UMA ESTRATÉGIA LOCAL BRANCHING PARA O PROBLEMA DE SEQUENCIAMENTO EM PROJETOS COM RESTRIÇÃO EM RECURSOS E MÚTIPLOS MODOS DE EXECUÇÃO

Gustavo Alves Fernandes
Centro Federal de Educação Tecnológica de Minas Gerais - CEFETMG
Av. Amazonas, 7675 - Nova Gameleira- CEP 30510-000 - Belo Horizonte, MG, Brasil
gustavo.alfer@gmail.com

Sérgio Ricardo de Souza
Centro Federal de Educação Tecnológica de Minas Gerais - CEFETMG
sergio@dppg.cefetmg.br

Elisângela Martins de Sá
Centro Federal de Educação Tecnológica de Minas Gerais - CEFETMG
elisamamatematica_licenciatura@hotmail.com

RESUMO

O problema de Sequenciamento em Projetos com Restrição em Recursos e Múltiplos Modos de Execução (*Multi-mode Resource Constrained Project Scheduling Problem* - MRCPSP) consiste em sequenciar todas as atividades de um projeto, respeitando a ordem de precedência dessas atividades, que, uma vez iniciadas, não podem ser interrompidas. As atividades possuem várias combinações de modos de execução, na forma de diferentes combinações de tempo de processamento e consumo de recursos. O objetivo é minimizar o tempo de duração do projeto, encontrando tempos de início para cada atividade do projeto. Trata-se de um problema NP-Difícil e este artigo estuda sua solução pela estratégia *Local Branching* (LB). Foram realizados testes computacionais utilizando instâncias da biblioteca PSPLIB. Os melhores resultados conhecidos para as instâncias usadas nos testes são apresentados como referência para validar os resultados obtidos pelo LB.

PALAVRAS CHAVE. Sequenciamento em Projetos, Local Branching, Matheurísticas.

Tópicos (indique, em ordem de PRIORIDADE, o(s) tópicos(s) de seu artigo)

ABSTRACT

In this paper the nonpreemptive Multi-mode Resource Constrained Project Scheduling Problem (MRCPSP) is addressed. This problem aims to scheduling all jobs of a project subject to the precedence relations among these jobs, which, once started, can not be interrupted. The jobs have combinations of execution modes, which consist of different processing times and resource consumption. The goal is to minimize the project makespan and determine start times for each project activity. For solving this NP-Hard problem, this article presents an implementation of the Local Branching Method (LB). Computational tests were performed for classes of instances of the PSPLIB public library. The known results of these instances were used as a reference to validate the results obtained by the LB.

KEYWORDS. Project Scheduling. Local Branching. Matheuristics.

Paper topics (indicate in order of PRIORITY the paper topic(s))

https://proceedings.science/p/85311?lang=pt-br

1. Introdução

O problema de Sequenciamento em Projetos com Restrição em Recursos e Múltiplos Modos de Execução (*Multi-mode Resource Constrained Project Scheduling Problem* - MRCPSP) é caracterizado da seguinte maneira. Cada tarefa $j = 1, ..., n$ do projeto deve ser processada sem interrupção e sujeita à relação de precedência entre outras tarefas, isto é, cada tarefa j deve ser iniciada somente após todas as suas predecessoras $p \in \mathcal{P}_j$, sendo $\mathcal{P}_j$ o conjunto de tarefas que precedem j. Por convenção, $j = 1$ e $j = n$ são tarefas fictícias, tal que $j = 1$ e $j = n$ deverão ser, respectivamente, a primeira e a última tarefas no sequenciamento. Cada tarefa j é executada em um modo de execução $m \in M_j$, que estabelece o tempo de processamento d_{jm} e a demanda de recursos renováveis r^{ρ}_{jmk} e de recursos não-renováveis r^{υ}_{jmk} do tipo k, para $k = 1, ..., |K|$. Os recursos renováveis são restaurados em sua capacidade máxima R^{ρ}_k a cada período de tempo t. Já os recursos não-renováveis possuem uma capacidade disponível R^{υ}_k para toda a duração do projeto e não são restaurados. Todos os parâmetros são inteiros não-negativos. O objetivo para o MRCPSP é encontrar uma combinação factível dos modos de execução para as tarefas assim como determinar um sequenciamento factível sujeito às restrições de recursos e precedência, tal que a duração do projeto seja mínima.

O MRCPSP é um problema NP-Difícil, conforme [Blazewicz et al., 1983]. Kolisch e Drexl [1997] demonstraram que encontrar uma solução viável para o MRCPSP, com pelo menos dois tipos de recursos não-renováveis e no mínimo dois modos de execução, é um problema NP-Difícil.

