



FACULDADE SENAI DE TECNOLOGIA MECATRÔNICA

INTELIGÊNCIA ARTIFICIAL APLICADA NA MELHORIA DO PROCESSO DE CUBAGEM DE CARGAS: UMA ABORDAGEM EMPREGANDO ALGORITMOS GENÉTICOS

ARTIFICIAL INTELLIGENCE APPLIED IN THE IMPROVEMENT OF THE LOAD CUBING PROCESS: AN APPROACH EMPLOYING GENETIC ALGORITHMS

Eduardo Bueno da Silva ^{1 i}

João Gomes Zardo ^{2 ii}

Daniel Otavio Tambasco Bruno ^{3 iii}

RESUMO

O objetivo deste trabalho é desenvolver um sistema computacional empregando conceitos matemáticos, técnicas de aprendizado de máquina tais como algoritmo genético, linguagem Prolog, servidores WEB e projeção 3D com JavaScript, HTML5 e CSS, com o objetivo de otimizar a utilização de espaço por parte dos sólidos (caixas) dentro de um invólucro, tais como o baú de um caminhão, possibilitando diminuir perdas por mal planejamento de posicionamento e por deslizamento das cargas. Este sistema possui interface gráfica amigável, de fácil utilização, é independente de plataforma computacional, pois é executado em um servidor Web e pode ser amplamente utilizado em diferentes tamanhos de armazéns, desde um contêiner de um navio ao porta-malas de um carro. Através dos testes realizados, até o presente momento o sistema tem se mostrado eficiente no posicionamento de cargas retangulares.

ABSTRACT

The goal of this paper is develop a computational system employing mathematical concepts, machine learning techniques such as genetic algorithm, Prolog Language, WEB servers and 3D projection with JavaScript, HTML5 and CSS, in order to optimize space utilization by solids (boxes) inside a casing, such as a truck trunk, making it possible to reduce losses due to poor positioning planning and sliding loads. This system has a user-friendly, user-friendly graphical interface, it is platform independent as it runs on a Web server and can be widely used in different warehouse sizes, from a ship's container to a car's trunk. Through the tests performed, until the present moment the system has been efficient in the positioning of rectangular loads.

¹ Tecnólogo em Mecatrônica Industrial. E-mail: edubs_bueno@hotmail.com

² Tecnólogo em Mecatrônica Industrial. E-mail: joaogzardo@gmail.com

³ Mestre em Engenharia da Informação e Professor da Faculdade SENAI de Tecnologia Mecatrônica Industrial. E-mail: daniel.bruno@sp.senai.br

1 INTRODUÇÃO

Para Fleury; Wanke; Figueiredo (2000), O transporte é o principal componente de um sistema logístico. Sua importância pode ser medida através de pelo menos três indicadores financeiros: custos, faturamento e lucro. O transporte representa, em média, 60% dos custos logísticos, 3,5% do faturamento, e em alguns casos, mais que o dobro do lucro. Além disso, o transporte tem um papel fundamental na qualidade dos serviços logísticos, pois impacta diretamente o tempo de entrega, a confiabilidade e a segurança dos produtos.

Segundo Alves (1997), a atividade do transporte é definida como aquela que assegura o fluxo físico dos produtos entre as empresas, quando ocorre transferência de propriedade do produto, através de uma transação comercial.

Decker e Trisch (2004) ressaltam a importância do setor de transportes de cargas na economia do País e afirmam que, "por mais que ocorram avanços tecnológicos, o ato de levar um produto de um lugar ao outro é, e permanecerá sendo por muito tempo, a única forma de dispor um bem em um local diferente do que é produzido".

Um problema recorrente ao armazenamento das cargas em baús de caminhões, observado empiricamente pelos autores é em muitos casos, a transportadora ter que realizar várias viagens para transportar uma determinada carga. Outro fator observado foi o de deslocamento interno das cargas, ocasionando a desorganização e eventualmente a quebra de alguns itens armazenados. O problema é mais frequente quando é realizado o transporte de líquidos engarrafados, como: refrigerantes, água e bebidas em geral. Normalmente esses produtos são embalados por engarrafados, sendo que cada fardo é produzido de uma maneira específica, isso implica que cada ele receba dimensões de largura, altura e comprimento diferentes uns dos outros.

Normalmente os responsáveis pelo armazenamento das cargas no caminhão tem que avaliar empiricamente a disposição destas cargas, encaixando o maior número possível de fardos dentro do baú, e com a mesma frequência não era possível armazená-los da melhor maneira possível, acarretando assim um número maior de viagens, com um grande espaço inutilizado no caminhão.

Nota-se que a partir desta situação é possível verificar um problema que empresas de logística possuem no transporte de suas cargas. A fim de verificar a viabilidade deste projeto e seguir estudando este problema de pesquisa, os autores realizaram uma pesquisa de campo junto a uma empresa de logística, situada na cidade de São Bernardo do Campo (SP), um grande polo industrial e logístico do Brasil. Em contato com especialistas desta empresa, foi constatado a necessidade de uma investigação mais aprofundada e o desenvolvimento de um sistema computacional baseado em inteligência artificial para ajudar a solucionar estes problemas.

