

Integer Linear Programming Formulations for the RCPSP with Practical Constraints

L. V. Melo and T. A. Queiroz

Abstract— The project scheduling problem is essential both in the theoretical part, as in the field of operational research, and practice, with the project management in corporate environments. Integer linear programming formulations indexed on time are studied for the Resource-Constrained Project Scheduling Problem (RCPSP). Moreover, the multi-skill, multiple modes, and time lag constraints are taken into consideration. The objective of the RCPSP is to minimize the makespan. The formulations are solved with the default branch-and-cut algorithm of the solver Gurobi Optimizer. The formulations and solver are analyzed concerning the runtime, the number of optimal solutions, and the gap on the resolution of more than 2000 instances. Results indicate the solver can have better performance when instances with up to 50 activities are solved. Then, to develop models to handle hard instances of this problem is a challenge. Moreover, it can bring significant advantages to the corporate environment, helping managers to make accurate decisions and reduce costs.

Index Terms— Mathematical model, optimal scheduling, optimization.

I. INTRODUÇÃO

O problema de Programação de Projetos com Restrição de Recursos (*Resource Constrained Project Scheduling Problem* – RCPSP) considera um conjunto de atividades, um conjunto de tipos de recursos e um conjunto de pares ordenados para estabelecer a relação de precedência entre as atividades [1]. Cada atividade possui um tempo estabelecido de execução e consome certa quantidade de unidades de cada recurso, considerando que os recursos são limitados. O RCPSP consiste em determinar os tempos de início de cada atividade respeitando as relações de precedência e a disponibilidade máxima de cada recurso, de forma que o *makespan*, isto é, a duração total do projeto, seja mínima [2].

O RCPSP pode ser encontrado em indústrias de manufatura e de serviços. Em ambos os contextos, as decisões são tomadas no nível operacional [3]. Na literatura é possível encontrar diferentes métodos para resolver o RCPSP aplicado a situações reais, como o estudo de caso realizado em uma empresa especializada em projetos de [4]. Em [5] há um algoritmo baseado no comportamento de abelhas proposto para o RCPSP com aplicação em indústrias de moldagem por

injeção. Tem-se também um estudo de caso em uma planta de análise química de minério de ferro em [6].

Encontrar uma solução ótima dentro de um tempo computacional satisfatório para o RCPSP é um desafio de pesquisadores que têm proposto modelos matemáticos e métodos exatos para resolvê-lo [3]. Outra consideração bastante importante dentro do RCPSP é a inclusão de restrições que modelam situações reais, como o caso de múltiplos modos, diferentes habilidades, janelas de tempo e tempo de atraso entre atividades, projetos não renováveis, atividades que podem ser interrompidas e depois continuadas, tempo de preparo (*setup*) entre a execução de uma atividade e outra, ambientes de trabalho com máquinas diferentes, atividades que dependem de certos recursos para serem processadas, entre outros [7].

Neste artigo, a principal contribuição consiste em desenvolver e comparar modelos de programação linear inteira para o RCPSP na presença das restrições de múltiplas habilidades, múltiplos modos e tempos de atraso. Em particular, a restrição de múltiplas habilidades (*multi-skill*) impõe que as atividades tenham mais de uma habilidade e, assim, mais de um recurso poderá ser necessário durante o seu processamento. Na restrição de múltiplos modos (*multi-mode*), as atividades podem ter vários modos de execução e, assim, consumir quantidades diferentes dos recursos conforme o modo escolhido. A restrição de tempo de atraso (*minimal and maximal time lags*) entre as atividades considera que há um instante de tempo mínimo e máximo para iniciar uma atividade logo após o término de outra.

Embora problemas de escalonamento e programação de projetos venham sendo investigados já há algum tempo na literatura, a adição dessas restrições não é tão comum [7], geralmente por trazer dificuldades nas etapas de modelagem e resolução. Além disso, uma vez que os modelos desenvolvidos são indexados no tempo, propõe-se discretizar o horizonte de tempo em função dos tempos de processamento das atividades. Também, propõe-se a resolução de uma versão relaxada dos modelos para obter um limitante inferior válido.

Essa discretização e o limitante podem ocasionar uma redução significativa no tamanho dos modelos. Com isso, agilizar a obtenção da solução ótima pelo algoritmo *branch-and-cut* do *solver* Gurobi Optimizer, escolhido pela sua performance para resolver os modelos. Outra contribuição deste trabalho decorre da comparação dos modelos quanto ao tempo de otimização e qualidade das soluções quando resolvendo uma grande variedade de instâncias.

A estrutura do artigo é a seguinte. Na seção II revisa-se brevemente trabalhos que utilizaram métodos exatos e heurísticos para resolver o RCPSP com restrições reais; na

Submitted to review on July 16th, 2019.

This work was supported by the Brazilian funding agencies CNPQ (under Grant. 308312/2016-3) and FAPEG.

L. V. Melo and T. A. Queiroz are with the Programa de Pós-graduação em Engenharia de Produção, Universidade Federal de Goiás – Regional Catalão, Catalão, Goiás, 75708-560, Brasil (e-mail: luciana_vieira.92@hotmail.com, taq@ufg.br).

Corresponding author: L. V. Melo.

seção III são apresentadas a definição formal do RCPSP e os modelos elaborados; na seção IV analisam-se os resultados obtidos nos experimentos computacionais realizados; na seção V apresentam-se as conclusões e trabalhos futuros a serem realizados; e, no Anexo estão as tabelas completas contendo os resultados obtidos com os modelos desenvolvidos.

II. REVISÃO DA LITERATURA

Para resolver o RCPSP na presença da restrição de múltiplos modos e de tempos de atraso (ou intervalos de tempo mínimo e máximo) entre as atividades, Heilmann [8] apresentou um método exato, o *branch-and-bound* com busca em profundidade. O objetivo dessa variante do RCPSP é definir um modo de execução e um instante de tempo para iniciar cada atividade, respeitando as restrições impostas, além de minimizar o *makespan*. Os experimentos computacionais mostraram a dificuldade do método em resolver instâncias com muitas atividades.

Yamashita e Morabito [9] propuseram um algoritmo para resolver o RCPSP com custo de disponibilidade de recursos e atividades com múltiplos modos de execução. O objetivo do estudo foi determinar um planejamento do processamento das atividades, a fim de minimizar o custo total de destinação de recursos, sem violar a data de conclusão do projeto, as relações de precedência entre as atividades e a disponibilidade de recursos. Os resultados comprovaram que o algoritmo proposto é viável computacionalmente para resolver instâncias com até 10 atividades.

Correia e Saldanha-da-Gama [10] desenvolveram um modelo de programação linear inteira mista para o RCPSP com múltiplas habilidades. O projeto foi composto por diversas atividades e cada atividade demanda mais de uma habilidade e, assim, mais de um recurso necessário para processá-la. Os custos estão relacionados com o consumo de recursos e envolvem custos fixos e variáveis. Um dos objetivos do artigo foi investigar a viabilidade de utilizar um *solver* para resolver o problema. Além disso, os autores buscaram comparar as soluções alcançadas com o objetivo direcionado a custos com a minimização do *makespan*. Experimentos computacionais foram realizados sobre instâncias com 20 atividades. Os resultados demonstraram que tanto o custo ótimo, quanto o *makespan* final, possuem relações com insumos, componentes de custo, quantidade de recurso e tempo para o projeto.

Para resolver o RCPSP com múltiplas habilidades, dois modelos de programação linear inteira mista foram desenvolvidos por Javanmard *et al.* [11] a fim de minimizar o custo total de recrutamento para níveis de habilidades distintos. Os níveis mais altos de habilidades possuem um processamento mais rápido, logo o custo é maior, e cada atividade pode demandar uma ou mais habilidades. Os modelos foram validados a partir de 30 instâncias.

Além dos trabalhos acima citados, que resolveram o RCPSP por algum método exato, há também algumas revisões que discutiram aspectos gerais do problema e de restrições reais. A revisão feita por Hartmann e Briskorn [7], por exemplo, discute sobre a restrição de tempos de atraso mínimo e máximo, que compreendem atrasos ou intervalos de tempo

mínimo e máximo entre o término de uma atividade i e o início de outra j .

Quanto aos métodos heurísticos, desenvolvidos para resolver de modo satisfatório instâncias grandes, Almeida *et al.* [12] propuseram uma heurística paralela para o RCPSP com múltiplas habilidades. Para tanto, dois conceitos foram abordados, o peso do recurso e o agrupamento de atividades.

Silva *et al.* [13] desenvolveram uma heurística que combina uma parte construtiva, para gerar soluções iniciais de qualidade, e uma busca tabu, para explorar conveniente o espaço de soluções. A heurística, chamada *Shifting Bottleneck Procedure*, foi utilizada para resolver um problema do tipo *job shop*, visando a minimização do *makespan*. Os resultados foram promissores conforme os autores relataram.