Seja a variável de decisão denotada por x_{jmt}, sendo igual a 1 se a atividade j é executada no modo m e possui término no período t. Caso contrário, x_{jmt} é igual a 0. A formulação matemática recorrente na literatura e apresentada por Talbot [1982] para o MRCPSP é dada por:

$$(P) \quad \min \sum_{m=1}^{|M_n|} \sum_{t=EFT_n}^{LFT_n} t x_{nmt} \tag{1}$$

s.a.

$$\sum_{m=1}^{|M_j|} \sum_{t=EFT_j}^{LFT_j} x_{jmt} = 1, j = 1, ..., n \tag{2}$$

$$\sum_{m=1}^{|M_i|} \sum_{t=EFT_i}^{LFT_i} t x_{imt} \leq \sum_{m=1}^{|M_j|} \sum_{t=EFT_j}^{LFT_j} (t - d_{jm}) x_{jmt}, \forall (i,j) \in A \tag{3}$$

$$\sum_{j=1}^{n} \sum_{m=1}^{|M_j|} \sum_{s=\max\{t, EFT_j\}}^{\min\{t+d_{jm}-1, LFT_j\}} r^{\rho}_{jmk} x_{jms} \leq R^{\rho}_k, k = 1, ..., R^{\rho}; t = 1, ..., T \tag{4}$$

$$\sum_{j=1}^{n} \sum_{m=1}^{|M_j|} \sum_{t=EFT_j}^{LFT_j} r^{\nu}_{jmk} x_{jmt} \leq R^{\nu}_k, k = 1, ..., R^{\nu} \tag{5}$$

$$x_{jmt} \in \{0,1\} (j = 1, ..., n; m = 1, ...|M_j|; t = EFT_j, ..., LFT_j) \tag{6}$$

Nessa formulação, a expressão (1) apresenta o objetivo de minimizar o término da última atividade; a expressão (2) representa a execução de um modo por atividade, assim como a atribuição de um único tempo de término; a expressão (3) apresenta a relação de precedência entre atividades; a expressão (4) assegura que a disponibilidade de recursos renováveis por período não seja violada; a expressão (5) corresponde à restrição de recursos não renováveis; finalmente, a expressão (6) mostra a característica binária da variável de decisão x_{jmt}. Os valores EST_j, LST_j, EFT_j e LFT_j são obtidos pelo método do Caminho Crítico, conforme Kelley Jr e Walker [1959], após a atribuição do modo com menor duração, para cada atividade.

https://proceedings.science/p/85311?lang=pt-br

Neste artigo é apresentada uma variação do *Local Branching* (LB) proposta por Fischetti e Lodi [2003]. A estratégia LB se assemelha às conhecidas metaheurísticas de busca local, mas, diferentemente destas, as vizinhanças são obtidas por meio de desigualdades lineares conhecidas como *local branching cuts*. Essas desigualdades são inseridas no *Mixed-Integer Programming* (MIP) para garantir otimalidade local. Algumas abordagens recentes baseadas em LB para a solução de problemas de otimização inteira são, dentre outros, Acuna-Agost et al. [2011] para sequenciamento ferroviário, Rodríguez-Martín e Salazar-González [2010] para problemas de fluxo em rede, Li et al. [2012] para planejamento de cargas, e Zhu et al. [2006] para o MRCPSP.

A proposta do presente artigo constitui da resolução de um MIP para encontrar uma solução inicial $\bar{x}$, baseada no menor tempo de duração das tarefas. Essa solução é, então, utilizada pela estratégia LB, que, para cada vizinhança gerada em torno de $\bar{x}$, busca pela otimalidade local. Para cada nova solução ótima local x^* encontrada, a região em torno da solução $\bar{x}$ é excluída, e, então, a busca recomeça em torno de x^*. Para apresentar a adaptação aqui proposta, inicialmente uma revisão literária é apresentada na Seção 2. Em seguida, a adaptação da estratégia LB para o MRCPSP é discutida na Seção 3. Os resultados computacionais encontrados são discutidos na Seção 4. A Seção 5 apresenta conclusões sobre a proposta apresentada neste trabalho.

2. Revisão de Literatura

Nessa seção são apresentadas as abordagens exatas para resolução do MRCPSP e uma discussão das metaheurísticas recentes destacadas pela literatura.

2.1. Soluções usando Métodos Exatos

Patterson et al. [1989] propuseram um algoritmo *Branch-and-Bound* (B&B), conhecido como Árvore de Precedência, cujo sequenciamento é contruído por meio de uma árvore. Em seguida, Sprecher [1994]; Sprecher e Drexl [1998] aprimoraram esta abordagem, incluindo outros critérios de *branching*. Já Sprecher et al. [1997] apresentaram uma nova abordagem B&B, introduzindo o conceito de Alternativas de Modo e Atraso (*Mode and Delay Alternatives*). Essa abordagem também constrói um sequenciamento sobre uma árvore. Posteriormente, utilizando o conceito de Alternativas de Modo e Atraso, Hartmann e Drexl [1998] criaram o conceito de Alternativas de Modo e Extensão (*Mode and Extension Alternatives*). Essas três abordagens B&B podem ser encontradas, detalhadamente, com comparações entre si, em Hartmann e Drexl [1998].

