

Um algoritmo genético multiobjetivo para a programação integrada de veículos e tripulações

Bruno de Athayde Prata¹

Resumo: O *Vehicle and Crew Scheduling Problem* (VCSP) é um difícil problema de Otimização Combinatória, objeto de pesquisa continuada ao longo dos últimos anos. Tendo em consideração a gama de variáveis relacionadas com o VCSP, há uma série de características práticas do problema que não têm sido contempladas nas soluções geradas computacionalmente. Os modelos existentes na literatura focam somente na minimização de custos. No entanto, outros objetivos ou critérios devem ser considerados como, por exemplo, a redução nos intervalos de lanche dos tripulantes. Este artigo tem como objetivo reportar o desenvolvimento de uma abordagem multiobjetivo, baseada em um Algoritmo Genético, para a otimização integrada da programação de veículos e tripulações em sistemas de transporte público. Experimentos computacionais são apresentados e discutidos. Os resultados obtidos apontam para a possibilidade de, com o uso da abordagem proposta, se obter ganhos significativos em termos de custos de operação e em termos da redução dos tempos de planejamento.

Palavras-chave: Transporte Público; Otimização Combinatória Multiobjetivo; Pareto-based Selection Algorithm.

Abstract: The integrated vehicle and crew scheduling problem is a hard, widely studied Combinatorial Optimization problem over the years. Taking into consideration the range of variables related to the planning process of vehicles and drivers, there are several practical characteristics of the problem that are not reflected in the solutions generated computationally. The existing models focus on minimizing costs. However, other objectives must be considered as for example the reduction of meal breaks for the crews. This paper aims at presenting a multiobjective approach for the integrated vehicle and crew scheduling problem in public transport systems based on Genetic Algorithms. Computational results with real instances are presented and discussed. These results indicate that this new approach has a considerable potential for achieving significant gains in terms of operation costs and reduction in planning times.

Keywords: Public Transport, Multiobjective combinatorial optimization, Pareto-based Selection Algorithm.

1. INTRODUÇÃO

No que concerne aos ônibus, o planejamento dos transportes urbanos pode ser decomposto nas seguintes principais etapas (Ceder, 2002): definição da rede de transportes, definição da tabela de horários, programação dos veículos, programação das tripulações e rotação das tripulações. A programação de veículos e de tripulações têm sido objeto de inúmeras atividades de pesquisa, devido aos ganhos significativos que podem resultar da sua otimização (Daduna e Paixão, 1995; Wren e Rosseau, 1995).

Diversos pesquisadores têm relatado a forte interação entre os problemas de escalonamento de veículos e de motoristas e os ganhos em considerá-los conjuntamente (Friberg e Haase, 1999; Gaffi e Nonato, 1999; Freling *et al.*, 1999). O problema integrado é usualmente denominado *Vehicle and Crew Scheduling Problem* – VCSP.

Para resolução do VCSP, várias abordagens vem sendo reportadas na literatura. Diversos trabalhos apresentam modelos matemáticos para o VCSP com um único depósito, tais como Freling *et al.*, (1999), Friberg e Haase (1999), Haase *et al.* (2001) e Huisman (2004). Outros trabalhos apresentam modelos matemáticos para o VCSP com múltiplos depósitos, tais como Gaffi e Nonato (1999), Fischetti *et al.* (2001), Freling *et al.* (2003), Huisman (2004), Huisman

et al. (2005), Weider (2007), Fleurent e Rosseau (2007), Groot e Huisman (2008), Mesquita e Paias (2008), Mesquita *et al.* (2009), Steizen *et al.* (2010) e Kliever *et al.* (2012). Foram apresentados testes em instâncias reais e geradas aleatoriamente e os resultados apontam que estas formulações tendem a não permitir a obtenção de soluções exatas para problemas reais de grande porte.

Abordagens baseadas em heurísticas e meta-heurísticas são reportadas em Ball *et al.* (1983), Falkner e Ryan (1992), Patrikalakis e Xerocostas (1992), Wren e Gualda (1999), Valouxis e Housos (2002), Rodrigues *et al.* (2006), Laurent e Hao (2007), Laurent e Hao (2008), Bartodziej *et al.* (2007), Steinzen *et al.* (2007), Steizen (2007) e Laurent e Hao (2008). As abordagens heurísticas reportadas padecem de pouca generalidade, no sentido em que são muito específicas aos problemas para os quais foram desenvolvidas.

Diante da complexidade de tratar o VCSP de forma exata, algoritmos heurísticos vêm sendo propostos na literatura (Ball *et al.*, 1983; Patrikalakis e Xerocostas, 1992; Valouxis e Housos, 2002). Tais abordagens primam pela decomposição do problema integrado nos problemas isolados (programação de veículos e de tripulantes), sendo buscada uma integração *a posteriori*.

Alguns *softwares* também foram desenvolvidos para a resolução do VCSP em casos reais (Falkner e Ryan, 1992; Wren e Gualda, 1999; Rodrigues *et al.*, 2006). Esses sistemas empregam regras práticas usadas pelos planejadores de empresas de transporte público para a obtenção de boas soluções em termos práticos.

Dada a dificuldade de obtenção de soluções de elevada qualidade para o VCSP, o uso de heurísticas se mostra bastante limitado. Deste modo, diversos trabalhos apre-

¹ Bruno de Athayde Prata, Departamento de Engenharia de Produção da Universidade Federal do Ceará. (baprata@ufc.br)

Manuscrito recebido em 12/08/2015 e aprovado para publicação em 02/12/2015.

Este artigo é parte de TRANSPORTES v. 24, n. 1, 2016 ISSN: 2237-1346 (online). DOI: 10.14295/transportes.v24i1.975

sentam meta-heurísticas para a resolução do problema em foco, tais como Laurent e Hao (2007), Laurent e Hao (2008), Bartodziej *et al.* (2007), Steinzen *et al.* (2007), Steinzen (2007), Laurent e Hao (2008), Silva e Cunha (2010), Silva e Prates (2014) e Silva e Silva (2015). As abordagens heurísticas reportadas padecem de pouca generalidade, no sentido em que são muito específicas aos problemas para os quais foram desenvolvidas.