Um algoritmo baseado no comportamento de abelhas foi proposto por Oztemel e Selam [5] para o RCPSP com recursos únicos e limitados, e múltiplos modos de execução, com o objetivo de encontrar o menor tempo de conclusão de projetos de moldagem por injeção. Os autores usaram um gerador de projetos para definir as limitações do projeto de fabricação de moldes. O algoritmo foi eficiente, gerando cronogramas apropriados para os projetos dessa natureza, mesmo na presença de muitas atividades e recursos limitados.

Muritiba *et al.* [14] desenvolveram um algoritmo *Path-Relinking* para resolver o RCPSP com a restrição de múltiplos modos, no qual as atividades que fazem parte de um projeto dispõem de um conjunto de modos de execução. Experimentos computacionais foram realizados sobre três conjuntos distintos de instâncias, com o número de atividades variando entre 10 e 100, visando comparar os resultados com métodos da literatura.

Wang e Zheng [15] utilizaram um algoritmo multiobjetivo, que simula o comportamento da mosca-de-fruta, para resolver o RCPSP com múltiplas habilidades, com o objetivo de minimizar o *makespan*. Os resultados computacionais mostraram a eficácia do algoritmo quando comparado com outros métodos da literatura.

Schimidt *et al.* [16] apresentaram um modelo de programação linear inteira e um método *local branching* para um problema de escalonamento em máquinas paralelas considerando tempos de *setup* entre as atividades. Os autores compararam a performance das abordagens em instâncias com até 100 atividades e 5 máquinas, relatando a superioridade do *local branching* nas instâncias com mais atividades. Em Gomes *et al.* [17] faz-se o uso de simulação contínua para aprimorar o desenvolvimento de *softwares*, levando em consideração aspectos de gerenciamento de risco e programação de atividades.

Observando as contribuições da literatura, o presente trabalho segue na linha do desenvolvimento e comparação de modelos e métodos exatos para o RCPSP, buscando obter soluções ótimas [8, 10, 11, 16], com o desenvolvimento de modelos de programação linear inteira resolvidos pelo *solver Gurobi Optimizer*. Os modelos incluem aspectos práticos do problema, relativamente pouco investigados na literatura, mas de grande interesse real [7], em particular, as restrições de múltiplas habilidades, múltiplos modos e tempos de atraso. Essas restrições consideram modelos que minimizam a duração total do projeto, diferente daquelas que consideraram a minimização de custos [10, 11] ou que lidaram com um

problema multiobjetivo [15]. Para uma comparação cuidadosa dos modelos, logo não se limitando a relativamente poucas instâncias ou àquelas de pequeno porte [9, 10, 11], realizam-se experimentos sobre mais de 2000 instâncias de diferentes tamanhos (com até 202 atividades, 20 diferentes recursos e 15 tipos de habilidades).

III. FORMULAÇÕES

Para resolver o RCPSP é necessário que as relações de precedência entre as atividades sejam satisfeitas; a quantidade de cada recurso consumida durante cada instante de tempo não ultrapasse a disponibilidade total do respectivo recurso; e, que o instante de início de cada atividade seja determinado; tal que o objetivo é minimizar o *makespan*, que resulta no tempo de conclusão total do projeto, ou no tempo de conclusão da última atividade. A partir disso, têm-se os seguintes conjuntos definidos para o problema:

- $V = \{0, 1, 2, \dots, n, n+1\}$: possui as n atividades que fazem parte do projeto, além das atividades fictícias 0 e $n+1$;
- $R = \{1, 2, \dots, r\}$: conjunto dos r recursos renováveis;
- $Pred = \{(i, j) | i \in V, j \in V, i < j\}$: conjunto de pares ordenados (i, j) de atividades com relação de precedência, indicando que a atividade j só poderá iniciar após o término da atividade i ;
- C_{ib} : quantidade do recurso b que a atividade i consome durante o seu processamento, sendo que as atividades fictícias não consomem recursos;
- R_b : disponibilidade total do recurso b , que é renovável;
- d_i : tempo de processamento (isto é, duração) da atividade i , assumido como um valor inteiro, sendo que as atividades fictícias possuem duração zero;
- $S = \{1, 2, \dots, K\}$: possui as k habilidades que um recurso consegue atender.

Além disso, o problema requer que cada atividade, uma vez iniciada, não seja interrompida até o término do seu processamento, ou seja, não é permitida a interrupção de atividades; as atividades podem ocorrer em paralelo, desde que o consumo dos recursos seja respeitado em cada instante de tempo; a partir do tempo zero, as atividades podem iniciar a qualquer momento, desde que as relações de precedência sejam respeitadas; as atividades não podem ser fracionadas; e, o tempo de preparo (*setup*) entre as atividades é desprezível, sendo considerado igual à zero.

As formulações desenvolvidas para o RCPSP com restrições práticas consideram que as atividades são programadas em um horizonte de tempo, variando de $[0, T]$, em que T é um limitante superior para o *makespan* obtido pela soma dos tempos de processamento de todas as atividades.

Ao analisar o horizonte de tempo $[0, T]$ e, sabendo que o valor do tempo de processamento d_i é um inteiro não-negativo, nota-se que não é preciso considerar todos os valores inteiros no intervalo $[0, T]$. Conforme a estratégia definida em relação aos *canonical dissections* [18], torna-se suficiente discretizar o horizonte de tempo em valores inteiros que são combinações cônicas (isto é, combinações binárias não negativas) entre os tempos de processamento das atividades.

Com isso, define-se o conjunto P com os pontos de discretização para os modelos desenvolvidos adiante. Um

instante de tempo $p \in P$ advém de um subconjunto de atividades cuja soma de seus tempos de processamento seja exatamente p . Para determinar os elementos do conjunto P , pode-se empregar o algoritmo *Discretization using Dynamic Programming* (DDP) de [19].

A. Múltiplas Habilidades

O RCPSP com múltiplas habilidades considera um subconjunto de habilidades, enquanto cada recurso dispõe de um subconjunto de habilidades que consegue atender. Diferente de outras propostas da literatura [10, 12], que consideram custos associados as habilidades e recursos, assumem-se os seguintes conjuntos:

- S_b : quantidade total do recurso b que está disponível para atender as atividades;
- l_{ik} : quantidade da habilidade k que a atividade i requer;
- Q_{bk} : nível da habilidade k que o recurso b consegue atender.

A formulação para resolver o RCPSP com múltiplas habilidades é de programação linear inteira, a qual considera um único tipo de variável de decisão binária, isto é:

- x_{ipb} : recebe valor 1 quando a atividade i inicia no tempo p e faz uso do recurso b , caso contrário, recebe valor zero.

A formulação (1)-(8) do Modelo 1, definido para o RCPSP com múltiplas habilidades, é:

$$\text{Minimizar } \sum_{b \in R} \sum_{p \in P} p x_{(n+1)pb} \quad (1)$$

$$\sum_{b \in R} \sum_{p \in P} x_{ipb} = 1, \quad \forall i \in V \quad (2)$$

$$\sum_{i \in V} \sum_{p \in \{t-d_i+1 \leq p \leq t\}} x_{ipb} \leq 1, \quad \forall t \in P, \quad \forall b \in R \quad (3)$$

$$\sum_{b \in R} \sum_{p \in P} p x_{jpb} - \sum_{b \in R} \sum_{p \in P} p x_{ipb} \geq d_i, \quad \forall (i, j) \in pred \quad (4)$$

$$\sum_{i \in V} \sum_{p \in P} \sum_{k \in S} l_{ik} x_{ipb} \leq S_b, \quad \forall b \in R \quad (5)$$

$$\sum_{b \in R} \sum_{p \in P} Q_{bk} x_{ipb} \geq l_{ik}, \quad \forall i \in V, \quad \forall k \in S \quad (6)$$

$$\sum_{b \in R} \sum_{p \in P} p x_{(n+1)pb} \geq LI \quad (7)$$

$$x_{ipb} \in \{0, 1\}, \quad \forall i \in V, \quad \forall p \in P, \quad \forall b \in R \quad (8)$$

A função objetivo (1) está relacionada com uma solução de *makespan* mínimo. As restrições (2) indicam que cada atividade deve ser iniciada em exatamente um instante de

tempo p e ser atendida por exatamente um recurso b . Nota-se que dentre todos os recursos com habilidades suficientes para suprir cada atividade, apenas um deles deve ser escolhido. As restrições em (3) definem que em um instante de tempo p e para um dado recurso b , no máximo uma atividade poderá requerer o recurso b para ser processada, sendo que este recurso deve ser capaz de atender todos os níveis de habilidade que a atividade exige. Caso existam atividades sendo processadas em paralelo, elas não podem compartilhar o mesmo recurso b .

Nas restrições em (4), a relação de precedência entre as atividades é garantida, tal que a atividade j inicia somente após o término das atividades i que são suas predecessoras. As restrições (5) e (6) estão relacionadas com a restrição prática de múltiplas habilidades.

As restrições (5) asseguram que o consumo do recurso b seja respeitado, ou seja, dentre todas as atividades que requerem o recurso b , a soma da quantidade das habilidades deve ser menor ou igual ao total disponível do recurso. Isso significa que um recurso tem capacidade limitada e é não-renovável, restringindo a quantidade de atividades que o recurso consegue atender.