A última abordagem exata para resolução do MRCPSP com minimização do *makespan*, conhecida pelos autores do presente artigo, é devida a Zhu et al. [2006], no qual é apresentado um algoritmo *Branch-and-Cut* para o MRCPSP. Os autores afirmam que a abordagem que utilizaram consome menor tempo de execução na determinação de sequenciamentos ótimos em relação ao trabalho de Sprecher e Drexl [1998]. Entretanto, verificando a diferença no esforço computacional de ambos os trabalhos, o último apresenta uma enumeração mais rápida. Como destaque do trabalho de Zhu et al. [2006], os autores encontraram valores ótimos para 506 de 552 instâncias da maior classe (J30) da biblioteca pública PSPLIB [Kolisch e Sprecher, 1997], mantendo um tempo de 3600 segundos como critério de parada para o B&C.

2.2. Soluções usando Metaheurísticas

Bouleimen e Lecocq [2003] apresentam uma implementação da metaheurística *Simulated Annealing* (SA) como busca local em dois estágios. No primeiro estágio, a vizinhança é baseada somente em modos de execução, e, nos casos em que é possível melhorar o *makespan*, o segundo estágio é executado somente em vizinhanças baseadas em tarefas. Zhang et al. [2006] discutem a metaheurística evolucionaria *Particle Swarm Optimization* (PSO) com a adição de um procedimento para reparar soluções inviáveis. A metaheuristica PSO também foi usada em Jarboui et al. [2008] para atribuir modos de execução para as tarefas, mas, neste caso, um procedimento de busca local determina o sequencianento das tarefas. Em Mika et al. [2008] foi proposta uma implementação da metaheurística *Tabu Search* (TS) para o MRCPSP, considerando tempos de *setup*. Uma abordagem baseada na metaheurística Evolução Diferencial (DE) é apresentada por Damak et al. [2009], no

https://proceedings.science/p/85311?lang=pt-br

qual os autores comparam os resultados encontrados em relação aos resultados de Jarboui et al. [2008] e Bouleimen e Lecocq [2003]. Uma abordagem via Artificial Immune System (AIS) foi utilizada por Van Peteghem e Vanhoucke [2009], que utiliza o princípio da seleção clonal para explorar várias regiões do espaço de busca.

Lova et al. [2009] apresenta uma implementação de um Algoritmo Genético (AG) com a heurística *Multi-mode Forward/Backward Improvement* (MFBI). A mesma técnica de hibridização é usada em Van Peteghem e Vanhoucke [2010], sendo que, neste caso, os autores destacam a contribuição da MFBI para o GA. A adaptação do GA de Van Peteghem e Vanhoucke [2010] recebe destaque como o melhor trabalho na resolução de grandes instâncias de acordo com o estudo de Van Peteghem e Vanhoucke [2014]. O trabalho de Okada et al. [2014] também apresenta um GA hibridizado com a MFBI e procedimentos de busca local baseados no Caminho Crítico. O método Scatter-search foi implementado por Van Peteghem e Vanhoucke [2011], que também utiliza uma heurística na tentativa de construir soluções viáveis. Um procedimento baseado no Algoritmo de Colonia de Formigas foi usado por Chiang et al. [2008]; Li e Zhang [2013], sendo que Li e Zhang [2013] usa Regras de Prioridade para guiar as "formigas" na escolha de tarefas enquanto programa o sequenciamento. Em Muller [2011] a metaheurística *Adaptive Large Neighborhood Search* (ALNS) foi utilizada. Essa metaheurística utiliza várias estruturas de vizinhanças, e, durante a busca, as vizinhanças competem umas com as outras para melhorar a solução corrente. Wauters et al. [2011] apresenta uma abordagem multi-agente, no qual heurísticas competem para apresentar o melhor sequenciamento. Em Fernandes e Souza [2017], os autores apresentam uma estratégia hibridizada entre as metaheurísticas AIS e ALNS, obtendo resultados significativos para as instâncias da PSPLIB.

Em Van Peteghem e Vanhoucke [2014] é apresentado uma comparação entre algumas das metaheurísticas citadas para resolução do MRCPSP sobre a biblioteca publica PSPLIB, ao mesmo tempo em que propõe uma nova biblioteca de instâncias, denominada MMLIB, contendo novas classes de problemas.