Levando em consideração a gama de variáveis relacionadas à programação de veículos e tripulações, existem diversas características práticas que nem sempre são refletidas nas soluções geradas.

Como objetivos mais comuns para a programação de veículos, podem ser destacados: minimização do número de veículos; minimização do número de mudanças de linhas; minimização do número e do comprimento das viagens em vazio; e minimização dos tempos ociosos dos veículos nos terminos (Baita *et al.*, 2000).

Como objetivos mais comuns para a programação de tripulações, podem ser destacados: minimização do custo dos serviços selecionados (Wren e Rosseau, 1995); minimização do número de trocas de motoristas por veículo (*changeovers*); minimização do número de tramos descobertos (*leftovers*); minimização do número de serviços em uma solução (Lourenço *et al.*, 2000); minimização da duração dos *leftovers*; minimização dos tempos ociosos; e, minimização do número de tripulantes (Dias, 2005).

Diante do exposto, pode-se observar a necessidade do desenvolvimento de abordagens que contemplem múltiplos objetivos na resolução do VCSP.

Este trabalho tem por objetivo reportar o desenvolvimento de um Algoritmo Genético Multiobjetivo para o VCSP, baseado no algoritmo *Pareto-based Selection Algorithm II* (PESA – II).

As próximas seções do artigo são estruturadas como segue. Na segunda seção, é apresentada a definição do problema. Na terceira seção, é apresentada a formulação matemática proposta para o problema. Na quarta seção, é apresentado o Algoritmo Genético Multiobjetivo proposto. Na quinta seção é apresentado o Estudo de Caso, inerente à cidade de Fortaleza, Brasil. Na sexta seção, são apresentados os resultados obtidos. Por fim, na sétima seção, são apresentadas as principais conclusões e recomendações para estudos futuros.

2. DEFINIÇÃO DO PROBLEMA

Dada uma tabela de horários com um conjunto de viagens, a programação de veículos (*vehicle scheduling*) consiste na alocação de veículos às viagens, usualmente objetivando a minimização dos custos operacionais. Na programação das tripulações (*crew scheduling*), um tramo (*piece-of-work*) consiste na menor unidade de trabalho de um tripulante. Um serviço (*duty*) é um conjunto de tramos cobertos por um tripulante.

O problema em estudo consiste no atendimento simultâneo de viagens por veículos e de tramos por tripulantes. A programação de veículos é de complexidade menor do que o problema de programação de tripulantes (Daduna e Paixão, 1995; Wren e Rosseau, 1995). Como a programação de tripulantes tende a ser a maior fonte de complexida-

de do problema integrado, devem ser buscadas estratégias para facilitar a sua resolução.

Em problemas de planejamento operacional de sistemas de transporte público, abordagens baseadas em cobertura (*Set Covering Problem – SCP*) e particionamento de conjuntos (*Set Partitioning Problem – SPP*) são correntemente utilizadas (Dias, 2005).

Abordagens baseadas no SCP têm a vantagem de permitirem uma solução mais rápida; todavia, devido a poderem ocorrer *overcovers* (cobertura de um tramo por mais de um serviço), tais soluções necessitam de correções para serem aplicadas na prática. Abordagens baseadas no SPP são de resolução mais difícil, porém, por não permitirem *overcovers*, são mais atrativas do ponto de vista prático.

Huisman (2004) salienta que, para o caso particular do VCSP em que *changeovers* não são permitidos, pode-se adotar uma abordagem baseada no SCP. Por conseguinte, uma abordagem baseada no SPP também pode ser utilizada. Conforme Klabjan *et al.* (2001), os principais motivos pelos quais o problema de *scheduling* de tripulações baseado no SPP torna-se difícil, são o grande número de serviços admissíveis, a sua estrutura complexa e a natureza de custos não-linear.

A função objetivo do SPP consiste em um produto escalar de um vetor de custos e um vetor de variáveis de decisão. Logo, o problema tem uma estrutura de custos *explícita* de caráter linear. Contudo, de um modo *implícito*, esta estrutura pode ser não-linear. Tal particularidade se evidência em problemas do tipo *crew scheduling*.

Em problemas de planejamento operacional de transportes públicos, um serviço pode ser visto como a cobertura de um conjunto de tramos. O custo de um serviço está relacionado com o pagamento da tripulação. Devido às legislações trabalhistas em vigência na maioria das grandes cidades, os serviços de tripulações não são pagos por hora trabalhada, mas por jornada de trabalho. Deste modo, serviços que cubram quantidades de tramos diferentes podem vir a ter o mesmo custo. Por conseguinte, a relação entre o custo do serviço e a quantidade de tramos cobertos não é diretamente proporcional. Em outras palavras, tal relação é não-linear.

Outros aspectos estão associados a essa não-linearidade: serviços de diferentes tipos (serviços seguidos ou serviços partidos) têm custos diferentes (não explícitos), assim como a percentagem de diferentes tipos de serviço na solução final. Esta relação é claramente não-linear e tem um custo na qualidade da solução

Outra fonte de não-linearidade da estrutura implícita de custos do SPP é o pagamento de horas-extras. Ao se exceder a duração da jornada de trabalho, para que um serviço possa cobrir mais tramos é necessária a adição de um valor extra a ser pago, o qual é proporcionalmente mais oneroso do que uma hora trabalhada em jornada normal. Tem-se, novamente, uma evidência da não-linearidade de custos do SPP.

Uma forma de evitar, na medida do possível, sua natureza de custos não-linear, é trabalhar com uma função objetivo que, em vez de procurar minimizar os custos dos serviços a serem selecionados, procure minimizar os tramos não cobertos.

O problema tratado neste artigo possui duas funções objetivo, descritas a seguir:

- Minimização de tramos descobertos (*leftovers*). A ocorrência de tramos que não são cobertos por serviços regulares requer a adição de horas extras, que são pagas com acréscimo de 50%. Deste modo, ao se minimizar este não-cobrimto, são geradas soluções que reduzem os custos operacionais.
- Minimização de tempos ociosos. Os tempos em que um veículo fica parado entre a operação de duas viagens consecutivas gera diversos problemas. O primeiro deles é o uso ineficiente dos veículos, que consistem em recursos onerosos. Quando um veículo possui um tempo ocioso superior a duas horas, este é recolhido para a garagem (movimento de *pull-out*). Assim, percebe-se que o aumento dos tempos ociosos também corrobora para o incremento das viagens em vazio, as quais incorrem em diversos problemas, tais como ineficiência operacional e problemas de congestionamento viário.