As restrições (6) garantem que o nível de habilidade k que a atividade i exige deve ser atendido por algum recurso b . Na restrição (7), impõe-se que o *makespan* seja pelo menos igual a um limitante inferior (LI). Por fim, tem-se as restrições (8) que representam o domínio das variáveis. O limitante inferior (LI) usado na restrição (7) é obtido por meio da resolução do modelo relaxado com função objetivo (1) e restrições (2), (3), (4), (6) e (8). Conforme discutido em [20], as restrições de consumo de recursos acabam dificultando a resolução da formulação, por isso são relaxadas.

B. Múltiplos Modos

Para o RCPSP com múltiplos modos foram adicionadas restrições convenientes, considerando que as atividades podem ter vários modos de execução. Cada modo determina uma forma diferente de processar a atividade. Além disso, a formulação considera dois tipos de recursos, os renováveis e os não-renováveis [21].

A formulação (9)-(15) do Modelo 2 a seguir, definido para o RCPSP com múltiplos modos, possui variáveis de decisões binárias:

- x_{imp} : tem valor 1 quando a atividade i possui modo de execução m e inicia no instante de tempo p , caso contrário, recebe valor zero.

Além disso, têm-se os seguintes conjuntos:

- M_i : modos de execução que a atividade i possui. As atividades fictícias 0 e $n+1$ possuem apenas um modo de execução, têm duração igual a zero e não consomem recursos;
- $NR = \{1, 2, \dots, nr\}$: conjunto dos nr recursos não-renováveis;
- NR_b : disponibilidade total do recurso b , que é não-renovável;
- C_{imb} : quantidade do recurso b que a atividade i consome durante seu processamento no modo de execução m ;
- d_{im} : duração da atividade i no modo de execução m .

$$\text{Minimizar } \sum_{m \in M_i} \sum_{p \in P} px_{(n+1)mp} \quad (9)$$

$$\sum_{m \in M_i} \sum_{p \in P} x_{imp} = 1, \quad \forall i \in V \quad (10)$$

$$\sum_{i \in V} \sum_{m \in M_i} \sum_{p \in \{t-d_i+1 \leq p \leq t\}} C_{imb} x_{imp} \leq R_b, \quad \forall t \in P, \quad \forall b \in R \quad (11)$$

$$\sum_{i \in V} \sum_{m \in M_i} \sum_{p \in P} C_{imb} x_{imp} \leq NR_b, \quad \forall b \in NR \quad (12)$$

$$\sum_{m \in M_j} \sum_{p \in P} px_{jmp} - \sum_{m \in M_i} \sum_{p \in P} (p + d_{im}) x_{imp} \geq 0, \quad (13)$$

$$\forall (i, j) \in \text{pred}$$

$$\sum_{m \in M_i} \sum_{p \in P} px_{(n+1)mp} \geq LI \quad (14)$$

$$x_{imp} \in \{0, 1\}, \quad \forall i \in V, \quad \forall m \in M_i, \quad \forall p \in P \quad (15)$$

A função objetivo (9) calcula o *makespan*, que se busca o mínimo possível. As restrições em (10) estão diretamente relacionadas com a restrição prática de múltiplos modos, impondo que cada atividade deve ser iniciada em exatamente um instante de tempo p e processada em um único modo m . As restrições em (11) garantem que o consumo de recurso pelas atividades executadas no modo m , em cada instante de tempo, respeite a disponibilidade de cada recurso. Nas restrições (12), impõem-se que a quantidade total consumida, considerando todo o horizonte de tempo, de cada recurso não-renovável, seja respeitada.

As restrições em (13) garantem a relação de precedência entre as atividades. A restrição (14) assegura que o *makespan* seja pelo menos igual a um limitante inferior (LI). Por último, têm-se as restrições (15) que apresentam o domínio das variáveis. O limitante inferior (LI) é obtido pela resolução de um modelo relaxado, sem as restrições de consumo de recursos renováveis (11) e não-renováveis (12). Neste caso, o modelo relaxado tem função objetivo (9) e restrições (10), (13) e (15).

C. Tempo de Atraso

No RCPSP com tempos de atraso mínimo e máximo para a execução das atividades, tem-se restrições que consideram uma janela de tempo que define o instante de tempo mínimo e máximo para o início de uma atividade j dado o término de uma atividade i . Neste caso, consideram-se os seguintes parâmetros/conjuntos:

- $\text{lag} = \{(i, j) \mid i, j \in V, i \neq j\}$: conjunto de pares ordenados (i, j) de atividades que possuem relações de atraso mínimo e máximo;

- d_{ij}^{\min} : duração mínima para a preparação (isto é, atraso) após o término da atividade i para que a atividade j possa ser iniciada, para os pares de atividades (i, j) em lag ;
- d_{ij}^{\max} : máximo instante após o término da atividade i para que a atividade j seja iniciada, para os pares de atividades (i, j) em lag ;

A formulação (16)-(23) do Modelo 3 a seguir, baseada em [7] e [22], é definida para o RCPSP com tempos de atraso e utiliza variáveis de decisões binárias, isto é:

- x_{ip} : recebe valor 1 quando a atividade i inicia no instante de tempo p , caso contrário, recebe valor zero.

$$\text{Minimizar } \sum_{p \in P} px_{(n+1)p} \quad (16)$$

$$\sum_{p \in P} x_{ip} = 1, \quad \forall i \in V \quad (17)$$

$$\sum_{i \in V} \sum_{p \in \{t-d_i+1 \leq p \leq t\}} C_{ib} x_{ip} \leq R_b, \quad \forall t \in P, \quad \forall b \in R \quad (18)$$

$$\sum_{p \in P} px_{jp} - \sum_{p \in P} px_{ip} \geq d_i, \quad \forall (i, j) \in pred \quad (19)$$

$$\sum_{p \in P} px_{jp} \geq \sum_{p \in P} x_{ip} + d_i + d_{ij}^{\min}, \quad \forall (i, j) \in lag \quad (20)$$

$$\sum_{p \in P} px_{jp} \leq \sum_{p \in P} x_{ip} + d_i + d_{ij}^{\max}, \quad \forall (i, j) \in lag \quad (21)$$

$$\sum_{b \in R} \sum_{p \in P} px_{(n+1)p} \geq LI \quad (22)$$

$$x_{ip} \in \{0,1\}, \quad \forall i \in V, \quad \forall p \in P \quad (23)$$

A função objetivo (16) está associada com a minimização do *makespan*. As restrições (17) garantem que toda atividade deve ser iniciada em exatamente um instante de tempo p , com a possibilidade de atividades em paralelo. As restrições (18) garantem que a quantidade de cada recurso consumida em cada instante de tempo respeite a quantidade disponível do referido recurso. As relações de precedência entre as atividades são satisfeitas por meio das restrições (19). As restrições (20) e (21) modelam os tempos de atraso, de forma que eles sejam respeitados entre as atividades. Nas restrições (20), tem-se o tempo mínimo que a atividade j deve esperar para iniciar após o término do processamento da atividade i .

Por outro lado, em (21), tem-se o máximo instante de tempo permitido, após a conclusão da atividade i , para que o início da atividade j ocorra. A restrição em (22) define que o *makespan* deve ser pelo menos igual a um limitante inferior (LI). Finalmente, as restrições (23) indicam o domínio das variáveis. O limitante inferior (LI) é alcançado pela resolução do modelo relaxado, que não inclui as restrições de consumo de recursos, isto é, resolve-se o modelo relaxado com função objetivo (16) e restrições (17), (19), (20), (21) e (23).

IV. RESULTADOS E DISCUSSÕES

As formulações para o RCPSP com restrições reais foram codificadas na linguagem C++ e resolvidas pelo *solver Gurobi Optimizer* na versão 8.0, com seu algoritmo *branch-and-cut* com configurações e parâmetros *default*. O computador adotado para os experimentos possui sistema operacional Ubuntu 16.04 LTS, com processador Intel Core i5 de 3,4 GHz e 8 GB de RAM. Cada instância teve um tempo de limite de 3.600 segundos para ser resolvida, sendo que as tabelas completas com os resultados se encontram no Apêndice.

Embora nos resultados reportados adiante as instâncias tenham sido obtidas de bases para o RCPSP [23, 24], tornou-se necessário adaptações conforme a modelagem das restrições, procedimento usual na literatura do problema quando há restrições práticas [5, 8, 9, 11, 12]. Outra dificuldade em realizar uma comparação direta com resultados já publicados advém de critérios diversos usados na literatura, como o tempo limite de resolução, por exemplo, de até 10 horas para resolver cada instância [10]. No que tange uma comparação justa do tempo computacional requerido pelo método, tem-se ainda as diferenças de *hardware* (computadores de configurações diversas) e *software* (linguagens de programação e sistemas operacionais distintos) utilizados na literatura. Por esses motivos, os modelos desenvolvidos são comparados quanto ao tempo de otimização requerido pelo *solver* e qualidade das soluções quanto a otimalidade, para um total de 2076 instâncias.