3. Estratégia Local Branching

Seja o um problema de Programação Inteira Mista (*Mixed-Integer Program*-MIP) (PI):

$$
\begin{aligned}
(PI) \quad \min \quad & c'x && (7)\\
\text{s.a.} \quad & \\
& Ax \geq b && (8)\\
& x_j \geq 0, j \in \mathbb{N} && (9)\\
& x_j \in \{0,1\}, j \in \mathbb{I} && (10)
\end{aligned}
$$

tendo, associada, uma solução viável de referência $\bar{x}$ de (PI).

Fischetti e Lodi [2003] apresentaram a heurística *Local Branching* (LB) para para a solução de MIPs baseado na fixação de variáveis e na tentativa de garantir a otimalidade na vizinhança de $\bar{x}$. Eles discutem a introdução da restrição de ramificação *branching*, com o objetivo de definir uma vizinhança para uma solução viável de referência, e, também, a possibilidade de gerar pseudo-cortes, para excluir partes do espaço de soluções.

A estratégia LB fixa algumas variáveis, isto é, exige que uma certa quantidade de variáveis assuma os mesmos valores que a solução de referência, sem fixar as outras variáveis. Sendo assim, tem-se a adição de uma restrição linear ao problema original. A restrição adicionada $\delta(x, \bar{x}) \leq k$ define, de acordo com Fischetti e Lodi [2003], uma vizinhança k-ótima N^k de $\bar{x}$. Seja $\bar{S} = \{j : \bar{x_j} = 1\}$. Então:

$$
\delta(x, \bar{x}) = \sum_{j \in \bar{S}} (1 - x_j) + \sum_{j \setminus \bar{S}} x_j \qquad (11)
$$

https://proceedings.science/p/85311?lang=pt-br

A vizinhança N^k, para algum k inteiro, é então definida como:

$$N^k(\bar{x}) = \{x \texttt{ obtida de } (PI) : \delta(x, \bar{x}) \leq k\}$$

Como discutido em Fischetti e Lodi [2003], a expressão (11) pode ser usada com uma regra de *branching* para (PI). Dado $\bar{x}$, o espaço de soluções associado ao nó atual em algum esquema de enumeração, pode ser particionado pelas regras:

$$\delta(x, \bar{x}) \leq k \textit{ (left branch) } \textbf{ou } \delta(x, \bar{x}) \geq k + 1 \textit{ (right branch)}$$

O parâmetro k deve ser escolhido de forma apropriada a garantir um subproblema mais fácil de resolver após a adição do corte *left branch*, isto é, a vizinhança $N^k(\bar{x})$ deve ser suficientemente pequena para ser otimizada em tempo aceitável, bem como grande o suficiente na tentativa de encontrar soluções melhores que $\bar{x}$. Essa abordagem do LB descrita até o momento, corresponde a um método exato. A conversão do LB para uma heurística pode ser descrita ao adicionar o parâmetro `node_time_limit`, que é usado para incorporar um tempo limite para a exploração do nó *left branch*. Esse parâmetro define o tempo limite para resolver cada subproblema gerado pelo corte *left branch*.

3.1. Construção de uma Solução Inicial

Uma solução viável para o MRCPSP consiste na seleção de modos de execução, tal que o consumo de recursos não-renováveis deve ser respeitado de acordo com a disponibilidade desses recursos. Seja $x_{jm} = 1$ se a tarefa j é executada no modo m; caso contrário $x_{jm} = 0$. O seguinte problema binário é utilizado para obter uma solução inicial factível.

$$(P') \quad \min \quad \sum_{j=1}^{n} d_{jm} x_{jm} \tag{12}$$

s.a.

$$\sum_{m=1}^{|M_j|} x_{jm} = 1, j = 1, ..., n \tag{13}$$

$$\sum_{j=1}^{n} \sum_{m=1}^{|M_j|} r^{\nu}_{jmk} x_{jm} \leq R^{\nu}_k, k = 1, ..., R^{\nu} \tag{14}$$

$$x_{jm} \in \{0, 1\} (j = 1, ..., n; m = 1, ...|M_j|) \tag{15}$$

De acordo com essa formução, a expressão (12) procura pelos modos de execução que apresentam os menores tempo de duração. A expressão (13) garante um modo de execução por tarefa e a expressão (14) garante que os modos de execução respeitem a disponibilidade dos recursos não-renováveis. Encontrar uma uma solução para (P') é um problema NP-Difícil, pois esse problema corresponde ao conhecido Problema da Mochila Multidimensional 0-1. Entretanto, *solvers* atuais de programação inteira podem resolver esse problema de forma ótima em tempo computacional surpreendentemente rápido. Essa mesma estratégia foi usada eficientemente por Santos et al. [2014].