De acordo com especialistas entrevistados, muitas empresas de logística possuem deficiência com relação ao armazenamento quando considerado grande quantidade de volumes das amostras a serem armazenados e o volume do armazém, ou seja, em muitos casos, as empresas não possuem uma solução eficaz para tal problema, em muitos casos efetuando tais cálculos manualmente.

Para buscar uma solução para tais problemas, os autores iniciaram suas pesquisas buscando referências bibliográficas sobre cubagem de cargas. Segundo Nascimento; Gallon e Beuren (2009), a cubagem da carga pode ser definida como a relação entre o tamanho, o peso de um produto e o espaço que ele ocupará dentro do veículo. Essa relação é obtida por meio de um cálculo matemático utilizando o fator de cubagem.

O objetivo geral deste estudo é desenvolver um sistema computacional baseado em técnicas de inteligência artificial que permita que o usuário possa inserir as medidas dos volumes de uma carga e do local a ser carregado. O algoritmo proposto irá realizar a busca pela melhor combinação de disposição possível destas cargas em relação ao local de transporte e ou de armazenamento.

Os objetivos específicos são:

- a) Investigar o comportamento dos algoritmos genéticos e linguagem Prolog para busca da melhor solução possível;
- b) Investigar e implantar servidores WEB para interpretar as solicitações do usuário e comunicar com o algoritmo de inteligência artificial;
- c) Desenvolver em Linguagem HTML 5, CSS e Java Script interface de integração entre o usuário, o servidor Web e algoritmo de inteligência artificial;
- d) Apresentar de forma gráfica a melhor combinação obtida.

Este trabalho está organizado da seguinte forma, esta seção apresentou a problemática de pesquisa, os objetivos gerais e específicos desta pesquisa. Na seção 2, será apresentada a fundamentação teórica e o estado da arte. A Seção 3 apresenta a metodologia desenvolvida. Já a Seção 4, apresenta os resultados parciais e discussões dos experimentos realizados. Em seguida, são apresentadas as conclusões parciais obtidas e as referências consultadas.

2 FUNDAMENTAÇÃO TEÓRICA

Para discorrer sobre as técnicas utilizadas para geração de resultados e processamento das informações coletadas é necessário apresentar os conceitos teóricos sobre as técnicas computacionais e linguagens utilizadas que basicamente são: Prolog, Java Script, inteligência artificial, algoritmos evolucionários, uso da biblioteca Tree.js. Também serão apresentados trabalhos relacionados com esta pesquisa.

2.1 Definição matemática do problema

Dado um espaço amostral E tal que exista $N \in N$ elementos de largura $w_1, w_2, w_3, \dots, w_n \in R$, altura $h_1, h_2, h_3, \dots, h_n \in R$, comprimento $l_1, l_2, l_3, \dots, l_n \in R$, e volume $v_1, v_2, v_3, \dots, v_n \in R$, e seus objetos sejam paralelepípedos retangulares sólidos. Considerando também o espaço de armazenamento um paralelepípedo retangular sólido de dimensões, e volume $V \in R$ respectivamente referente a largura, comprimento e altura, determinar $k \in N$ tal que $V - (v_1 + v_2 + v_3 + \dots + v_k)$ onde k é máximo.

2.2 Inteligência Artificial (IA)

Segundo Norvig e Russell (2010), ao longo da história humana, tentamos definir o que é inteligência humana e artificial, só que como essa é uma área do conhecimento que não permite a realização de testes físicos, as definições tomam carácter filosófico, e por terem tal carácter normalmente elas seguem a tomam características essenciais da linha de pesquisa do pesquisador que as está estudando no caso temos quatro principais:

1. Pensamento Humano: A automação de atividades que nós associamos com pensamento humano atividades capazes de tomar decisões resolver problemas, e ainda mesmo aprender (BELLMAN (1978) *apud* NORVIG; RUSSELL (2010));
2. Pensamento racional: O estudo que possibilita os computadores serem perceptivos, capazes de pensamento racional e ação. (WINSTON (1972) *apud* NORVIG; RUSSELL (2010));
3. Ação Humana: A arte de criar máquinas que realizam funções que requerem por sua vez inteligência, essa uma vez já realizada pela ação humana. (KURZWEIL (1990));
4. Ação Racional: Inteligência artificial é o estudo que cria agentes inteligentes. (POOLE; MACKWORTH; GOEBEL (1998)).

Segundo Lima (2014) e Coley (1999), ao longo dos anos inevitavelmente as áreas e linhas de pesquisas se ajudam e formam um pensamento unificado, ou divergem para um mesmo caminho ou ramo científico, no caso da IA nas últimas décadas obtivemos avanços significativos, com ajuda da biologia, a teoria da evolução das espécies, obtivemos os algoritmos evolucionários, e também da biologia mas mais especificamente da neurociência, obtivemos técnicas como as redes neurais, para entender esses avanços é primeiro necessário entender o que cada carácter filosófico quer dizer, e saber com precisão sobre aquilo que estamos discutindo para isso é apresentado a seguir a definição de software:

Para Carvalho et al. (2011), um programa de computador ou *software* é uma máquina abstrata que produz e manipula entidades abstratas (dados) tornando-se a descrição da máquina, passando a ser a própria máquina quando em execução, sendo o meio físico de execução o próprio computador.

