



UNIVERSIDADE FEDERAL DO CEARÁ
CAMPUS DE RUSSAS
CURSO DE GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

JARDEL TORRES CHAGAS

**UMA METAHEURÍSTICA ADAPTADA AO PROBLEMA DE DETERMINAÇÃO DO
CAMINHO DE CORTE DE UM LEIAUTE COM TEMPO MÍNIMO**

RUSSAS

2022

JARDEL TORRES CHAGAS

UMA METAHEURÍSTICA ADAPTADA AO PROBLEMA DE DETERMINAÇÃO DO
CAMINHO DE CORTE DE UM LEIAUTE COM TEMPO MÍNIMO

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da computação do CAMPUS DE RUSSAS da Universidade Federal do Ceará, como requisito parcial à obtenção do grau de bacharel em Ciência da computação.

Orientador: Prof. Dr. Bonfim Amaro Júnior

RUSSAS

2022

Dados Internacionais de Catalogação na Publicação
Universidade Federal do Ceará
Sistema de Bibliotecas
Gerada automaticamente pelo módulo Catalog, mediante os dados fornecidos pelo(a) autor(a)

C424m Chagas, Jardel Torres.
Uma Metaheurística Adaptada Ao Problema De Determinação Do Caminho De Corte De Um Leiaute Com Tempo Mínimo / Jardel Torres Chagas. – 2022.
29 f. : il. color.

Trabalho de Conclusão de Curso (graduação) – Universidade Federal do Ceará, Campus de Russas, Curso de Ciência da Computação, Russas, 2022.
Orientação: Prof. Dr. Bonfim Amaro Júnior.

1. caminho de corte. 2. golden ball. 3. algoritmos genético. I. Título.

CDD 005

JARDEL TORRES CHAGAS

UMA METAHEURÍSTICA ADAPTADA AO PROBLEMA DE DETERMINAÇÃO DO
CAMINHO DE CORTE DE UM LEIAUTE COM TEMPO MÍNIMO

Trabalho de Conclusão de Curso apresentado ao
Curso de Graduação em Ciência da computação
do CAMPUS DE RUSSAS da Universidade
Federal do Ceará, como requisito parcial à
obtenção do grau de bacharel em Ciência da
computação.

Aprovada em:

BANCA EXAMINADORA

Prof. Dr. Bonfim Amaro Júnior (Orientador)
Universidade Federal do Ceará (UFC)

PROF. DR. MARCIO COSTA SANTOS
Universidade Federal do Ceará (UFC)

PROF.DR. PLÁCIDO ROGÉRIO PINHEIRO
UNIVERSIDADE DE FORTALEZA (UNIFOR)

Mãe, seu cuidado e dedicação foi que deram, em alguns momentos, a esperança para seguir. Pai, sua presença significou segurança e certeza de que não estou sozinho nessa caminhada. Rebeca por está comigo em todos os momentos nessa jornada.

AGRADECIMENTOS

Primeiramente a Deus, Pois dele, por ele e para ele são todas as coisas. A ele seja a glória para sempre! Amém.

Ao Prof. Dr. Bonfim Amaro Júnior por ser um ótimo professor/orientador aprendi muito. Prof. Ms. Eurinaldo Rodrigues Costa por mostrar que tudo é FUP com um pouco de criatividade. Profa. Dra. Tatiane Fernandes Figueiredo por mais desafiador que algo possa parecer, olhe, encare e ache lindo. Prof. Dr. Marcio Costa Santos por toda paciência e ao Prof. Dr. Pablo Luiz Braga Soares e suas frases de Naruto, será que agora me tornarei um Hokage?. Obrigado a todos vocês e os demais professores por tudo ensinado tenho uma vasta gratidão a vocês.

Ao Thiago Felipe meu primeiro incentivador na computação provavelmente sem ele não estaria aqui. A todos os amigos que de alguma forma participou dessa jornada. A todos os colegas da faculdade, que fizemos trabalhos juntos. A todos os amigos do ônibus, antes da pandemia, pela resenhas e toda espera.

A minha família que mesmo muita vezes sem entender o que eu tanto faço na frente do computador sempre esteve do meu lado apoiando o que eu queria ser, desde professor de matemática a cientista da computação. A Rebeca por tudo! Em especial por todo apoio ao tentar entender o meu TCC e ajudar no que for possível.

RESUMO

Uma miríade de aplicações industriais necessitam extrair objetos menores de insumos com tamanhos preestabelecidos. Esse trabalho apresenta uma variação do problema determinação do caminho de corte, evidenciado em uma fase subsequente à definição de um leiaute. O principal objetivo é estabelecer uma ordem de corte para as arestas de todos os polígonos que são as representações computacionais dos itens. Com o intuito de apurar soluções otimizadas para esse problema, apresentamos a abordagem do algoritmo *Golden Ball*. Comparando os resultados obtido com os resultados apresentados do algoritmo genético e o algoritmo genético com chaves tendenciosas. Para validar nossas soluções, utilizamos uma máquina de corte a laser para comparar os tempos requeridos com o software original, algoritmos genéticos e o algoritmo genético com chaves tendenciosas. Com a saída produzida pelo algoritmo aplicado sobre um conjunto de testes escolhido. Obtendo resultados semelhantes em instancias pequenas e uma melhora em instancias maiores, porem com tempo de execução maior que a dos algoritmos evolutivos abordado nesse trabalho.

Palavras-chave: Caminho de Corte, *Golden Ball*, algoritmos genético.

ABSTRACT

A myriad of industrial applications needs to extract smaller objects from inputs with preset sizes. This work presents a variation of the determination problem of the cutting path, evidenced in a phase subsequent to the definition of a layout. The main objective is to establish a cutting order for the edges of all the polygons that are the computational representations of the items. In order to find optimal solutions for this problem, we present the Golden Ball algorithm approach. Comparing the results obtained with the presented results of the genetic algorithm and the genetic algorithm with biased keys. To validate our solutions, we use a laser cutting machine to compare the times required with the original software, genetic algorithms, and genetic algorithm with biased keys. With the output produced by the algorithm applied to a chosen test set. Getting similar results on small instances is an improvement in more significant instances but with a longer execution time than the algorithms developments discussed in this work.

Keywords: Cutting Path, Golden Ball, evolutionary algorithms.