A obtenção de abordagens gerais para os problemas de programação de veículos e de tripulantes é praticamente inviável, visto que as empresas e cidades tendem a adotar seus conjuntos de regras operacionais e de legislação trabalhista (Daduna e Paixão, 1995; Wren e Rosseau, 1995). O problema em estudo leva em consideração as características do sistema de transporte público por ônibus de Fortaleza, Brasil.

As principais características da programação de veículos e tripulantes em Fortaleza são apresentadas a seguir: (i) a programação é realizada para cada linha, individualmente; (ii) não são permitidas trocas de veículos entre linhas durante os intervalos para folgas; (iii) cada linha é operada por um conjunto de empresas; (iv) os tempos de viagens em vazio (*deadheads*) no início e no término da operação de um veículo (*pull-in* e *pull-out*) não são considerados na programação, uma vez que não se sabe, a priori, quais as empresas que vão operar uma determinada linha; (v) a jornada de trabalho é de 07h20min, que corresponde ao valor mínimo remunerado à tripulação, sendo permitida uma jornada máxima de 09h20min, com a adição de horas extras, que são pagas com acréscimo de 50%; (vi) é obrigatório um intervalo (não remunerado) para lanche, que deve ter uma duração mínima de 30min. O intervalo de lanche deve, preferencialmente, estar compreendido num intervalo de 30min a 1h. Intervalos entre 1h e 2h são permitidos, mas geram reclamações entre os tripulantes; (vii) devido a um acordo entre as empresas e os trabalhadores, um valor máximo de 30% das viagens pode possuir intervalos superiores a 2h; e (viii) *changeovers* não são permitidos.

3. FORMULAÇÃO MATEMÁTICA

A seguir é apresentada a definição e a notação dos elementos para o modelo proposto para o VCSP. Nesse modelo, as trocas de tripulante por veículo (*changeovers*) são proibidas. Seja uma matriz, denotada por A , a qual representa o conjunto de serviços gerados. O número de linhas de

esta matriz representa o número de tramos (*piece-of-works*, isto é, menor unidade de trabalho de uma tripulação) e o número de colunas o número de serviços (conjunto de atividades efetuadas por uma tripulação na sua jornada diária de trabalho). O elemento $a_{ij}=1$ se o serviço j cobre o tramo i , sendo 0 caso contrário.

Dada uma matriz A , de ordem $m \times n$, com elementos $a_{ij} \in \{0,1\}$; uma matriz B , de ordem $m \times p$, com elementos $b_{ik} \in \{0,1\}$. A matriz A , a qual possui m linhas e n colunas, contém serviços viáveis que respeitem a legislação trabalhista, enquanto a matriz B , que possui m linhas e p colunas, contém serviços inviáveis. É importante destacar que não existe penalização em não se cobrir uma linha da matriz B , tendo em vista que este tipo de serviço não é prioritário.

Seja x_j uma variável de decisão binária que é igual a um se o j -ésimo serviço da matriz A está na solução, sendo zero caso contrário; y_k uma variável de decisão binária que é igual a um se o k -ésimo serviço da matriz B está na solução, sendo zero caso contrário; z_i uma variável de decisão binária que é igual a um se a i -ésimo tramo não é coberto, sendo zero caso contrário e γ o número máximo de serviços da matriz B permitidos. O modelo matemático para o VCSP sem *changeovers* é definido como segue:

$$\text{Minimizar } \sum_{i=1}^m z_i \quad (1)$$

$$\text{Minimizar } \sum_{i=1}^n t_i \quad (2)$$

Sujeito a:

$$\sum_{j=1}^n x_j + \sum_{k=1}^p y_k \leq d \quad (3)$$

$$\sum_{j=1}^n a_{ij} x_j + \sum_{k=1}^p b_{ik} y_k + z_i = 1 \quad \forall i = 1, \dots, m. \quad (4)$$

$$\sum_{k=1}^p y_k \leq \gamma \quad \forall k = 1, \dots, p. \quad (5)$$

$$x_j \in \{0,1\} \quad \forall j = 1, \dots, n. \quad (6)$$

$$y_k \in \{0,1\} \quad \forall k = 1, \dots, p. \quad (7)$$

$$z_i \in \{0,1\} \quad \forall i = 1, \dots, m. \quad (8)$$

$$t_i \geq 0 \text{ e inteiro} \quad \forall i = 1, \dots, m. \quad (9)$$

A função objetivo representada pela equação (1) procura minimizar o número de tramos descobertos. Caso o i -ésimo tramo não possa ser coberto, $z_i = 1$. A função objetivo representada pela equação (2) procura minimizar o tempo ocioso dos veículos entre duas viagens consecutivas. A restrição (3) impõe que um número máximo d de colunas das matrizes A e B sejam selecionadas na solução. O conjunto de restrições do tipo (4) garante que os tramos serão cobertos por colunas sem interseções de linhas em colunas distintas, e, caso isso não seja possível, $z_i = 1$. A restrição do

tipo (5) impõe que a quantidade máxima de serviços inviáveis não seja excedida. Por serviço inviável entende-se um serviço que não atenda plenamente às regras trabalhistas. As expressões (6), (7), (8) e (9) impõem o caráter discreto das variáveis. Deve-se observar que as variáveis z_i e t_i são auxiliares, sendo que não representam, de fato, decisões.

Sobre a questão da inviabilidade dos serviços em uma dada solução, podem ser apresentados os seguintes comentários. A legislação trabalhista impõe um intervalo não remunerado para lanche, com duração mínima de 30 min e duração máxima de 2h. Na prática, os tripulantes preferem intervalos entre 30min e 1h. No entanto, para as empresas, tais intervalos dificultam a programação, pois para a cobertura de todos os tramos acaba sendo requerida uma maior quantidade de tripulantes. Gerando serviços com intervalos compreendidos entre 1h e 2h (serviços viáveis, mas indesejáveis pelos tripulantes), bem como serviços com intervalos superiores à 2h (serviços inviáveis e indesejáveis pelos tripulantes), consegue-se cobrir mais facilmente os tramos.

