**UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO**

**CENTRO DE CIÊNCIAS EXATAS, NATURAIS E DA SAÚDE**

**DEPARTAMENTO DE COMPUTAÇÃO**

**LUCAS DOS SANTOS CARVALHO**

***CLUSTERING SEARCH* PARA O PROBLEMA DE ESCALONAMENTO DE PROJETOS COM RESTRIÇÕES DE RECURSOS**

ALEGRE - ES

2022

LUCAS DOS SANTOS CARVALHO

***CLUSTERING SEARCH* PARA O PROBLEMA DE ESCALONAMENTO DE PROJETOS COM RESTRIÇÕES DE RECURSOS**

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Orientador: Prof. Dr. Geraldo Regis Mauri.

ALEGRE - ES

2022

**LUCAS DOS SANTOS CARVALHO**

***CLUSTERING SEARCH* PARA O PROBLEMA DE ESCALONAMENTO DE PROJETOS COM RESTRIÇÕES DE RECURSOS**

Trabalho de conclusão de curso apresentado ao Departamento de Computação do Centro de Ciências Exatas, Naturais e da Saúde da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do grau de Bacharel em Sistemas de Informação.

Aprovado em \_\_\_\_ de agosto de 2022.

**ORIENTADOR**

**\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_**

**Prof. Dr. Geraldo Regis Mauri**

**Universidade Federal do Espírito Santo**

**RESUMO**

O escalonamento de projetos é um fator importante para empresas de fabricação em ordem. Um problema comum de escalonamento é o *Resource-Constrained Project Scheduling Problem* (RCPSP), que é um problema dotado como uma generalização do *Job Shop Scheduling Problem* (JSSP). O RCPSP envolve o escalonamento das atividades de um ou mais projetos, considerando seus recursos renováveis, e visando reduzir a duração total do projeto. Neste trabalho, pretende-se utilizar a meta-heurística *Clustering Search* (CS) para a resolução da abordagem clássica do RCPSP. Essa meta-heurística apresentou bons resultados para outros problemas de otimização encontrados na literatura, e ainda não foi explorada como alternativa para o RCPSP.

Palavras-chave: *Resource Constrained Project Scheduling Problem.* *RCPSP*. *Clustering Search*.

**SUMÁRIO**

**1.** [**INTRODUÇÃO**](#_30j0zll) **6**

[1.1. O PROBLEMA E SUA IMPORTÂNCIA](#_1fob9te) 6

[1.2. OBJETIVOS](#_3znysh7) 8

[1.2.1. Objetivo Geral](#_2et92p0) 8

[1.2.2. Objetivos Específicos](#_tyjcwt) 9

[1.3. ESTRUTURA DO TRABALHO](#_3dy6vkm) 9

**2.** [**REVISÃO DA LITERATURA**](#_1t3h5sf) **10**

[2.1. TRABALHOS RELACIONADOS AO RCPSP](#_2s8eyo1) 14

[2.2. TRABALHOS RELACIONADOS AO CS](#_17dp8vu) 16

[2.3. O ESTADO DA ARTE DO RCPSP](#_3rdcrjn) 18

**3.** [**METODOLOGIA**](#_26in1rg) **21**

[3.1. CLUSTERING SEARCH](#_lnxbz9) 21

[3.1.1. Search Metaheuristic (SM)](#_35nkun2) 22

[3.1.2. Iterative Clustering Component (IC)](#_1ksv4uv) 23

[3.1.3. Analyser Module (AM)](#_44sinio) 23

[3.1.4. Local Search Heuristic (LS)](#_2jxsxqh) 23

[3.2. EXPERIMENTOS COMPUTACIONAIS](#_td8fke2cj4gj) 23

**4.** [**RESULTADOS ESPERADOS**](#_z337ya) **24**

**5.** [**REFERÊNCIAS**](#_3j2qqm3) **25**

**6.** [**CRONOGRAMA**](#_1y810tw) **28**

# INTRODUÇÃO

O escalonamento de projetos é um fator importante para empresas de fabricação em ordem, onde as capacidades são reduzidas para lidar com os conceitos de gerenciamento enxuto (BRUCKER et al., 1999).

O problema de escalonamento de projetos com restrições de recursos, também conhecido como *Resource-Constrained Project Scheduling Problem* (RCPSP), envolve o escalonamento das atividades de um ou mais projetos, do início ao fim, sem atraso, com recursos renováveis e com intenção de minimizar a duração total do projeto. Esse problema é denotado como sendo uma generalização do *Job Shop Scheduling Problem* (JSSP) (HERROELEN; DEMEULEMEESTER; DE REYCK, 1999).