A. RCPSP com Múltiplas Habilidades

Os testes realizados sobre a formulação para o RCPSP com múltiplas habilidades ocorreram sobre instâncias da base IMOPSE¹ (*Intelligent Multi Objective Project Scheduling Environment*) proposta por [21]. Essas instâncias possuem custos para o uso de recursos pelas atividades e, assim, o problema é geralmente tratado como multiobjetivo.

Utilizaram-se um total de 36 instâncias para os testes. A quantidade de atividades presente nas instâncias é de 12, 17, 102 ou 202, sendo a primeira e a última atividade fictícias; a quantidade de recursos é de 3, 5, 6, 7, 9, 10 ou 20; a quantidade de tipos de habilidades é de 3, 5, 6, 7, 9 ou 15. As quantidades disponíveis de cada recurso e das habilidades que as atividades requerem foram gerados aleatoriamente a partir das informações originais de custos, seguindo a modelagem assumida para as restrições de múltiplas habilidades.

O Modelo 1, definido em (1)-(8), tem a melhor solução encontrada reportada na Tabela I. Os dados em cada linha dessa tabela são: nome da instância, limitante inferior (LI) obtido ao resolver o modelo relaxado pelo *Gurobi*; valor da solução retornado pelo *Gurobi*, que é o *makespan*; tempo de otimização em segundos (não está incluso o tempo para obter o limitante inferior); e, o valor do *gap* em porcentagem, retornado pelo *Gurobi*. No final da tabela há a média, o desvio padrão, o mínimo e o máximo dos resultados.

Observando a Tabela I, das 36 instâncias, 18 foram resolvidas de forma ótima pelo Modelo 1, o que corresponde a 50,0%. Essas instâncias compreendem até 102 atividades,

¹<http://imopse.ii.pwr.wroc.pl/download.html>

sendo apenas uma de 202 atividades (incluindo as duas fictícias). Apenas para uma das instâncias com 102 atividades e 5 tipos de recursos (isto é, instância 100_5_64_15), que não foi possível obter qualquer solução viável, notando que esta instância possui 64 relações de precedência e 15 tipos de habilidade. Para as demais instâncias, com 102 ou mais atividades e 10 ou mais tipos de recursos, não foi possível obter qualquer solução dado o tempo limite considerado.

Com relação ao limitante inferior na Tabela I, obtido ao resolver com o *solver* o modelo relaxado associado ao Modelo 1, nota-se que ele coincidiu com a solução retornada pelo *Gurobi* ao resolver o Modelo 1, indicando que os limitantes são justos. O tempo computacional médio, considerando as 18 instâncias com solução apresentada, é de 893,7 segundos, mas houve uma instância cujo tempo é superior aos 3.000 segundos (que é a 100_10_47_9). Portanto, observa-se que o *Gurobi* teve dificuldades, dentro do tempo computacional imposto, com o Modelo 1 sobre instâncias com mais de 102 atividades.

B. RCPSP com Múltiplas Modos

As instâncias para o RCPSP com múltiplos modos foram obtidas da base MMLIB² (*Multi-mode Library*) de [24]. Utilizaram-se dois conjuntos, MMLIB50 e MMLIB100, para um total de 1080 instâncias. As instâncias do conjunto MMLIB50 possuem 52 atividades, sendo a primeira e a última fictícias; dois recursos renováveis e dois não-renováveis; tempo de processamento das atividades e quantidade de consumo de recurso, por cada atividade, para cada um dos três modos de execução. O conjunto MMLIB100 possui as mesmas características do conjunto MMLIB50, exceto a quantidade de atividades que neste caso é de 102, incluindo as duas fictícias.

Considera-se o Modelo 2, definido em (9)-(15), sendo que a melhor solução encontrada pelo *solver* é reportada. As Tabelas II e III contemplam os resultados das 540 instâncias do conjunto MMLIB50 e 540 instâncias do conjunto MMLIB100, respectivamente.

Os resultados nas Tabelas II e III estão organizados por grupos de 5 instâncias, ou seja, MM501 refere-se ao primeiro grupo contendo 5 instâncias, de forma que os valores apresentados são a média para o referido grupo. Os dados em cada linha da tabela são: nome do grupo; média do limitante inferior (LI), obtido ao resolver o modelo relaxado pelo *Gurobi*; número de instâncias do grupo resolvidas de forma ótima; média do valor de solução, isto é, do *makespan*; média do tempo de otimização em segundos (não está incluso o tempo gasto para obter o limitante inferior); e, a média do valor do *gap*, em porcentagem, retornado pelo *Gurobi*. No final dessas tabelas, tem-se também a média geral, o desvio padrão, o mínimo e o máximo, considerando todos os grupos.

Conforme os resultados nas Tabelas II e III, 431 instâncias foram resolvidas de forma ótima, correspondendo a 79,8% das instâncias para o conjunto MMLIB50 e sendo este resultado de 372 (isto é, 68,9%) instâncias para o conjunto MMLIB100. Com relação a quantidade de grupos que obtiveram todas as instâncias resolvidas de forma ótima, para o conjunto

MMLIB50 foi possível resolver 75 dos 108 grupos, enquanto para o conjunto MMLIB100 foi possível resolver 63 dos 108 grupos. Para os demais grupos de instâncias, o *solver* alcançou o tempo de otimização de 3.600 segundos para pelo menos uma instância, com o *gap* diferente de 0,0%, gerando apenas uma solução viável e não provada ótima.

Com relação ao *gap* médio reportado nas Tabelas II e III, para o conjunto MMLIB50, este valor foi 2,1% (com desvio padrão de 4,9%), enquanto para o conjunto MMLIB100 foi de 6,5% (com desvio padrão de 14,5%). Ao observar o maior *gap* no conjunto MMLIB50, este foi de 27,0% para o grupo MM5080, enquanto para o conjunto MMLIB100, tal *gap* foi de 79,4% para o grupo MM10043. Nota-se, assim, que o *Gurobi* sobre o Modelo 2 trouxe resultados de qualidade para o conjunto MMLIB50, enquanto necessitaria de mais tempo computacional para o conjunto MMLIB100, que envolve o dobro de atividades.

Embora Muritiba *et al.* [14] resolveram essas mesmas instâncias, os autores não reportaram os valores individuais de soluções, dificultando uma comparação precisa com eles. Por outro lado, os autores informaram o desvio percentual médio do seu método heurístico, considerando todas as instâncias de cada conjunto, sendo esses valores acima dos 23,0%, o que é relativamente superior aos 2,1% e 6,5% encontrado pelo *Gurobi* ao resolver o Modelo 2.

Ao considerar os limitantes inferiores reportados nas Tabelas II e III a partir da resolução do modelo relaxado associado ao Modelo 2, tem-se que para 32 grupos do conjunto MMLIB50, o valor médio desse limitante coincidiu com o valor médio da solução para o grupo. Por outro lado, no caso do conjunto MMLIB100, o número de grupos com essa característica foi de 34. Embora isso tenha ocorrido para muitos grupos, o *Gurobi* ainda precisou de elevado tempo computacional para provar a otimalidade das soluções e, em vários casos, não conseguiu essa prova dentro do tempo limite imposto. Em geral, o tempo médio foi de 773,3 segundos (com desvio padrão de 1.306,9 segundos) para o conjunto MMLIB50, ao passo que foi de 1.207,9 segundos (com desvio padrão de 1.456,1 segundos) para o MMLIB100. Percebe-se ainda que o tempo limite foi atingido para todas as instâncias de 15 grupos do conjunto MMLIB50 e para 22 grupos do conjunto MMLIB100.

C. RCPSP com Tempos de Atraso

Para os testes que consideram o RCPSP com as restrições de tempos de atraso mínimo e máximo, adaptaram-se as instâncias da PSPLIB³ (*Project Scheduling Problem Library*), em particular, os conjuntos J30 e J60, resultando nos respectivos conjuntos L30 e L60. A adaptação consistiu em definir, para todas as atividades que têm relação de precedência (isto é, faz-se o conjunto *lag* ser igual ao *pred*), um valor de d_{ij}^{min} e o d_{ij}^{max} . Neste caso, considerou-se d_{ij}^{min} como sendo o mínimo valor entre a duração d_i da atividade i e a duração d_j da atividade j , depois dividido por 3. Para o valor de d_{ij}^{max} , considerou-se o máximo entre d_i e d_j , depois multiplicado por 10.

Nas Tabelas IV e V estão os resultados para as 960 instâncias do conjunto L30 e L60, dada a resolução do Modelo

²http://www.projectmanagement.ugent.be/research/project_scheduling/multi_mode

³<http://www.om-db.wi-tum.de/psplib/>

3, definido em (16)-(23), pelo Gurobi. Os valores nessas tabelas seguem a mesma interpretação das Tabelas II e III, porém os resultados são organizados em grupos de 10 instâncias.

