

# Multi-Period Service Scheduling Problems: A New Model and Heuristic Approaches of Relax-and-Fix and Fix-and-Optimize

J. Oliveira, and C. Scarpin

**Abstract**— **Multi-Period Service Scheduling Problem – MSSP** is described as a category of problems that arise in the context of scheduling services. In these problems, customers have a periodic service to be attended, a finite number of times, within the time horizon. The **Multi-Period Multi-Service Scheduling Problem – MMSSP** is presented as an alternative formulation for MSSP, in which the resolution of an MMSSP is equivalent to the resolution of multiple MSSP, demonstrating the complexity of solving this pattern. The presented formulation can lead several applications in different areas. To solve this new pattern, the **Relax-and-Fix** and **Fix-and-Optimize** strategies are used. In these heuristics, part of the entire variables are fixed and/or relaxed interactively. To assess the efficiency of the strategies, a set of 150 small, medium and large instances are generated and resolved. The results of extensive experiments demonstrate that heuristics produce quality solutions, highlighting a good balance between optimization and computational time.

**Index Terms**— Heuristic, Multi-period service scheduling, Relax-and-fix, Fix-and-optimize, Scheduling.

## I. INTRODUÇÃO

Problemas de Agendamentos de Serviços Multiperíodos - MSSP (do inglês: *Multi-Period Service Scheduling Problem*), são descritos pela primeira vez no trabalho seminal de [1] e posteriormente nos trabalhos [2], [3]. Nessa classe de problemas um conjunto de clientes possui demandas (serviços) periódicas a serem atendidas em número finito de vezes dentro de um horizonte de planejamento. Os autores destacam que os serviços são vistos como visitas, podendo ser para coleta, entrega ou manutenção. Nenhum cliente possui mais de uma demanda por período e a execução de um serviço não excede um período do horizonte de tempo. É dado, ainda, um conjunto de operadores de capacidades fixas para satisfazer as demandas. A capacidade refere-se ao número de clientes que um operador pode atender em um único período. O tempo de duração de cada serviço é considerado homogêneo, ou seja, todas as demandas possuem o mesmo tempo de duração. Toda vez que um operador é acionado para atender um ou mais serviços (respeitando sua capacidade) gera-se um custo fixo de operação.

J. D. Oliveira, Universidade Estadual do Centro-Oeste (UNICENTRO), membro Research Group of Technology Applied to Optimization (GTAO), Irati, Paraná, Brasil (e-mail: jailson-de-oliveira@hotmail.com).

C. T. Scarpin, Universidade Federal do Paraná (UFPR), coordenador Research Group of Technology Applied to Optimization (GTAO), Curitiba, Paraná, Brasil (e-mail: cassiusts@gmail.com).

O objetivo é determinar para cada cliente os períodos nos quais ele terá sua demanda atendida por um operador (período de serviço), de tal forma que todas as demandas dos clientes sejam satisfeitas e o número total de operadores utilizados ao longo do horizonte de tempo seja mínimo [3].

O MSSP surge em uma variedade de aplicações, desde o agendamento de entregas até a telecomunicação. Os autores [1], [2], [3] apresentam uma série de práticas, nas quais os MSSP aparecem como componentes centrais ou secundários. A principal aplicação destacada pelos autores está relacionada à logística de entrega ou coleta de produtos, matéria prima ou resíduos. Nesses problemas, os clientes possuem demandas de um produto ou recolhimento de resíduos, que devem ser atendidos periodicamente devido às suas capacidades de armazenamento limitadas. Essa aplicação surge, por exemplo, na logística reversa e no reabastecimento de estoque do comércio varejista.

Outras aplicações ocorrem no agendamento de visitas para a inspeção e manutenção preventiva de equipamentos, em que técnicos são requeridos para execução de inspeções periódicas; no gerenciamento e reabastecimento de estoque, sob a responsabilidade do fornecedor; e na transmissão de páginas de informações, as quais devem ser programadas periodicamente nos canais de transmissão. Em algumas aplicações o MSSP não surge como componente central mas, como subproblema do problema geral [2]. Por exemplo, em problemas de designer de territórios de serviços multiperíodos (*Multi-period Service Territory Design Problem*) [4], [5], um dos subproblemas abordados é o de agendamento. Neste subproblema, os distritos já foram estabelecidos e uma agenda válida de visitas deve ser determinada para cada distrito individualmente, respeitando as especificidades de cada cliente. Outros problemas nos quais o MSSP é abordado como componente secundário são: Problemas de Roteamento Periódico de Veículos [6]; Problemas de Roteamento Periódico em Arcos Capacitado [7]; e em Problemas de Roteamento de Inventário [8].

Um dos diferenciais do MSSP é a característica periódica do problema. Ao invés das decisões serem tomadas apenas no nível operacional, tem-se decisões de níveis tático e operacionais, para horizontes de tempo maiores do que um período e com restrições de frequência. A frequência é a principal característica dos problemas com serviços periódicos. A frequência abordada nos modelos descritos na literatura é baseada em intervalos de tempo, isto é, são aquelas que utilizam um tempo fixo (política regular) entre dois tratamentos consecutivos. Em [9] é apresentado um problema de roteamento periódico em arcos capacitados (*Periodic Capacitated Arc*

*Routing Problem*) que emprega intervalos fixos entre dois serviços sucessivos. O modelo proposto em [9] é criado visando a manutenção preditiva e preventiva de linhas férreas. Uma série de aplicações para problemas de programação de intervalos fixos (*Fixed Interval Scheduling*) são descritas no trabalho de [8]. Destaca-se: planejamento de frota; atribuição de aeronaves a portões [10]; planejamento de manutenção em plataformas de petróleo [11]; programação de motoristas [12]; transmissão de dados; e comparação em biologia molecular [13].

Outra característica do MSSP refere-se às demandas que não possuem níveis de prioridade e são consideradas de período único. Isto significa que a execução de um serviço não excede um período do horizonte de tempo e, nenhum cliente possui mais de uma demanda por período. Em problemas periódicos, a demanda interfere na periodicidade do atendimento em consequência da maneira como as frequências são abordadas.

Os modelos descritos por [1], [2] para o MSSP com política de serviço periódica não são capazes de abordar mais de um serviço (tarefa). Neste artigo é introduzido uma nova formulação que generaliza os modelos já propostos em [1], [2], aproximando o modelo de aplicações reais e complexas. Essa nova formulação foi denominada Problema de Agendamento de Múltiplos Serviços Multiperíodo – MMSSP (*Multi-Period Multi-Service Scheduling Problem*). O agendamento de vários serviços é mais complexo em comparação ao agendamento de um único serviço, uma vez que a resolução de um MMSSP equivale a resolução de múltiplos MSSP.

O MMSSP envolve a atribuição de um conjunto de operadores (recursos) para atender um conjunto de atividades requeridas periodicamente por um cliente, dentro de um horizonte de planejamento. Neste trabalho, considerou-se que os operadores possuem diferentes qualificações e capacidades para atender os clientes. A execução de um serviço não excede um período do horizonte de tempo, sendo que, um cliente pode possuir mais de uma demanda por período.