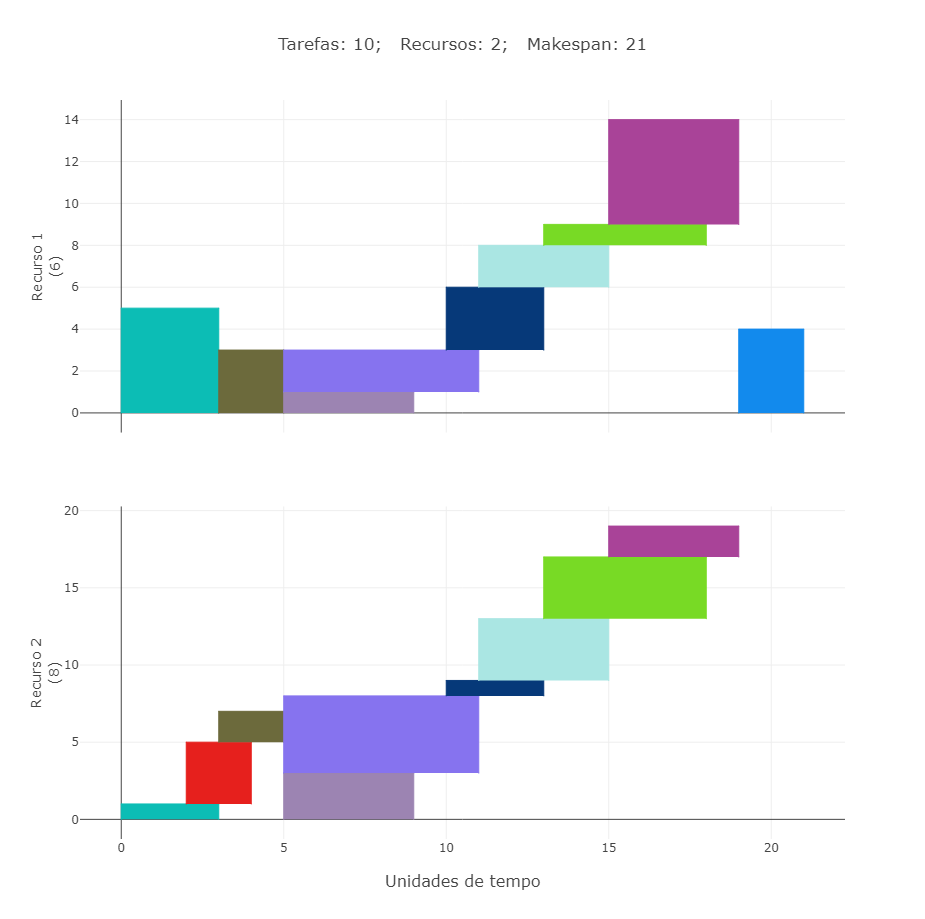
De acordo com Kolisch e Hartmann (1999), o objetivo do RCPSP é encontrar a precedência e a viabilidade para que se complete todas as suas atividades dentro do tempo dos projetos, de maneira que os mesmos tenham seu tempo máximo minimizado.

Dessa maneira, neste trabalho é proposta a utilização de uma meta-heurística híbrida para a resolução do RCPSP, conhecida como *Clustering Search* (CS). A escolha do CS se baseou em seu bom resultado na resolução de outros problemas encontrados na literatura, como *Berth Allocation Problem* (DE OLIVEIRA; MAURI; LORENA, 2012), *Point-Feature Cartographic Label Placement Problem* (RABELLO et al., 2014) e *Bi-objective Flexible Job Shop Scheduling Problem* (ALTOÉ et al., 2018), além de não ter sido utilizado para a resolução do RCPSP.

## 1.1. O PROBLEMA E SUA IMPORTÂNCIA

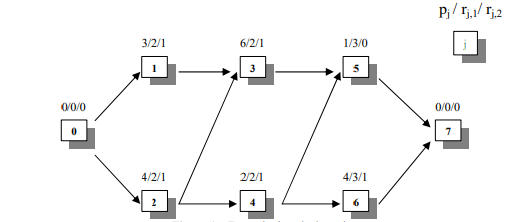
Esse problema é importante para praticantes e também para várias aplicações, variando de *software* de manejo de projetos a sistemas para planejamento de produção e escalonamento (PELLERIN; PERRIER; BERTHAUT, 2020).

Como ilustrado na Figura 1, o RCPSP consiste em encontrar uma escala de atividades, com tempos de início, a fim de minimizar a duração dos projetos (*makespan*). O *makespan* é definido como a diferença entre o início e o final dos projetos, restringido por cada atividade que é executada uma única vez, e cada atividade inicia se e apenas se todas as suas atividades predecessoras estiverem completas, e contando com a disponibilidade do recurso renovável necessário (PELLERIN; PERRIER; BERTHAUT, 2020).

Figura 1 – Exemplo de gráfico de solução para um RCPSP, resultante da instancia “j10.sm”.

Fonte: O autor.

Para ilustrar o RCPSP, a Figura 2 retrata um exemplo com 2 recursos e 6 atividades. A Figura 3 ilustra a utilização destes dois recursos para a solução final.

Figura 2 – Exemplo de RCPSP, onde ***p*** representa a duração da etapa e ***r1*** e ***r2*** representam a utilização dos respectivos recursos.

Fonte: Mendes e Gonçalves (2004).

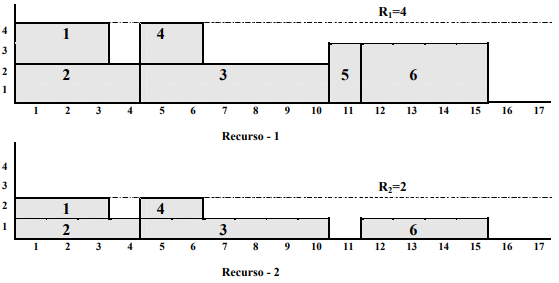


Figura 3 – Ilustração da utilização dos recursos do exemplo demonstrado na Figura 2.

Fonte: Mendes e Gonçalves (2004).

## 1.2. OBJETIVOS

A seguir são apresentados os objetivos gerais e específicos relacionados a este trabalho.