De acordo com os resultados das Tabelas IV e V, das 480 instâncias para o conjunto L30, 438 foram resolvidas de forma ótima (ou seja, 91,3%) e 344 para o conjunto L60, o que equivale a 71,9%. Além disso, todos os 48 grupos do conjunto L30 tiveram pelo menos uma instância resolvida de forma ótima pelo *solver*, sendo que 25 grupos tiveram todas as 10 instâncias do grupo resolvidas na otimalidade. Para o conjunto L60, para 44 grupos, pelo menos uma instância teve a solução ótima obtida, enquanto para 8 grupos, todas as 10 instâncias do grupo tiveram a solução ótima encontrada.

Com relação ao valor do *gap*, o valor médio para o conjunto L30 foi de 0,2% (com desvio padrão de 0,8%), enquanto o do conjunto L60 foi de 1,6% (com desvio padrão de 5,1%). O pior *gap* no L30 foi para o grupo L3029 e corresponde a 4,5%, enquanto no L60, o pior *gap* foi de 27,7%, para o grupo L6013. Nota-se que para dois grupos não foi possível obter uma solução viável para qualquer instância do grupo, devido ao valor do multiplicador utilizado, que foi de 10 para o tempo máximo, sendo preciso adotar um valor maior.

Quanto ao tempo computacional, nota-se que o *solver* atingiu o tempo limite para todas as instâncias de cinco grupos do conjunto L60. O tempo médio do *Gurobi* foi de 129,0 segundos (com desvio padrão de 471,5 segundos) para o conjunto L30 e de 589,9 segundos (com desvio padrão de 1.232,9 segundos) para o conjunto L60.

V. CONCLUSÕES

O problema de programação de projetos com restrição de recursos é investigado neste artigo. Apresentaram-se três formulações de programação linear inteira para o RCPSP com a inclusão de restrições de importância prática, quais sejam: múltiplas habilidades, múltiplos modos de execução e tempos de atraso entre atividades. Além disso, os modelos tiram proveito de uma discretização do horizonte de tempo em função do tempo de processamento das atividades. Os modelos foram resolvidos pelo *solver Gurobi*, utilizando seu algoritmo *branch-and-cut*.

Os resultados computacionais indicam que o *solver*, ao resolver os modelos desenvolvidos, teve um comportamento satisfatório. Embora não tenha sido realizada uma comparação direta com outros métodos da literatura, no caso das restrições de múltiplas habilidades e tempos de atraso, os modelos e as soluções reportadas podem ser utilizados como ponto de partida para a proposta de outros métodos pela literatura especializada, além de fomentar a criação de possíveis *benchmarks* para o RCPSP na presença dessas restrições.

Com relação ao RCPSP com a restrição de múltiplas habilidades, o *solver* obteve a solução ótima para 50,0% das instâncias, com um tempo computacional médio de 2.246,9 segundos. Para as demais instâncias em que o *solver* não obteve qualquer solução dentro do tempo limite de uma hora, o número de atividades é superior a 102 e com pelo menos 10 tipos de recursos (exceto para uma instância que tem 5 tipos de recursos, porém um número elevado de relações de

precedência e tipos de habilidades), sendo instâncias relativamente grandes para um modelo matemático.

A restrição de múltiplos modos foi também considerada no RCPSP, dando origem a segunda formulação. Os resultados computacionais sobre as 1080 instâncias, divididas em dois conjuntos, um com 52 e o outro com 102 atividades, mostram soluções ótimas pelo *solver* para 79,8% das instâncias com 52 atividades e 68,9% para o conjunto com 102 atividades, com um tempo computacional médio de 773,3 e 1.207,9 segundos, respectivamente. Além disso, o *gap* médio geral foi de, respectivamente, 2,1% e 6,5%, que é menor do que o desvio percentual médio apresentado por um método heurístico proposto recentemente na literatura. Assim, o *solver* sobre o modelo desenvolvido teve um bom desempenho para as instâncias com 52 atividades, começando a ter dificuldade nas instâncias com 102 atividades.

A restrição de tempos de atraso, que modela uma janela de tempo mínimo e máximo para que uma atividade possa começar após o término de outra, faz parte do terceiro modelo. Os resultados computacionais do *solver*, sobre 980 instâncias adaptadas da literatura, mostram soluções ótimas para 91,3% das instâncias com 32 atividades e de 71,9% das instâncias com 62 atividades, com um tempo computacional médio respectivo de 129,0 e de 589,9 segundos. Nota-se ainda que o *gap* médio geral foi de 0,2% para o primeiro conjunto de instâncias e de 1,6% para o segundo conjunto.

A partir da análise dos três modelos desenvolvidos para o RCPSP com as três distintas restrições, conclui-se que a proposta de um modelo de programação matemática para lidar com instâncias maiores, considerando um tempo limite de uma hora de execução, torna-se um desafio para aqueles que buscam por soluções ótimas a fim de tomar suas decisões. Embora esses modelos possam ser interessantes para o ambiente corporativo, acredita-se que há oportunidades de melhorias, como por exemplo: considerar um limitante superior para o *makespan*, obtido por meio de uma heurística; desenvolver desigualdades válidas, bem como calcular outros limitantes inferiores (por exemplo, pela relaxação Lagrangeana); e a proposta de *math*-heurísticas.

APÊNDICE

As Tabelas I a V, contendo os resultados analisados na seção IV, são apresentadas neste apêndice. O símbolo “-” indica que a instância não foi resolvida, ou seja, o *solver Gurobi* não conseguiu apresentar qualquer solução viável dentro do tempo limite de 3.600 segundos. O símbolo “*” representa que pelo menos uma instância do grupo não foi resolvida, ou seja, o *Gurobi* não conseguiu encontrar qualquer solução viável dentro do tempo limite imposto.

TABELA I
RESULTADOS DO RCPSP COM MÚLTIPLAS HABILIDADES

Instância	Modelo 1			
	LI	Sol.	Tempo	GAP
10_3_5_3	93,0	93,0	0,5	0,0
10_5_8_5	80,0	80,0	0,5	0,0
10_7_10_7	104,0	104,0	0,8	0,0
15_3_5_3	230,0	230,0	1,5	0,0
15_6_10_6	102,0	102,0	2,0	0,0
15_9_12_9	90,0	90,0	4,3	0,0
100_5_20_9_D3	389,0	389,0	857,5	0,0
100_5_22_15	493,0	493,0	1.466,2	0,0
100_5_46_15	2.566,0	2.566,0	188,7	0,0
100_5_48_9	498,0	498,0	1.690,3	0,0
100_5_64_9	492,0	492,0	1.210,9	0,0
100_10_26_15	2.253,0	2.253,0	1.927,0	0,0
100_5_64_15	-	-	3.600,0	-
100_10_27_9_D2	-	-	3.600,0	-
100_10_47_9	304,0	304,0	3.323,6	0,0
100_10_48_15	244,0	2.395,0	3.600,0	89,0
100_10_64_9	255,0	-	3.600,0	-
100_10_65_15	-	-	3.600,0	-
100_20_22_15	156,0	156,0	2.303,4	0,0
100_20_23_9_D1	172,0	172,0	1.226,4	0,0
100_20_46_15	2.612,0	2.612,0	501,5	0,0
100_20_47_9	-	-	3.600,0	-
100_20_65_9	-	-	3.600,0	-
100_20_65_15	2.401,0	2.401,0	543,3	0,0
200_10_50_9	4.809,0	4.809,0	838,3	0,0
200_10_50_15	-	-	3.600,0	-
200_10_84_9	-	-	3.600,0	-
200_10_85_15	-	-	3.600,0	-
200_10_128_15	-	-	3.600,0	-
200_10_135_9_D6	-	-	3.600,0	-
200_20_54_15	-	-	3.600,0	-
200_20_55_9	-	-	3.600,0	-
200_20_97_9	-	-	3.600,0	-
200_20_97_15	-	-	3.600,0	-
200_20_145_15	-	-	3.600,0	-
200_20_150_9_D5	-	-	3.600,0	-
Média	917,2	1.065,2	2.246,9	4,7
Desvio Padrão	1.293,8	1.347,4	1.526,9	20,4
Mínimo	80,0	80,0	0,5	0,0
Máximo	4.809,0	4.809,0	3.600,0	89,0

TABELA II
RESULTADOS PARA OS CONJUNTOS MMLIB50 OBTIDOS AO RESOLVER O
MODELO 2

Grupo	MMLIB50				
	LI	QTD	Sol.	Tempo	GAP
MM501	17,2	4	27,4	748,4	0,8
MM502	19,2	4	23,6	974,1	0,8