O MMSSP é um problema encontrado especialmente no contexto de empresas prestadoras de serviços terceirizados. A terceirização de serviços frequentemente faz parte das estratégias corporativas por permite que os gestores permaneçam focados na atividade principal da empresa. Em alguns casos, uma única empresa pode prestar diversos serviços para seus clientes tais como: serviços de manutenção, limpeza, jardinagem, recepção, portaria, segurança, serviços de recursos humanos, reabastecimento e/ou coleta entre outros. Dessa forma, a terceirização de serviços pode garantir que a empresa alcance melhor produtividade na sua área de atuação. Referência [14] destaca que a terceirização de serviços nos últimos anos se tornou um importante recurso estratégico para a gestão dos custos. Outras aplicações para o MMSSP surgem em problemas de agendamento de múltiplas consultas em hospitais, agendamento de vários processadores (em computação) e agendamento de serviços de atendimento social.

Resolver de forma exata problemas que envolvem restrições operacionais que limitam a capacidade de atendimento como os descritos é uma tarefa complexa e difícil, mesmo para *solvers* modernos e específicos para resolução de Problemas de Programação Linear Inteira – PLI. Sob a ótica da otimização, sabe-se que a dificuldade de resolução de um PLI pode estar associada ao número de variáveis inteiras. Quanto maior esse

número, mais difícil pode ser sua resolução de forma exata. Consequentemente o uso de *solvers* tornam-se, em muitos casos, não operacional para problemas reais e instâncias de grande porte, como os casos tratados nesse trabalho.

Apesar de sua alta relevância prática, o MMSSP carece de técnicas e métodos de resolução eficazes em tempos computacionais aceitáveis para cada problema e sua tomada de decisão. Em virtude dessa dificuldade na obtenção de soluções para o modelo proposto, foram abordados métodos heurísticos. Especificamente abordou-se as heurísticas *relax-and-fix* (R&F) e *fix-and-optimize* (F&O) propostas em [15]. Ambas as heurísticas são métodos de decomposição do problema em subproblemas menores e mais fáceis de serem resolvidos com os mesmos recursos computacionais iniciais.

As principais contribuições desse trabalho são: a) Introduzir uma formulação mais próxima da realidade para o MSSP; b) Propor de forma inédita o uso da heurística *relax-and-fix* e *fix-and-optimize* para o MMSSP; c) Mostrar a eficiência das heurísticas por meio de extensas experiências computacionais em instâncias de pequeno, médio e grande porte.

O artigo está organizado da seguinte forma. Na próxima seção, a formulação para o MSSP proposta na literatura em [1], [2] é pormenorizada, bem como uma formulação alternativa apresentada. A seção 3 descreve as heurísticas R&F e F&O e métodos de decomposições abordados. A metodologia utilizada é exposta na seção 4. Na seção 5 são apresentados os resultados computacionais para mostrar a eficiência das heurísticas e, por fim, na seção 6 as considerações finais.

## II. MODELOS

Nesta seção é descrita a formulação MSSP e a formulação MMSSP, sendo este o foco principal deste artigo. Dado um horizonte de planejamento  $T$ ,  $|T|$  expressa a duração do horizonte de tempo. O conjunto de clientes é denotado por  $I$  e  $t_i \in \mathbb{N}$  indica a duração entre dois atendimentos consecutivos ao cliente  $i \in I$ . O conjunto de operadores (recursos) é representado por  $K$  e  $Q$  expressa a capacidade dos operadores por período  $t \in T$ , a capacidade é fixa e idêntica para todos os operadores.

O objetivo no MSSP é definir uma agenda de serviços que minimize o número total de operadores, onde para cada cliente  $i$  é determinada uma sequência ordenada de visitas. O período entre duas visitas sucessivas não pode exceder o intervalo de serviço  $t_i$ . Essa sequência é denominada calendário de visitas do cliente  $i$  ( $C_i \subset T$ ).

A seguir é detalhado o modelo abordado em [1], [2].

Variáveis de decisão:

$$\forall i \in I, t \in T,$$

$$x_i^t = \begin{cases} 1, & \text{se o cliente } i \text{ é visitado no período } t \\ 0, & \text{caso contrário} \end{cases}$$

$$\forall i \in I, k \in K, t \in T,$$

$$y_{ik}^t = \begin{cases} 1, & \text{se o cliente } i \text{ é visitado pelo operador } k \\ & \text{no período } t \\ 0, & \text{caso contrário} \end{cases}$$

$$\forall k \in K, t \in T$$

$$z_k^t = \begin{cases} 1, & \text{se o operador } k \text{ é utilizado no período } t \\ 0, & \text{caso contrário} \end{cases}$$

Modelo MSSP:

$$\min \sum_{t \in T} \sum_{k \in K} z_k^t \quad (1)$$

Sujeito à:

$$\sum_{t=1}^{t_i} x_i^t = 1 \quad \forall i \in I \quad (2)$$

$$x_i^t = x_i^{t+t_i} \quad \forall i \in I, t \in \{1, \dots, |T| - t_i\} \quad (3)$$

$$x_i^t = \sum_{k \in K} y_{ik}^t \quad \forall i \in I, t \in T \quad (4)$$

$$\sum_{i \in I} y_{ik}^t \leq Qz_k^t \quad \forall k \in K, t \in T \quad (5)$$

$$Qz_k^t \leq \sum_{i \in I} y_{i,k-1}^t \quad \forall k \in K \setminus \{1\}, t \in T \quad (6)$$

$$x_i^t, y_{ik}^t, z_k^t \in \{0,1\} \quad \forall i \in I, k \in K, t \in T \quad (7)$$

O objetivo (1) minimiza o número total de operadores utilizados no horizonte de tempo  $T$ . As restrições (2) garantem que a primeira visita deva ocorrer até  $t_i$ , já as visitas consecutivas, são asseguradas em (3). As restrições (4) garantem que, se um cliente é visitado em um período  $t$ , sua visita é atribuída a um operador  $k$ . As restrições (5) são referentes a capacidade dos operadores, que deve ser respeitada em todos os períodos de tempos  $t \in T$ . As restrições (6) impõem que o operador  $k$  só seja acionado no período  $t$  se os operadores  $1, \dots, k-1$  tiverem suas capacidades esgotadas no período  $t$ . Por fim, em (7) define-se o domínio das variáveis.

A seguir é descrito a formulação matemática que generaliza o modelo MSSP, sendo esta a contribuição proposta nesse artigo. Para tal, faz-se necessário redefinir algumas notações e definir novos conjuntos: o conjunto  $J$  indica o conjunto de atividades/serviços,  $t_{ij} \in \mathbb{N}$  indica a duração máxima entre duas visitas consecutivas ao cliente  $i$  para a execução da atividade  $j$ ;  $Q_{kj}$  é a capacidade do operador  $k$  para atender a atividade  $j$ ,  $Q_{kj}$  expressa o número de clientes que o operador  $k$  pode atender em um período  $t \in T$ , podendo variar de acordo com o tipo de serviço atendido pelo operador. Pressupõe-se que todos os clientes  $i$  foram atendidos antes do início do horizonte de tempo  $T$ . Sejam, então, os seguintes conjuntos:

$A_i$ : conjunto de atividades  $j$  demandadas pelo cliente  $i$ ,  $\forall i \in I$ ;

$B_j$ : conjunto de operadores  $k$  que podem atender ao serviço  $j$ ,  $\forall j \in J$ ;

$C_k$ : conjunto de atividades  $j$  servidas pelo operador  $k$ ,  $\forall k \in K$ ;

$D_j$ : conjunto de clientes  $i$  que requerem a atividade  $j$ ,  $\forall j \in J$ .