Assim, a inviabilidade dos serviços gerados diz respeito apenas à existência de um percentual máximo de 30% serviços com intervalos superiores à 2h, conforme acordo firmado entre as empresas e os trabalhadores. No que concerne às demais restrições operacionais e trabalhistas, nenhuma delas é violada pelo modelo proposto.

Na Figura 1, é apresentada uma ilustração das matrizes A e B objetivando uma melhor compreensão do modelo matemático proposto. Seja uma instância do VCSP com $m=7$, $n=4$ e $p=3$, isto é, tem-se uma matriz A com 4 serviços viáveis e uma matriz B com 3 serviços inviáveis para a cobertura de 7 tramos. Uma solução para o problema seria

$\mathbf{x} = (1 \ 1 \ 1 \ 0)$ e $\mathbf{y} = (0 \ 0 \ 1)$, sendo cobertos todos os tramos e sem sobreposição de serviços (*overcovers*).

<i>Matriz A</i>				<i>Matriz B</i>		
1	0	0	0	1	0	0
0	1	0	0	0	1	0
0	0	1	0	0	0	0
1	0	0	0	0	0	1
0	1	0	0	1	0	0
0	0	1	0	0	1	0
0	0	0	1	0	0	1

Figura 1: Ilustrando o conceito de múltipla cobertura do modelo proposto.

4. ALGORITMO GENÉTICO MULTIOBJETIVO PROPOSTO

A otimização multiobjetivo procura obter soluções denominadas soluções não-dominadas. Graficamente, tais soluções assumem a forma de uma curva denominada fronteira de Pareto. Considere-se um problema de maximização, onde \mathbf{x} é um vetor de variáveis de decisão, $f_j(\mathbf{x})$ um conjunto de funções objetivo e D o conjunto de soluções viáveis. Uma solução $\mathbf{x} \in D$ é não-dominada (ótima de Pareto) se não existe nenhum $\mathbf{x}' \in D$ tal que $\forall j, f_j(\mathbf{x}') \geq f_j(\mathbf{x})$ e $f_j(\mathbf{x}') > f_j(\mathbf{x})$ para pelo menos um j .

De acordo com Deb (2001), a otimização multiobjetivo possui duas finalidades distintas: (i) descobrir soluções tão próximas quanto possível da frente de Pareto; e (ii) encontrar soluções tão diversas quanto possível na fronteira de Pareto. Estes objetivos são conflitantes: o primeiro reflete a proximidade das soluções não-dominadas relativamente à frente de Pareto e o segundo refere-se à diversidade das soluções não-dominadas ao longo da aproximação da fronteira de Pareto. Ainda de acordo com Deb (2001), os algoritmos genéticos são extremamente propícios para serem aplicados na otimização multiobjetivo por trabalharem com uma população de soluções.

Konak *et al.* (2006) apresenta um levantamento dos algoritmos genéticos multiobjetivos existentes. Com base na análise da literatura existente, procurou-se optar por um dos algoritmos com base nos seguintes critérios: qualidade das soluções, desempenho computacional e facilidade de implementação algorítmica.

Nos diversos resultados reportados em Zitzler e Thiele (1999), Corne *et al.* (2000), Zitzler *et al.* (2001) e Corne *et al.* (2001), pode-se ressaltar que os algoritmos que apresentaram os melhores resultados foram SPEA-II, NSGA-II e PESA-II. Diante do levantamento realizado, com base em critérios de eficácia e eficiência, não se pode concluir sobre um melhor algoritmo. Entretanto, sob o prisma da implementação, a abordagem PESA-II é mais simples do que SPEA-II e NSGA-II. Com base neste último critério, justifica-se a adoção do PESA-II como abordagem a ser adotada.

O PESA (CORNE *et al.*, 2000) consiste numa metaheurística evolucionária multiobjetivo baseada no conceito de hipergrade (*hiper grid*). Uma hipergrade é uma grade no hiperplano, a qual tem por objetivo controlar a densidade de soluções no espaço normalizado dos objetivos. Tal mecanismo é de fundamental importância para a eficiência dos operadores de seleção, bem como para a manutenção da diversidade da população.

A estratégia de subdividir o espaço de soluções em hipercubos tem a finalidade de permitir a avaliação da densidade de soluções em uma dada célula do *grid*. Células com menor densidade de soluções são preferíveis a células com maior densidade de soluções. Isto ocorre porque hipercubos com menor densidade de soluções representam regiões promissoras da frente de Pareto que apresentam uma baixa representatividade na população corrente.

A densidade de um hipercubo é dada pelo número de soluções não dominadas nele contido. Cada hipercubo é definido por faixas de valores admissíveis das funções objetivo. Ao avaliar a aptidão de uma solução com base nos objetivos em análise, pode-se determinar a qual hipercubo a solução pertence. Explorar células com menor densidade de soluções, além de incorrer em uma melhor exploração das regiões próximas da fronteira de Pareto, ajuda também a evitar uma possível convergência prematura da população.

O conceito de densidade das células do PESA é ampliado como um critério para a qualidade das soluções, gerando o algoritmo PESA-II (CORNE *et al.*, 2001). Nesta abordagem, na aplicação de um operador de seleção, como, por exemplo, o torneio binário, uma célula que é esparsamente ocupada tem maior chance de ser selecio-

nada do que uma célula densamente ocupada. Uma vez que uma dada célula é selecionada, uma dentre as soluções que compõem a célula é selecionada aleatoriamente para o emprego dos operadores genéticos.

O algoritmo que se propõe neste trabalho é baseado na heurística PESA-II. No Quadro 1, é ilustrado o algoritmo PESA-II proposto para resolução do VCSP.

Quadro 1: Algoritmo PESA-II proposto para o VCSP.