MM503	19,2	5	19,6	25,2	0,0
MM504	17,8	4	19,8	731,1	0,9
MM505	18,6	5	18,6	3,0	0,0
MM506	22,0	5	22,0	3,0	0,0
MM507	19,6	0	46,2	3.600,0	11,2
MM508	18,8	0	39,0	3.600,0	12,8
MM509	17,6	0	28,6	3.600,0	8,1
MM5010	17,2	0	32,4	3.600,0	12,4
MM5011	18,6	0	27,2	3.600,0	3,7
MM5012	17,8	0	26,8	3.600,0	4,4
MM5013	19,4	5	26,0	8,7	0,0
MM5014	19,4	5	21,0	5,5	0,0
MM5015	20,4	5	20,4	1,6	0,0
MM5016	19,2	5	19,2	1,8	0,0
MM5017	17,2	5	17,2	1,6	0,0
MM5018	19,2	5	19,2	1,5	0,0
MM5019	18,0	5	28,0	36,6	0,0
MM5020	22,8	4	25,8	1.156,9	0,8
MM5021	18,6	5	19,0	407,5	0,0
MM5022	20,0	4	20,4	723,6	0,9
MM5023	18,0	5	18,8	4,1	0,0
MM5024	18,6	5	18,6	15,4	0,0
MM5025	18,4	5	26,6	9,0	0,0
MM5026	18,2	5	20,6	5,1	0,0
MM5027	18,0	5	18,0	1,5	0,0
MM5028	21,2	5	21,2	1,7	0,0
MM5029	18,0	5	18,0	1,5	0,0
MM5030	18,2	5	18,2	1,6	0,0
MM5031	17,8	5	26,4	14,8	0,0
MM5032	18,2	5	22,2	9,7	0,0
MM5033	20,8	5	20,8	1,8	0,0
MM5034	21,2	5	21,4	2,5	0,0
MM5035	20,2	5	20,2	1,9	0,0
MM5036	20,2	5	20,2	2,0	0,0
MM5037	26,6	5	41,4	300,9	0,0
MM5038	27,6	5	34,8	637,1	0,0
MM5039	27,8	5	28,0	7,7	0,0
MM5040	26,4	5	28,2	51,9	0,0
MM5041	28,2	5	28,6	3,5	0,0
MM5042	27,2	5	27,6	10,9	0,0
MM5043*	26,8	0	70,3	3.600,0	16,3
MM5044	26,6	0	51,2	3.600,0	14,5
MM5045	27,4	0	43,4	3.600,0	11,0
MM5046	25,8	0	43,4	3.600,0	12,2
MM5047	25,6	0	42,0	3.600,0	10,3
MM5048	28,0	0	38,0	3.600,0	9,3
MM5049	24,2	5	38,8	84,5	0,0
MM5050	29,0	5	34,8	14,9	0,0
MM5051	25,6	5	25,6	2,5	0,0
MM5052	26,2	5	27,6	7,1	0,0
MM5053	27,0	5	27,0	1,8	0,0
MM5054	24,2	5	24,4	1,9	0,0
MM5055	25,0	4	40,0	936,2	0,5

MM5056	24,8	2	34,8	2.763,5	3,8
MM5057	27,4	4	27,8	744,2	0,6
MM5058	29,2	4	31,4	743,8	0,7
MM5059	27,2	5	27,8	11,2	0,0
MM5060	28,0	5	28,0	8,8	0,0
MM5061	28,8	5	40,0	27,2	0,0
MM5062	28,2	5	31,4	21,3	0,0
MM5063	28,0	5	28,0	2,1	0,0
MM5064	27,2	5	27,4	2,9	0,0
MM5065	27,0	5	27,0	1,6	0,0
MM5066	27,6	5	27,6	2,1	0,0
MM5067	26,2	5	38,6	64,4	0,0
MM5068	25,0	5	30,4	30,8	0,0
MM5069	24,4	5	24,4	2,5	0,0
MM5070	25,0	5	26,0	8,3	0,0
MM5071	27,6	5	27,6	2,4	0,0
MM5072	25,6	5	25,6	2,5	0,0
MM5073	44,4	4	72,2	1.304,9	0,9
MM5074	43,2	2	57,0	2.207,4	1,4
MM5075	45,4	5	46,2	15,5	0,0
MM5076	38,2	5	43,0	207,4	0,0
MM5077	44,0	5	45,0	10,8	0,0
MM5078	45,2	5	46,0	13,7	0,0
MM5079*	-	-	-	-	-
MM5080	43,0	0	94,0	3.600,0	27,0
MM5081	50,0	1	69,4	2.915,7	17,8
MM5082	44,0	0	66,2	3.600,0	16,6
MM5083	45,2	2	57,0	2.685,0	6,8
MM5084	41,6	0	61,4	3.600,0	13,7
MM5085	42,8	5	73,2	231,0	0,0
MM5086	46,0	5	57,2	101,6	0,0
MM5087	43,0	5	43,0	5,9	0,0
MM5088	43,4	5	45,4	64,2	0,0
MM5089	47,4	5	47,4	6,0	0,0
MM5090	46,4	5	46,4	8,3	0,0
MM5091	40,6	4	67,0	1.133,5	0,3
MM5092*	41,8	2	57,5	2.762,2	1,3
MM5093	44,4	5	46,0	104,1	0,0
MM5094	39,6	3	46,6	2.092,8	1,6
MM5095	44,4	5	46,2	34,2	0,0
MM5096	41,6	4	43,6	807,6	0,8
MM5097	42,8	5	65,4	157,7	0,0
MM5098	44,8	5	56,4	101,9	0,0
MM5099	38,6	5	38,6	8,8	0,0
MM50100	44,8	5	46,2	17,5	0,0
MM50101	39,4	5	39,4	4,1	0,0
MM50102	38,4	5	38,4	5,7	0,0
MM50103	43,4	5	68,2	146,6	0,0
MM50104	45,4	5	54,2	113,6	0,0
MM50105	40,6	5	40,6	7,2	0,0
MM50106	44,0	5	47,4	52,2	0,0
MM50107	43,8	5	43,8	6,1	0,0
MM50108	43,8	5	43,8	7,4	0,0

Média	29,6	4,0	36,4	773,3	2,1
Desvio Padrão	10,3	1,8	15,7	1.306,9	4,9
Mínimo	17,2	0,0	17,2	1,5	0,0
Máximo	50,0	5,0	94,0	3.600,0	27,0

TABELA III
RESULTADOS PARA OS CONJUNTOS MMLIB100 OBTIDOS AO RESOLVER O
MODELO 2

Grupo	MMLIB100				
	LI	QTD	Sol.	Tempo	GAP
MM1001	23,0	0	35,4	3.600,0	7,4
MM1002	23,2	1	27,6	3.229,9	5,1
MM1003	22,4	4	22,6	734,4	0,9
MM1004	24,0	4	24,6	737,1	1,7
MM1005	20,8	4	21,6	1.382,7	1,0
MM1006	22,4	4	22,6	733,5	1,0
MM1007	23,8	0	53,0	3.600,0	12,0
MM1008	23,4	0	54,4	3.600,0	19,2
MM1009	23,6	0	34,6	3.600,0	5,5
MM10010	20,8	0	33,0	3.600,0	8,9
MM10011	23,6	1	35,8	3.004,4	4,2
MM10012	22,2	0	82,0	3.600,0	23,8
MM10013	22,4	5	30,8	45,7	0,0
MM10014	23,0	5	24,2	31,7	0,0
MM10015	24,2	5	24,2	8,7	0,0
MM10016	23,6	5	23,6	12,1	0,0
MM10017	21,4	5	21,4	8,4	0,0
MM10018	20,6	5	20,6	9,5	0,0
MM10019	22,4	5	29,2	119,9	0,0
MM10020	21,6	4	25,4	1.081,6	0,8
MM10021	22,0	5	22,4	501,3	0,0
MM10022	21,2	4	22,4	1.193,3	0,8
MM10023	23,0	5	23,2	17,3	0,0
MM10024	23,8	5	23,8	21,6	0,0
MM10025	26,4	5	31,2	46,9	0,0
MM10026	20,2	5	22,8	26,6	0,0
MM10027	24,2	5	24,2	8,9	0,0
MM10028	21,6	5	21,6	10,9	0,0
MM10029	23,4	5	23,4	8,5	0,0
MM10030	21,6	5	21,6	9,2	0,0
MM10031	24,2	5	29,6	79,1	0,0
MM10032	20,0	5	23,0	70,7	0,0
MM10033	22,8	5	22,8	12,0	0,0
MM10034	20,6	5	20,6	17,6	0,0
MM10035	21,2	5	21,2	9,9	0,0
MM10036	24,2	5	24,2	12,3	0,0
MM10037*	36,0	1	50,5	1.931,6	4,9
MM10038	32,2	1	41,6	2.984,9	5,1
MM10039	33,2	4	33,6	1.282,2	0,6
MM10040	30,8	1	33,8	2.891,6	3,6
MM10041	30,4	5	30,6	78,5	0,0
MM10042	34,2	5	34,2	41,4	0,0
MM10043*	36,0	0	321,0	3.600,0	79,4