As variáveis para MMSSP são:

$$\forall i \in I, j \in J, t \in T$$

$$x_{ij}^t = \begin{cases} 1, & \text{se o cliente } i \text{ tem a atividade } j \text{ servida no} \\ & \text{período } t \\ 0, & \text{caso contrário} \end{cases}$$

$$\forall i \in I, k \in K, j \in J, t \in T$$

$$y_{ikj}^t = \begin{cases} 1, & \text{se o cliente } i \text{ é visitado pelo operador } k \\ & \text{para atender a atividade } j \text{ em } t \\ 0, & \text{caso contrário} \end{cases}$$

$$\forall k \in K, j \in J, t \in T$$

$$z_{kj}^t = \begin{cases} 1, & \text{se o operador } k \text{ é utilizado para realizar a} \\ & \text{atividade } j \text{ no período } t \\ 0, & \text{caso contrário} \end{cases}$$

$$\forall k \in K, t \in T$$

$$o_k^t = \begin{cases} 1, & \text{se o operador } k \text{ é utilizado no tempo } t \text{ para} \\ & \text{realizar qualquer atividade } j \\ 0, & \text{caso contrário} \end{cases}$$

Modelo MMSSP baseado em clientes:

$$\min \sum_{t \in T} \sum_{k \in K} o_k^t \quad (8)$$

Sujeito à:

$$\sum_{t=1}^{t_{ij}} x_{ij}^t = 1 \quad \forall i \in I, j \in A_i \quad (9)$$

$$x_{ij}^t = x_{ij}^{t+t_{ij}} \quad \forall i \in I, j \in A_i, t \in \{1, \dots, |T| - t_{ij}\} \quad (10)$$

$$x_{ij}^t = \sum_{k \in B_j} y_{ikj}^t \quad \forall i \in I, j \in A_i, t \in T \quad (11)$$

$$\sum_{i \in D_j} y_{ikj}^t \leq Q_{kj} z_{kj}^t \quad \forall k \in K, j \in C_k, t \in T \quad (12)$$

$$\sum_{i \in I} \sum_{j \in A_i} y_{ikj}^t \leq M * o_k^t \quad \forall k \in K \setminus \{1\}, t \in T \quad (13)$$

$$x_{ij}^t, y_{ikj}^t, z_{kj}^t, o_k^t \in \{0,1\} \quad \forall i \in I, j \in J, k \in K, t \in T \quad (14)$$

Na formulação proposta, a sentença (8) descreve a função objetivo que consiste em minimizar o número total de operadores, utilizados no horizonte de tempo  $T$ . As restrições (9) garantem que a primeira visita deve ocorrer até  $t_{ij}$  para cada cliente  $i$  para a atividade  $j$  e as demais visitas devem ser respeitadas devido as restrições (10). As restrições (11) asseguram que se um cliente  $i$  é visitado em um período  $t$  para ter sua atividade  $j$  servida, algum operador  $k$  deve realizar a tarefa. As restrições (12) são referentes a limitação das capacidades dos operadores. As restrições (13) efetivam a utilização do operador  $k$ , se o mesmo realizou o atendimento de qualquer atividade  $j$  no período  $t$ , é somado a função objetivo o uso do operador  $k$ . O valor  $M$  é limitado inferiormente pela soma de todos os clientes que dependem da atividade  $j$ , as quais são atendidas pelo operador  $k$ ,  $\forall j \in C_k$ . As demais restrições (14) definem o domínio das variáveis.

As restrições (6) do modelo MSSP apresentado em [1] são excluídas nos modelos aqui propostos, uma vez que, em algumas instâncias se faz necessário o uso de um operador  $k$ , antes mesmo da capacidade do operador  $k - 1$  ser totalmente utilizada. Por exemplo, dado uma instância onde o operador  $k - 1$  não atenda a atividade  $j$ , logo é necessário acionar um operador  $k$  ( $k, k + 1, \dots, K$ ) para atender esta atividade.

### III. ABORDAGEM DE SOLUÇÃO

A heurística *relax-and-fix* (R&F) descrita em [15] consiste na decomposição de um modelo de programação inteira mista (*Mixed Integer Problem* - MIP), de larga escala, em vários subproblemas menores disjuntos. Estratégias do tipo R&F utilizam-se de relaxação linear como parte da abordagem de solução para tornar a resolução dos subproblemas mais “fácil”. No entanto, a heurística não garante a resolução do problema de forma ótima. A heurística R&F tem sido amplamente utilizada na solução de diferentes problemas. Por exemplo, problemas de dimensionamento e sequenciamento da produção em dois estágios [16], problemas de dimensionamento de lote capacitado de múltiplos produtos [17], problemas de árbitro itinerante [18], problemas de roteamento em arcos capacitados e periódicos [19], entre outros.

Seja um MIP, em que um conjunto de variáveis binárias e/ou inteiras são particionadas em  $R$  conjuntos disjuntos  $Q^1, Q^2, \dots, Q^R$ . Então, resolve-se sequencialmente os  $R$  MIP's, denotados por  $MIP^r$  com  $1 \leq r \leq R$ . Observa-se que  $R$  determina o número iterações da heurística.

Primeiro resolve-se o problema  $MIP^1$  semi-relaxado, as variáveis pertencentes  $Q^1$  são mantidas inteiras e relaxa-se as demais variáveis. Ao término desta resolução fixa-se as variáveis do conjunto  $Q^1$ . No problema  $MIP^2$ , as variáveis pertencentes a  $Q^2$  são mantidas inteiras, relaxa-se as demais e, após a resolução do  $MIP^2$ , fixa-se as variáveis do conjunto  $Q^2$ . O processo encerra quando todos os subproblemas  $MIP^r$  tiverem sido resolvidos ou quando um subproblema for inatível. Uma descrição formal do processo pode ser observada no Algoritmo 1.

Nesse trabalho foram implementadas três heurísticas R&F. Todas as propostas são baseadas na decomposição por períodos:

---

#### Algoritmo 1: Relax-and-Fix Clássico

---

Particione as variáveis em  $R$  conjuntos disjuntos  $Q^r, r = 1, \dots, R$

**Para**  $r = 1, \dots, R$  **faça**

Relaxe as variáveis pertencentes a  $Q^i, i = r + 1, \dots, R$

**se**  $r > 1$

Fixe as variáveis pertencentes a  $Q^i, i = 1, \dots, r - 1$ , conforme a solução dos  $MIP^r$  anteriores

**Fim Se**

Resolva o  $MIP^r$  gerado

**se**  $MIP^r$  inatível **então**

O particionamento não foi capaz de determinar uma solução viável

**Exit para**

**Fim Se**

$r = r + 1$

**Fim Para**

---

*Relax-and-Fix Forward* (R&F-F): o MIP é particionado em  $|T|$  conjuntos disjuntos e os  $MIP^r$  ( $r = 1, \dots, |T|$ ) são solucionados em ordem cronológica do horizonte de tempo.