### 1.2.1. Objetivo Geral

Este trabalho tem como objetivo geral aplicar a meta-heurística *Clustering Search* (CS) para resolução do Problema de Escalonamento de Projetos com Restrições de Recursos – RCPSP.

### 1.2.2. Objetivos Específicos

a) Implementar o CS para resolver o RCPSP;

b) Realizar de experimentos computacionais utilizando as instâncias disponibilizadas pela PSPLIB (2022);

c) Avaliar o desempenho do CS comparando com os resultados apresentados na literatura.

## 1.3. ESTRUTURA DO TRABALHO

O Capítulo 2 deste trabalho apresenta uma revisão das principais obras na literatura, os trabalhos com abordagem similar ao RCPSP, utilizando ou não o CS. No capítulo 3 é apresentada a metodologia abordada. No Capítulo 4 são apresentados os resultados esperados, seguido pelas referências bibliográficas apresentadas no Capítulo 5 e, por último, é apresentado um cronograma de execução.

# REVISÃO DA LITERATURA

A busca de trabalhos correlatos ao Problema de Escalonamento de Projetos com Restrições de Recursos (RCPSP) foi realizada nas plataformas *Google Scholar1, ScienceDirect2, IEEEXplore3 e SpringerLink4,* utilizando como parâmetros de busca as palavras-chaves: “*Clustering Search*”, “*Resource Constrained Project Scheduling Problem”* e “*RCPSP”*; e em português: “Problema de Escalonamento de Projetos com Restrições de Recursos” e “Meta-heurística RCPSP”. A busca de cada palavra-chave foi realizada de maneira independente, de modo a atrelar a palavra-chave ao seu respectivo número de resultados.

Para a seleção dos trabalhos, utilizou-se uma análise baseada inicialmente na leitura do título de cada um dos resultados da busca, verificando a existência de relação entre o mesmo e o assunto tratado nesta pesquisa. Para os trabalhos selecionados, foi realizada a leitura do seu resumo.

As Tabelas 1 e 2 apresentam o resultado da análise das buscas realizadas na plataforma *Google Scholar*. Deve-se ressaltar que esta plataforma limita o número páginas para 10 a 50 *links* por página, o que reduz o número total de resultados acessíveis em menos de 1000 por pesquisa. Logo, não foi possível analisar todos os artigos das palavras-chaves que tiveram o resultado maior do que 1000.

Tabela 1 - Resultados da busca na plataforma *Google Scholar* utilizando as palavras-chaves em inglês.

| **Palavra-chave** | **Resultados** | **Número de trabalhos selecionados** |
| --- | --- | --- |
| *“Clustering Search”* | 4.280 | 1 |
| *“Resource Constrained Project Scheduling Problem”* | 469.000 | 1 |
| “RCPSP” | 7.310 | 0 |
| Total | 480.590 | 2 |

Fonte: O autor.

Devido à quantia elevada de resultados, os artigos passíveis de análise foram tidos como os disponíveis nas primeiras páginas de busca, como dito anteriormente, e dos 480.590 resultados apresentados na Tabela 1, 4 artigos haviam sido selecionados, porém um foi descartado pela impossibilidade de acesso, mesmo utilizando as credenciais institucionais da UFES, portanto apenas 2 foram selecionados.

Como pode ser observado na Tabela 2, 2 dos 11.901 resultados obtidos foram selecionados por estarem diretamente relacionados com o tema tratado nesta pesquisa, enquanto os outros resultados encontrados nas primeiras páginas de pesquisa não satisfaziam as condições para seleção e, na melhor das hipóteses, tangenciam o assunto.

Tabela 2 - Resultados da busca na plataforma *Google Scholar* utilizando as palavras-chaves em português.

| **Palavra-chave** | **Resultados** | **Número de trabalhos selecionados** |
| --- | --- | --- |
| “Problema de Escalonamento de Projetos com Restrições de Recursos” | 11.800 | 1 |
| “Meta-heurística RCPSP” | 101 | 1 |
| Total | 11.901 | 2 |

Fonte: O autor.

As Tabelas 3 e 4 apresentam o resultado da análise das buscas realizadas na plataforma *ScienceDirect*.

A Tabela 3 também apresenta uma quantidade elevada de resultados, totalizando 18.837. Na primeira palavra-chave foram selecionados 4 trabalhos. Na segunda e terceira etapa nenhum resultado foi selecionado. Por fim, 5 trabalhos foram selecionados, com todos eles sendo artigos de outros problemas que utilizam CS em sua solução.

Tabela 3 - Resultados da busca na plataforma *ScienceDirect* utilizando as palavras-chaves em inglês.

| **Palavra-chave** | **Resultados** | **Número de trabalhos selecionados** |
| --- | --- | --- |
| *“Clustering Search”* | 759 | 4 |
| *“Resource Constrained Project Scheduling Problem”* | 17.372 | 0 |
| “RCPSP” | 706 | 0 |
| Total | 18.837 | 4 |

Fonte: O autor.

A Tabela 4 não apresenta nenhum resultado devido a plataforma *ScienceDirect* favorecer a publicação de artigos na língua inglesa e, dessa maneira, nenhum artigo foi selecionado.

Tabela 4 - Resultados da busca na plataforma *ScienceDirect* utilizando as palavras-chaves em português.

| **Palavra-chave** | **Resultados** | **Número de trabalhos selecionados** |
| --- | --- | --- |
| “Problema de Escalonamento de Projetos com Restrições de Recursos” | 0 | 0 |
| “Meta-heurística RCPSP” | 0 | 0 |
| Total | 0 | 0 |

Fonte: O autor.