Após definir os modos de execução, basta então sequenciar as tarefas, respeitando as restrições de precedência e consumo de recursos renováveis. O Esquema de Geração de Sequenciamento Serial (*Serial Scheduling Scheme - SSS*), descrito em Kolisch [1996], foi utilizado para definir os tempos de início (e fim) das tarefas. O SSS precisa de uma lista de tarefas L_J para criar o sequenciamento, respeitando a ordem em que as tarefas aparecem na lista, já que esse esquema de sequenciamento atribui o instante mais cedo possível para uma determinada tarefa. Sendo assim, com o mesmo conjunto de tarefas, porém considerando as tarefas em ordem diversa e respeitando a

https://proceedings.science/p/85311?lang=pt-br

restrição de precedência, é possível construir mais de um sequenciamento distinto [Demeulemeester, 2002].

Seja então $\bar{x}$ a solução construída. A codificação $S(\bar{x}) = (L_J, L_M, L_T)$, representa a solução $\bar{x}$ em listas, ou seja, L_J para as tarefas, L_M para modos de execução e L_T para os tempos de término. Logo, essa codificação em listas indica que durante o processsamento do SSS para uma tarefa j na posição i de L_J, seu modo de execução na posição i de L_M é verificado para atribuir um tempo de término na posição i de L_T.

3.2. Adaptação Local Branching para o MRCPSP

O esquema de busca local utilizado neste trabalho é uma adaptação da heurística *Local Branching* descrita anteriormente, utlizando, como solução inicial de referência $\bar{x}$ de (P'), para $\bar{x}$ ótima obtida como citado no item de seção anterior. A expressão (11) é reescrita para o MRCPSP da seguinte maneira:

$$\delta(x, \bar{x}) = \sum_{j=1}^{n} \sum_{m=1}^{|M_j|} \sum_{t=EFT_j}^{LFT_j} (x_{jmt}(1 - \bar{x}_{jmt}) + \bar{x}_{jmt}(1 - x_{jmt})) \tag{16}$$

Sendo assim, a busca em uma vizinhança $N^k(\bar{x})$ pode ser definida adicionando o corte $k_{min} \leq \delta(x, \bar{x}) \leq k$; já o novo conjunto de soluções $X = X \backslash N^k(\bar{x})$ é representado eliminando-se $k_{min} \leq \delta(x, \bar{x}) \leq k$ e adicionando o corte $\delta(x, \bar{x}) \geq k + 1$. Seja $\bar{x}$ a solução factível inicial construída e X o conjunto de todas as soluções factíveis de (P). O Algoritmo 1 apresenta a adaptação da heurística LB para a solução do MRCPSP.

Algoritmo 1: Local Branching - (LB)

Entrada: $\bar{x}$, k_0, `node_time_limit`

```
1  k = k_0 ;
2  k_min = 1 ;
3  enquanto critério de parada não satisfeito faça
4      Resolver MIP_SOLVER(P_1, X ∩ N^k(x̄), node_time_limit) ;
5      se solução x* ∈ X ∩ N^k(x̄) é ótima ou viável && é melhor que x̄ então
6          se x* é solução ótima então
7              X = X \ N^k(x̄) ;
8              inicializar k e k_min ;
9          fim
10         x̄ = x* ;
11     senão
12         se solução x* ∈ X ∩ N^k(x̄) é inviável || x* é ótima e não é melhor que x̄
             então
13             k_min = k ;
14             k = k + max(1, ⌈k/2⌉) ;
15         senão
16             k = max(1, ⌈k/2⌉) ;
17             se k ≤ k_min então
18                 perturbarSolucao(x̄) ;
19                 reinicia X ;
20             fim
21         fim
22     fim
23 fim
```

O algoritmo começa com uma solução viável $\bar{x}$ de (P). Em seguida, o nó correspondente ao *left branch* é adicionado, criando um subproblema resolvido por um MIP *solver*. Se uma solução

https://proceedings.science/p/85311?lang=pt-br

ótima e de melhora x^* é encontrada, então o processo realiza um *backtrack* ao nó anterior e o *left branch* é substituído pelo *right branch* em (P). A partir desse instante, a solução de referência é atualizada para x^* e um novo *left branch* é adicionado. Se o tempo limite é alcançado e x^* não é uma solução melhor que $\bar{x}$, a cardinalidade da vizinhança é diminuída, na tentativa de garantir otimalidade local em uma vizinhança menor gerada pelo novo *left branch* para k reduzido. Caso a solução x^* seja ótima e sem apresentar melhora em relação a $\bar{x}$, ou nenhuma solução viável é encontrada, a vizinhança $N^k(\bar{x})$ se torna maior, devido ao aumento de k e, por consequência, espera-se realizar uma busca local em uma vizinhança aumentada, sem a adição de um *right branch*, pois não há critérios matemáticos que permitem excluir a região criada pelo *left branch*. Entretanto, um novo *left branch* será criado para k aumentado.