*Relax-and-Fix Overlapping* (R&F-O): Essa estratégia é baseada na sobreposição de partições. Para um  $MIP^r$ , com  $1 \leq r \leq R - 1$ , as variáveis pertencentes as partições  $Q^r$  e  $Q^{r+1}$  são mantidas binárias e as demais variáveis subsequentes a esses períodos são relaxadas. Na iteração seguinte, as variáveis do conjunto  $Q^r$  são fixadas, as variáveis de  $Q^{r+1}$  e  $Q^{r+2}$  mantidas binárias e as demais variáveis relaxadas. O processo ocorre até que o último subproblema  $MIP^{R-1}$  seja resolvido.

*Relax-and-Fix Overlapping 2* (R&F-O2): Similar à estratégia anterior. Para um subproblema  $MIP^r$ , com  $1 \leq r \leq R - 2$ , as variáveis pertencentes as partições  $Q^r, Q^{r+1}$  e  $Q^{r+2}$  são mantidas binárias e as demais variáveis subsequentes a esses períodos são relaxadas. Para o próximo subproblema  $MIP^{r+1}$ , as variáveis do conjunto  $Q^r$  são fixadas, as variáveis de  $Q^{r+1}, Q^{r+2}$  e  $Q^{r+3}$  são definidas como binárias e as demais variáveis relaxadas. A heurística é finalizada com a resolução do  $MIP^{R-2}$ . Para maiores detalhes das estratégias R&F consultar [15], [19].

A heurística *Exchange* foi descrita em [20] e, posteriormente, denominada de *fix-and-optimize* (F&O) nos trabalhos de [21], [22]. Assim como na heurística R&F, esta heurística também é baseada na decomposição do problema original em subproblemas menores, mas ao contrário das estratégias R&F que utilizam relaxação linear, as estratégias F&O utilizam a fixação de um conjunto de variáveis em seus valores inteiros, dado por uma solução inicial. A heurística F&O tem sido utilizada na melhoria de soluções de diferentes problemas, principalmente em problemas de dimensionamento de lotes [17], [21].

A heurística F&O é constituída dos seguintes procedimentos: 1) dada uma solução inicial e um conjunto de partições para as variáveis inteiras, inicia-se o processo iterativo; 2) escolhe-se uma partição das variáveis, as quais serão liberadas para otimização, enquanto que as demais variáveis são mantidas com os valores fixados pela solução incumbente inicial. Caso a solução do subproblema seja melhor que a solução atual, esta será substituída; 3) na iteração seguinte, uma nova partição é escolhida para otimização e as demais permanecem com os valores fixados no valor da solução incumbente atual; 4) a

heurística é finalizada quando um critério preestabelecido é atingido, por exemplo, não houver mais melhorias nas soluções dos subproblemas, um tempo limite é atingido ou atingiu o final das partições. O Algoritmo 2 descreve a heurística F&O.

---

**Algoritmo 2:** Fix-and-Optimize Clássico
 

---

$S \leftarrow$  Solução inicial  
 Particione as variáveis em  $R$  conjuntos disjuntos  $Q^r$ ,  $r = 1, \dots, R$   
**Para**  $r = 1, \dots, R$  **faça**  
   Libere para otimização variáveis pertencentes a  $Q^r$   
   Fixe as variáveis pertencentes a  $Q - Q^r$  no valor da solução atual  $S$   
   Resolva o MIP<sup>r</sup> gerado, obtendo a solução  $S^r$   
   **Se**  $S^r < S$  **então**  
      $S \leftarrow S^r$   
   **Fim Se**  
    $r = r + 1$   
**Fim Para**

---

Nesse trabalho, optou-se em aplicar quatro estratégias do tipo F&O sob a melhor solução factível determinada pelas estratégias R&F descritas anteriormente. Três estratégias são baseadas na decomposição por clientes e uma na decomposição por atividades. A ordem que os calendários são liberados para otimização respeitam a sequência  $i = 1, \dots, I$ .

*Fix-and-Optimize Forward Customers* (F&O-FC): Havendo uma solução inicial factível, os calendários dos clientes  $C_{i,j}$  são liberados sequencialmente para otimização. Na primeira iteração o calendário do cliente  $C_{1,j}, \forall j \in J$ , é liberado para otimização. O procedimento é repetido até que o calendário do último cliente seja liberado para otimização;

*Fix-and-Optimize Overlapping Customers* (F&O-OC): A estratégia R&F-OC é baseada na sobreposição dos clientes. São liberados dois calendários consecutivos de clientes para otimização  $C_{i,j}$  e  $C_{i+1,j}$ ,  $1 \leq i \leq I - 1$  e  $\forall j \in J$ . Na iteração subsequente os calendários  $C_{i+1,j}$  e  $C_{i+2,j}$ ,  $\forall j \in J$  são liberados para otimização. O objetivo nessa estratégia é aumentar a melhoria obtida na resolução de cada subproblema. A estratégia encerra-se quando os calendários  $C_{I-1,j}$  e  $C_{I,j}$  tenham sido otimizados;

*Fix-and-Optimize Overlapping Customers 2* (F&O-OC2): Essa estratégia é similar à estratégia F&O-OC. Nessa proposta são liberados três calendários consecutivos para otimização  $C_{i,j}$ ,  $C_{i+1,j}$  e  $C_{i+2,j}$ ,  $1 \leq i \leq I - 2$  e  $\forall j \in J$ . Observa-se que, aumentar o número de calendários livres para otimização, consequentemente, aumenta a dificuldade de resolução do subproblema;

*Fix-and-Optimize Activities* (F&O-A): A decomposição nessa estratégia ocorre por atividade, sendo elas liberadas sequencialmente para otimização ( $j = 1, \dots, J$ ). Na primeira iteração todas as variáveis dependentes do índice  $j = 1$  são liberadas para otimização e as demais fixadas. O procedimento é repetido até que a última atividade seja liberada para otimização.

#### IV. METODOLOGIA

Nessa seção é descrito a metodologia de solução utilizado para o modelo MMSSP, assim como, instâncias e parâmetros adotados para as heurísticas R&F e F&O.

Gerou-se um conjunto de 150 instâncias de *benchmark* de forma aleatória, uma vez que, no melhor de nosso conhecimento, não há instâncias disponíveis na literatura. Para o número de clientes foi determinado  $I \in \{10, 30, 50, 80, 100\}$  e o horizonte de tempo está relacionado a um mês  $|T| = 30$ . Para os possíveis intervalos de atendimentos  $t_{ij}$  dos clientes foi estabelecido  $t_{ij} \in \{4, 5, 7\}$ . Utilizou-se duas capacidades distintas para os operadores,  $Q \in \{6, 12\}$  e, por fim, o conjunto de atividades  $J \in \{2, 3, 6\}$ .

Para cada combinação  $|T|, |I|, |J|$  e  $|Q|$ , foram geradas 5 instâncias distintas e os intervalos de serviços para cada cliente foram determinados de forma aleatória uniformemente distribuídos. Os conjuntos de instâncias são indicados por Tempo\_nºcliente\_nºatividade\_capacidade. Por exemplo, 50 clientes, 2 atividades e capacidade 6 é denotado por T30\_I50\_J2\_Q6.

A metodologia utilizada é exposta na fig.1. A partir do modelo MMSSP foram aplicadas 3 estratégias R&F, como o objetivo de gerar soluções iniciais para o modelo. As soluções obtidas são comparadas entre si para determinar a solução de melhor qualidade, essa solução é adotada como solução inicial para a heurística F&O. O tempo total para a execução das duas heurísticas combinadas é limitado em 3600 s.