<p>Passo 1: Gerar aleatoriamente e avaliar uma população de soluções.</p> <p>Passo 2: Dividir o espaço normalizado dos objetivos em $ngrid^k$ hipercubos.</p> <p>Passo 3: Atualizar o conjunto de soluções não-dominadas P_e.</p> <p>Enquanto $geração\ atual \leq número\ máximo\ de\ gerações$, faça</p> <p>Passo 4: Efetuar seleção por torneio entre hipercubos.</p> <p>Passo 5: Realizar o cruzamento uniforme com $p_c=100\%$.</p> <p>Passo 6: Realizar mutação do filho gerado com $p_m=100\%$.</p> <p>Passo 7: Aplicar os procedimentos DROP e ADD.</p> <p>Passo 8: Avaliar a aptidão da solução gerada.</p> <p>Passo 9: Atualizar P_i e P_e. Incrementar o número de gerações.</p> <p>Fim-do-enquanto</p> <p>Imprimir o conjunto de soluções não-dominadas obtido</p>

A população de soluções é denominada P_i e a população de soluções não-dominadas é denominada P_e . Após a geração da população inicial, é gerada a população de soluções eficientes, a qual é atualizada a cada geração. A seleção é efetuada com base nos hipercubos e não nas soluções, como ocorre com o GA tradicional.

A cada geração é gerada uma nova prole, a qual sofre mutação. Em seguida, são aplicados os procedimentos de melhoria (reparo da viabilidade e inserção de novas colunas, caso possível). Por fim, atualizam-se P_i e P_e . Todas as etapas do algoritmo proposto são detalhadas a seguir.

Codificação

Considere-se um vetor de dimensão m , em que m é o número de variáveis de decisão (ou seja, quantidade de serviços gerados). Se $x_j = 1$ a j -ésima variável de decisão fará parte da solução; caso contrário, se $x_j = 0$, a j -ésima variável de decisão não fará parte da solução.

Formação dos hipercubos

A formação de hipercubos é controlada por um parâmetro $ngrid$ que define o número de subdivisões dos eixos do espaço das funções objetivo. O espaço das funções objetivo deve ser limitado e normalizado pelos valores mínimos e máximos das funções objetivo em análise.

Geração da população inicial

A população inicial é gerada de forma aleatória,

levando em conta a viabilidade das soluções geradas, de modo que são gerados apenas indivíduos viáveis, ou seja, sem a sobreposição de serviços (*overcovers*), atendendo ao conjunto de restrições do tipo (4). Sobre a questão dos serviços com folgas superiores à 2h, o algoritmo respeita os conjuntos de restrições dos tipos (3) e (5).

No início do algoritmo todas as colunas pertencem à lista de colunas disponíveis. Em seguida, uma coluna x_j , selecionada aleatoriamente, é adicionada à solução. A lista é então atualizada, sendo removida a coluna inserida e as demais colunas que cobrirem as mesmas linhas cobertas pela coluna inserida, de modo que as próximas inserções satisfaçam as restrições apresentadas.

Avaliação das soluções não-dominadas

Após a geração da população, esta é subdividida em duas populações: uma população interna P_i e uma população externa P_e . As soluções não dominadas são armazenadas em P_e enquanto as demais soluções são armazenadas em P_i .

Seleção

A seleção dos pais a serem cruzados é efetuada por um torneio com características variáveis, as quais se relacionam com a densidade dos hipercubos. No algoritmo proposto, os pais a serem selecionados não são soluções, mas sim hipercubos. Os hipercubos mais aptos, ou seja, aqueles com menores valores de densidade, serão escolhidos como pais.

A seleção será feita por torneio binário tradicional sempre que existirem quatro ou mais hipercubos com densidade maior ou igual a 1. Neste caso, selecionam-se 4 hipercubos e são realizados torneios com dois deles.

Quando o número de hipercubos com densidades maiores ou iguais a 1 é inferior a 4, situação esta que ocorre mais frequentemente nas primeiras gerações da busca, procede-se conforme descrito a seguir. Selecionam-se 2 hipercubos para a realização de um torneio. Após a seleção do hipercubo, seleciona-se uma solução pertencente ao hipercubo vencedor do torneio. Esta solução será recombinada com uma solução pertencente à população interna, escolhida aleatoriamente.

No caso em que só existe um hipercubo com densidade maior ou igual a 1, seleciona-se aleatoriamente uma solução deste. Em seguida, seleciona-se aleatoriamente uma solução da população interna.

Recombinação

O cruzamento é efetutado com probabilidade igual a 100%, ou seja, a cada geração, é gerada uma nova solução. O tipo de cruzamento utilizado foi o uniforme. Cada gene da solução filho a ser gerada é gerado com base numa distribuição de Bernoulli, na qual uma variável aleatória discreta b recebe um valor 0 ou 1. Se b é igual a 0, o filho receberá o valor correspondente ao gene do primeiro pai. Se b é igual a 1, o filho receberá o valor correspondente ao gene do segundo pai.

Mutação

A operação de mutação ocorre da seguinte forma: é gerado um número aleatório compreendido no intervalo

$[0,100]$ e se este é menor ou igual à probabilidade de mutação considerada, o operador de mutação é acionado. É gerado um número aleatório r compreendido no intervalo $[0,m]$ e o bit correspondente ao número aleatório gerado é invertido (por exemplo, se $x_r = 1$ muda-se x_r para 0 e se $x_r = 0$ muda-se x_r para 1).

Procedimento de melhoria

Os procedimentos DROP e ADD foram adaptados do *Set Partitioning Problem* (BEASLEY e CGU, 1998). O Procedimento DROP consiste em corrigir uma solução inviável, selecionando aleatoriamente uma coluna a descobrir a linha e remove a coluna na solução. O Procedimento ADD consiste em selecionar aleatoriamente uma coluna e inseri-la na solução, sem infringir nenhuma restrição.

O procedimento DROP é descrito no Quadro 2. DROP tem por objetivo eliminar os *overcovers*. Enquanto a soma dos elementos de w maiores que 1 for diferente de zero, existirão *overcovers* a serem corrigidos. A seleção das colunas a serem removidas da solução é realizada de forma aleatória, de modo a evitar alguma convergência prematura de colunas na solução.