Quando a cardinalidade da vizinhança é diminuída por meio de k, há a possibilidade da heurística LB entrar em um ciclo, isto é, quando k é igual (ou menor) a k_{min}. Caso isso aconteça, é necessário um procedimento para diversificação na heurística LB. Esse procedimento é destacado por Fischetti e Lodi [2003] como *strong diversification*. Esse procedimento de perturbação é necessário para retirar a LB do ótimo local, gerando uma nova solução de referência a partir da codificação em listas $S(\bar{x})$ discutida na Seção 3.1. O procedimento para diversificação utilizado neste trabalho inclui duas estruturas de vizinhanças discutidas em [Fernandes e Souza, 2017]:

(i) `NJ-SH`: Uma tarefa j é selecionada na posição i de L_J. Então, seu predecessor $p \in \mathcal{P}_j$ e seu sucessor mais próximos $s \in \mathcal{S}_j$ são encontrados, respectivamente, nas posições i_p e i_s da lista L_J, para determinar o quão longe j pode ser movida para frente ou para trás em L_J. Após, j é colocada na nova posição $\bar{i}$, tal que $\bar{i} \neq i$ e $i_p < \bar{i} < i_s$;

(ii) `NM-CM`: Uma tarefa j é selecionada na posição i de L_J. Em seguida, um novo modo de execução $\bar{m} \neq m$, sendo m o modo corrente, é selecionado para j na posição i de L_M, desde que a solução permaneça viável em relação aos recursos não-renováveis de acordo com o valor $ERR(s)$ dado por:

$$ERR(s) = \sum_{j=1}^{n} \max \left\{ 0, \sum_{k=1}^{K^\nu} r_{jmk} - R_k^\nu \right\}$$

Para realizar a perturbação, a melhor solução $\bar{x}$ encontrada pela heurística LB é codificada por meio de $S(\bar{x})$. A lista L_J é ordenada de forma crescente pelos tempos de término, de acordo a lista L_T. Então, as listas L_J e L_M são modificadas por meio das estruturas `NJ-SH` e `NM-CM`, aleatoriamente. O número de movimentos para cada estrutura de vizinhaça é $n - 2$, para n o número de tarefas no projeto. Por fim, o sequenciador SSS cria o novo sequenciamento e é realizada a decodificação das listas L_J, L_M e L_T para a nova solução de referência $\bar{x}$.

4. Resultados Computacionais

A abordagem deste trabalho, baseada na estratégia *Local Branching*, foi implementada em linguagem C++, utilizando o *solver* CPLEX, e compilador GNU G++ 5.4.0. Os testes foram realizados em um computador com processador Intel Core i3-3217U; 1.80GHz; 4GB RAM; e sistema operacional Ubuntu Linux 64 bits. Para os testes, foram utilizadas as classes J20 e J30 da biblioteca pública PSPLIB. Para cada subproblema, foi considerado um tempo de 180 segundos de resolução, isto é, `node_time_limit = 180`. Já o valor inicial para k foi definido como 15. Esses valores estão de acordo com o observado em Fischetti e Lodi [2003]. O critério de parada da LB foi de 1800 segundos para a classe J20 e 3600 segundos para a classe J30. Esses tempos foram escolhidos por critério de comparação com o trabalho de Zhu et al. [2006], que é referência na literatura. Seja OPT o número de ótimos encontrados; GAP_{opt} o desvio médio para os ótimos encontrados; GAP_{cpm} o desvio médio para o *lower bound* Caminho Crítico e GAP_{cpm}^{const} o desvio médio para o *lower bound* Caminho Crítico da solução inicial construída. Os valores de GAP mostram o quão longe, em média, as soluções estão de um outro valor de referência, isto é, para este trabalho, os

https://proceedings.science/p/85311?lang=pt-br

valores de referência são os valores ótimos ou os valores do *lower bound*. A Tabela 1 apresenta estes resultados.

Tabela 1: Resultados para a bibliotea pública PSPLIB

classe	OPT	GAP_{opt} **(%)**	GAP_{cpm} **(%)**	GAP_{cpm}^{const} **(%)**
J20 (554 instâncias)	498	0.5230	17.8684	38.1709
J30 (552 instâncias)	445	1.3689	20.2880	43.3982

De acordo com a Tabela 1, percebe-se que o valor GAP_{opt} é menor que 1% para a classe J20 e de 1.4% para a classe J30, isto é, quando a solução ótima (para a classe J20) ou a melhor solução conhecida (para a classe J30) não foi encontrada, pelo GAP_{opt} percebe-se que a solução encontrada não está distante da solução ótima (ou melhor solução conhecida). Destaca-se também que 89% das soluções ótimas foram encontradas para a classe J20 e 80% das melhores soluções conhecidas para a classe J30 foram encontradas no tempo limite adotado.