LISTA DE ILUSTRAÇÕES

Figura 1 – Problema de determinação do caminho de corte.	13
Figura 2 – Fluxograma algoritmo genético	15
Figura 3 – Como os pais são selecionados	16
Figura 4 – Fluxograma do algoritmo <i>Golden Ball</i>	17
Figura 5 – Resultados obtidos no trabalho (OSABA <i>et al.</i> , 2016)	20
Figura 6 – Resultados obtidos no trabalho (NEPOMUCENO <i>et al.</i> , 2020)	21
Figura 7 – Representação do jogador	22

LISTA DE TABELAS

Tabela 1 – divisão dos jogadores	17
Tabela 2 – Características das Instâncias	24
Tabela 3 – algoritmo original vs golden ball	25
Tabela 4 – BRKGA vs GB	25
Tabela 5 – Instâncias Empacotadas	26
Tabela 6 – Instâncias Separadas	26

SUMÁRIO

1	INTRODUÇÃO	11
2	FUNDAMENTAÇÃO TEÓRICA	13
2.1	Problema de determinação do caminho de corte	13
2.2	Algoritmos Evolutivos	14
2.2.1	<i>Algoritmo genético Clássico</i>	14
2.2.2	<i>Algoritmo Genético de Chaves Aleatórias Tendenciosas (BRKGA)</i>	16
2.3	<i>Metaheurística Golden Ball</i>	16
3	TRABALHOS RELACIONADOS	19
4	GOLDEN BALL APLICADO AO PROBLEMA DE DETERMINAÇÃO DO CAMINHO DE CORTE	22
5	RESULTADOS	24
5.1	Instancias	24
5.2	Resultados Golden Ball	24
6	CONCLUSÕES	28
	REFERÊNCIAS	29

1 INTRODUÇÃO

Após a revolução industrial a manufatura vem sendo substituída pela maquinofatura. Há destacados investimentos em máquinas para a realização de trabalhos, que até então, eram feitos de maneira artesanal. Em tese, as máquinas reduzem riscos, melhoram a qualidade de trabalho e diminuem o tempo de produção fabril. Ainda assim, é possível investigar a maquinofatura além da melhora dos dispositivos (*hardware*). Por exemplo, empresas que operam nos setores de construção civil, fabricação e transporte de móveis, do vestuário e na indústria naval, deparam-se, diariamente, com problemas relacionados ao corte de itens menores que devem ser extraídos de uma peça maior.

Nesse contexto, surgiram pesquisas voltadas para a camada de *software* inerente dos equipamentos automatizadores considerando algoritmos exatos ou heurísticos. Profusas abordagens representando problemas evidenciados nas indústrias pertencem a classe NP-difícil (WÄSCHER *et al.*, 2007), ou seja o problema não é resolvido em tempo polinomial. Caso do problema de determinação do caminho de corte de um leiaute buscando tempo mínimo. Em escala industrial, a solução sobre instâncias desse problema se torna inviável em tempo computacional hábil, no escopo de métodos computacionais exatos (modelos matemáticos).

Dessa forma, estudos voltados para algoritmos heurísticos e metaheurísticas são motivados na busca de soluções factíveis e de qualidade destacada. Esse trabalho apresenta uma variante do problema determinação do caminho de corte, focado na fase posterior à definição de um leiaute considerando, ainda, parâmetros de um dispositivo de corte: a velocidade de movimentação da cabeça quando está cortando ou não.

Diante do exposto, é fundamental definir a trajetória de corte que minimize o tempo em uma variação de máquinas de corte como: laser, plasma, oxicorte e jato d'água. Esses equipamentos possuem variações de acordo com o tipo do material a ser cortado, a potência significativa, bem como o tempo de deslocamento para a mover o bico de corte, ou seja, a movimentação sem que esteja cortando, que decorre da capacidade de translação para todos os eixos, comumente agindo independentemente (DEWIL *et al.*, 2016).

Dada essas ponderações, esse trabalho apresentará um estudo fundamentado em outra pesquisa sobre duas abordagens para a determinação do caminho de corte de um leiaute com tempo mínimo: um algoritmo genético clássico e, em especial, o algoritmo genético de chaves aleatórias tendenciosas (BRKGA). Este, por sua vez, obteve melhores resultados.

Na pesquisa abordada iremos utilizar o algoritmo *Golden Ball*(seção 3.3) para a

resolução do problema de determinação do caminho de corte, em que cada jogador é uma representação de uma resposta (seção 5.1) inicialmente gerada de forma aleatória. Para a execução do algoritmo necessita de duas variáveis de entradas V_m e V_c que representa respectivamente, velocidade de movimentação da cabeça, ou seja, velocidade em que a cabeça se movimenta sem cortar algo, e a velocidade de Corte. Criado os times (seção 3.3) variáveis de entradas definidas agora basta seguir o fluxo da metaheurística (Figura 4)

Uma comparação entre os resultados obtidos pela adaptação da metaheurística multipopulacional *Golden Ball* (OSABA *et al.*, 2013) aplicada ao problema desse estudo também será realizada (NEPOMUCENO *et al.*, 2020).

A proposta da pesquisa científica representam, além das metas propostas para o estudo do projeto, mecanismo de obtenção de resultados mediante o trabalho realizado. Dessa forma, nessa seção foi introduzida a investigação da meta-heurística *Golden Ball* (GB) no contexto do problema de determinação de corte com tempo mínimo

Esse documento está organizado da seguinte forma: a fundamentação teórica é apresentada (seção 2). Em seguida pelo problema de determinação do caminho de corte, algoritmos Evolutivos, Algoritmo genético clássico, Algoritmo genético com chaves tendenciosas (BRKGA) seguido pela meta-heurística *Golden Ball*, a fim de realizar um comparativo entre o que esses trabalhos e essa pesquisa abrangem (seção 3). Por fim, uma proposta de solução relacionada ao contexto desse trabalho é apresentada, assim como as etapas percorridas para a elaboração da mesma (seção 4).

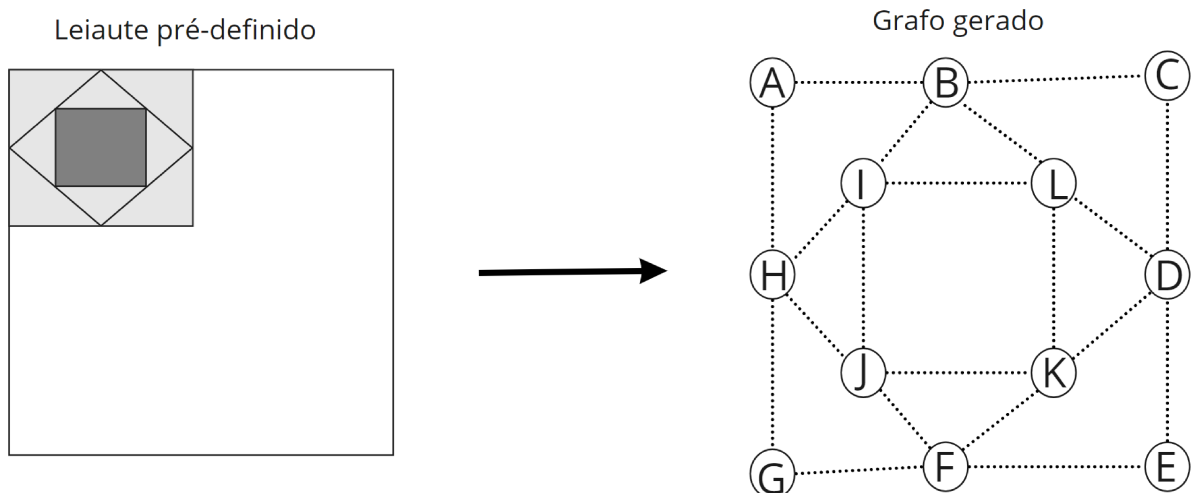
2 FUNDAMENTAÇÃO TEÓRICA

Nessa seção será demonstrada todos os fundamentos teóricos necessário para a execução do trabalho. Primeiramente, a seção 3.1 mostrará o Problema de determinação do caminho de corte que será problema investigado pelo trabalho, posteriormente será introduzido os Algoritmos Evolutivos seção 3.2 dando uma ênfase nos algoritmos genéticos clássico seção 3.2.1 e nos algoritmo Genético de Chaves Aleatórias Tendenciosas seção 3.2.2. Por fim, a seção 3.3 demonstrando a teoria por trás da Metaheurística *Golden Ball*, que será aplicada ao problema de determinação do caminho de corte com as mesmas instancias.