2.2.1 Ação Humana

De acordo com Norvig e Russell (2010) e Kurzweil (1999), o teste de Turing, proposto por Alan Turing em 1950, foi criado para teste das definições de inteligência artificial. Consistindo de um computador e um “júri” humano que tem a função de propor algumas perguntas ao computador e a algumas pessoas e depois avaliar todas as respostas juntas respostas, dizendo se elas são respostas de um computador ou de uma pessoa, o computador passa no teste caso o “júri” não souber se responder se a resposta avaliada é de um computador ou de uma pessoa, mas para isso o computador necessita de algumas capacidades:

1. Processamento de Linguagem natural: ser capaz de se comunicar na linguagem natural;
2. Representação de Conhecimento: guardar o conhecimento de alguma maneira;
3. *Automated Reasoning*: usar o conhecimento armazenado para constituir novas respostas cada vez mais inteligentes;
4. Aprendizado de máquina: se adaptar a novas situações e detectar padrões.

2.2.2 Pensamento Humano

Segundo Norvig e Russell (2010) para dizer que um computador pensa como um ser humano primeiramente precisamos definir como um ser humano pensa, para isso alguns métodos científicos foram criados.

- a) Introspecção do pensamento, entender como o pensamento humano funciona;

- b) Experimentos de psicológicos, observar uma pessoa agindo e pensando;
- c) Observar o cérebro humano em ação.

Observar o cérebro humano em ação é importante para entender de forma analítica como processamos informação e caso as entradas e saídas de um programa de computador se comportem de maneira semelhante ao cérebro humano, isso evidencia que o programa poderia também operar em seres humanos, claramente que dentro das condições cabíveis, visto que o fisicamente ambos são diferentes um do outro.

2.2.3 Pensamento Racional

Segundo Norvig e Russell (2010) o filósofo grego Aristóteles foi o primeiro a se esforçar em tentar explicar o “Pensar correto”, baseado no uso da lógica. Seus silogismos forneceram padrões de estrutura de argumentação que sempre se deduziam em uma conclusão correta, quando dadas as premissas de valor lógico verdadeiro. Esses silogismos foram supostos como principal meio de pensamento humano, seus estudos iniciaram o campo de estudos da lógica.

Em 1965, foi desenvolvido um programa que em princípio poderia resolver qualquer problema baseado em lógica de primeiro grau. Chamado de *Logicist* com a esperança de criar um sistema inteligente.

Existem dois obstáculos principais primeiramente não é fácil transformar um conhecimento informal em termos formais que a lógica de primeiro grau requer para dedução de suas conclusões, segundo existe uma diferença entre resolver um problema em princípio e resolvê-lo na prática.

2.2.4 Ação Racional

Segundo Norvig e Russell (2010) um agente é algo que faz alguma coisa, mas um agente computacional deve fazer mais do que simplesmente agir, eles devem operar de forma autônoma, adaptando-se para mudanças, criando e solucionando objetivos. Um agente racional é aquele agente que a ação tenta atingir a melhor solução, para um objetivo ou problema.

Todas as habilidades requeridas para o teste de Turing admitem a ação racional de um agente. Agir racionalmente depende de uma base de dados e pensamento crítico, sobre o processamento deles.

2.2.5 Definindo IA

Segundo Norvig e Russell (2010) com o caráter filosófico que a inteligência artificial tem em suas discussões, não é possível definir com precisão o que ela é mas é possível abstrair a ideia do que um programa de computador inteligente deve fazer, baseado nos tópicos acima e na definição de software também apresentada podemos concluir que o objetivo ao tornar um computador inteligente é fazer a máquina abstrata processe informações de maneira que ela seja capaz de tomar decisões de forma automática e racional, buscando sempre a melhor solução, mesmo quando o problema é não determinístico e prevê mais de uma resposta correta.

2.3 Algoritmos Genéticos

Segundo Coley (1999) para a solução do problema apresentado foi escolhido a técnica dos algoritmos evolucionários, para isso antes de explicar como esta técnica foi aplicada solução do problema é necessário primeiro defini-la e explicar como ela funciona.

Genetic algorithms (GA) – Algoritmos Genéticos (AG) iniciam com aquilo que é chamado de população que são usualmente randômicos e se espalham pelo espaço de busca de solução do algoritmo. Um algoritmo típico usa três operações seleção, crossover e mutação.

A população são os dados iniciais do programa normalmente gerados de forma randômica e que sofrem com as três operações acima apresentadas.

Seleção: Testa a população classificando seus membros em fracos e fortes, nivelando-os de acordo com o problema ser solucionado assim como na teoria da evolução de Darwin onde sistemas biológicos selecionam indivíduos para próximas gerações, deixando apenas aqueles que se adaptam bem ao meio.

Crossover: Método similar ao usado pelo DNA humano quando em reprodução, um par de indivíduos é selecionado e randomicamente a característica de uma passa para o outro, essa definição ficará mais clara ao estudarmos um exemplo.