As Tabelas 5 e 6 apresentam o resultado da análise das buscas realizadas na plataforma *SpringerLink*.

A Tabela 5 possui uma quantidade elevada de resultados, totalizando 48.919. Na primeira palavra-chave foram selecionados 3 trabalhos. Na segunda e terceira etapas foram selecionados apenas 2 resultados, uma em cada etapa. Porém, 3 deles não puderam ser acessados, mesmo com as credenciais institucionais. Portanto dos 5 resultados escolhidos, apenas 2 trabalhos foram obtidos em sua integridade e assim sendo selecionados.

Tabela 5 - Resultados da busca na plataforma *SpringerLink* utilizando as palavras-chaves em inglês.

| **Palavra-chave** | **Resultados** | **Número de trabalhos selecionados** |
| --- | --- | --- |
| *“Clustering Search”* | 1.146 | 0 |
| *“Resource Constrained Project Scheduling Problem”* | 46.727 | 1 |
| “RCPSP” | 1.046 | 1 |
| Total | 48.919 | 2 |

Fonte: O autor.

A Tabela 6 também não apresenta nenhum resultado devido à plataforma *SpringerLink* favorecer a publicação de artigos na língua inglesa e, dessa maneira, nenhum artigo foi selecionado.

Tabela 6 - Resultados da busca na plataforma *SpringerLink* utilizando as palavras-chaves em português.

| **Palavra-chave** | **Resultados** | **Número de trabalhos selecionados** |
| --- | --- | --- |
| “Problema de Escalonamento de Projetos com Restrições de Recursos” | 0 | 0 |
| “Meta-heurística RCPSP” | 0 | 0 |
| Total | 0 | 0 |

Fonte: O autor.

As Tabelas 7 e 8 apresentam o resultado da análise das buscas realizadas na plataforma *IEEEXplore*.

A Tabela 7 possui 1.398 resultados. Apesar da quantia elevada de resultados, apenas dois preencheram satisfatoriamente as condições e, portanto, foram selecionados.

Tabela 7 - Resultados da busca na plataforma *IEEEXplore* utilizando as palavras-chaves em inglês.

| **Palavra-chave** | **Resultados** | **Número de trabalhos selecionados** |
| --- | --- | --- |
| *“Clustering Search”* | 885 | 1 |
| *“Resource Constrained Project Scheduling Problem”* | 367 | 0 |
| “RCPSP” | 146 | 1 |
| Total | 1.398 | 2 |

Fonte: O autor.

A Tabela 8 não apresenta nenhum resultado para artigos em língua portuguesa devido ao problema abordado anteriormente e, dessa maneira, nenhum artigo foi selecionado.

Tabela 8 - Resultados da busca na plataforma *IEEEXplore* utilizando as palavras-chaves em português.

| **Palavra-chave** | **Resultados** | **Número de trabalhos selecionados** |
| --- | --- | --- |
| “Problema de Escalonamento de Projetos com Restrições de Recursos” | 0 | 0 |
| “Meta-heurística RCPSP” | 0 | 0 |
| Total | 0 | 0 |

Fonte: O autor.

Após a análise de todos os resultados, 12 trabalhos relacionados ao tema tratado nesta pesquisa foram selecionados, sendo: 5 relacionados ao RCPSP, 6 relacionados ao CS e 1 de revisão da literatura. Além disso, outros 10 trabalhos foram identificados por meio da referência cruzada. Todos os trabalhos selecionados são apresentados em detalhes a seguir.

## 2.1. TRABALHOS RELACIONADOS AO RCPSP

Pritsker, Watters e Wolfe (1969) apresentam um algoritmo para a resolução da variante de múltiplos projetos do RCPSP que utiliza uma solução orientada a programação linear 0-1 (*zero-um)*. O algoritmo foi capaz de acomodar uma grande gama de condições, e se provava ser mais eficiente que outros algoritmos, tidos até 1968, em termos de número de variáveis e constantes.

Ikonomou, Galletly e Daniel (1998) apresentam um algoritmo para a resolução do RCPSP utilizando uma heurística adaptativa chamada *Tabu Search* (TS). Os autores submeteram o algoritmo a instâncias baseadas em problemas reais obtidos no projeto PAMIPS e obtiveram resultados demonstrando que para instâncias de pequena e média escala é possível obter resultados viáveis, porém para uma escala maior o TS não se mostrou capaz de produzir uma solução viável.

Lorenzoni, Albuquerque e Nascimento (2001) apresentam uma implementação da meta-heurística *Simulated Annealing* (SA) para a resolução de MRCPSP em conjunto com uma nova forma de representação do escalonamento em árvores. Os resultados obtidos através de 15 instâncias padrões apresentaram bons resultados na busca de soluções viáveis.

Mendes e Gonçalves (2004) apresentam uma abordagem que combina um GA com um esquema de geração de soluções para a resolução de RCPSP. Os testes realizados nas instâncias de conjuntos J30, J60 e J120 demonstraram que esta abordagem obteve excelentes resultados.

Zhang, Xu e Peng (2008) propõem um algoritmo baseado em GA para a resolução de RCPSP. Os autores apresentam uma nova permutação de esquema de codificação baseado em prioridade, combinando os esquemas de codificação por permutação com a codificação baseada em prioridade. Os testes foram realizados com instâncias disponíveis na PSPLIB e demonstram, quando comparados com outros algoritmos, que esse método é efetivo para a resolução do RCPSP.