2.1 Problema de determinação do caminho de corte

Com destacada fundamentação para esta pesquisa, (LEE; KWON, 2006) e (HAN; NA, 1999) apresentaram o problema da determinação do caminho de corte (CPDP). O principal objetivo desta aplicação prática é estabelecer uma sequência de movimentos para um equipamento, de modo que todos os itens posicionados na superfície sejam devidamente separados (NEPOMUCENO *et al.*, 2020). Neste processo existem duas fases importantes, a primeira é a geração de um leiaute otimizado, ou seja, o encaixe das peças a serem retiradas de forma a minimizar o desperdício de insumos, e a segunda fase consiste em otimizar a definição da ordem do corte, ou seja, definir qual sequência uma determinada máquina de corte deve seguir para minimizar o tempo de extração de peças. Na figura 1 apresenta a representação para uma instância do problema supracitado.

Figura 1 – Problema de determinação do caminho de corte.



Dada uma demanda que já passou pela fase de geração do leiaute, no trabalho gera-se um grafo não direcionado, como mostrado na Figura 1. Um grafo não direcionado é um tipo abstrato de dado (TAD) que é a relação de adjacência é simétrica, por exemplo, a resta (A, B) é igual a (B, A). Seja um grafo $G = (V, E)$ em que G é formado por uma 2-upla, onde V é um conjunto de vértices ou nós, tomando a figura 1 como exemplo, o conjunto de vértice é definido por $V = A, B, C, D, E, F, G, H, I, J, K, L$ o conjunto de aresta é representado por um conjunto de pares, seja $E = (A, B), (A, H), (B, C), (B, I), (B, L), (C, D), (D, L), (D, K), (D, E), (E, F), (F, K), (F, J), (F, G), (G, H), (H, I), (I, L), (I, J), (L, K), (K, J)$. A aplicação de metaheurísticas evolutivas é uma ferramenta de destacada aplicação para resolver problemas com características combinatórias. Assim, este trabalho foca na investigação desse paradigma, bem como, a vertente multipopulacional do algoritmo *Golden Ball*.

2.2 Algoritmos Evolutivos

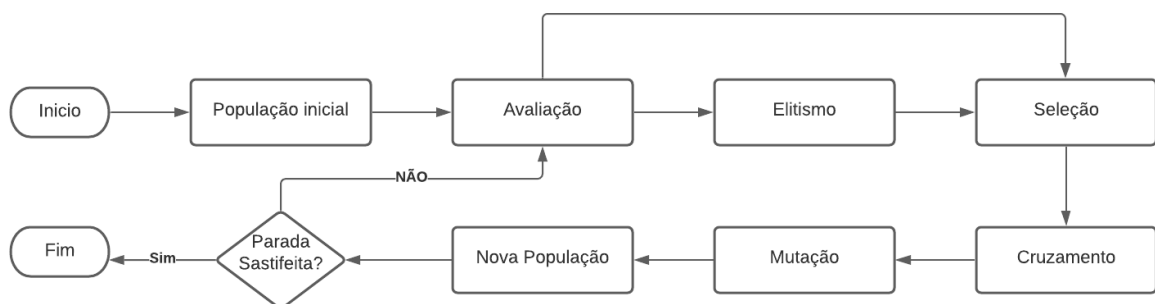
Existem muitas variantes diferente de algoritmos evolutivos, mas todos partem de uma ideia em comum: “dada uma população de indivíduos dentro de algum ambiente com recursos limitados, a competição por esses recursos causa a seleção natural (sobrevivência do mais apto).” (EIBEN; SMITH, 2003) O algoritmo genético clássico (seção 2.2.1) é um algoritmo evolutivo fundamental para o entendimento dos processos evolutivos computacionais, tendo inúmeras variações dentre delas o (BRKGA) algoritmo genético de chaves aleatória tendenciosas (seção 2.2.2).

2.2.1 Algoritmo genético Clássico

Algoritmo genético Clássico, ou simplesmente Algoritmo genético (AG) representam uma analogia computacional em relação à teoria da evolução natural de Darwin. Os AG's foram introduzidos por J. Holland em 1975 e funcionam da seguinte forma: geralmente inicializa a população com indivíduos cujo seus cromossomos são aleatórios e o tamanho da população varia de acordo com a necessidade do problema e está diretamente ligado ao fato de que quanto maior a população maior será o tempo de execução, ou seja, no tempo computacional. Ainda, para avaliar quais são os melhores indivíduos, passa-se por uma função de *fitness*, função de avaliação nos AG's, que dá uma pontuação para cada indivíduo com base nos critérios modelados na função nessa avaliação. Além disso, há as fases de elitismo e seleção, cada indivíduo com

sua avaliação devida, será selecionado para o cruzamento e mutação, gerando assim uma nova população. Uma estratégia interessante para selecionar geracionalmente esses indivíduos pode ser por meio do elitismo, que se configura com o descarte de soluções com a avaliação baixa. É importante frisar que a fase de elitismo pode ser removida (opcional), portanto vai depender da estratégia para o problema. Para selecionar os pais existem alguns métodos como o da roleta (LIPOWSKI; LIPOWSKA, 2011) e torneio (THIERENS; GOLDBERG, 1994). O da roleta consiste em rodar uma roleta e cada indivíduo ocupa um espaço nessa roleta com forme a relação estabelecida ponto do indivíduo por ponto total da população e, no torneio, o indivíduo mais forte, com a melhor função de avaliação é o vencedor, ou seja, o escolhido na imagem a seguir mostra uma variante do fluxograma de um AG.

Figura 2 – Fluxograma algoritmo genético



Fonte: Adaptado Usida *et al.* (2021)

Na fase do cruzamento (DESHPANDE; KELKAR, 2008) geram-se dois filhos combinados em um determinado intervalo dos pais, assim como na natureza existe uma pequena chance de mutação nos seres de uma espécie, porem devemos ter cuidado para não ter uma taxa muito pequena pois pode deixar o processo preso em ótimos locais, no AG também deve-se ter esse operador primordial, mas também tendo cuidado de não colocar uma taxa de mutação muito alta pois pode gerar um efeito de aleatoriedade. assegurando uma varredura mais expansiva do espaço de busca. Após ser aplicado os operadores de mutação gera-se uma nova população, caso o critério de parada seja alcançado o algoritmo termina, tendo como resultado o melhor indivíduo da última população, caso contrário volta-se para a fase de avaliação e fica nesse ciclo o critério de parada ser alcançado.