Mutação: Depois de aplicada a seleção e mutação na população inicial de dados do programa, será formada uma nova população, nessa nova população é aplicada uma mudança ou alteração em uma determinada quantidade de membros, essa modificação é determinada pelo próprio problema, a mutação tem o objetivo principal de manter a diversidade entre os membros da população.

Exemplo:

Considere que queremos otimizar a função $f(x) = x^2$, tal que $f: N \rightarrow N$ e $0 \leq X \leq 4095$

Utilizaremos a seguinte estratégia:

1. Formar uma população de números binários de comprimento 8 com 20 membros.
2. Decodificar cada binário em um valor inteiro.
3. Testar esses números como solução do problema.
4. Selecionar a melhor metade da população e passá-la para próxima geração.
5. Selecionar um par de binários aleatoriamente, para aplicar o crossover entre eles, trocando os 4 bits mais significativos entre ambos e mantendo os 4 bits menos significativos.
6. Aplicar mutação entre os membros restantes da população, que será inverter o bit menos significativo (Antes: 10001, Depois: 10000).
7. Manter os membros que sofreram mutação e crossover juntos e passá-los para próxima geração
8. Retornar para o passo 2 e repetir uma quantidade determinada de vezes.

2.4 PROLOG - *Programming in Logic*

De acordo com Bratko (1986) Prolog é uma linguagem programação que se enquadra no paradigma lógico de forma simbólica e não numérica, é uma ótima opção para problemas que envolve objetos e relações entre estes objetos, o uso do Prolog puro foi originalmente restrito em provas de teorema da resolução com cláusulas de Horn no formato $H: -B_1, \dots, B_n$.

O nome Prolog vem de “*Programmation en logique*”, foi criada em 1972 por Philippe Roussel e Alain Colmerauer.

O Prolog é uma linguagem de programação daquelas muito diferentes encontradas no mercado é mais voltada a sistemas inteligentes e processamento de linguagem natural, como seu paradigma é lógico a maneira de apresentação do problema ao computador torna-se descritivo em vez de imperativo como acontece com linguagens comuns como C++ ou Java.

2.4.1 Tipos de dados

Para Bratko (1986), os tipos de dados em PROLOG são organizados da seguinte forma:

1. **Átomos:** Sequência de Letras Números ou *Underscore*, sempre iniciado com letra minúscula, caso a sequência não seja alfanumérica é preciso usar aspas simples:
2. **Número:** Um número é uma sequência de dígitos (0 a 9) permitindo também sinais negativo e positivo.
3. **Variáveis:** Tem a mesma estrutura de um átomo mas deve ser iniciada com letra maiúscula, em Prolog variáveis se comportam de maneira diferente por causa do estilo de processamento do seu motor de inferência baseado em uma *Heap*, o comportamento dela está mais para incógnita do que para variável, não há declaração de tipo pois a memória é gerenciada pelo próprio Prolog em tempo de processamento.
4. **Listas:** Uma lista não é um tipo de dado a parte, mas sim definida por uma construção recursiva.
 - a) O átomo [] é uma lista vazia sem elementos;
 - b) Se T é uma lista e H é um elemento, então o termo ‘.(H,T) é uma Lista. O primeiro elemento que é H é chamado de head e T é o Tail, a representação de lista é [1,2,3,4,...],
- 5 **Predicados:** São compostos de argumento e um identificador que é um átomo e um argumento deve vir entre parênteses exemplo:
 - a) %box(ID, width, Length).
 - b) box(45, [3cm, 5cm]).

Existem outros tipos de dados disponíveis, mas esses são os mais importantes para as implementações, como métodos de repetição o Prolog conta com *Backtracking* (É um algoritmo genérico que busca, por força bruta, soluções possíveis para problemas computacionais) e recursão, que é o ato de uma função de programação chamar a si própria.

2.3 Aplicação Web

Segundo Duckett (2016), em computação, aplicação *web* designa, de forma geral, sistemas de informática projetados para utilização através de um navegador, através da internet ou aplicativos desenvolvidos utilizando tecnologias web HTML, CSS e JavaScript. Pode ser executado a partir de um Servidor HTTP (*Web Host*) ou localmente, no dispositivo do usuário.

2.3.1 HTML e CSS

Conforme Duckett (2016) HTML (*HyperText Markup Language* - Linguagem de Marcação de Hipertexto) é uma linguagem de marcação utilizada na construção de páginas Web. Juntamente com o CSS (*Cascading Style Sheets* – Folhas de Estilo em Cascata) é um mecanismo para adicionar estilo (cores, fontes, espaçamento etc.) a um documento web, podendo assim alterar a aparência e modificar os documentos HTML vinculados. Através da junção de ambos (HTML e CSS) é possível criar uma página própria com estilização e funcionamento a escolha do programador, fazendo assim completamente modificável e reprogramável.

2.3.2 JavaScript

Para Duckett (2016) JavaScript, frequentemente abreviado como JS, é uma linguagem de programação interpretada de alto nível. Juntamente com HTML e CSS, o *JavaScript* é uma das três principais tecnologias da WWW (*World Wide Web*). *JavaScript* permite páginas da Web interativas e, portanto, é uma parte essencial dos aplicativos da web. A grande maioria dos sites usa, e todos os principais navegadores têm um mecanismo JavaScript dedicado para executá-lo.