MM10044	32,4	0	144,4	3.600,0	51,2	MM10097	52,6	4	77,2	1.701,2	0,3
MM10045*	32,0	0	108,5	3.600,0	32,5	MM10098	52,2	5	63,0	1.130,0	0,0
MM10046	30,0	0	73,4	3.600,0	26,8	MM10099	54,8	5	54,8	35,4	0,0
MM10047	36,6	0	103,8	3.600,0	24,5	MM100100	54,4	5	54,6	59,1	0,0
MM10048	31,4	0	108,0	3.600,0	30,5	MM100101	57,4	5	57,4	30,2	0,0
MM10049	30,6	5	43,4	215,3	0,0	MM100102	53,0	5	53,0	34,3	0,0
MM10050	29,8	5	36,0	405,1	0,0	MM100103*	52,5	2	81,5	1.414,6	0,0
MM10051	33,2	5	33,2	37,2	0,0	MM100104	54,4	4	67,8	1.596,2	0,3
MM10052	33,0	5	33,6	50,3	0,0	MM100105	58,0	5	58,0	55,2	0,0
MM10053	35,2	5	35,2	14,6	0,0	MM100106	57,0	4	57,2	776,8	0,3
MM10054	31,4	5	31,4	25,7	0,0	MM100107	57,0	4	57,2	776,8	0,3
MM10055*	31,0	0	48,3	3.600,0	4,6	MM100108	53,6	5	53,6	63,4	0,0
MM10056	31,0	1	40,6	3.053,6	2,4	Média	36,6	3,5	53,1	1.207,9	6,5
MM10057	34,2	5	34,2	73,3	0,0	Desvio	13,9	2,1	43,1	1.456,1	14,5
MM10058	31,4	3	34,2	1.506,5	1,8	Padrão					
MM10059	32,6	4	33,2	1.058,6	0,6	Mínimo	20,0	0,0	20,6	8,4	0,0
MM10060	30,8	5	31,0	96,7	0,0	Máximo	61,6	5,0	321,0	3.600,0	79,4
MM10061	30,8	5	44,4	216,5	0,0						
MM10062	32,0	5	35,6	323,5	0,0						
MM10063	33,2	5	33,2	29,7	0,0						
MM10064	31,2	5	31,2	30,2	0,0						
MM10065	33,0	5	33,0	17,0	0,0						
MM10066	31,8	5	31,8	17,9	0,0						
MM10067	34,6	5	44,2	861,2	0,0						
MM10068	31,6	5	36,4	511,1	0,0						
MM10069	31,8	5	31,8	41,8	0,0						
MM10070	31,0	5	31,0	47,5	0,0						
MM10071	32,2	5	32,2	30,3	0,0						
MM10072	32,6	5	32,6	33,7	0,0						
MM10073*	54,5	0	105,0	3.600,0	18,4						
MM10074*	57,0	0	92,0	3.600,0	24,2						
MM10075	59,8	5	60,0	239,9	0,0						
MM10076	54,8	1	62,6	2.894,5	7,5						
MM10077	61,6	5	62,2	66,2	0,0						
MM10078	51,8	5	52,4	321,2	0,0						
MM10079	-	-	-	-	-						
MM10080	51,6	0	228,6	3.600,0	62,3						
MM10081*	53,0	0	115,0	3.600,0	39,1						
MM10082	54,6	0	163,4	3.600,0	51,1						
MM10083	52,0	0	107,3	3.600,0	34,5						
MM10084	53,0	0	151,0	3.600,0	45,8						
MM10085*	59,3	3	75,7	1.814,2	0,0						
MM10086	59,6	5	67,8	847,9	0,0						
MM10087	58,4	5	58,4	43,0	0,0						
MM10088	51,0	5	52,6	129,7	0,0						
MM10089	55,2	5	55,2	31,9	0,0						
MM10090	57,6	5	57,6	35,5	0,0						
MM10091*	56,0	0	115,0	3.600,0	27,0						
MM10092*	56,5	0	78,8	3.600,0	10,5						
MM10093	58,4	5	58,4	78,6	0,0						
MM10094	53,6	2	61,0	2.401,4	7,0						
MM10095	56,0	5	56,0	188,2	0,0						
MM10096	52,0	2	55,0	2.185,3	2,1						

TABELA IV					
RESULTADOS PARA O CONJUNTO L30					
Grupo	L30				
	LI	QTD	Sol.	Tempo	GAP
L301*	51,3	9	55,9	0,5	0,0
L302	53,0	10	54,3	0,4	0,0
L303	70,0	10	71,1	0,3	0,0
L304*	62,6	9	62,6	0,2	0,0
L305*	57,2	9	72,8	22,6	0,0
L306	54,3	10	56,5	1,7	0,0
L307	51,4	10	53,2	0,5	0,0
L308	59,4	10	59,4	0,4	0,0
L309	59,1	10	78,8	99,4	0,0
L3010	55,4	10	56,3	2,4	0,0
L3011	64,9	10	65,2	1,9	0,0
L3012	58,1	10	58,1	0,7	0,0
L3013*	50,8	3	76,4	2.529,5	3,6
L3014*	53,9	9	55,3	3,7	0,0
L3015	61,2	10	61,4	1,0	0,0
L3016	53,0	10	53,0	0,6	0,0
L3017*	63,6	7	69,1	1,2	0,0
L3018	58,7	10	61,1	0,6	0,0
L3019	59,0	10	59,9	0,3	0,0
L3020	58,5	10	58,5	0,3	0,0
L3021*	59,8	9	76,0	12,4	0,0
L3022	59,6	10	61,6	1,4	0,0
L3023	64,2	10	65,1	0,5	0,0
L3024*	60,4	9	60,4	0,4	0,0
L3025	56,7	10	82,5	209,8	0,0
L3026*	62,0	9	63,6	1,3	0,0
L3027	65,7	10	65,7	0,7	0,0
L3028	67,6	10	67,6	0,6	0,0
L3029	59,8	6	92,8	1.804,1	4,5
L3030*	60,6	9	62,0	3,9	0,0
L3031*	64,4	9	64,7	1,2	0,0
L3032*	65,1	9	65,1	0,9	0,0

L3033	63,5	10	69,3	1,2	0,0
L3034*	66,3	9	68,9	0,6	0,0
L3035	68,1	10	68,7	0,3	0,0
L3036*	68,2	9	68,2	0,3	0,0
L3037*	63,7	9	84,4	40,4	0,0
L3038*	67,4	7	70,3	1,3	0,0
L3039	67,4	10	69,5	0,7	0,0
L3040*	66,1	7	66,1	0,4	0,0
L3041*	66,6	9	97,6	208,9	0,0
L3042	70,2	10	72,0	2,9	0,0
L3043*	67,1	9	67,7	0,9	0,0
L3044*	63,0	9	63,0	0,6	0,0
L3045*	69,9	7	103,1	1.222,0	1,4
L3046	65,7	10	67,1	5,0	0,0
L3047	65,9	10	66,5	1,5	0,0
L3048*	64,4	7	64,4	0,9	0,0
Média	61,8	9,1	67,4	129,0	0,2
Desvio Padrão	5,3	1,4	10,7	471,5	0,8
Mínimo	50,8	3,0	53,0	0,2	0,0
Máximo	70,2	10,0	103,1	2.529,5	4,5

TABELA V
RESULTADOS PARA O CONJUNTO L60

Grupo	L60				
	LI	QTD	Sol.	Tempo	GAP
L601*	81,7	9	85,9	14,4	0,0
L602	78,2	10	78,5	2,8	0,0
L603*	81,4	9	82,4	3,6	0,0
L604*	81,3	8	81,3	1,7	0,0
L605*	75,5	5	88,8	1.941,9	1,6
L606	79,3	10	79,5	7,5	0,0
L607*	82,8	9	82,8	3,0	0,0
L608*	82,3	9	82,3	3,8	0,0
L609*	82,1	0	112,7	3.600,0	12,0
L6010*	85,0	9	85,0	9,7	0,0
L6011*	80,3	8	80,3	7,6	0,0
L6012*	77,0	9	77,0	4,9	0,0
L6013*	74,4	0	146,7	3.600,0	27,7
L6014*	74,9	9	74,9	13,8	0,0
L6015*	85,9	8	85,9	11,0	0,0
L6016	75,7	10	75,7	8,1	0,0
L6017	83,0	10	87,5	24,1	0,0
L6018*	87,9	8	89,0	7,1	0,0
L6019	86,4	10	88,7	4,8	0,0
L6020*	89,0	7	89,0	2,1	0,0

L6021*	81,6	4	100,0	2.131,4	0,0
L6022*	83,0	9	84,3	8,4	0,0
L6023	86,8	10	87,1	5,6	0,0
L6024	84,1	10	84,1	4,0	0,0
L6025	81,0	0	129,4	3.600,0	12,5
L6026*	83,7	7	84,0	12,5	0,0
L6027*	87,9	9	87,9	7,2	0,0
L6028*	88,0	6	88,0	6,6	0,0
L6029	86,5	0	156,0	3.600,0	5,8
L6030	89,9	10	90,4	167,5	0,0
L6031*	84,7	9	84,7	11,8	0,0
L6032*	96,8	8	96,8	10,8	0,0
L6033*	99,4	9	105,1	32,8	0,0
L6034*	85,4	8	85,9	4,4	0,0
L6035*	89,3	9	91,6	6,1	0,0
L6036*	88,7	9	88,7	3,0	0,0
L6037*	91,3	2	114,0	1.739,5	1,6
L6038*	89,1	9	90,2	13,7	0,0
L6039*	94,9	7	95,0	7,3	0,0
L6040*	97,8	9	97,8	5,9	0,0
L6041*	97,1	1	135,8	3.414,4	14,9
L6042*	91,9	8	92,9	59,0	0,0
L6043*	89,5	6	89,7	10,7	0,0
L6044*	91,6	7	91,6	6,1	0,0
L6045	87,3	-	-	3.600,0	-
L6046*	88,3	9	90,1	552,2	0,0
L6047*	87,7	6	87,7	13,0	0,0
L6048*	91,0	7	91,0	8,4	0,0
Média	85,8	7,3	93,1	589,9	1,6
Desvio Padrão	6,0	3,0	17,3	1232,9	5,1
Mínimo	74,4	0,0	74,9	1,7	0,0
Máximo	99,4	10,0	156,0	3600,0	27,7