Quadro 2: Procedimento DROP.

```

w ← A × xT
enquanto (soma(w>1)≠0) faça
  para i=1 até n
    se wi ≥ 2 então
      seleciona aleatoriamente coluna a descobrir a linha
      remove a coluna na solução
      w ← A × xT
    fim-do-se
  fim-do-para
fim-do-enquanto

```

Pode-se observar que nesta heurística de reparos não é analisada a restrição inerente ao número máximo de colunas. Esta análise é realizada pelo procedimento ADD, descrito no Quadro 3, que atualiza a lista de colunas disponíveis (AV) e a quantidade de colunas da solução (col). ADD permite a inserção de colunas que não infringem as restrições de igualdade na prole gerada, cuja viabilidade foi devidamente corrigida.

Quadro 3: Procedimento ADD.

```

Atualiza AV
Atualiza col
enquanto AV ≠ ∅ ou col ≤ d faça
  selecionar aleatoriamente uma coluna xj
  solução ← xj
  Atualiza AV
  col ← col + 1
fim-do-enquanto

```

5. ESTUDO DE CASO

A cidade de Fortaleza, capital do Ceará, Estado situado na Região Nordeste do Brasil, é uma das me-

trópoles regionais brasileiras em crescente expansão. É a cidade brasileira com maior densidade populacional, possuindo cerca de 2,5 milhões de habitantes, distribuídos em apenas dois milésimos do território cearense, os quais representam aproximadamente um terço da população do Estado. Considerando sua Região Metropolitana, a população de Fortaleza é superior aos três milhões e oitocentos mil de habitantes.

O transporte público em Fortaleza é composto por quatro modos: os ônibus, o transporte complementar (topics ou vans), os táxis e os mototáxis. Os modos que realizam o transporte de uma grande quantidade de passageiros são os ônibus e o transporte complementar (também chamado de transporte alternativo).

A Empresa de Transporte Urbano de Fortaleza – ETUFOR é a empresa responsável pela programação dos veículos que operam no serviço de transporte alternativo em Fortaleza. O sistema de transporte coletivo por ônibus em Fortaleza é denominado Sistema Integrado de Transportes – SITFOR. Atualmente, a rede de atendimento do SITFOR é composta por: sete estações de ônibus fechadas e integradas; duas estações de ônibus abertas e não integradas; rede de mais de 200 linhas, em sistema tronco-alimentado; e cerca de 3700 paradas para ônibus. Apesar de tal complexidade, a programação dos veículos ainda é efetuada manualmente, o que incorre num elevado tempo de planejamento e em soluções cujas qualidades ainda poderiam ser significativamente melhoradas.

De um modo geral, as linhas que operam no SITFOR podem ser agrupadas em três grandes classes: linhas que fazem integração de caráter bairro-terminal, linhas que fazem integração entre um terminal e o centro da cidade e linhas que integram duas ou mais estações.

Diante do fato de que a programação de veículos em Fortaleza requer a geração de um conjunto de serviços inviáveis, um processo enumerativo, tal como *Depth-First Search*, poderia incorrer em custos computacionais proibitivos no processo de otimização. Portanto, optou-se por gerar serviços heurísticamente, conforme ilustrado no Quadro 4.

O algoritmo procura gerar serviços com uma folga para lanche, alocando as demais viagens com a menor folga possível (se possível, com folga nula). Deste modo, a heurística procura gerar serviços com folgas tão pequenas quanto possível. A matriz A foi gerada com folgas com duração no intervalo $[30;60]$, enquanto a matriz B foi gerada superiores a 60 minutos. Não foram gerados serviços com horas-extras, mas apenas serviços com uma duração máxima de 07h20min.

Quadro 4: Procedimento para geração de serviços.

```

Enquanto k ≤ w faça
  para i=1 até m faça
    para j=1 até m faça
      alocar viagens aos serviços com a menor
      folga possível, alocando uma folga
      por serviço
    fim-do-para
  fim-do-para
fim-do-enquanto

```

Outro problema que os tempos ociosos acarretam é a ineficiência do uso das tripulações, visto que, como os *changeovers* são proibidos, veículos e tripulações formam um só recurso. Durante a jornada de trabalho, obrigatoriamente ocorre uma parada (não-remunerada) para refeição. Os demais tempos em que os veículos ficam ociosos incorrem em tempos que os tripulantes também não executam atividades, mesmo sendo remunerados.

A outra questão referente aos tempos ociosos é que jornadas com paradas longas, ou seja, entre uma e duas horas, geram reclamações entre as tripulações. Portanto, os tempos ociosos dos veículos também influenciam a qualidade dos serviços gerados, sob a ótica dos tripulantes.

A otimização multiobjetivo possui duas finalidades distintas (DEB, 2001): (i) descobrir soluções tão próximas quanto possível da fronteira de Pareto; e (ii) encontrar soluções tão diversas quanto possível na fronteira de Pareto. Estes objetivos são conflitantes: o primeiro reflete a proximidade das soluções não dominadas relativamente à frente de Pareto e o segundo refere-se à diversidade das soluções não dominadas ao longo da aproximação da fronteira de Pareto.

Uma dificuldade da avaliação de metaheurísticas multiobjetivo é que, em muitos casos, a fronteira de Pareto não é conhecida. Deste modo, a determinação de quão próximas e diversificadas são as soluções geradas pelas metaheurísticas pode ser um pouco imprecisa.

Existem diversas métricas para avaliação de métodos de otimização multiobjetivo, as quais são classificadas em três grupos (DEB, 2001): (i) métricas que avaliam a proximidade à fronteira de Pareto; (ii) métricas que avaliam a diversidade das soluções não-dominadas; e (iii) métricas que avaliam proximidade e diversidade. Nesse contexto, optou-se por adotar uma métrica que avaliasse tanto a proximidade quanto a diversidade, visto que estes dois objetivos são fundamentais quando se trata de otimização multiobjetivo.

A métrica escolhida foi a do hipervolume, devido sua fácil implementação e ao fato de que, mesmo desconhecendo com exatidão a frente de Pareto para um dado problema, o hipervolume de um conjunto de soluções não-dominadas pode propiciar informação acerca da qualidade das soluções geradas.