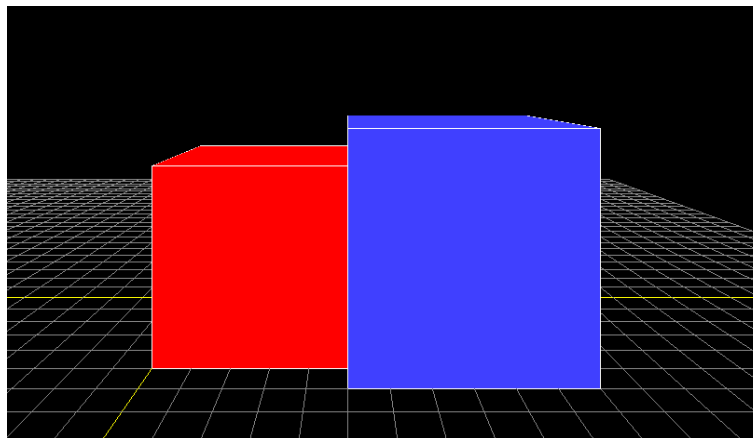
Com o JS é possível utilizar bibliotecas criadas por usuários com diversas funcionalidades utilizando apenas um servidor local e uma página web. Tornando-o assim extremamente dinâmico com diversas aplicações diferentes.

2.3.3 Three.js

A partir da necessidade de apresentar uma resposta com uma imagem 3D para melhor entendimento do posicionamento dos sólidos e um rápido processamento de dados, foi escolhida a biblioteca *Three.js*⁴. Esta biblioteca permite a criação de sólidos, alterar as dimensões, a visualização e cores dos itens (Duckett, 2016).

Após otimizar algumas predefinições, como a grade no eixo X com dimensões de 1 unidade por 1 unidade, e duas caixas, uma com dimensões de 5 unidades de altura 5 unidades de largura e 5 unidades e a outra com 6 unidades de altura 6 unidades de comprimento e 6 unidades de largura criou-se um programa que será utilizado como base para testes. A figura 1 ilustra uma tela de saída do sistema, com o posicionamento de duas caixas.

Figura 1 – Tela de saída da solução



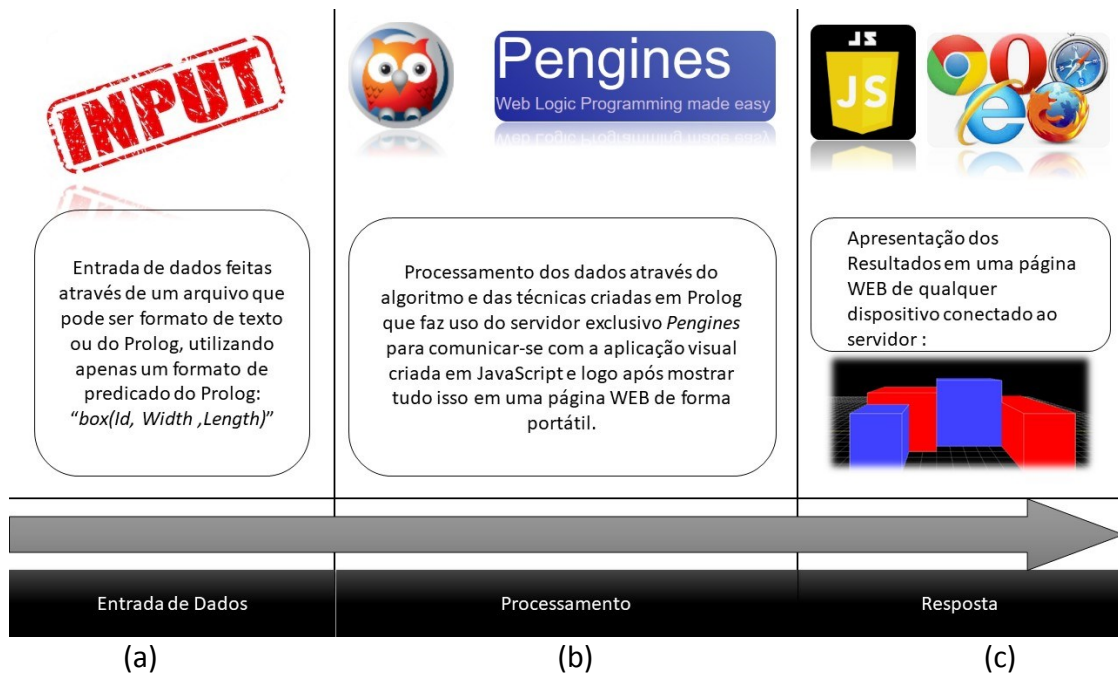
Fonte: Autores (2019)

⁴ Todas as referências da biblioteca *tree.js* foram retiradas do site: <https://threejsfundamentals.org/>

3 MATERIAIS E MÉTODOS

O diagrama da figura 2, apresenta a visão geral do sistema implementado, onde em figura 2 (a), o usuário irá entrar com os dados da carga; Já em figura 2 (b), o servidor Web Pengines irá processar as informações e utilizar a API (*Application Programming Interface* - Interface de Programação de Aplicativos) do SWI Prolog. Na Figura 2 (c) é apresentada a saída, com o posicionamento das caixas no *browser* do usuário.