Cravo (2009) apresenta uma implementação da meta-heurística *Greedy Randomized Adaptive Search Procedure* (GRASP) para a resolução do *Multi-Mode Resource-Constrained Project Scheduling Problem* (MRCPSP), uma extensão do RCPSP. Os resultados obtidos para a solução das instâncias apresentadas na PSPLIB demonstram que o GRASP consegue ser um método competitivo e, em um estudo de caso específico, apresentou bom desempenho quando comparado com resultados da literatura.

Muller (2009) apresenta um algoritmo de *Adaptive Large Neighborhood Search* (ALNS), unificando diversas técnicas dentro deste *framework*, para a resolução de RCPSP. Os resultados computacionais nas instâncias de conjuntos J30, J60 e J120 demonstram a competitividade com outros algoritmos estado-da-arte.

Muller (2011) novamente apresenta um algoritmo de ALNS, porém para a resolução de MRCPSP. O algoritmo proposto apresenta três novos *bounds* específicos para o multimodo. Seus testes foram executados em instâncias disponibilizadas na PSPLIB. Os resultados dos testes demonstraram que esse algoritmo foi competitivo, mas incapaz de superar os melhores algoritmos.

Azevedo, Pessoa e Torres (2012) apresentam um modelo de *Genetic Algorithm* (GA) para solução de um RCPSP, representado por um estudo de caso de uma empresa de petróleo e gás. Esse algoritmo utilizava uma representação cromossômica com decodificação por SGS Paralelo e de mutação com cruzamentos parametrizados seguidos de mutação aleatória. Como resultado os autores concluíram a impossibilidade de se manter o escopo do projeto original uma vez que este possuía uma expectativa contratual de 1 ano enquanto sua expectativa real girava em torno de 2 anos.

Crawford et al. (2015) propõe uma abordagem baseada no algoritmo *Artificial Bee Colony* (ABC) para a resolução do RCPSP. Os autores apresentam o modelo matemático e descrevem o comportamento do ABC, acreditando ser possível que este algoritmo seja efetivo em situações competitivas.

## 2.2. TRABALHOS RELACIONADOS AO CS

Chaves e Lorena (2010) apresentam um algoritmo para resolver o *Capacitated Centered Clustering Problem* (CCCP) utilizando a meta-heurística CS com um algoritmo SA para a geração de soluções. Os autores compararam os testes com um algoritmo SA sem o CS e com um *Variable Neighborhood Search* (VNS) proposto na literatura. O CS foi melhor que ambos em quase todas as instâncias, produzindo diversas novas melhores soluções.

De Oliveira, Mauri e Lorena (2012) apresentam um algoritmo para a resolução do *Berth Allocation Problem* (BAP) utilizando CS com o uso de SA para a geração de soluções. Os autores demonstram que o CS é uma alternativa útil para encontrar boas soluções através de testes comparativos com outros métodos (como AS + RA) propostos na literatura. Quando comparado com outras abordagens recentes na literatura, os resultados do algoritmo foram superiores.

Nagano, Da Silva e Lorena (2012) aplicam uma meta-heurística híbrida GA-CS em um algoritmo chamado de *Evolutionary Clustering Sea*rch (ECS) para solucionar o *no-wait flow shop problem with set-up times.* Os autores, através dos testes experimentais, chegam as conclusões de que o CS é superior ao melhor algoritmo conhecido na literatura (*Iterated Local Search* – ILS), que o ECS obteve uma performance melhor que o melhor método para instâncias de larga escala, e que novas melhores soluções foram obtidas pelo CS, totalizando 1441 instâncias.

Rabello et al. (2014) apresentam um algoritmo para a resolução do *Point-Feature Cartographic Label Placement* (PFCLIP) utilizando CS com SA para gerar soluções. Os experimentos computacionais realizados em instâncias de até 750 pontos resultaram em soluções ótimas e os testes realizados em instâncias de até 13.206 pontos obtiveram resultados melhores que outros encontrados na literatura.

Altoé et al. (2018) apresentam algoritmos para a resolução do *Flexible Job Shop Scheduling Problem* (FJSP) e para o *bi-objective* FJSP(BOFJSP). Esses problemas são extensões do *Job Shop Scheduling* (JSP). Os autores propõem uma solução baseada na meta-heurística híbrida CS, usando SA como método gerador de soluções. Os resultados mostram que o algoritmo foi capaz de encontrar a melhor solução para 6 dos 10 problemas clássicos considerados para o FJSP, e para o BOFJSP o algoritmo foi capaz de encontrar 94 soluções não-dominadas em um conjunto de 20 instâncias propostas.

González et al. (2019) propõem duas abordagens para resolver *Traffic Counting Location Problem* (TCLP), um algoritmo B&C e uma heurística CS. O algoritmo B&C foi capaz de obter resultados ótimos dentro de 7200 segundos, além de ser capaz de retornar soluções de alta qualidade nos casos onde não se obteve o resultado ótimo. O CS implementado foi comparado com um GA estado-da-arte e, após os testes, se demonstrou superior em todos os aspectos. A comparação do B&C com o CS demonstrou que o segundo foi capaz de encontrar soluções melhores em menos tempo para 11 instâncias, demonstrando que o mesmo tinha redefinido o estado-da-arte para este problema.