A métrica do hipervolume pode ser explicada como segue: considere-se um problema de otimização multiobjetivo no qual todos os objetivos são de minimização. Dado um ponto de referência W , cada solução não-dominada delimita um hipercubo com relação à W . O ponto de referência W pode representar as coordenadas equivalentes aos piores valores possíveis no espaço das funções objetivo. Seja $|Q|$ a cardinalidade do conjunto de soluções não dominadas, a união dos i volumes (mutuamente exclusivos) destes hipercubos compreende o hipervolume HV :

$$HV = \text{volume} \left(\bigcup_{i=1}^{|Q|} v_i \right) \quad (10)$$

Deve-se observar que quanto maior o valor de HV , mais distante do ponto W se encontram as soluções geradas, e, por conseguinte, mais próximas da frente e mais e diversificadas ao longo da frente elas estão.

Os parâmetros utilizados no algoritmo PESA-II são apresentados a seguir: tamanho da população (*maxpop*): 100; número de gerações (*maxgen*): 10000; probabilidade de mutação (*p_{mut}*): 100%; espaçamento da grade (*ngrid*): 25. Tais parâmetros foram ajustados empiricamente após alguns testes computacionais, permitindo a análise do comportamento dos algoritmos.

Foram escolhidas três instâncias a serem testadas, no caso as linhas 905, 122 e 070, com vista a avaliar os benefícios da abordagem multiobjetivo em relação à prática corrente. A linha 905 é uma linha de caráter circular (bairro – centro da cidade), e as linhas 122 e 070 são linhas que partem de um terminal para um bairro.

Na Tabela 1, são apresentadas as características das instâncias analisadas. A coluna 1 consiste na identificação da instância, a qual é representada pelo número da linha. Na coluna 2 apresenta-se o número de viagens da linha. Na coluna 3 apresenta-se o número de serviços viáveis gerados, enquanto na coluna 4 apresenta-se o número de serviços inviáveis gerados.

Na coluna 4 é ilustrado o tempo de ciclo da linha. Deve-se ressaltar que, no sistema de transporte público por autocarros de Fortaleza, o tempo de ciclo varia ao longo do dia. Assim, é apresentada a moda da variável tempo de ciclo. Na coluna 5 é apresentada a quantidade máxima de serviços viáveis a comporem a solução. Na coluna 6, a quantidade máxima de serviços inviáveis a comporem a solução.

Tabela 1: Características das instâncias.

1.	2.	3.	6.	4.	5.	6.
Instância	<i>m</i>	<i>n</i>	<i>p</i>	ciclo (min)	<i>d</i>	γ
905	49	258	258	72	8	2
122	64	440	440	51	8	2
070	83	295	302	108	22	7

6. RESULTADOS OBTIDOS

Para a compreensão dos resultados reportados, deve-se recordar que $|Q|$ representa a cardinalidade do conjunto de soluções não dominadas, t representa o tempo de execução do algoritmo e HV denota o hipervolume compreendido pelo conjunto de soluções não-dominadas obtido em cada execução do PESA-II.

Tendo em vista que a metaheurística PESA-II é um processo de amostragem aleatória, optou-se por executar o algoritmo 10 vezes, para avaliar o comportamento médio do mesmo, em termos de qualidade de solução obtida e de tempo de processamento. O algoritmo proposto foi implementado em MATLAB. Os testes foram realizados num processador Genuine Intel 1.86 GHz com 1GB de memória RAM. Os resultados computacionais, para as três instâncias selecionadas, são apresentados na Tabela 2.

Tabela 2: Resultados computacionais.

<i>execução</i>	<i>905</i>			<i>122</i>			<i>070</i>		
	$ Q $	$t(s)$	HV	$ Q $	$t(s)$	HV	$ Q $	$t(s)$	HV
1	17	100,1	148776	21	277,2	190434	8	329,7	707863
2	9	104,5	145814	20	275,9	189644	12	339,7	726740
3	15	102,8	151593	20	281,7	189412	11	324,2	743178
4	18	104,7	152129	18	265,4	183209	10	371,9	725934
5	13	104,3	152540	18	239,2	174032	9	350,2	704636
6	15	102,1	149063	25	262,8	196520	5	318,7	698300
7	19	105,1	155748	24	249,7	183668	10	305,4	669251
8	12	96,3	145056	19	245,4	183722	11	318,6	700386
9	17	99,1	155068	19	277,5	188619	9	334,0	706808
10	17	102,4	152218	24	232,1	166091	11	326,7	696232
média	15	102,1	150800,5	21	260,7	184535,1	10	331,9	707932,8

Tabela 3: Soluções obtidas para as instâncias analisadas.

<i>Instância 905</i>			<i>Instância 122</i>			<i>Instância 070</i>		
<i>Solução</i>	f_1	f_2	<i>Solução</i>	f_1	f_2	<i>Solução</i>	f_1	f_2
ETUFOR	8	367	ETUFOR	10	676	ETUFOR	12	1691
ND1	4	235	ND1	6	341	ND1	5	581
ND2	5	199	ND2	7	340	ND2	6	574
ND3	6	193	ND3	8	257	ND3	7	497
ND4	7	157	ND4	9	256	ND4	8	377
ND5	8	153	ND5	10	251	ND5	9	324

