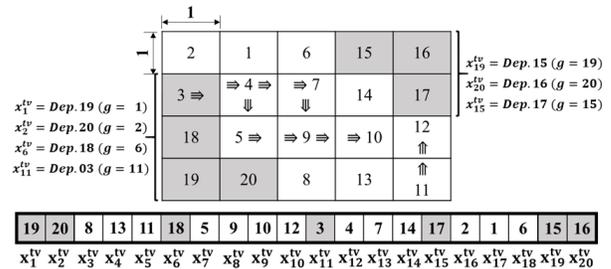


Fig. 18. Resultado do experimento 18, 33, 34 e 50 do 2º grupo. Custo igual a 11.489.

evolutivo) e na Fig. 10 (segundo algoritmo evolutivo) com *fitness* igual a 11.489 unidades monetárias e a posição dos departamentos na grade de cada uma das configurações.

VIII. CONCLUSÃO

O principal indicador quantitativo obtido através de [4] é o número de trabalhos encontrados com o uso de Algoritmo Genético (AG) para a resolução do problema FLP: 50 trabalhos, mas nenhum trata do problema de atribuição de facilidades do *fixed position layout*. O que permite afirmar que há na literatura um número significativo de artigos científicos com contribuição relevante para a resolução do problema FLP utilizando AG de vários outros tipos de sistemas de manufatura mas não de *layout* de estaleiros. A contribuição científica de [1] é a divisão da resolução do problema FLP com o uso de duas técnicas entre os dois estágios do procedimento proposto pelos autores: 1. Otimização topológica da grade dos Departamentos com o uso de um AG e 2. Otimização geométrica da grade dos Departamentos com o uso de uma heurística de crescimento estocástico para o caso específico de *layout* de estaleiros. No caso deste trabalho, de acordo com o argumento de [15], que considera a função de *crossover* um procedimento importante de um AG, a contribuição científica encontra-se fundamentada através do uso da técnica mapeamento parcial e do algoritmo recursivo aplicado para ambos os procedimentos genéticos propostos neste trabalho para a solução do primeiro estágio de [1]. Durante a etapa de reprodução em um AG dois cromossomos são combinados para gerar um novo indivíduo. Nesse momento inúmeras técnicas de *crossover* podem ser aplicadas, cada uma possuindo características específicas que as tornam mais ou menos efetivas. A técnica *PMX* (primeiro algoritmo evolutivo) foi escolhida pois respeita as restrições

do problema, que apresenta cromossomos ordenados sem repetição, ou seja, cada gene possui um escopo limitado de valores possíveis e não podem haver dois ou mais genes com o mesmo valor em um único cromossomo. Portanto um algoritmo totalmente aleatório não atuaria com a eficiência requerida, tornando necessária a criação de uma função que caminhe pelo cromossomo a ser criado e escolha aleatoriamente um valor para cada posição, levando em consideração um escopo específico do gene em questão, possuindo apenas valores que ao serem utilizados naquela posição não desrespeitariam nenhuma restrição. Em situações como essa um algoritmo recursivo se prova muito mais eficiente que um iterativo, pois facilita o *backtracking* (técnica que consiste em voltar ao passo anterior do algoritmo caso o próximo passo não consiga ser executado). O que pôde ser comprovado nos dois algoritmos genéticos apresentados neste trabalho. Como resultado dos procedimentos propostos deve ser destacado que com apenas 100 iterações o custo mínimo de 11.489 unidades foi obtido, enquanto o procedimento proposto por [1] alcançou o mesmo custo mínimo com 3740 iterações.

AGRADECIMENTOS

Agradecemos a Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) e ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelo apoio financeiro, aos pesquisadores Minjoo Choi, Sang Hyun Kim & Hyun Chung da Division of Ocean Systems Engineering, School of Mechanical Aerospace & Systems Engineering, Korea Advanced Institute of Science and Technology, Daejeon, Republic of Korea e ao pesquisador Leandro Resende Mundim do Centro de Matemática e Estatística Aplicadas à Indústria (CEPID - CeMEAI/ICMC/USP) pela gentil orientação, a Astilleros Navales Ecuatorianos EP (Astinave EP) pela disponibilidade de dados reais para o desenvolvimento de trabalhos futuros do tema de pesquisa e a SIEMENS PLM Software Corp.