REFERÊNCIAS

- [1] E. R. Gafarov, A. A. Lazarev, and F. Werner, “On lower and upper bounds for the resource-constrained project scheduling problem”, Fakultät für Mathematik, Otto-von-Guericke-Universität, Tech. Rep. Series: 2010-08, 2010.
- [2] J. J. M. Mendes and J. F. Gonçalves, “Um algoritmo genético para o problema do sequenciamento de projetos com recursos limitados”, *Inv. Op.*, vol. 23, no. 2, pp. 179-195, 2003.
- [3] M. L. Pinedo, *Scheduling: theory, algorithms, and systems*, 4th ed., New York, NY, USA: Springer, 2012.

- [4] A. Singh, "Resource constrained multi-project scheduling with priority rules & analytic hierarchy process", *Procedia Eng.*, vol. 69, pp. 725-734, Mar. 2014, DOI. 10.1016.2014.03048.
- [5] E. Oztemel and A. A. Selam, "Bees algorithm for multi-mode, resource-constrained project scheduling in molding industry", *Compt. Ind. Eng.*, vol. 112, pp. 187-196, Oct. 2017, DOI. 10.1016.2017.08012.
- [6] P. J. Silva, C. S. Vieira, and A. L. Silva, "Modelagem e solução de problemas de sequenciamento em projetos com restrição de recursos", *Prod. Prod.*, vol. 18, no. 1, pp. 12-24, Feb. 2017, DOI. 10.22456.1983-8026.65686.
- [7] S. Hartmann and D. Briskorn, "A survey of variants and extensions of the resource-constrained project scheduling problem", *Eur. J. Oper. Res.*, vol. 207, no. 1, pp. 1-14, Nov. 2010, DOI. 10.1016.2009.11005.
- [8] R. Heilmann, "A branch-and-bound procedure for the multi-mode resource-constrained project scheduling problem with minimum and maximum time lags", *Eur. J. Oper. Res.*, vol. 144, pp. 348-365, Jan. 2003, DOI. 10.1016.S0377-2217(02)00136-4.
- [9] D. S. Yamashita and R. Morabito, "Um algoritmo branch-and-bound para o problema de programação de projetos com custo de disponibilidade de recursos e múltiplos modos", *Gest. Prod.*, vol. 14, no. 3, pp. 545-555, Sept. 2007, DOI. 10.1590.S0104-530X2007000300010.
- [10] I. Correia and F. Saldanha-da-Gama, "The impact of fixed and variable costs in a multi-skill project scheduling problem: An empirical study", *Compt. Ind. Eng.*, vol. 72, pp. 230-238, Jun. 2014, DOI. 10.1016.2014.03020.
- [11] S. Javanmard, B. A. Nadjafi, and S. T. A. Niaki, "Preemptive multi-skilled resource investment project scheduling problem: Mathematical modelling and solution approaches", *Comput. Chem. Eng.*, vol. 96, pp. 55-68, Jan. 2017, DOI. 10.1016.2016.11001.
- [12] B. F. Almeida, I. Correia, and F. Saldanha-da-Gama, "Priority-based heuristics for the multi-skill resource constrained project scheduling problem", *Expert Syst. Appl.*, vol. 57, pp. 91-103, Sept. 2016, DOI. 10.1016.2016.03017.
- [13] R. M. Silva, C. Cubillos, and D. C. Paniagua, "A constructive heuristic for solving the Job-Shop Scheduling Problem", *IEEE Lat. Am. T.*, vol. 14, no. 6, pp. 2758-2763, Aug. 2016, DOI. 10.1109.2016.7555250.
- [14] A. E. F. Muritiba, C. D. Rodrigues, and F. A. Costa, "A path-relinking algorithm for the multi-mode resource constrained project scheduling problem", *Comput. Oper. Res.*, vol. 92, pp. 145-154, Apr. 2018, DOI. 10.1016.2018.01001.
- [15] L. Wang and X. Zheng, "A knowledge-guided multi-objective fruit fly optimization algorithm for the multi-skill resource constrained project scheduling problem", *Swarm Evol. Comput.*, vol. 38, pp. 54-63, Feb. 2018, DOI. 10.1016.2017.06001.
- [16] T. Schmidt, C. Scarpin, G. Loch, and N. da Silva, "An Application of the Local Branching to the Identical Parallel Machines Scheduling Problem", *IEEE Lat. Am. T.*, vol. 17, no. 6, pp. 1047-1054, Nov. 2019, DOI. 10.1109.2019.8896828.
- [17] J. Z. Gomes, J. L. Montenegro, J. V. C. Santos, J. L. V. Barbosa, and C. A. Costa, "A Strategy Using Continuous Simulation to Mitigate Effort Estimation Risks in Software Projects", *IEEE Lat. Am. T.*, vol. 17, no. 8, pp. 1390-1398, Aug. 2019, DOI. 10.1109.2019.8932373.
- [18] J. C. Herz, "Recursive computational procedure for two-dimensional stock cutting", *IBM J. Res. Dev.*, vol. 16, no. 5, pp. 462-469, Sept. 1972, DOI. 10.1147.165.0462.
- [19] G. F. Cintra, F. K. Miyazawa, Y. Wakabayashi, and E. C. Xavier, "Algorithms for two-dimensional cutting stock and strip packing problems using dynamic programming and column generation", *Eur. J. Oper. Res.*, vol. 191, no. 1, pp. 61-85, Nov. 2008, DOI. 10.1016.2007.08007.
- [20] A. Drexler and A. Kimms, "Optimization guided lower and upper bounds for the resource investment problem", *J. Oper. Res. Soc.*, vol. 52, no. 3, pp. 340-351, Oct. 2001, DOI. 10.1057.2601099.
- [21] M. Mika, R. Rózycki, and G. Waligóra, "Some new concepts of setup costs in multimode resource-constrained project scheduling problem", in *17th International Conference on Methods & Models in Automation & Robotics (MMAR)*, Miedzyzdroje, Poland, Aug. 2012, pp. 470-474.
- [22] A. A. B. Pritsker, L. J. Watters, P. M. Wolfe, "Multi-project scheduling with limited resources: a zero-one programming approach", *Manag. Sci.* vol. 16, no. 1, pp. 93-108, Sept. 1969.
- [23] P. B. Myszkowski, M. Laszczyk, I. Nikulin, and M. Skowroński, "iMOPSE: a library for bicriteria optimization in multi-skill resource-constrained project scheduling problem", *Soft Comput.*, vol. 22, no. 10, pp. 3397-3410, May 2019, DOI. 10.1007.s00500-017-2997-5.

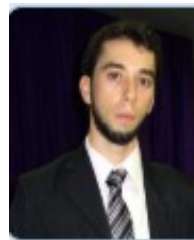
- [24] V. V. Peteghem and M. Vanhoucke, "An experimental investigation of metaheuristics for the multi-mode resource-constrained project scheduling problem on new dataset instances", *Eur. J. Oper. Res.*, vol. 235, no. 1, pp. 62-72, May 2014, DOI. 10.1016.2013.10.012.



L. V. Melo nasceu em Catalão, Goiás, Brasil. Ela possui graduação em Engenharia de Produção (2016) e mestrado em Engenharia de Produção (2018), ambos obtidos na Universidade Federal de Goiás, Regional Catalão.

Ela possui experiência como Assistente Administrativa e atualmente é professora na Universidade Federal de Goiás,

Regional Catalão. Seus interesses de pesquisa estão relacionados a resolução de problemas de programação da produção.



T. A. Queiroz nasceu em Trindade, Goiás, Brasil. Ele possui graduação em Ciência da Computação pela Universidade Federal da Goiás (2007), mestrado em Engenharia Mecânica pela Universidade Federal de Uberlândia (2008) e doutorado em Ciência da Computação pelo Instituto de Computação, da Universidade Estadual de Campinas (2010).

Ele é professor na Universidade Federal de Goiás, Regional Catalão, desde 2010. Seus interesses de pesquisa estão relacionados com o desenvolvimento de métodos exatos e heurísticos para problemas de otimização combinatória, em especial, problemas de corte e empacotamento, roteamento de veículos, programação da produção, localização de facilidades e suas integrações. Ele é editor associado do periódico Pesquisa Operacional para o Desenvolvimento.