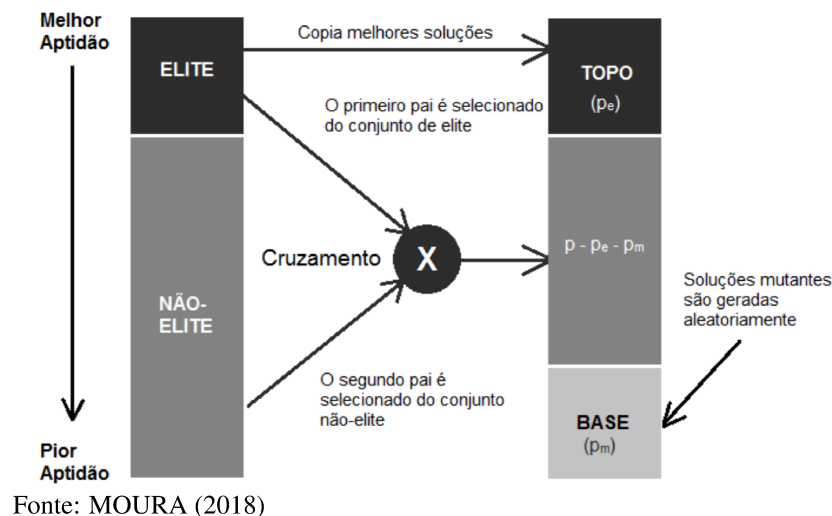
A figura 2 representa uma variação do algoritmo genético, a fase de elitismo não é uma fase obrigatória para os algoritmos genéticos clássicos, mas é uma fase primordial para o Algoritmo Genético de Chaves Aleatórias Tendenciosas (BRKGA).

2.2.2 Algoritmo Genético de Chaves Aleatórias Tendenciosas (BRKGA)

Os algoritmos genéticos com chaves aleatórias tendenciosas (BRKGA) foram propostos por (RESENDE, 2010) contemplando uma variação dos algoritmos genéticos com chaves aleatórias (RKGA)(BEAN, 1994), sua principal distinção é a forma como os pais são selecionados, pois sempre será selecionado um pai que faz parte do conjunto dos indivíduos da elite e o outro pai selecionado pertence ao grupo de não elite, a Figura 3 demonstra como isso é feito. Esse método descrito com fase de elitismo no processo de seleção é responsável por expandir a chance de interação dos indivíduos mais aptos nos métodos de cruzamento. Uma das grandes vantagens foi descrita por (NEPOMUCENO *et al.*, 2020)

Uma das vantagens de utilização do BRKGA vem da sua tolerância a diversos problemas de otimização, assumindo que dois elementos do seu arcabouço devem ser representados: um decodificador compatível e uma função de aptidão representativa.

Figura 3 – Como os pais são selecionados

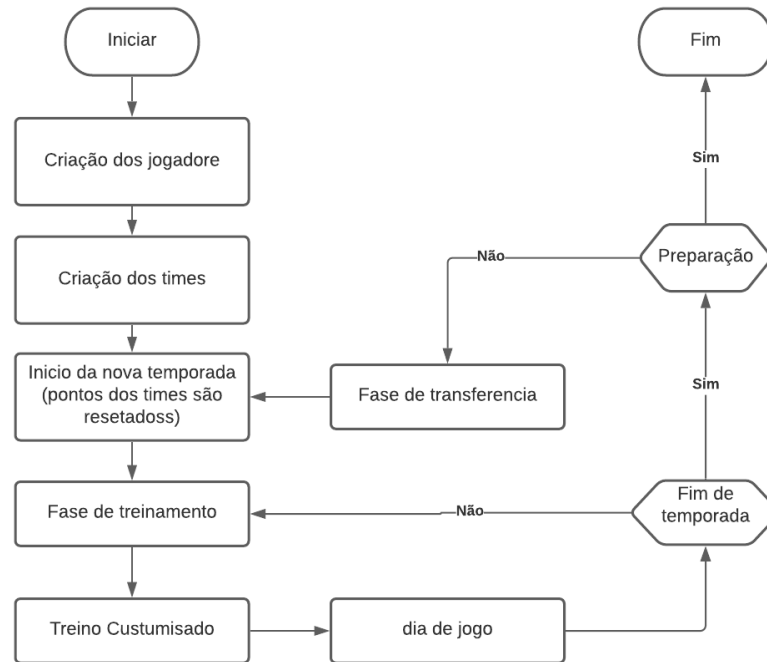


2.3 Metaheurística *Golden Ball*

O algoritmo *Golden Ball* é uma metaheurística que opera sobre o conceito de metaheurísticas multipopulacionais e estratégias relacionadas ao futebol. Bola de Ouro (no original em francês *Ballon d'Or*) é um prêmio de futebol introduzido pela revista francesa *France Football*. Curiosamente, essa distinção já foi conhecida mundialmente como Futebolista do Ano na Europa, sendo entregue de forma independente entre 1956 e 2009 e novamente a partir de 2016, após fim da parceria com a Federação Internacional de Futebol (FIFA). A figura 4 apresenta o fluxograma

do método (OSABA *et al.*, 2013).

Figura 4 – Fluxograma do algoritmo *Golden Ball*



Fonte: adaptado (OSABA *et al.*, 2013).

Em termos computacionais, preliminarmente cria-se a população inicial que é um conjunto de soluções introdutórias, como qualquer outro algoritmo evolutivo, mas no *Golden Ball*, esse processo refere-se à criação de jogadores, e a divisão em subpopulação é responsável pela formação dos times.

De acordo com (OSABA *et al.*, 2013), a definição do tamanho da população é calculada por $P = T_n * P_t$. O termo T_n informa os números de times na liga e P_t o número de jogadores de cada time. O próximo passo é referente à fase da divisão dos jogadores formando, assim, os times que podem ser representados por uma matriz, veja a tabela 1. Cada jogador é representado por P_{ij} , ou seja, jogador i no plantel j .

Tabela 1 – divisão dos jogadores

L/C	1	2	3	...	P_t
1	P_{11}	P_{12}	P_{13}	...	$P_1 P_t$
2	P_{21}	P_{22}	P_{23}	...	$P_2 P_t$
3	P_{31}	P_{32}	P_{33}	...	$P_3 P_t$
...
T_n	PT_{n1}	PT_{n2}	PT_{n3}	...	$P_{TN} P_t$

Fonte: Adaptado Osaba *et al.* (2013)

Agora precisamos calcular a força do time, essa valor será usada nas partidas para

determinar o vencedor do jogo, conceito semelhante de uma função de custo, de outro modo, a função de aptidão dos algoritmos genéticos. O jogador com maior qualidade (ou aptidão) será denominado de capitão do time.

A fase de treinamento tem objetivo de qualificar os jogadores, ou seja, tenta elevar a força individual e, conseqüentemente, aumentar a força do time. Os times jogam entre si como em uma liga, que irar ranquear influenciando, assim, uma fase posterior, a fase de transferência.