Figura 2 - Diagrama apresentando a visão geral do sistema



Fonte: Autores (2019)

3.1 Materiais

Para o desenvolvimento da metodologia proposta, foi utilizado um microcomputador padrão IBM-PC equipado com microprocessador Intel Core i5 M480 com 2,67 GHz de frequência, 8 GB de memória RAM e placa de vídeo Nvidia Geforce GT 640. Os softwares utilizados foram: Sistema Operacional Windows 10 Professional, SWI-Prolog *development release* versão 8.1.14, Web Server Pengines versão 0.9.3 e navegador Google Chrome Versão 77.0.3865.90 com suporte a HTML 5, CSS e JavaScript.

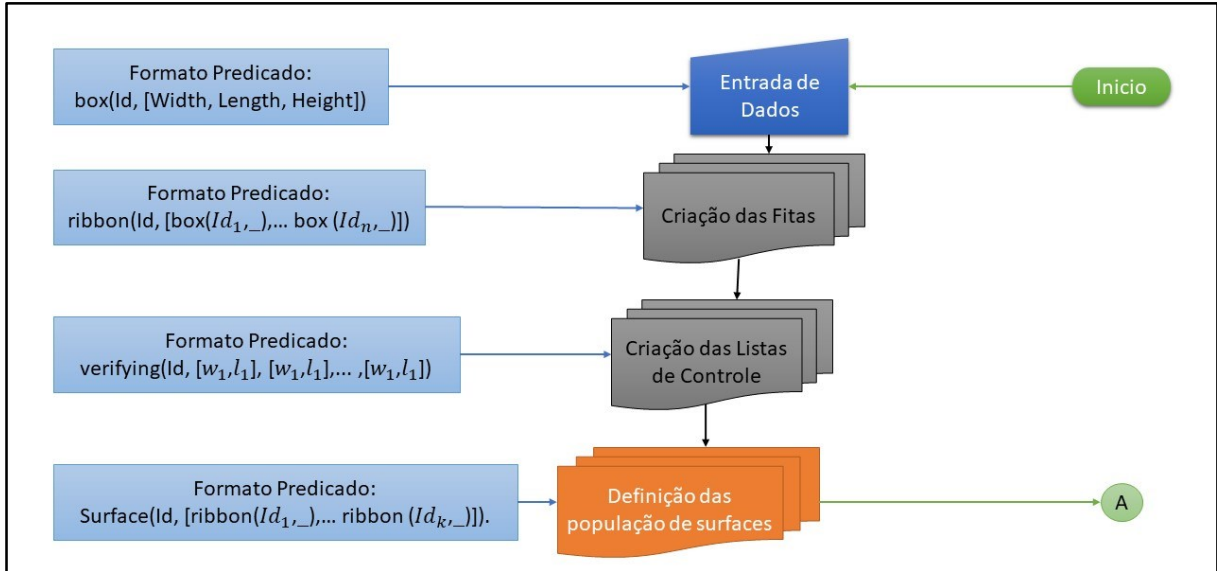
3.2 Métodos

O Diagrama da Figura 3, apresenta o funcionamento detalhado da primeira parte do algoritmo, onde o usuário deverá informar as dimensões das caixas e as dimensões do espaço de armazenamento. Nesta figura é possível visualizar como a população é criada. O método utilizado é uma transformação de dados, partindo de volumes separados em predicados *box* separados para o predicado *surface* que é a própria população.

As figuras retangulares em azul representam o tipo de dado que o Prolog receberá e também produzirá, as partes em cinza correspondem a como o algoritmo trabalha, e as partes

em laranja são as partes essenciais sendo que a primeira denota o início do algoritmo genético já a segunda parte em laranja marca o fim de uma geração.

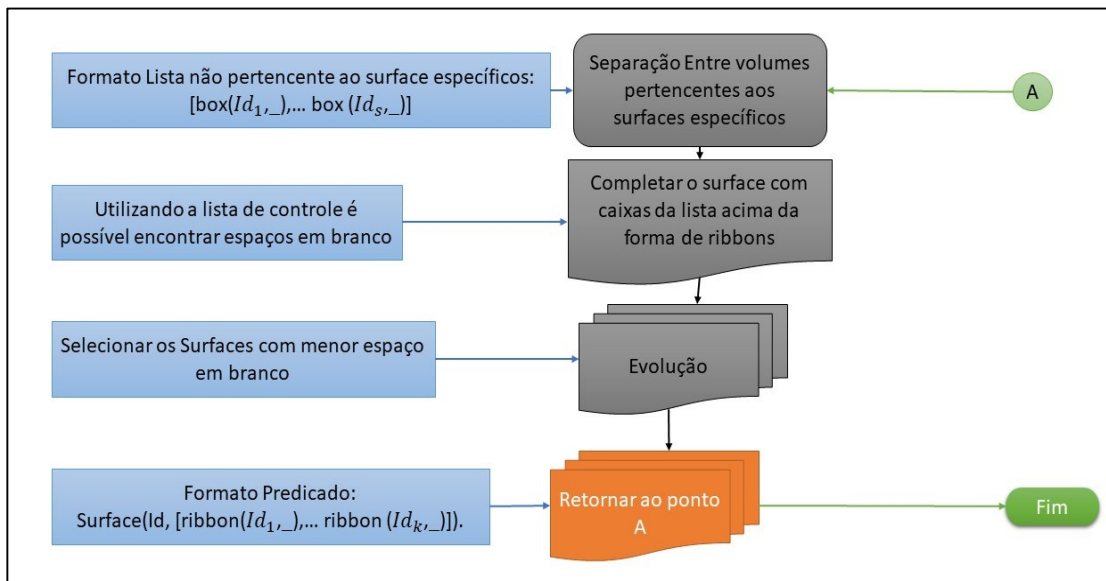
Figura 3 - Diagrama apresentando o detalhamento do algoritmo proposto – Parte 1