Para discussão dos resultados obtidos pela adaptação LB apresentada no Algoritmo 1, os resultados de Zhu et al. [2006] foram utilizados como referência. Esses autores também utilizaram as classes J20 e J30 com o mesmo tempo para critério de parada do B&C em um computador cuja frequência de processamento também é de 1.8GHz. A Tabela 2 apresentada os resultados.

Tabela 2: Resultados para as classe J20 e J30 da bibliotea pública PSPLIB

	J20		**J30**	
Trabalho	OPT	GAP_{opt}	OPT	GAP_{cpm}
Zhu et al. [2006]	554	0	506	0.09
LB adaptada	498	0.52	445	1.3689

Há uma diferença entre o B&C de Zhu et al. [2006] para a adaptação LB deste trabalho. A justificativa para essa diferença é que as desigualdades geradas por Zhu et al. [2006] são válidas para o MRCPSP, o que apresenta melhores limites para resolução do problema (P) do que as desigualdade do tipo (16).

5. Conclusão

Este trabalho apresenta a implementação da estratégia *Local Branching* para a solução do problema de Sequenciamento em Projetos com Restrição em Recursos e Múltiplos Modos de Execução (*Multi-mode Resource Constrained Project Scheduling Problem* - MRCPSP). Foram discutidas as adaptações das restrições geradas pela LB padrão para o problema em discussão, assim como foi apresentada e discutida a implementação do algoritmo LB para solucionar o MRCPSP. As duas maiores classes de instâncias da bilioteca pública PSPLIB foram solucionadas. Os resultados foram comparados com uma abordagem *Branch-and-Cut* que utiliza geração de desigualdades válidas para o MRCPSP. Há uma diferença a favor do *Branch-and-Cut*, devido à qualidade dos limites gerados pelas desigualdades desse método. Percebe-se que a adaptação da LB pode ser aprimorada para gerar outros cortes além dos pseudo-cortes padrões. Além disso, uma abordagem de perturbação usando busca local pode ser inserida na LB para a fase de *strong diversification*, quando ocorre a ciclagem, o que também pode implicar em melhoria de desempenho.

Agradecimentos

Os autores agradecem a CAPES, FAPEMIG, CNPq e ao CEFET-MG pelo suporte durante o desenvolvimento deste trabalho.

Referências

Acuna-Agost, R., Michelon, P., Feillet, D., e Gueye, S. (2011). A mip-based local search method for the railway rescheduling problem. *Networks*, 57(1):69–86.

https://proceedings.science/p/85311?lang=pt-br

Blazewicz, J., Lenstra, J. K., e Kan, A. R. (1983). Scheduling subject to resource constraints: classification and complexity. *Discrete Applied Mathematics*, 5(1):11–24.

Bouleimen, K. e Lecocq, H. (2003). A new efficient simulated annealing algorithm for the resource-constrained project scheduling problem and its multiple mode version. *European Journal of Operational Research*, 149(2):268–281.

Chiang, C.-W., Huang, Y.-Q., e Wang, W.-Y. (2008). Ant colony optimization with parameter adaptation for multi-mode resource-constrained project scheduling. *Journal of Intelligent & Fuzzy Systems: Applications in Engineering and Technology*, 19(4, 5):345–358.

Damak, N., Jarboui, B., Siarry, P., e Loukil, T. (2009). Differential evolution for solving multi-mode resource-constrained project scheduling problems. *Computers & Operations Research*, 36 (9):2653–2659.

Demeulemeester, E. L. (2002). *Project scheduling: a research handbook*, volume 102. Springer.

Fernandes, G. A. e Souza, S. R. (2017). A CLONALG inspired algorithm with adaptive large neighborhood for the multi-mode resource-constrained project scheduling problem. In *Proceedings of the 12th edition of the Metaheuristics International Conference (MIC 2017), Barcelona, July 4-7*, p. 561–570. MIC/MAEB.

Fischetti, M. e Lodi, A. (2003). Local branching. *Mathematical programming*, 98(1-3):23–47.

Hartmann, S. e Drexl, A. (1998). Project scheduling with multiple modes: a comparison of exact algorithms. *Networks*, 32(4):283–297.

Jarboui, B. et al. (2008). A combinatorial particle swarm optimization for solving multi-mode resource-constrained project scheduling problems. *Applied Mathematics and Computation*, 195 (1):299–308.

Kelley Jr, J. E. e Walker, M. R. (1959). Critical-path planning and scheduling. In *Papers presented at the December 1-3, 1959, Eastern Joint IRE-AIEE-ACM computer conference*, p. 160–173. ACM.