De Abreu et al. (2020) apresentam um modelo matemático e propõem um algoritmo para o *Network Sensor Location Problem* (NSLP) usando um algoritmo *Branch-and-Cut* (B&C) e a heurística CS para encontrar soluções para o problema. Para avaliar a eficácia do algoritmo, testes computacionais foram realizados em instâncias reais correspondendo a estados brasileiros. Os resultados de B&C demonstraram que as soluções viáveis foram encontradas em 22,22% dos casos e boas soluções foram encontradas para 47,01%, enquanto os resultados do CS demonstraram que soluções ótimas foram encontradas para 18,38% dos casos e boas soluções em 92,31%. Quando comparados, o CS encontrou soluções melhores que o B&C para 82,48% dos cenários.

## 2.3. O ESTADO DA ARTE DO RCPSP

De acordo com Kolisch e Hartmann (1999), até 1999 as estratégias mais adotadas na resolução do RCPSP consistiam em *Simulated Annealing*, *Tabu Search* e *Genetic Algorithm*. Porém, Brucker et al. (1999), ainda em 1999, demonstram em suas pesquisas que o foco das pesquisas se deslocou para heurísticas mais elaboradas, como *Truncated Branch-and-Bound*, heurísticas baseadas em *integer programming*, conceitos de *disjunctive arc*, análise local baseada em constantes, técnicas de amostragem e técnicas de busca local.

Porém, de acordo com Pellerin, Perrier e Berthaut (2020), pode-se observar uma mudança no cenário dos principais métodos usados baseando-se no fato de que os melhores métodos exatos são capazes de resolver instâncias de até 60 atividades. Desta maneira, o cenário das estratégias mais recentes se vê dominado por meta-heurísticas, híbridas ou não.

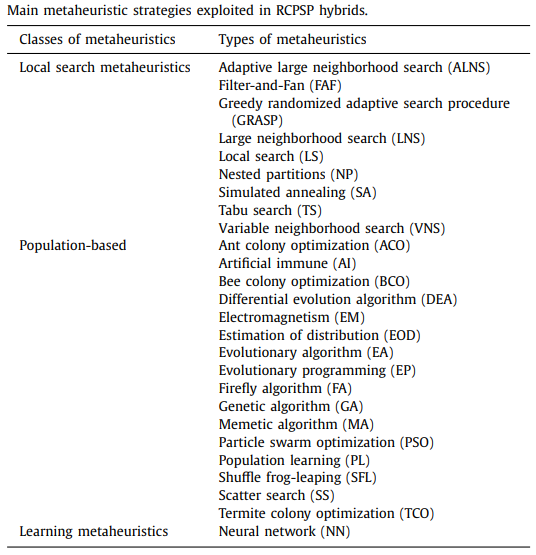
Ainda de acordo com Pellerin, Perrier e Berthaut (2020), meta-heurísticas se provaram excelentes para resolver o RCPSP. *Simulated Annealing*, *Tabu Search*, *Artificial Immune*, *Bee Colony*, *Genetic Algorithm*, *Particle Swarm*, *Scatter Search* e *Ant Colony*, estão, dentre outras, mais comumentes listadas em pesquisas recentes. 

Figura 4 - Principais estratégias híbridas para resolução de RCPSP.  
Fonte: Pellerin, Perrier e Berthaut (2020).

Porém, segundo Pellerin, Perrier e Berthaut (2020), em anos recentes, um número crescente de meta-heurísticas híbridas vêm sendo usadas para a resolução de problemas de otimização. Os autores informam que possuem a ciência de ao menos 27 tipos de meta-heurísticas híbridas que foram desenvolvidas para a resolução do RCPSP. Estes resultados são exibidos na Figura 4.

# METODOLOGIA

A metodologia deste trabalho se baseia na utilização da meta-heurística híbrida *Clustering Search* (CS) para a resolução do *Resource-Constrained Project Scheduling Problem* (RCPSP). Em princípio, a meta-heurística SA será utilizada como algoritmo de busca local para o CS.

## 3.1. *CLUSTERING SEARCH*

Segundo Oliveira, Chaves e Lorena (2013), CS implementa *clustering* para detectar áreas promissoras no espaço de busca. Um *cluster* é definido por um centro que é, geralmente, inicializado de maneira aleatória e, posteriormente, tende a progredir para pontos promissores no espaço de busca. O número de *clusters* pode ser fixo *a priori* ou determinado dinamicamente de acordo com amplitude da busca nas áreas que serão exploradas pela meta-heurística. Nos casos onde se é determinado dinamicamente, os *clusters* podem ser criados de uma maneira que todas as soluções candidatas são cobertas por pelo menos um *cluster*.

Ainda segundo Oliveira, Chaves e Lorena (2013), a sua cobertura é definida por uma distância métrica que computa a similaridade entre uma solução e o centro do *cluster*, e deve considerar a natureza do problema. Um *cluster* também possui densidade, chamada de volume, que é uma indicação do nível de atividade dentro do *cluster* respectivo.

O CS pode ser dividido conceitualmente em quatro partes independentes: a *Search Metaheuristic* (SM), o *Iterative Clustering Component* (IC), o *Analyser Module* (AM), e a *Local Search Heuristic* (LS), como ilustrado na Figura 5 (OLIVEIRA; CHAVES; LORENA, 2013).