Após esse processo, as soluções finais, são comparadas com as soluções obtidas pelo *software* GUROBI 7.0 (configuração *default*) limitado de 3600 s por instância. Por padrão as configurações MIPFocus e MIPGap são definidas em 0 e  $1.E^{-4}$ , respectivamente.

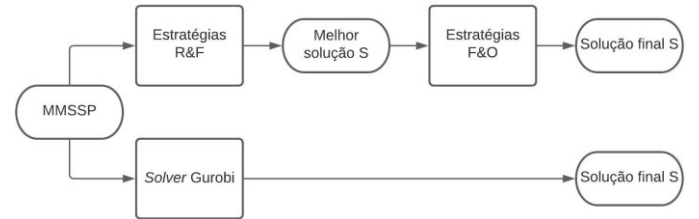


Fig. 1. Metodologia.

Devido ao grande volume de dados, optou-se por usar o método de perfis de desempenho, propostos em [23]. O método consiste em avaliar a robustez de diferentes estratégias ou *softwares*, quando utilizado para o mesmo propósito. A técnica pode ser utilizada para comparar diferentes métricas como tempo computacional, *Gap*, erro da solução obtida, número de falhas, entre outras.

Seja um conjunto  $P$  de  $n_p$  problemas a serem resolvidos, por um conjunto  $S$  de  $n_s$  estratégias distintas. Para cada problema  $p \in P$  e estratégia  $s \in S$ , tem-se  $Z_{p,s}$ , solução determinada pela estratégia  $s$  na resolução do problema  $p$ . O coeficiente de desempenho, é calculado por meio da expressão (15). Nessa expressão, é realizada a comparação do desempenho da estratégia  $s \in S$  na resolução do problema  $p \in P$ , em relação ao desempenho da melhor estratégia para o problema  $p$ .

$$r_{p,s} = \frac{Z_{ps}}{\min\{Z_{p,s}; \forall s \in S\}} \quad (15)$$

A razão  $r_{p,s}$  avalia o desempenho de uma estratégia na resolução de um problema  $p$ . Todavia, o que se pretende é uma

avaliação de forma geral do desempenho da estratégia. Para esse fim, [23] define o perfil de desempenho  $\rho_s(\tau)$ :

$$\rho_s(\tau) = \frac{|\{p \in P: r_{p,s} \leq \tau\}|}{n_p} \quad (16)$$

A função  $\rho_s(\tau)$ , associa a um valor dado  $\tau \in \mathbb{R}$ , a fração de problemas resolvidos, pela estratégia, com um desempenho dentro de um fator  $\tau$  do melhor desempenho obtido.

V. RESULTADOS COMPUTACIONAIS

Todos os experimentos foram codificados em *Microsoft Visual Studio 2015*, utilizando pacote de otimização *GUROBI 7.0*, e executados em um processador Intel Core i7-5500U, 2.4GHz, 8 GB de RAM, sob sistema operacional *Windows 10*.

A fig. 2 apresenta os perfis de desempenhos para a métrica função objetivo ( $Z$ ), quando aplicado as estratégias R&F no modelo MMSSP. Observando as curvas de desempenhos nota-se que, todas as estratégias atingiram o valor  $\rho_s(\tau) = 1$ , indicando que as estratégias e o *solver* foram capazes de resolver 100% das instâncias propostas. O *solver* *GUROBI* obteve o melhor desempenho em 83,33% das instâncias ( $\tau = 1$ ). Além disso, o *GUROBI* convergiu mais rápido para o valor 1, atingindo  $\rho_s(\tau) = 1$  para  $\tau = 1,256$ , ou seja, o pior resultado dessa estratégia é 25,6% maior do que o melhor resultado geral obtido.

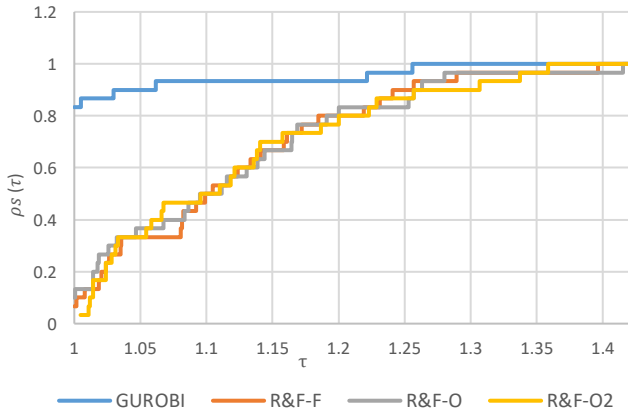


Fig. 2. Perfil de desempenho Z estratégias R&F.

Quando observado as estratégias R&F propostas, nota-se que a estratégia R&F-O2 apresentou o melhor perfil de desempenho entre as heurísticas, seguida pela R&F-F que, atingiram  $\rho_s(\tau) = 1$  em  $\tau = 1,3585$  e  $\tau = 1,3962$ , respectivamente. Em contrapartida, o pior perfil de desempenho foi obtido pela estratégia R&F-O, que somente obteve  $\rho_s(\tau) = 1$  em  $\tau = 1,4151$ .

A Tabela I apresenta o *gap*  $G_s(\%)$  fornecido pelo *software* *GUROBI* e o *gap*  $G(\%)$  de acordo com a expressão (17). O  $G(\%)$  leva em consideração a diferença entre a solução factível determinada pelas estratégias R&F ( $Z_h$ ) e o melhor limitante inferior fornecido pelo *GUROBI* ( $Z_g$ ), na configuração e tempo descritos na seção metodologia. A coluna S indica o número de problemas resolvidos na sua otimalidade pelo *solver*.

$$G(\%) = \frac{Z_h - Z_g}{Z_g} * 100 \quad (17)$$

TABELA I  
RESUMO DOS RESULTADOS OBTIDOS COM A APLICAÇÃO DAS ESTRATÉGIAS R&F.

I	J	GUROBI		R&F-F	R&F-O	R&F-O2
		$G_s(\%)$	S	$G(\%)$	$G(\%)$	$G(\%)$
10	2	0,00	10	26,48	34,75	27,92
	3	0,00	10	24,89	25,49	29,71
	6	0,85	7	26,03	21,38	24,68
30	2	4,95	3	18,99	16,53	18,46
	3	13,55	1	14,50	15,13	17,21
	6	0,00	10	17,15	15,45	14,80
50	2	7,26	1	10,78	11,80	7,69
	3	83,69	0	8,96	10,09	10,09
	6	50,21	1	11,13	11,70	11,42
80	2	17,62	0	1,95	1,81	1,28
	3	63,01	0	2,95	2,00	2,95
	6	52,25	0	-4,95	-4,73	-5,87
100	2	47,00	0	2,86	0,46	2,14
	3	95,45	0	-4,28	-4,32	-2,69
	6	52,25	0	-4,62	-6,73	-8,88

Observa-se que o *software* *GUROBI* determinou pelo menos uma solução factível para todas as instâncias dentro do tempo estabelecido. Determinou solução ótima para 28,67% dos problemas propostos. Especificamente, o *solver* apresentou dificuldade para provar a otimalidade para instâncias com  $I \geq 50$ . O menor percentual (0%) de problemas resolvidos otimamente ocorre para instâncias 80 e 100 clientes, evidenciando a dificuldade na obtenção de solução ótima para instâncias de grande porte. Em contrapartida, as estratégias R&F apresentam melhores desempenhos em instâncias maiores ( $I \geq 50$ ).