A liga dura $2T_n - 2$ jogos. Cada time se enfrenta duas vezes e tem P_t chances de fazer gol. Cada chance é calculada por meio do método de torneio entre p_{ij} . Cada jogador da equipe se enfrenta de acordo com um ranque interno de cada time, esse ranque é quem define o capitão, pois quem tem mais pontos é o primeiro no ranque e conseqüentemente é o capitão, os capitães dos times vão se enfrentar, os segundos no ranque vão se enfrentar e assim em diante até chegar no jogador na posição P_t . O jogador que ganhar o torneio ganha o gol. A liga como na vida real premia o ganhador da partida com três pontos e o perdedor fica com zero pontos e em caso de empates cada equipe fica com um ponto, esses pontos são somados até o fim da temporada, definindo os melhores times. Após o final da temporada verifica se o critério de parada foi alcançado caso não tenha sido alcançado começa o período de transferências, os times da metade superior fazem aquisição do melhor jogador dos times da metade inferior da tabela e os times da metade inferior se contenta com os menos bons dos times da parte de cima

3 TRABALHOS RELACIONADOS

Nessa seção será apresentada uma revisão da literatura existente sobre o problema investigado nesse artigo. De antemão, avisamos ao leitor que esta revisão é focada nos métodos para resolver o problema de determinação do caminho de corte e o uso do algoritmo *Golden Ball*.

O problema de determinação do caminho de corte é objeto de estudo de abundantes pesquisas. (HAN; NA, 1999) tratam o problema dessa forma: percorrer os contornos do corte com um comprimento mínimo de percurso e minimizar o efeito do calor gerado pelo equipamento (tubo) na sequência do percurso de corte. Para isso, a temperatura crítica deve ser evitada durante o procedimento. Uma estrutura similar ao (*Traveling Salesman Problem - TSP*) é resolvida por meio de um algoritmo fundamentado na meta-heurística Têmpera Simulada (*Simulated Annealing*) para otimizar o planejamento do caminho de corte. O efeito do calor é incorporado à função de custo com base no modelo analítico modificado de condução de calor, também, apresentado em (HAN; NA, 1999).

No trabalho de (OSABA *et al.*, 2016) a meta-heurística *Golden Ball* (GB) foi comparada com os algoritmos *Evolutionary Simulated Annealing* (ESA) e o *Tabu Search* (TS) aplicado ao problema do caixeiro viajante. Os objetivos do artigo era além de descrever o funcionamento do golden ball de maneira clara , mas também demonstrar que essa meta-heurística, o GB pode competir com alguns das técnicas mais utilizadas da literatura. A Figura 5, a seguir mostra a tabela de resultados obtidos pela pesquisa. Nos testes foram usados 16 instancias com 50 a 144 nós, geradas por (REINELT, 1991). Cada instancia foi executada 30 vezes e o GB obteve melhores resultados do que o ESA e TS em 81,25% dos testes realizados na pesquisa. Por se mostrar competitivo escolhi o *Golden Ball*, melhor descrito na seção 3.3, para comparar os resultados do GB com os (NEPOMUCENO *et al.*, 2020)

Figura 5 – Resultados obtidos no trabalho (OSABA *et al.*, 2016)

TSP.									
Instância	ESA			TS			GB		
Nome	Avg.	S. dev.	Tempo	Avg.	S. dev.	Tempo	Avg.	S. dev.	Tempo
Eilon50	427.6	1.7	1.5	427.8	1.6	1.4	427.2	1.5	1.1
Eil51	431.6	2.9	1.7	428.9	1.4	1.5	428.6	1.3	1.4
Berlin52	7542.0	0.0	2.1	7542.0	0.0	2.2	7542.0	0.0	2.1
St70	679.1	2.8	3.9	679.2	3.4	4.1	679.4	3.5	4.2
Eilon75	550.2	3.9	4.5	552.3	3.7	4.7	544.3	3.3	5.4
Eil76	553.7	4.2	5.1	545.1	3.4	5.4	547.3	3.7	5.5
KroA100	21481.7	150.1	10.6	21495.4	120.9	10.1	21386.7	99.7	9.5
KroB100	22602.2	210.2	11.1	22612.8	205.8	10.7	22311.0	139.6	9.7
KroC100	21170.4	188.7	12.0	21192.6	166.6	12.3	20968.2	111.3	9.3
KroD100	21726.5	156.9	11.7	21761.3	155.4	11.2	21485.8	188.2	9.7
KroE100	22499.7	171.4	11.4	22534.8	175.5	10.9	22266.8	158.1	9.8
Eil101	656.4	4.4	11.7	658.0	5.1	11.4	645.7	4.3	8.9
Pr107	44821.5	179.3	12.1	44834.3	227.0	11.8	44693.0	210.7	10.1
Pr124	59593.6	367.8	18.5	59560.4	453.1	17.9	59348.2	190.3	16.2
Pr136	99864.4	655.7	23.4	99802.3	914.2	23.0	98906.5	986.7	23.7
Pr144	58807.3	220.9	33.9	58973.1	762.4	32.7	58712.0	247.7	34.1

Fonte: (OSABA *et al.*, 2016)

No trabalho de (NEPOMUCENO *et al.*, 2020), os autores utilizaram uma variante do problema de determinação do caminho de corte utilizando o algoritmo genético clássico e o algoritmo genético com chaves tendenciosas para gerar os caminhos de corte, com sua execução em uma máquina de corte a laser, assim como esse trabalho. Os melhores resultados obtidos foram usando o algoritmo BRKGA, definido na seção 3.2.2, tendo uma melhora significativa de 60% em leiautes empacotadas, e uma melhora de 20% em leiautes contendo itens separados no espaço, como demonstrado pela Figura 6. Pelo fato do BRKGA ter se saído melhor nosso principal foco será compara o BG com o BRKGA. A melhor configuração para a execução do algoritmo foi, tamanho da população com 5000 indivíduos com 30% desses indivíduos pertencente ao grupo elite e com 10% dos indivíduos mutáveis. Utilizando a distância de Chebyshev para calcular o tempo entre os vértices de corte e seus possíveis deslocamentos. A maneira na qual irei tratar os indivíduos e calcular a distancia será igual ao do (NEPOMUCENO *et al.*, 2020), para que os testes possam ser o mais fidedignos o possível.