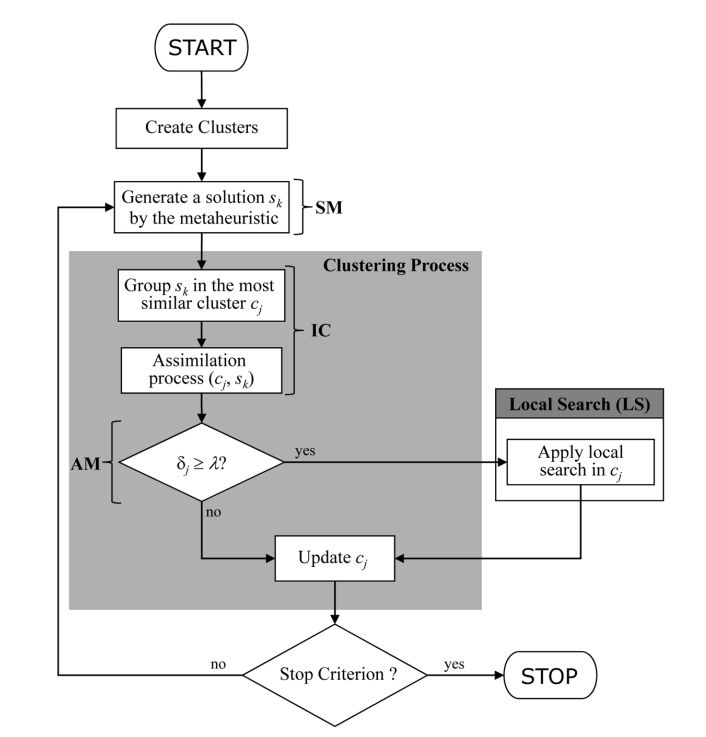


Figura 5 - Os componentes do CS.

Fonte: Oliveira, Chaves e Lorena (2013).

### 3.1.1. *Search Metaheuristic* (SM)

O componente SM pode ser implementado por qualquer algoritmo de otimização que consiga gerar soluções diversificadas no espaço de busca. Ele deve funcionar como um gerador completo de solução, explorando o espaço por manipulação de soluções, de acordo com suas estratégias específicas de busca (OLIVEIRA; CHAVES; LORENA, 2013).

### 3.1.2. *Iterative Clustering Component* (IC)

Este componente possui a finalidade de agrupar soluções similares em grupos, mantendo um centro de *cluster* representativo. Uma distância métrica deve ser definida, *a priori*, permitindo se mensurar a similaridade para o processo de *clustering* (OLIVEIRA; CHAVES; LORENA, 2013).

### 3.1.3. *Analyser Module* (AM)

O componente AM examina cada *cluster*, em intervalos regulares, indicando um provável *cluster* promissor. Ele conta o número de soluções geradas pelo SM e agrupadas dentro do *cluster*, e, sempre que o volume deste *cluster* alcança um limiar 𝜆, o mesmo precisará ser melhor investigado (OLIVEIRA; CHAVES; LORENA, 2013).

### 3.1.4. *Local Search Heuristic* (LS)

Por último, o componente LS é um módulo de busca interna que provê a exploração de uma suposta área promissora, enquadrada em um *cluster* (OLIVEIRA; CHAVES; LORENA, 2013).

## 3.2. EXPERIMENTOS COMPUTACIONAIS

Após a implementação dos algoritmos para a solução do RCPSP, serão realizados os testes seguindo as instâncias disponibilizadas na PSBLIB, nas categorias de j30, j60, j90 e j120. Os resultados serão comparados com os também disponíveis na PSBLIB e apresentados de maneira normalizada.

# RESULTADOS ESPERADOS

Como apresentado no decorrer deste trabalho, a meta-heurística CS obteve excelentes resultados quando aplicada a outros problemas de otimização encontrados na literatura. Dessa forma, espera-se que essa meta-heurística apresente bons resultados quando aplicada ao RCPSP, gerando soluções viáveis e de boa qualidade.

Por fim, espera-se comparar os resultados computacionais obtidos com a aplicação do CS para o RCPSP com a de outros trabalhos encontrados na literatura, de abordagem semelhante ao problema considerado nesta pesquisa.

# REFERÊNCIAS

ALTOÉ, W. A. S. et al. A Clustering Search Metaheuristic for the Bi-objective Flexible Job Shop Scheduling Problem. **2018 XLIV Latin American Computer Conference (CLEI)**, p. 158–166, 2018.

AZEVEDO, G. H. I. DE; PESSOA, A. A.; TORRES, C. R. R. Problemas de Escalonamento de Projetos com Restrição de Recursos: Um Estudo de Caso no Setor de Petróleo e Gás. **Pesquisa Operacional para o Desenvolvimento**, v. 4, n. 3, p. 288–303, 2012.

BRUCKER, P. et al. Resource-constrained project scheduling: Notation, classification, models, and methods. **European Journal of Operational Research**, v. 112, n. 1, p. 3–41, 1999.

CHAVES, A. A.; LORENA, L. A. N. Clustering search algorithm for the capacitated centered clustering problem. **Computers & Operations Research**, v. 37, n. 3, p. 552–558, 2010.

CRAVO, G. L. **Escalonamento de Projetos com Restrições de Recursos e Múltiplos Modos de Processamento**: Soluções Heurísticas e uma Aplicação à Programação de Manutenção Industrial. Dissertação (Pós-Graduação em Informática) — Programa de Pós-Graduação em Informática, Universidade Federal do Espírito Santo, Vitória, 2009.