Fonte: Autores (2019)

Na figura 4, o algoritmo genético entra em ação realizando o *crossover*, a mutação (o encaixe de volumes que estão fora do próprio *surface*) e a seleção, escolhendo os *surfaces* que possuem menor espaço em branco.

Figura 4 - Diagrama apresentando o detalhamento do algoritmo proposto – Parte 2



Fonte: Autores (2019)

4 RESULTADOS

A fim de avaliar a metodologia proposta, foram realizados 12 experimentos, com caixas e depósito de tamanho variável, conforme apresentados na Tabela 1.

Tabela 1 - Detalhamento dos experimentos realizados com o algoritmo proposto

Testes	Espaço Amostral	Fitas	Superfícies	Gerações	Tempo (ms)
1	100	64	20	15	~100
2	100	63	17	19	~100
3	100	57	22	23	~100
4	100	62	19	17	~100
5	50	26	10	7	~50
6	50	28	12	6	~50
7	50	24	9	7	~50
8	50	30	13	5	~50
9	1000	657	285	153	~750
10	1000	648	264	175	~750
11	1000	623	278	198	~750
12	1000	689	215	140	~750

Fonte: Autores (2019)

Para avaliação do algoritmo proposto foram aplicados testes com três espaços amostrais, com 50, 100 e 1000 volumes, sendo que o último serviria para um teste de exaustão.

Até aproximadamente 100 volumes o tempo de processamento é diretamente proporcional a quantidade de volumes. Em milissegundos totalizando para 50 volumes, 50ms e para 100 volumes 100ms.

No teste de exaustão realizado, observa-se que o tempo de execução teve uma queda de 25% aproximadamente do esperado. O que significa que, para volumes muito grandes o algoritmo tem uma performance ligeiramente melhor.

Nota-se no experimento realizado com 100 volumes que a quantidade de fitas criadas de forma aleatória pelo algoritmo, atingem uma faixa de 60% da quantidade de volumes no espaço amostral. A relação se mantém aparentemente constante para 50 volumes e no teste de exaustão.

Obviamente, que esses valores são aproximados, pois para melhor definição, eles têm que receber provas matemáticas para dedução da relação entre fita, superfície, espaço amostral e largura do espaço de encaixe, mas como o algoritmo ainda está em fase de testes no momento não é possível deduzir matematicamente uma fórmula determinística para a quantidade de fitas, superfícies e gerações do algoritmo genético.

Ao analisar as superfícies que são as faces já organizadas ou em processo de organização, podemos observar que elas correspondem aproximadamente de 15% a 20% do espaço amostral utilizado como entrada, o padrão se mantém para o teste de exaustão. Com relação ao número de gerações do algoritmo genético, podemos observar que elas seguem aproximadamente a quantidade de superfícies sendo aparentemente proporcionais.

4 CONCLUSÃO

A aplicação de *softwares* dentro da área da logística e armazenamento é de grande importância, auxiliando a tomada de decisão de forma eficiente, otimizando o processo de armazenamento com mais eficiência.

Uma característica positiva deste trabalho é seu uso em uma variedade de aplicações diferentes, não se limitando somente a grandes volumes de cargas, mas também para qualquer aplicação que necessite de otimização de espaço e mapeamento de objetos.

Existirão situações em que utilizar a solução proposta não seja viável pois nesta área existem diversos tipos de variáveis, dentre elas e a mais importante que é a limitação física que o espaço possui, como por exemplo a capacidade de peso suportada, ou ainda a área de entrega, se a carga é de risco ou não, dentre outras situações. Neste sentido é recomendado um estudo aprofundado da relação custo, volume e entrega para cada aplicação específica.

Como conclusão pode-se afirmar que o algoritmo desenvolvido trabalha melhor com espaços amostrais maiores pois existe uma diferença significativa entre a quantidade de população quando aplicado o teste de exaustão e aplicações corriqueiras.

Populações maiores do algoritmo genético permitem um desenvolvimento melhor do algoritmo evolucionário, pois ele ganha maior liberdade para evoluir e executar mutações nas populações.

Nota-se que a metodologia desenvolvida é bastante promissora. O estudo das técnicas de IA, como o Algoritmo Genético aplicado neste trabalho pode ser empregado nas mais diversas situações, como por exemplo na criação de horários escolares (NETTO, 2011), dentre muitas outras aplicações.