A fig. 3 apresenta os perfis de desempenho em relação a métrica tempo computacional  $t$ , limitado em 3600 s por instância. Devido à grande variabilidade ocorrida para  $\tau$ , foi utilizado uma escala logarítmica para gerar valores de forma mais concisa no eixo das abscissas ( $\tau \in [0, r_M]$  e  $r_M = \max \{\log_2(r_{p,s}): p \in P \text{ e } s \in S\}$ ) [23].

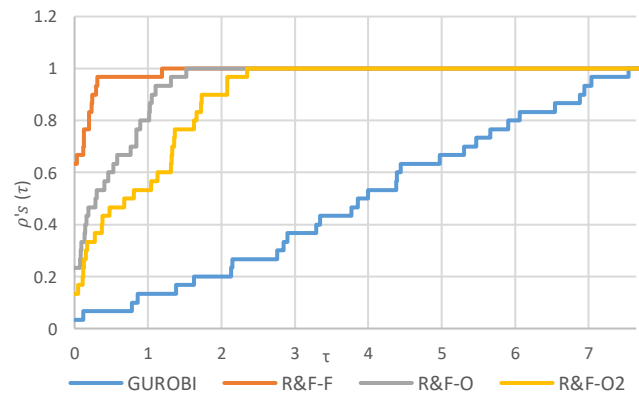


Fig. 3. Perfil de desempenho t estratégias R&F.

O desempenho da estratégia R&F-F no que concerne tempo de computação é superior às demais estratégias e ao *solver* GUROBI, uma vez que, essa atingiu  $\rho'_s(\tau) = 1$  para  $\tau = 1,1926$ , ou seja, possui  $r_{p,s} = 2,2857$ , indicando que o pior resultado dessa estratégia é 2,2857 vezes maior do que o melhor resultado geral obtido pelas demais. Em segundo lugar tem-se a R&F-O que alcançou  $\rho'_s(\tau) = 1$  para  $\tau = 1,5215$ , ou seja, possui  $r_{p,s} = 2,8709$ . É possível observar que a estratégia R&F-F obteve desempenho superior às demais em 63,33% dos problemas ( $\tau = 0$ ), seguida pela estratégia R&F-O que atingiu 23,33%. Por outro lado, o pior perfil de desempenho em relação ao tempo foi obtido pelo *solver* GUROBI, que alcançou  $\rho'_s(\tau) = 1$  para  $\tau = 7,5468$ , ou seja, possui  $r_{p,s} = 186,99$ .

Após a análise de perfil de desempenho da heurística *relax-and-fix*, aplicaram-se as estratégias *fix-and-optimize* (F&O-FC, F&O-OC, F&O-OC2 e F&O-A) na estratégia R&F que apresentou o melhor perfil de desempenho, neste caso a estratégia R&F-O2.

A Tabela II apresenta uma comparação entre o *software* GUROBI executado na configuração *default* e a combinação da heurística R&F-O2 com as estratégias F&O. Destaca-se que, para cada combinação T,I,J e Q é descrito a média de 5 instâncias. A coluna Z exibe o número médio de operadores utilizados em cada caso. Os tempos de processamento são apresentados na coluna t e  $t^*$ , a coluna  $t^*$  indica os tempos agregados das duas heurísticas R&F/F&O. As últimas linhas apresentam as médias: média de todos os testes; média S das instâncias com ótimo conhecido; e média  $\bar{S}$  dos problemas que não são conhecidos os valores ótimos.

Observa-se que todas as estratégias F&O apresentam melhorias na solução inicial fornecida pela heurística R&F-O2. A combinação das estratégias R&F-O2/F&O-A destaca-se entre as demais combinações, pois apresentou as menores médias, processamento (652,17 s) e função objetivo (119,98). Para os conjuntos de instâncias T30\_I80\_J2\_Q6, T30\_I80\_J2\_Q12, T30\_I80\_J3\_Q6, T30\_I80\_J6\_Q12, T30\_I100\_J2\_Q6, T30\_I100\_J2\_Q12, T30\_I100\_J3\_Q6, T30\_I100\_J3\_Q12 e T30\_I100\_J6\_Q12, a combinação R&F-O2/F&O-A determinou soluções melhores do que o *solver* limitado em 3600 s.

Na Tabela II é possível avaliar ainda a qualidade das estratégias no que concerne ao tempo de execução. O *software* GUROBI esgotou o tempo limite de 3600 s para 71,33% dos problemas e apresentou um tempo médio de execução de 2701,69 s. As estratégias R&F-O2 e F&O combinadas apresentam tempos médios agregados entre [644,51; 789,84] segundos.

Para instâncias mais difíceis, de grande porte, que não foram possíveis determinar o ótimo, as estratégias R&F/F&O exibem desempenho superior ao *software* GUROBI. A combinação R&F-O2/F&O-A apresenta o melhor desempenho médio  $\bar{S}$ , 146,79 para função objetivo e tempo médio de 853,07 s, valores inferiores aos determinados pelo GUROBI, 149,39 e 3600,00 s, respectivamente.

Aplicando o método de perfil de desempenho aos resultados obtidos gerou-se o gráfico apresentado na fig. 4. Ao observar as curvas de desempenho, nota-se que a curva da estratégia R&F-O2 é dominada pelas demais. Isto indica que todas as estratégias *fix-and-optimize* propostas melhoraram os resultados obtidos

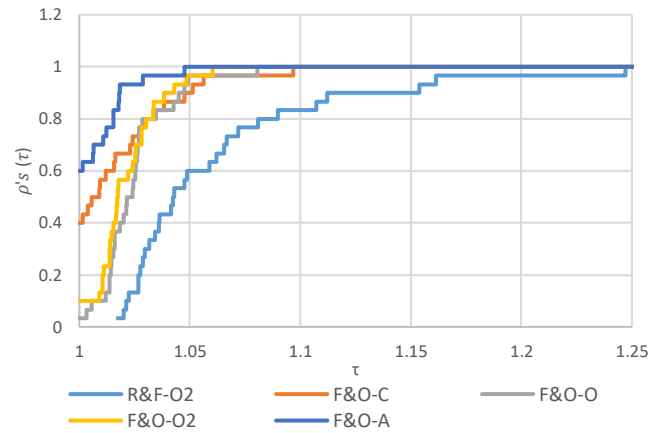


Fig. 4. Perfil de desempenho Z para as estratégias F&O.

pela R&F-O2. A estratégia F&O-A obteve o melhor desempenho para 60% das 150 instâncias ( $\tau = 1$ ). Além disso, apresentou melhor convergência  $\rho_s(\tau) = 1$  para  $\tau = 1,0476$ . Constata-se que a estratégia F&O-O obteve apenas um resultado melhor entre as demais estratégias. Isto não indica que a F&O-O teve o pior perfil de desempenho, pois ela atinge  $\rho_s(\tau) = 1$  para  $\tau = 1,0806$ , sugerindo que o pior desempenho dessa estratégia é 8,06% maior do que o melhor resultado geral obtido nos testes, ficando inferior a F&O-C que apresentou 9,68%. Observa-se ainda que a F&O-C obteve melhor desempenho em 40% das instâncias, enquanto que F&O-O e F&O-O2 apresentaram 3,33% e 10%, respectivamente.