CRAWFORD, B. et al. An Artificial Bee Colony Algorithm for the Resource Constrained Project Scheduling Problem. **Communications in Computer and Information Science**, p. 582–586, 2015.

DE ABREU, V. H. S. et al. Network sensor location problem with monitored lanes: Branch-and-cut and clustering search solution techniques. **Computers & Industrial Engineering**, v. 150, p. 106827, 2020.

DE OLIVEIRA, R. M.; MAURI, G. R.; LORENA, L. A. N. Clustering Search for the Berth Allocation Problem. **Expert Systems with Applications**, v. 39, n. 5, p. 5499–5505, 2012.

GONZÁLEZ, P. H. et al. New approaches for the traffic counting location problem. **Expert Systems with Applications**, v. 132, p. 189–198, 2019.

HERROELEN, W.; DEMEULEMEESTER, E.; DE REYCK, B. A Classification Scheme for Project Scheduling. **Project Scheduling. International Series in Operations Research & Management Science**, vol 14,, p. 1–26, 1999.

IKONOMOU, A.; GALLETLY, J. E.; DANIEL, R. C. Solving Resource-Constrained Project Scheduling Problems using Tabu Search. **Intelligent Systems for Manufacturing**, p. 311–322, 1998.

KOLISCH, R.; HARTMANN, S. Heuristic Algorithms for the Resource-Constrained Project Scheduling Problem: Classification and Computational Analysis. **Project Scheduling. International Series in Operations Research & Management Science**, vol 14,, p. 147–178, 1999.

LORENZONI, L. L.; ALBUQUERQUE, C.; NASCIMENTO, R. D. Têmpera Simulada Aplicada ao Problema de Escalonamento com Restrição de Recursos. A Pesquisa Operacional e as Cidades. Anais... In: **Simpósio Brasileiro de Pesquisa Operacional**. 2001.

MENDES, J. J. DE M.; GONÇALVES, J. F. Uma Meta-heurística para o Problema da Programação de Projectos com Recursos Limitados. In: **Jornadas Politécnicas de Engenharia: Mecânica, Automóvel, Organização e Gestão Industrial, Energia e Ambiente**. 2004.

MULLER, L. F. An Adaptive Large Neighborhood Search Algorithm for the Resource-constrained Project Scheduling Problem. **MIC 2009: The VIII Metaheuristics International Conference**, 2009.

MULLER, L. F. **A Multi-mode Resource-Constrained Project Scheduling Problem with Stochastic Nonrenewable Resource Consumption**. Lyngby, Denmark: DTU Management Engineering, 2011.

NAGANO, M. S.; DA SILVA, A. A.; LORENA, L. A. N. A new evolutionary clustering search for a no-wait flow shop problem with set-up times. **Engineering Applications of Artificial Intelligence**, v. 25, n. 6, p. 1114–1120, 2012.

OLIVEIRA, A. C. M.; CHAVES, A. A.; LORENA, L. A. N. Clustering search. **Pesquisa Operacional**, Rio de Janeiro, v. 33, n. 1, p. 105-121, 2013.

PSPLIB. **Project Scheduling Problem Library**. Disponível em: <https://www.om-db.wi.tum.de/psplib/main.html>. Acesso em: 6 jul. 2022.

PELLERIN, R.; PERRIER, N.; BERTHAUT, F. A survey of hybrid metaheuristics for the resource-constrained project scheduling problem. **European Journal of Operational Research**, v. 280, n. 2, p. 395–416, 2020.

PRITSKER, A. A. B.; WATTERS, L. J.; WOLFE, P. M. Multiproject Scheduling with Limited Resources: A Zero-One Programming Approach. **Management Science**, v. 16, n. 1, p. 93–108, 1969.

RABELLO, R. L. et al. A Clustering Search metaheuristic for the Point-Feature Cartographic Label Placement Problem. **European Journal of Operational Research**, v. 234, n. 3, p. 802–808, 2014.

ZHANG, H.; XU, H.; PENG, W. A Genetic Algorithm for Solving RCPSP. **2008 International Symposium on Computer Science and Computational Technology**, 2008.

# CRONOGRAMA

São apresentadas na Tabela 1 as atividades que deverão ser cumpridas para que se possa atingir os objetivos propostos neste trabalho.

Tabela 1 - Lista de atividades.

| **Lista de atividades** |
| --- |
| 1. Análise e estudo da meta-heurística CS. |
| 2. Implementação da meta-heurística CS. |
| 3. Realização de testes computacionais. |
| 4. Comparação dos resultados obtidos com os apresentados na literatura. |
| 5. Desenvolvimento da monografia final do TCC. |
| 6. Apresentação do TCC. |

Fonte: O autor.

O cronograma de execução é apresentado na Tabela 2, exibindo a distribuição das atividades listadas anteriormente e seu respectivo período de execução.

Tabela 2 - Cronograma de execução das atividades.

| Atividade | set/2022 | out/2022 | nov/2022 | dez/2022 | jan/2023 | fev/2023 |
| --- | --- | --- | --- | --- | --- | --- |
| 1 | X |  |  |  |  |  |
| 2 | X | X |  |  |  |  |
| 3 |  |  | X | X |  |  |
| 4 |  |  |  | X |  |  |
| 5 |  |  |  |  | X | X |
| 6 |  |  |  |  |  | X |

Fonte: O autor.