Kolisch, R. (1996). Serial and parallel resource-constrained project scheduling methods revisited: Theory and computation. *European Journal of Operational Research*, 90(2):320–333.

Kolisch, R. e Drexl, A. (1997). Local search for nonpreemptive multi-mode resource-constrained project scheduling. *IIE transactions*, 29(11):987–999.

Kolisch, R. e Sprecher, A. (1997). PSPLIB - a project scheduling problem library. *European Journal of Operational Research*, 96(1):205–216.

Li, H. e Zhang, H. (2013). Ant colony optimization-based multi-mode scheduling under renewable and nonrenewable resource constraints. *Automation in Construction*, 35:431–438.

Li, Z., Bookbinder, J. H., e Elhedhli, S. (2012). Optimal shipment decisions for an airfreight forwarder: Formulation and solution methods. *Transportation Research Part C: Emerging Technologies*, 21(1):17–30.

Lova, A. et al. (2009). An efficient hybrid genetic algorithm for scheduling projects with resource constraints and multiple execution modes. *International Journal of Production Economics*, 117 (2):302–316.

https://proceedings.science/p/85311?lang=pt-br

Mika, M., Waligóra, G., e Węglarz, J. (2008). Tabu search for multi-mode resource-constrained project scheduling with schedule-dependent setup times. *European Journal of Operational Research*, 187(3):1238–1250.

Muller, L. F. (2011). An adaptive large neighborhood search algorithm for the multi-mode RCPSP. Report 3, DTU Management.

Okada, I., Takahashi, K., Zhang, W., Zhang, X., Yang, H., e Fujimura, S. (2014). A genetic algorithm with local search using activity list characteristics for solving resource-constrained project scheduling problem with multiple modes. *IEEJ Transactions on Electrical and Electronic Engineering*, 9(2):190–199.

Patterson, J., Slowinski, R., Talbot, F., e Weglarz, J. (1989). An algorithm for a general class of precedence and resource constrained scheduling problems. In Slowinski, R. e Weglarz, J., editors, *Advances in Project Scheduling*, p. 3–28. Elsevier.

Rodríguez-Martín, I. e Salazar-González, J. J. (2010). A local branching heuristic for the capacitated fixed-charge network design problem. *Computers & Operations Research*, 37(3):575–581.

Santos, H. G., Soares, J., e Toffolo, T. (2014). Hybrid local search for the multi-mode resource-constrained multi-project scheduling problem. In *Proceedings of the 10th International Conference on the Practice and Theory of Automated Timetabling (PATAT'14)*, p. 397–407.

Sprecher, A. (1994). *Resource-Constrained Project Scheduling: Exact Methods for the Multi-Mode Case*, volume 409 of *Lecture Notes in Economics and Mathematics Systems*. Springer.

Sprecher, A. e Drexl, A. (1998). Multi-mode resource-constrained project scheduling by a simple, general and powerful sequencing algorithm. *European Journal of Operational Research*, 107(2): 431–450.

Sprecher, A., Hartmann, S., e Drexl, A. (1997). An exact algorithm for project scheduling with multiple modes. *Operations-Research-Spektrum*, 19(3):195–203.

Talbot, F. B. (1982). Resource-constrained project scheduling with time-resource tradeoffs: The nonpreemptive case. *Management Science*, 28(10):1197–1210.

Van Peteghem, V. e Vanhoucke, M. (2009). An artificial immune system for the multi-mode resource-constrained project scheduling problem. In *Evolutionary computation in combinatorial optimization*, p. 85–96. Springer.

Van Peteghem, V. e Vanhoucke, M. (2010). A genetic algorithm for the preemptive and non-preemptive multi-mode resource-constrained project scheduling problem. *European Journal of Operational Research*, 201(2):409–418.

Van Peteghem, V. e Vanhoucke, M. (2011). Using resource scarceness characteristics to solve the multi-mode resource-constrained project scheduling problem. *Journal of Heuristics*, 17(6):705–728.

Van Peteghem, V. e Vanhoucke, M. (2014). An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances. *European Journal of Operational Research*, 235(1):62–72.

Wauters, T. et al. (2011). Learning agents for the multi-mode project scheduling problem. *Journal of the operational research society*, 62(2):281–290.

https://proceedings.science/p/85311?lang=pt-br

Zhang, H., Tam, C., e Li, H. (2006). Multimode project scheduling based on particle swarm optimization. *Computer-Aided Civil and Infrastructure Engineering*, 21(2):93–103.

Zhu, G., Bard, J. F., e Yu, G. (2006). A branch-and-cut procedure for the multimode resource-constrained project-scheduling problem. *INFORMS Journal on Computing*, 18(3):377–390.

https://proceedings.science/p/85311?lang=pt-br