## VI. CONSIDERAÇÕES FINAIS

Nesse trabalho foi proposta uma formulação que generaliza os problemas de agendamento de serviços multiperíodos. Trabalhos envolvendo MSSP são recentes na literatura, sendo descrito pela primeira vez em [1] e posteriormente tratada nos trabalhos [2], [3]. Por se tratar de uma nova classe de problemas carecem de trabalhos que explorem suas possíveis variações.

O modelo MMSSP tratado considera a execução de múltiplos serviços periódicos, portanto a resolução de um MMSSP equivale a resolução de múltiplos MSSP. Exemplos de aplicações do MMSSP são: agendamento de múltiplas consultas, agendamento de serviços sociais, agendamento de diversos processadores, agendamento de múltiplos serviços prestados por uma empresa, entre outras aplicações.

Resolver na otimalidade o MMSSP é uma tarefa complexa e difícil, mesmo para *solvers* modernos. Para a resolução do modelo proposto, abordou-se métodos heurísticos especificamente as heurísticas *relax-and-fix* e *fix-and-optimize*. Foram propostos ao todo três métodos de decomposição para a heurística R&F e quatro métodos distintos para a heurística F&O. Os resultados computacionais indicam que a R&F é eficiente para determinar soluções iniciais para o MMSSP em um curto espaço de tempo. Além disso, a combinação das heurísticas R&F e F&O apresentou desempenho superior nas instâncias testadas no que concerne à função objetivo e tempo computacional quando comparada com o *software* GUROBI na configuração *default* limitado em 3600 s.

TABELA II  
 RESULTADOS OBTIDOS COM A APLICAÇÃO DAS ESTRATÉGIAS F&O COMBINADA COM A ESTRATÉGIA R&F-O2.

		GUROBI		R&F-O2		F&O-C		F&C-O		F&O-O2		F&O-A		
	<i>J</i>	<i>Q</i>	<i>Z</i>	<i>t</i>	<i>Z</i>	<i>t</i>	<i>Z</i>	<i>t*</i>	<i>Z</i>	<i>t*</i>	<i>Z</i>	<i>t*</i>	<i>Z</i>	<i>t*</i>
10 clientes	2	6	15,00	88,20	18,00	2,20	16,40	3,20	16,00	3,20	16,00	3,00	15,60	2,20
		12	10,60	1,40	14,40	1,20	13,60	1,20	13,40	1,20	13,00	1,20	12,40	1,20
	3	6	16,60	139,40	22,20	3,80	18,80	3,80	18,60	3,80	18,40	3,80	17,80	4,00
30 clientes		12	14,80	63,80	18,60	2,80	16,80	2,80	16,80	2,80	16,80	2,80	17,60	2,80
	6	6	45,00	2220,00	53,40	11,80	49,80	14,00	50,40	14,20	49,00	14,00	49,60	13,80
		12	22,80	787,80	29,80	8,80	27,80	10,40	28,40	9,80	27,80	9,80	28,60	10,00
50 clientes	2	6	39,80	3600,00	45,40	27,80	44,00	38,00	44,00	38,20	43,80	38,00	42,00	30,80
		12	21,00	2767,40	25,80	12,80	24,00	18,80	24,00	18,80	24,60	18,80	23,20	12,80
	3	6	61,00	3207,60	68,40	228,60	65,00	250,40	65,40	313,00	65,40	249,40	64,40	232,80
80 clientes		12	31,40	3600,00	38,40	27,60	36,00	39,60	36,20	40,00	36,40	39,60	36,00	29,60
	6	6	118,00	163,40	136,60	101,40	129,00	166,80	131,60	168,80	130,80	169,80	131,00	125,20
		12	120,00	260,40	136,60	103,60	132,80	175,80	134,60	175,40	135,00	187,00	135,20	129,20
100 clientes	2	6	65,20	2953,80	69,00	382,20	67,60	420,60	68,60	421,00	68,00	420,40	65,80	406,00
		12	35,60	3600,00	39,00	30,00	38,00	51,80	38,40	51,80	38,00	51,60	36,60	35,00
	3	6	97,00	3600,00	103,40	406,40	102,20	486,20	101,80	486,20	101,40	486,00	99,20	416,40
150 clientes		12	48,60	3600,00	55,20	182,00	53,20	225,20	54,40	226,60	54,60	226,20	53,00	186,60
	6	6	193,00	3597,60	215,80	521,00	206,00	806,20	211,00	812,80	208,80	800,20	209,20	583,00
		12	105,20	3600,00	116,80	687,40	112,00	834,60	113,80	829,20	113,00	821,80	113,20	713,00
200 clientes	2	6	128,40	3600,00	130,2	815,2	128,2	946,00	129,4	945,80	129,4	944,40	126,2	873,00
		12	68,60	3600,00	69,40	288,60	68,00	362,80	68,20	362,60	68,60	360,60	67,40	292,80
	3	6	200,20	3600,00	206,40	1239,80	204,00	1502,20	204,60	1499,40	204,00	1505,20	199,20	1294,20
300 clientes		12	99,60	3600,00	103,00	707,60	99,60	836,60	102,00	837,40	102,40	836,60	100,20	717,80
	6	6	368,20	3600,00	393,40	1924,40	383,20	2827,80	388,40	2869,20	389,80	2872,80	390,20	2137,00
		12	272,80	3600,00	222,20	1725,20	217,60	2183,40	221,60	2184,00	220,60	2181,60	219,00	1802,00
450 clientes	2	6	156,60	3600,00	161,80	1138,80	159,80	1389,80	160,40	1391,80	160,60	1392,20	156,20	1190,60
		12	84,20	3600,00	85,00	583,80	83,80	718,20	84,00	718,80	84,00	717,20	82,80	615,80
	3	6	264,20	3600,00	252,20	1668,40	246,80	2166,20	250,60	2163,60	248,00	2162,00	245,40	1719,20
600 clientes		12	131,60	3600,00	129,20	879,20	127,00	1125,60	127,40	1125,40	129,20	1124,00	127,20	896,20
	6	6	462,40	3600,00	486,60	2508,80	476,00	3000,05	483,60	3000,20	484,00	3070,20	484,40	2648,40
		12	309,80	3600,00	255,80	2111,20	251,20	2898,40	254,40	2980,20	253,40	2973,80	250,80	2216,80
Média			120,24	2701,69	123,40	611,08	119,94	783,54	121,40	789,84	121,16	789,47	119,98	644,61
Média <i>S</i>			47,70	464,13	56,12	48,40	53,26	73,21	53,63	73,59	53,33	74,30	53,16	56,09
Média <i>S̄</i>			149,39	3600,00	150,44	837,21	146,82	1064,37	148,78	1077,70	148,45	1076,86	146,79	853,07

REFERÊNCIAS

[1] C. Núñez-del-Toro, "Scheduling policies for multi-period services," M.S. thesis, Univ., Politècnica de Catalunya, Barcelona, Espanha, 2015.

[2] C. Núñez-del-Toro, E. Fernández, J. Kalcsics, and S. Nickel, "Scheduling policies for multi-period services," *Eur. J. Oper. Res.*, vol. 251, no. 3, pp. 751–770, 2016, doi: 10.1016/j.ejor.2015.12.002.