Figura 6 – Resultados obtidos no trabalho (NEPOMUCENO *et al.*, 2020)

Instância	Original		Melhor Solução		Porcentagem de Melhoria da Solução	
	Empacotado	Separado	Empacotado	Separado	Melhora Empacotado	Melhora Separado
albano	118,19	117,94	104,62	107,23	12,97%	9,99%
dighe1	113,64	114,54	70,53	96,03	61,12%	19,28%
fu	34,96	34,9	23,83	27,95	46,71%	24,87%
_2pol	37,6	37,68	33,12	36,4	13,53%	3,52%
_4pol	75,04	75,23	57,36	72,13	30,82%	4,30%
_5pol	94,53	94,28	69,61	90,13	35,80%	4,60%
_9pol	193,92	194,54	123,98	186,41	56,41%	4,36%
rco3	528,32	529,45	393,93	480,36	34,12%	10,22%
shapes2	246,1	242,87	214,73	227,79	14,61%	6,62%
shapes4	486,67	486,32	419,93	447,99	15,89%	8,56%

Fonte: (NEPOMUCENO *et al.*, 2020)

Já no trabalho de (SILVA, 2016) ele propôs diferentes modelos para a resolução do problema de determinação de caminho de corte (CPDP). Foram propostos três modelos matemáticos para tratar o problema. O primeiro modelo foi baseado no problema do carteiro rural (RPP), o segundo modelo foi baseado no problema do caixeiro viajante (TSP) e a terceira na adaptação foi baseada no problema do caixeiro viajante generalizado (GTSP). Em seus experimentos mostraram que o problema do carteiro rural utilizando o solver CPLEX apresentou os melhores resultados, encontrado todos os resultados ótimos para as instâncias testadas, que algumas delas se parecem com as que foram testadas nesse trabalho. Porém as instâncias testadas tinha como organização de suas peças empacotadas. No mesmo trabalho foi comparado também com o algoritmo de Lin-Kernighan que é uma meta-heurística que se assemelha ao método utilizado nos treinamentos dos jogadores, o 2-opt.

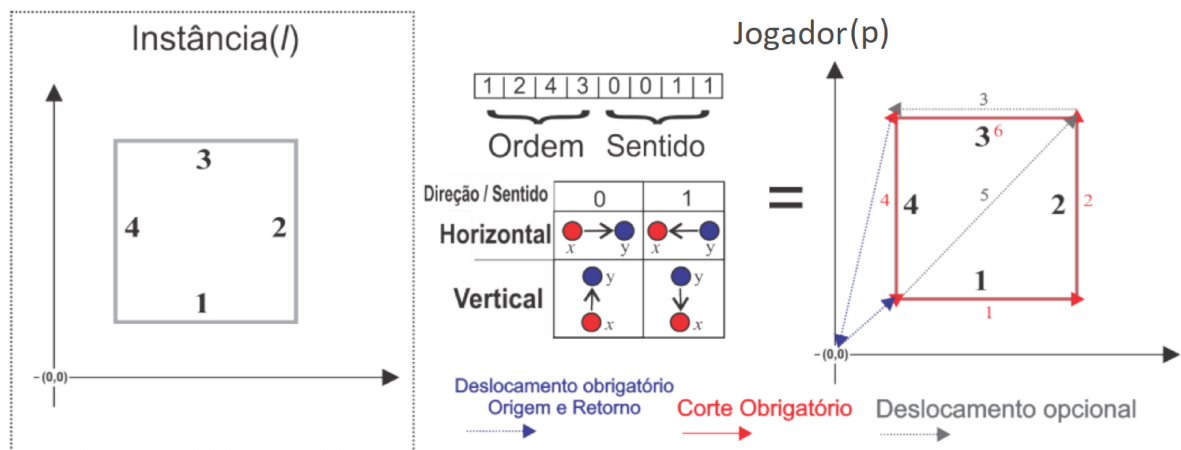
Utilizando o algoritmo demonstrado por (OSABA *et al.*, 2016) com as instancias usada por (NEPOMUCENO *et al.*, 2020) utilizando a mesma máquina de corte, irei comparar os algoritmos afim de constatar que o Golden Ball é um algoritmo que trás resultados competitivos em problemas de otimização combinatória, aplicando o mesmo ao problema de determinação de caminho de corte.

4 GOLDEN BALL APLICADO AO PROBLEMA DE DETERMINAÇÃO DO CAMINHO DE CORTE

Nessa seção será demonstrado os procedimentos metodológicos a serem desenvolvidos nesta pesquisa, como a estratégia *Golden Ball* é uma metaheurística evolutiva, o primeiro passo foi realizar uma revisão bibliográfica na literatura que tratava, não somente, da variação do problema de determinação do caminho de corte sob a ótica dos algoritmos mediante mesmo paradigma, mas também de possíveis adaptações. Com isso em mente, uma maneira de modelar o problema é suscitada nesta seção.

Assim como no trabalho de (NEPOMUCENO *et al.*, 2020) a execução será em uma máquina de corte a laser que desloca por coordenadas (x, y) . Como parâmetros de entrada será definido velocidade de movimentação da cabeça (V_m) e velocidade de corte (V_c). Esses valores podem variar conforme a necessidade dos equipamentos. Para calcular o tempo que uma máquina levaria para extrair peças dos insumos foi considerado a distância de Chebyshev e dependendo da ação tomada pela máquina será escolhido V_m representado pela seta cinza na figura 6 ou V_c representado pela seta vermelha ilustrado também pela figura 6.

Figura 7 – Representação do jogador



Fonte: Adaptado de Nepomuceno *et al.* (2020)

Definido as entradas, agora é a definição dos jogadores, cada jogador é criado aleatoriamente e são colocados em T_n times também de forma aleatória. A representação dos jogadores é dividida em duas partes, como apresenta a figura 6. A primeira metade do vetor informa a ordem de corte e a outra metade, o sentido do corte escolhido pela máquina. Dessa forma, a qualidade dos jogadores e, conseqüentemente, a qualidade dos times está ligada ao tempo de corte de um leiaute, ou seja, quanto mais rápido realizar o corte, maior será a qualidade

do jogador. Durante a competição há duas etapas fundamentais para o algoritmo, o método de treino e as partidas que darão *ranking* para os times e esse *ranking* é utilizado na fase de transferência.

Na fase de treino os jogadores são submetidos a aplicação métodos para melhorar sua performance nas partidas. Isso mantém um caráter semelhante ao mundo real. Cada equipe pode possuir um método de treino distinto, e dessa forma, esse mecanismo será atribuído de forma aleatória na criação dos times. "A saber, uma função sucessora que atua em uma determinada estrutura de vizinhança no espaço de solução para certos problemas de roteamento, uma função desse tipo poderia ser a conhecida 2-opt ou 3-opt"(OSABA *et al.*, 2013). A aplicação desta função pre escolhida pode ser realizada um numero x de vezes para cada treino objetivando melhorar um conjunto determinado de jogadores.

Cada equipe possui um treino personalizado que tem como objetivo fugir de ótimos locais, esse treino personalizado acontece quando um jogador não melhora a um numero determinado de treinos, o que faz ele treinar com o capitão do time, nesse treino acontece uma variação do "*crossover*". O jogador que está precisando de treino recebe metade (possivelmente parametrizado) das características do capitão do time.

Os jogos consistem em P_t chances de gols, cada chance se efetiva em gol por meio de um torneio entre os jogadores dos times. Como cada time tem a lista de jogadores ordenados pela aptidão, a lógica do processo é organizada com a relação entre melhores disputando em pares pelo gol (ponto).

Por fim, o período de transferência de jogadores. Nesse contexto, os times que obtiveram melhores posições na tabela escolhe os melhores jogadores dos times da metade de baixo, sendo que o melhor time pega o melhor jogador do pior time e o segundo time pega o melhor jogador do penúltimo time e assim por diante. Todos esses processos vão se repetir até que o critério de parada seja alcançado.