REFERÊNCIAS

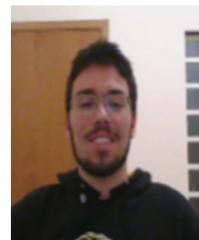
- [1] M. Choi, S. H. Kim, and H. Chung, "Optimal shipyard facility layout planning based on a genetic algorithm and stochastic growth algorithm," *Ships and Offshore Structures*. Vol. 12, no. 4, pp. 485-494. May. 2017, doi: 10.1080/17445302.2016.1176294.
- [2] J. G. Shin, Y. J. Song, D. K. Lee and J. H. Woo, "A Concept and Framework for a Shipyard Layout Design Based on Simulation," *Journal of Ship Production*, Vol. 25, no. 3, pp. 126-135. August. 2009.
- [3] A. Drira, H. Pierreval and S. Hajri-Gabouj, "Design of a robust layout with information uncertainty increasing over time: a fuzzy evolutionary approach," *Engineering Applications of Artificial Intelligence*. 2013, doi: 10.1016/j.engappai.2012.12.007.
- [4] H. Hosseini-Nasab, S. Fereidouni, S. M. T. F. Ghomi and M. B. Fakhrazad, "Classification of facility layout problems: a review study," *Eng. Rev., International Journal of Advanced Manufacturing Technology*. Vol. 94, pp. 957-977. July. 2017.
- [5] J. A. Tompkins, J. A. White, Y. A. Bozer and J. M. A. Tanchoco, "Introduction," in *Facilities Planning*, 2nd ed., vol. 1, New York, NY, USA: John Wiley & Sons, INC, 1996, pp. 3-23.
- [6] L. R. Mundim, "Mathematical models and heuristic methods for nesting problems," Doctorate Thesis, University of São Paulo, 2017.
- [7] T. C. Koopmans and M. Beckmann, "Assignment problems and the location of economic activities," *Econometrica*. Vol. 25, no. 1, pp. 53-76. Jan. 1957, doi: 10.2307/1907742.

- [8] T. D. Mavridou and P. M. Pardalos, "Simulated annealing and genetic algorithms for the facility layout problem: a survey," *Computational Optimization and Applications*. Kluwer Academic Publishers. Vol. 7, no. 1, pp. 111-126. Jan. 1997, doi: 10.1023/A:1008623913524.
- [9] J. J. Michalek, R. Choudhary and P. Y. Papalambros, "Architectural layout design optimization," Vol. 34, no. 5, pp. 461-484. Feb. 2002, doi: 10.1080=0305215021000033735.
- [10] B. Medjdoub and B. Yannou, "Separating topology and geometry in space planning," *CAD Computer Aided Design*. Vol. 32, no. 1, pp. 39-61. Jan. 2000, doi: 10.1016/S0010-4485(99)00084-6.
- [11] Y. Liu, M. Meng and S. Liu, "Layout design-based research on optimization and assessment method for shipbuilding workshop," *Journal of Marine Science and Application*. Vol. 12, no. 2, pp. 152-162. June. 2013, doi: 10.1007/s11804-013-1180-3.
- [12] A. De Freitas, (2012, Oct. 30). *GA_framework (MATLAB)*. [online]. Available: <https://sourceforge.net/projects/gatoolbox/files/>. File: *GA_Framework.zip*.
- [13] S. M. Bruschi and R. F. D. Mello, "Algoritmo genético para a tomada de decisão de particionamento de processos lógicos em simulação distribuída," *IEEE LATIN AMERICA TRANSACTIONS*. Vol. 6, no. 1, pp. 97-105. March. 2008.
- [14] A. E. Eiben and J. E. Smith, *Introduction to Evolutionary Computing*. 2nd ed., Natural Computing Series, G. Rozenberg, Th. Bäck, A. E. Eiben, J. N. Kok and H. P. Spaink, Series Editors. New York, NY, USA: Springer-Verlag, 2015, pp. 1-294, doi: 10.1007/978-3-662-44874-8.
- [15] M. Y. Orong, A. M. Sison and R. P. Medina, "A new crossover mechanism for genetic algorithm with rank-based selection method," in *ICBIR*, Bangkok, TH, 2018, pp. 83-88.
- [16] W. Azzolini, Jr, F. G. P. Azzolini, *Layout Stage 1*, [online]. Available: <https://github.com/fredgpa/LayoutStage1>



Walther Azzolini Junior nasceu em Araraquara, SP, BR em 1964. Ele concluiu o curso de Engenharia de Produção Mecânica pela Universidade Metodista de Piracicaba, SP, em 1991, o mestrado em Engenharia Mecânica pela Universidade Estadual de Campinas (UNICAMP), SP, em 1996 e o doutorado em Engenharia Mecânica pela Escola de Engenharia de São Carlos (EESC) da Universidade de São Paulo (USP), em 2004. Ele é professor da Escola de Engenharia de São Carlos da Universidade de São Paulo (USP) e efetivo. Tem experiência na área de Engenharia de

Produção e Engenharia Mecânica, com ênfase em Simulação Computacional de Sistemas de Manufatura, atuando principalmente nos seguintes temas: Planejamento e Controle da Produção, Pesquisa Operacional e Simulação Computacional. De 2015 a 2019, ele foi Presidente da Comissão de Apoio à Projetos Institucionais da EESC (Portaria 053/2015). Autor de oito livros e oito capítulos de livros. Ele é Líder do Grupo de Pesquisa: Grupo de Estudos e Pesquisas de Modelagem e Simulação de Sistemas Industriais (Infraestrutura: Núcleo de Ensino e Pesquisa em Automação e Simulação (NEPAS/EESC/USP)) e colaborador do Centro de Matemática e Estatística Aplicadas à Indústria (CEPID - CeMEAI/ICMC/USP).



Frederico Gomes Pires Azzolini nasceu em Araraquara, SP, BR em 1996. Ele é estudante de graduação do curso de Ciências da Computação, Universidade Estadual Paulista Júlio de Mesquita Filho, UNESP, Bauru, SP, BR. Frederico Gomes Pires Azzolini colabora com o Grupo de Estudos e Pesquisas de Modelagem e Simulação de Sistemas Industriais como aluno de iniciação científica.