O estudo e o desenvolvimento deste trabalho contribuem para a implantação de um dos pilares da Indústria 4.0, a Inteligência Artificial. Como trabalhos futuros, pretende-se aprimorar o algoritmo genético, incluindo novas variáveis reais a ele, como o peso máximo do invólucro, ordem de descarga e a fragilidade dos volumes. Também pretende-se avaliar o servidor Web Pengines em um sistema operacional Linux.

REFERÊNCIAS

ALVES, Maria Rita Pontes Assumpção. **Logística agroindustrial**. In: GEPAI - Grupo de Estudos e Pesquisas Agroindustriais. BATALHA, Mario Otavio (Org.). *Gestão agroindustrial*. São Paulo: Atlas, 1997. p. 139-214.

COLEY, David. **An introduction to Genetic Algorithms for scientists and Engineers**. Singapore: World Scientific Publishing Co., 1999.

BRATKO, Ivan. **Prolog programming for artificial intelligence**. England: Addison-Wesley Publishing Company, 1986.

CARVALHO, André et al. **Inteligência artificial: uma abordagem de aprendizado de máquina**. São Paulo: LTC, 2011.

DECKER, Sérgio Renato Ferreira; TRISCH, Vinícius Silva. Custos e finanças no transporte de cargas. In: CONGRESSO BRASILEIRO DE CUSTOS, 11., 2004, Porto Seguro BA. **Anais...** Porto Seguro: ABC, 2004. CD-ROM.

DUCKETT, Jon. **JavaScript e JQuery: desenvolvimento de interfaces web interativas**. Nova York: Alta Books, 2016. 460 p.

ELIODÓRIO, Keyv Pontes; CARMONA, Ricardo Alexandre; BRUNO, Daniel Otávio Tambasco. Machine learning: aplicações na indústria química. **Revista Brasileira de Mecatrônica**, São Caetano do Sul, v. 1, n. 4, p.34-49, abr. 2019. Disponível em: <http://revistabrmecatronica.com.br/ojs/index.php/revistabrmecatronica/article/view/37/38>. Acesso em: 06 out. 2019.

FLEURY, P.F.; WANKE, P.; FIGUEIREDO, K. F (org.) **Logística empresarial: a perspectiva brasileira**, Ed. Atlas, São Paulo, 2002.

JAVASCRIPT 3D LIBRARY. Página inicial. [c2018]. Disponível em: <https://threejsfundamentals.org/>. Acesso em: 07 de set. 2019.

KURZWEIL, R. **The age of Intelligent Machines**. MIT Press, 1990.

LIMA, Isaías. **Inteligência artificial**. São Paulo: Campus, 2014.

NASCIMENTO, Sabrina do; GALLON, Alessandra Vasconcelos; BEUREN, Ilse Maria. Formação de Preços em Empresa de Transporte Rodoviário de Cargas. **Pensar Contábil**, Rio de Janeiro, v. 11, n. 46, p.20-26, nov. 2009.

NETTO, Francisco Sobreira. **Aplicação de algoritmos genéticos na elaboração de horários escolares: o estudo do caso Inteliway**. Rio de Janeiro: Enanpad, 2011. Disponível em: <http://www.anpad.org.br/admin/pdf/ADI1563.pdf>. Acesso em: 7 out. 2019.

NORVIG, Peter; RUSSELL, Stuart J. **Artificial intelligence: a modern approach**. 3. ed. Boston: Prentice Hall, 2010.

POOLE, D.; MACKWORTH, A. K., GOEBEL, R. **Computational intelligence: a logical approach**. Oxford University Press, 1998.

AGRADECIMENTOS

Agradecemos a todos aqueles que contribuíram na realização deste trabalho especialmente ao Professor Orientador Daniel Otávio Tambasco Bruno, aos Professores Paulo Sebastião Ladivez, Professor Douglas Mazza e a empresa Júlio Simões Logística por nos receber, dar sugestões de melhorias e acreditar que este projeto é promissor.

Sobre os autores:

ⁱ EDUARDO BUENO DA SILVA



Tecnólogo em Mecatrônica Industrial pela Faculdade SENAI Armando de Arruda Pereira (2017-2019) e cursando Licenciatura em Matemática pelo Instituto Federal de São Paulo, possui experiência na área de automação residencial.

ⁱⁱ JOÃO GABRIEL GOMES ZARDO



Tecnólogo em Mecatrônica Industrial pela Faculdade SENAI Armando de Arruda Pereira (2017 - 2019).

ⁱⁱⁱ DANIEL OTAVIO TAMBASCO BRUNO



Doutorando e Mestre em Engenharia da Informação pela Universidade Federal do ABC (2013). Especialista em Análise, desenvolvimento de Sistemas e Banco de Dados pela Universidade de Ribeirão Preto (2007), Especialista em Educação a Distância pela Universidade Paulista (2012). Bacharel em Análise de Sistemas pela Universidade Paulista (2003). Atualmente é Técnico em Manufatura Digital e Professor da Faculdade SENAI de Tecnologia Mecatrônica Industrial. Tem experiência na área de Inteligência Artificial, Internet das coisas, Gestão de Tecnologia da Informação e Desenvolvimento de Sistemas de Informação.