5 RESULTADOS

Os experimentos computacionais foram realizados em uma máquina com Processador Intel i5 10ª geração de 2.9 GHz com 6 Núcleos, 12MB de cache e 16 GB de memória RAM e os códigos implementados na linguagem Python 3.10, sistema operacional Ubuntu 22.04.1 LTS.

5.1 Instancias

Para mensurar uma análise das abordagens, empregamos um conjunto de 10 instâncias, contemplando duas características fundamentais sobre espaçamento entre peças de um leiaute: Empacotado (E) e Separado (S). As configurações aplicadas para o Golden Ball foram as seguintes: quantidade de temporadas 1, 1 e 2; quantidade de times 5, 5 e 10; quantidade de jogadores por time 11, 5 e 11. Foi realizado 10 execuções para cada instância descrita na Tabela 2 que ainda define, a quantidade de vértices, o número de arestas: do leiaute Original e leiaute Adaptado com a junção das arestas em comum, além do total de polígonos. O critério de parada ocorre quando a quantidade de temporada chegar ao fim.

Tabela 2 – Características das Instâncias

Instância	Vértices	Arestas				Polígonos
		Original		Adaptado		
		(E)	(S)	(E)	(S)	
albano	156	164	164	173	164	24
dighe1	20	46	54	38	54	15
fu	37	43	43	51	43	12
instance_01_2pol	7	8	8	8	8	2
instance_01_4pol	10	16	16	13	16	4
instance_01_5pol	12	19	20	16	20	5
instance_01_9pol	18	35	36	26	36	9
rco3	82	108	108	116	108	21
shapes2	68	70	70	78	70	8
shapes4	127	140	140	147	140	16

Fonte: Adaptado de (NEPOMUCENO *et al.*, 2020)

5.2 Resultados *Golden Ball*

A Tabela 3 apresenta uma comparação entre os tempos do leiaute original e outro modificado a partir da melhor solução encontrada pelo Golden Ball. Ambas foram executadas em uma máquina de corte a laser com as seguintes configurações: Prisma produzida pela Automatisa

Laser Solutions, canhão de vidro com potência de 60w e área de trabalho máxima de 900 x 600 mm. O software utilizado é exclusivo da própria empresa. As velocidades V_c (corte) e V_m (deslocamento) são 16,67 mm/s e 400 mm/s, respectivamente. Por sua vez, esses valores foram , tanto no dispositivo, como na realização dos testes das abordagens. Todos os tempos estão em segundos.

Tabela 3 – algoritmo original vs golden ball

Instâncias	Original		Melhor Solução	
	Empacotado	Separado	Empacotado	Separado
albano	118.19	117.94	101.52	104,90
dighe1	113.64	114.54	70.35	97.64
fu	34.96	34.90	23.62	28.16
instance_01_2pol	37.60	37.68	33.12	36.27
instance_01_4pol	75.04	75.23	57.37	72.57
instance_01_5pol	94.53	94.28	69.62	90.72
instance_01_9pol	193.92	194.54	124	188.52
rco3	528.32	529.45	391.63	482.63
shapes2	246.10	242.87	213.94	223.74
shapes4	486.67	486.32	409.15	440.81

Como podemos observar houve uma mudança significativa comparando o algoritmo original com o Golden Ball, uma melhora em todas as instancias. Da mesma forma foi comparado com o algoritmo genético com chaves tendenciosas (BRKGA), nessa comparação o GB obteve resultados semelhantes, na Tabela 5 e na Tabela 6 podemos observar que em algumas instancias o resultado da FO foi melhor com o custo de tempo de execução mais lento, mesmos resultados com o tempo de execução mais rápida para o Golden Ball e resultados negativos para o GB com tempo de execução mais lento e com a FO pior.

Tabela 4 – BRKGA vs GB

Instâncias	Melhor Solução BRKGA		Melhor Solução GB	
	Empacotado	Separado	Empacotado	Separado
albano	104.62	107.23	101.52	104,90
dighe1	70.53	96.03	70.35	97.64
fu	23.83	27.95	23.62	28.16
instance_01_2pol	33.12	36.04	33.12	36.27
instance_01_4pol	57.36	72.13	57.37	72.57
instance_01_5pol	69.61	90.13	69.62	90.72
instance_01_9pol	123.98	186.41	124	188.52
rco3	393.93	480.36	391.63	482.63
shapes2	214.73	227.79	213.94	223.74
shapes4	419.93	447.99	409.15	440.81

Por fim na tabela 5 (instâncias com leiautes contendo peças juntas) e na tabela 6 (instâncias com leiautes contendo peças separadas) foi demonstrado a melhor configuração encontrada para o GB em ambas situações. O "P" remete ao tamanho da população, "E" representa a parcela da população "ELITE" e "M" a quantidade do bloco da mutação. "FO" o tempo requerido da melhor solução e, finalmente, o tempo de execução do algoritmo. O "J" define o número de jogadores por equipe, "Ti" a quantidade de equipes jogando, "Te" representa a quantidade de temporadas a serem jogadas pelas equipes

Tabela 5 – Instâncias Empacotadas

Instâncias	BRKGA					Golden Ball					
	P	E	M	FO	Tempo	J	Ti	Te	FO	Tempo	melhora
albano	5000	10%	10%	104.62	300s	5	5	1	101.82	7260.84	2,68%
dighe1	5000	30%	10%	70.53	124s	11	5	1	70.35	28.02	0,26%
fu	5000	30%	10%	23.83	164s	11	5	1	23.62	100.12	0,88%
_2pol	10000	20%	10%	33.12	51s	5	5	1	33.12	0.05	0%
_4pol	1000	10%	15%	57.36	07s	11	5	1	57.37	0.48	-0,02%
_5pol	1000	30%	10%	69.61	08s	11	5	1	69.62	1.07	-0,01%
_9pol	10000	10%	10%	123.98	136s	11	5	1	124	6.98	-0,02%
rco3	5000	20%	10%	393.93	300s	5	5	1	391.42	1410.34	0,64%
shapes2	10000	10%	10%	214.73	300s	5	5	1	214.71	311.77	0,01%
shapes4	5000	10%	15%	419.93	300s	5	5	1	409.15	4030.16	2,57%

Na Tabela 5 expõe que o Golden Ball se saiu melhor nas instâncias Empacotadas, com uma melhora de 2.9%, uma pequena melhora, sim é verdade! Mas demonstrando que o GB é uma meta-heurística extremamente competitiva com as demais meta-heurísticas em problemas de otimização combinatória.