[3] E. Fernández, J. Kalcsics, and C. Núñez-del-Toro, "A branch-and-price algorithm for the Aperiodic Multi-Period Service Scheduling Problem," *Eur. J. Oper. Res.*, vol. 263, no. 3, pp. 805–814, 2017, doi: 10.1016/j.ejor.2017.06.008.

[4] M. Bender, A. Meyer, J. Kalcsics, and S. Nickel, "The multi-period service territory design problem - An introduction, a model and a heuristic approach," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 96, pp. 135–157, 2016, doi: 10.1016/j.tre.2016.09.007.

[5] M. Bender, J. Kalcsics, S. Nickel, and M. Pouls, "A branch-and-price algorithm for the scheduling of customer visits in the context of multi-period service territory design," *Eur. J. Oper. Res.*, vol. 269, no. 1, pp. 382–396, 2018, doi: 10.1016/j.ejor.2018.01.047.

[6] C. Archetti, E. Fernández, and D. L. Huerta-Muñoz, "A two-phase solution algorithm for the Flexible Periodic Vehicle Routing Problem," *Comput. Oper. Res.*, vol. 99, pp. 27–37, 2018, doi: 10.1016/j.cor.2018.05.021.

[7] D. V. Thomaz, G. Valentim Loch, C. T. Scarpin, and C. M. Schenekemberg, "A Mathematical Model for the Periodic Capacitated Arc Routing Problem with Time Windows," *IEEE Lat. Am. Trans.*, vol. 16, no. 10, pp. 2567–2573, 2018, doi: 10.1109/TLA.2018.8795136.

[8] M. Y. Kovalyov, C. T. Ng, and T. C. E. Cheng, "Fixed interval scheduling: Models, applications, computational complexity and algorithms," *Eur. J. Oper. Res.*, vol. 178, no. 2, pp. 331–342, 2007, doi: 10.1016/j.ejor.2006.01.049.

[9] G. V. Batista and C. T. Scarpin, "Problema de roteamento em arcos capacitado e periódico aplicado a um contexto real," *Rev. Produção Online*, vol. 15, no. 3, p. 1080, 2015, doi: 10.14488/1676-1901.v15i3.1962.

[10] L. G. Kroon, M. Salomon, and L. N. Van Wassenhove, "Exact and approximation algorithms for the Tactical Fixed Interval Scheduling Problem," *Oper. Res.*, vol. 45, no. 4, pp. 624–638, 1997, doi: 10.1287/opre.45.4.624.

[11] B. Branda, J. Novotný, and A. Olstad, "Fixed interval scheduling under uncertainty - A tabu search algorithm for an extended robust coloring formulation," *Comput. Ind. Eng.*, vol. 93, pp. 45–54, 2016, doi: 10.1016/j.cie.2015.12.021.

[12] M. Fischetti, S. Martello, and P. Toth, "Approximation Algorithms for Fixed Job Schedule Problems Author (s): Matteo Fischetti, Silvano Martello and Paolo Toth Published by: INFORMS Stable URL: https://www.jstor.org/stable/3840840 REFERENCES Linked references are available on JSTOR for this," vol. 40, 2020.

[13] V. Veeramachaneni, P. Berman, and W. Miller, "Aligning two fragmented sequences," *Proc. - Int. Parallel Distrib. Process. Symp. IPDPS 2002*, vol. 127, pp. 187–194, 2002, doi: 10.1109/IPDPS.2002.1016591.

[14] P. R. Marcelino, "Afinal, o que é terceirização? Em busca de ferramentas de análise e de ação política," *PEGADA - A Rev. da Geogr. do Trab.*, vol. 8, no. 2, 2012, doi: 10.33026/peg.v8i2.1640.

[15] L. Wolsey, *Wiley-Interscience Series in Discrete Mathematics and Optimization: Integer Programmig*. New York: John Wiley & Sons, 1998.

[16] T. M. Schmidt, C. T. Scarpin, G. V. Loch, and C. M. Schenekemberg, "Heuristic approaches to solve a two-stage lot sizing and scheduling problem," *IEEE Lat. Am. Trans.*, vol. 17, no. 3, pp. 434–443, 2019, doi: 10.1109/TLA.2019.8863314.

[17] L. Li, S. Song, and C. Wu, "Solving a Multi-Level Capacitated Lot Sizing Problem with Random Demand via a Fix-and-Optimize heuristic," *2015 IEEE Congr. Evol. Comput. CEC 2015 - Proc.*, no. Liuxi Li, pp. 2721–2728, 2015, doi: 10.1109/CEC.2015.7257226.

[18] L. de Oliveira, C. C. de Souza, and T. Yunes, "Improved bounds for the traveling umpire problem: A stronger formulation and a relax-and-fix heuristic," *Eur. J. Oper. Res.*, vol. 236, no. 2, pp. 592–600, 2014, doi:



- 10.1016/j.ejor.2013.12.019.
- [19] J. D. Oliveira and C. T. Scarpin, "A Relax-and-Fix Decomposition Strategy Based on Adjacent Nodes Applied to the Periodic Capacitated Arc Routing Problem (PCARP)," *IEEE Lat. Am. Trans.*, vol. 18, no. 3, pp. 573–580, 2020.
- [20] Y. Pochet and L. A. Wolsey, *Production Planning by Mixed Integer Programming*. Springer, 2006.
- [21] F. Sahling, L. Buschkühl, H. Tempelmeier, and S. Helber, "Solving a multi-level capacitated lot sizing problem with multi-period setup carry-over via a fix-and-optimize heuristic," *Comput. Oper. Res.*, vol. 36, no. 9, pp. 2546–2553, 2009, doi: 10.1016/j.cor.2008.10.009.
- [22] S. Helber and F. Sahling, "A fix-and-optimize approach for the multi-level capacitated lot sizing problem," *Int. J. Prod. Econ.*, vol. 123, no. 2, pp. 247–256, 2010, doi: 10.1016/j.ijpe.2009.08.022.
- [23] E. D. Dolan and J. j. Moré, "Benchmarking optimization software with performance profiles," *Stud. Surf. Sci. Catal.*, vol. 91, no. A, pp. 201–213, 202AD, doi: 10.1007/s101070100263.



**Jailson Domingos de Oliveira** é Mestre (2017) em Pesquisa Operacional pelo Programa de Pós-Graduação em Métodos Numéricos em Engenharia, área de concentração: Programação Matemática, na Universidade Federal do Paraná; Licenciado em Matemática (2010) pela UNICENTRO. Atualmente é estudante de Doutorado no Programa de Pós-Graduação em Métodos Numéricos em Engenharia (PPGMNE) da UFPR e Professor substituto na UNICENTRO.



**Cassius Tadeu Scarpin** é Doutor (2012) e Mestre (2007) em Pesquisa Operacional pelo Programa de Pós-Graduação em Métodos Numéricos em Engenharia, área de concentração: Programação Matemática, na Universidade Federal do Paraná. Engenheiro de Produção (2010) e Licenciado em Matemática (2002) pela mesma universidade. Atualmente ocupa o cargo de Professor Adjunto no Departamento de Administração Geral e Aplicada (DAGA) da Universidade Federal do Paraná (UFPR) e coordena o Grupo de Pesquisa de Tecnologia Aplicada à Otimização (GTAO).