Tabela 6 – Instâncias Separadas

Instâncias	BRKGA					Golden Ball					
	P	E	M	FO	Tempo	J	Ti	Te	FO	Tempo	melhora
albano	5000	10%	10%	107.23	300s	5	5	1	104,90	5671.41	2,17%
dighe1	5000	30%	10%	96.03	124s	11	5	1	97.64	124.1	-1,68%
fu	5000	30%	10%	27.95	164s	11	5	1	28.16	48.76	-0,75%
_2pol	10000	20%	10%	36.04	51s	11	5	1	36.27	0.09	-0,64%
_4pol	1000	10%	15%	72.13	07s	11	5	1	72.57	1.08	-0,61%
_5pol	1000	30%	10%	90.13	08s	11	5	1	90.72	2.5	-0,65%
_9pol	10000	10%	10%	186.41	136s	11	5	1	188.52	26.13	-1,13%
rco3	5000	20%	10%	480.36	300s	5	5	1	482.63	982.22	-0,47%
shapes2	10000	10%	10%	227.79	300s	5	5	1	223.53	186.59	1,87%
shapes4	5000	10%	15%	447.99	300s	5	5	1	441.97	3337.81	1,36%

Tratando-se das instâncias separadas o GB não se saiu tão bem quanto nas instâncias

empacotadas, mas ainda sim os resultados foram bem semelhantes tendo resultado melhor nas 3 maiores, se tratando de vértices, resultado esse obtido por causa da função objetiva escolhida, irei dissertar melhor na seção 7

6 CONCLUSÕES

Neste trabalho, foi compreendido uma variante do problema de determinação do caminho de corte, atribuindo-se sua execução em uma máquina de corte a laser. O principal objetivo foi encontrar uma sequência de movimentos, de maneira que todos os polígonos de um determinado leiaute sejam cortados no tempo mínimo possível. Todos os testes foram realizados em 10 instâncias geradas por (JÚNIOR *et al.*, 2017). Para isso, desenvolvi o algoritmo *Golden Ball*. O tempo foi computado por meio do cálculo da distância de Chebyshev entre as arestas de corte, possíveis deslocamentos e dos parâmetros de entrada para realizar a simulação na máquina de corte a laser: V_c (velocidade de corte) e V_m (velocidade de deslocamento). Destacamos uma melhora de até 2.9% em leiautes empacotados e 2.14% em leiaute separados, comparando-se com o BRKGA que em trabalhos anteriores já havia se saído melhor que o algoritmo original e o algoritmo genético básico.

Apesar da meta heurística *Golden Ball* possuir melhores resultados na maioria dos casos dos empacotados, quando se tratou das instancias cuja os polígonos estavam separados obtive resultados piores para instancias pequenas e de tamanho médio, porém nas instancias com mais arestas obtive resultados melhores com tempo de execução mais elevado. Isso se deve ao fato da função objetiva, quem mais pontuava era individuo que conteve o menor número de deslocamento sem corte, todavia é melhor realizar x movimentos pequenos sem cortar do que um movimento só, uma vez que esse movimento maior contém o comprimento maior que a soma de todos os deslocamentos x . Por esse motivo essa função objetiva adquiriu melhores resultados para os casos em que não existem polígonos separados. Os resultados do algoritmo *Golden Ball* pode ser consultado no link: resultados

REFERÊNCIAS

- BEAN, J. C. Genetic algorithms and random keys for sequencing and optimization. **ORSA Journal on Computing**, v. 6, n. 2, p. 154–160, 1994. Disponível em: <<https://doi.org/10.1287/ijoc.6.2.154>>.
- DESHPANDE, A.; KELKAR, R. Advanced genetic operators and techniques: an analysis of dominance diploidy, reordering operator in genetic search. In: . [S.l.: s.n.], 2008. p. 27–33.
- DEWIL, R.; VANSTEENWEGEN, P.; CATTRYSSE, D. A review of cutting path algorithms for laser cutters. **The International Journal of Advanced Manufacturing Technology**, v. 87, p. 1865–1884, 11 2016.
- EIBEN, A. E.; SMITH, J. E. Genetic algorithms. In: _____. **Introduction to Evolutionary Computing**. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. p. 37–69. ISBN 978-3-662-05094-1. Disponível em: <https://doi.org/10.1007/978-3-662-05094-1_3>.
- HAN, G. C.; NA, S.-J. A study on torch path planning in laser cutting processes. part 2: Cutting path optimization using simulated annealing. **Journal of Manufacturing Systems**, v. 18, p. 62–70, 12 1999.
- JÚNIOR, B.; PINHEIRO, P.; COELHO, P. A parallel biased random-key genetic algorithm with multiple populations applied to irregular strip packing problems. **Mathematical Problems in Engineering**, v. 2017, p. 1–11, 09 2017.
- LEE, M.-K.; KWON, K.-B. Cutting path optimization in cnc cutting processes using a two-step genetic algorithm. **International Journal of Production Research**, Taylor Francis, v. 44, n. 24, p. 5307–5326, 2006. Disponível em: <<https://doi.org/10.1080/00207540600579615>>.
- LIPOWSKI, A.; LIPOWSKA, D. Roulette-wheel selection via stochastic acceptance. **Physica A: Statistical Mechanics and its Applications**, v. 391, 09 2011.
- MOURA, M. A. Algoritmo genético de chaves aleatórias segundo distribuição de levy para otimização global. Universidade Federal de Pernambuco, 2 2018.
- NEPOMUCENO, G.; SOARES, D. C.; SANTOS, K. B. dos; JUNIOR, B. A.; SANTOS, M. C. Abordagens evolutivas para determinação do caminho de corte de um leiaute com tempo mínimo. **ANAI DO LII SIMPÓSIO BRASILEIRO DE PESQUISA OPERACIONAL**, 9 2020.
- OSABA, E.; CARBALLEDO, R.; GARCÍA, P. L.; DIAZ, F. Comparison between golden ball meta-heuristic, evolutionary simulated annealing and tabu search for the traveling salesman problem. In: . [S.l.: s.n.], 2016.
- OSABA, E.; DÍAZ, F.; ONIEVA, E. Golden ball: A novel meta-heuristic to solve combinatorial optimization problems based on soccer concepts. **Applied Intelligence**, v. 41, p. 145–166, 12 2013.
- REINELT, G. Tsplib—a traveling salesman problem library. **ORSA Journal on Computing**, v. 3, n. 4, p. 376–384, 1991. Disponível em: <<https://doi.org/10.1287/ijoc.3.4.376>>.
- RESENDE, M. Biased random-key genetic algorithms with applications in telecommunications. **TOP**, v. 20, p. 130–153, 01 2010.

SILVA, E. F. da. Modelos matemáticos para um problema de caminho de corte. 04 2016.

THIERENS, D.; GOLDBERG, D. Convergence models of genetic algorithm selection schemes. In: DAVIDOR, Y.; SCHWEFEL, H.-P.; MÄNNER, R. (Ed.). **Parallel Problem Solving from Nature — PPSN III**. Berlin, Heidelberg: Springer Berlin Heidelberg, 1994. p. 119–129. ISBN 978-3-540-49001-2.

USIDA, W.; COURY, D.; FLAUZINO, R.; SILVA, I. Alocação de dispositivos indicadores de faltas em alimentadores primários usando algoritmos genéticos. 08 2021.

WÄSCHER, G.; HAUßNER, H.; SCHUMANN, H. An improved typology of cutting and packing problems. **European Journal of Operational Research**, v. 183, p. 1109–1130, 12 2007.