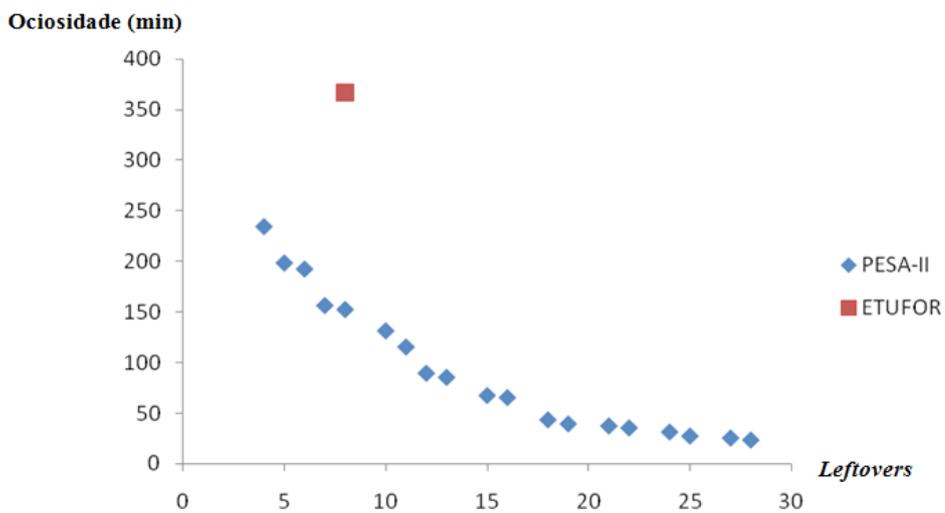
As soluções reais implementadas na prática do sistema de transporte público de Fortaleza, para as três instâncias em estudo, assim como as melhores soluções obtidas pelo algoritmo proposto, são apresentadas na Tabela 3. As soluções são avaliadas com relação às duas funções objetivo em questão: minimização de tramos descobertos (f_1) e minimização de tempos ociosos (f_2), expressos em minutos. As soluções geradas manualmente pelos planejadores da ETUFOR são comparadas com um conjunto de 5 soluções não-dominadas geradas pelo algoritmo PESA-II. Para cada instância, foram adotadas as soluções não-dominadas da execução do algoritmo que apresentou um maior hipervolume.

Com base na aproximação da frente de Pareto obtida para as três instâncias analisadas (Figura 2), pode-se afirmar que a solução manual da ETUFOR foi dominada por todas as soluções geradas pelo Algoritmo Genético Multiobjetivo proposto.

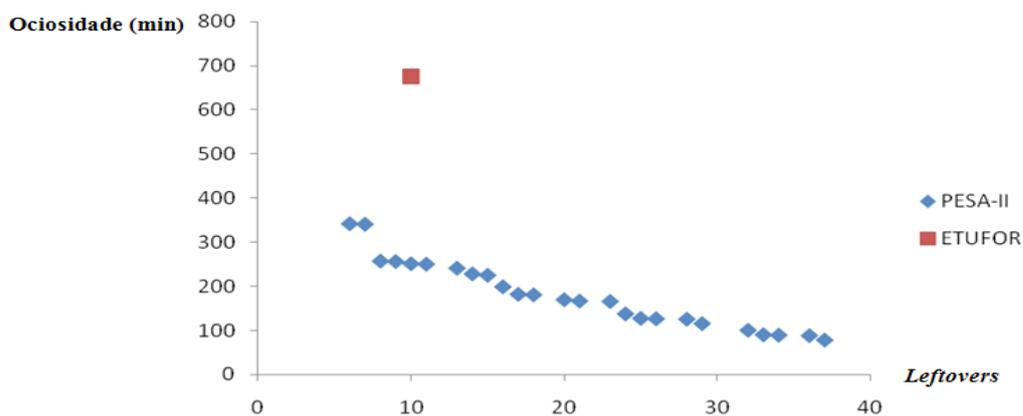
No que se refere à instância 905, a solução ND5 possui a mesma quantidade de tramos descobertos cons-

tante na solução da ETUFOR, mas possui um tempo ocioso que corresponde a 42% do tempo ocioso da solução da empresa supracitada. No que se refere à instância 122, a solução ND5 possui a mesma quantidade de tramos descobertos constante na solução da ETUFOR, mas possui um tempo ocioso que corresponde a 37% do tempo ocioso da solução da empresa supracitada. No que se refere à instância 070, as cinco soluções não-dominadas apresentadas apresentam melhor desempenho, em relação à solução manual gerada pela ETUFOR, para os dois critérios em análise. Dos resultados obtidos, percebe-se que há uma relação de *trade-off* entre o objetivo minimização do número de tramos descobertos e o objetivo minimização dos tempos ociosos.

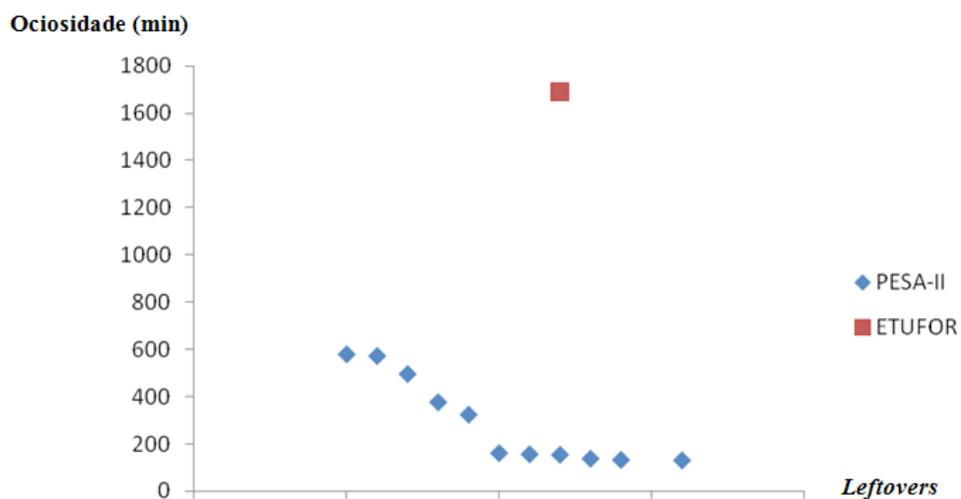
A abordagem proposta consiste em uma importante ferramenta de apoio à decisão para os planejadores do sistema, visto que podem fornecer soluções de qualidade, com baixo custo computacional e considerando múltiplos critérios, sem a necessidade da determinação de pesos para a ponderação destes.



(a) Instância 905



(b) Instância 122



(c) Instância 070

Figura 2: Aproximação da Fronteira de Pareto para as instâncias 905, 122 e 070.

7. CONCLUSÕES

Neste trabalho foi apresentada uma abordagem multiobjetivo para o problema de programação integrada de veículos e tripulações. Para a geração de aproximações da fronteira de Pareto, foram desenvolvidos algoritmos híbridos baseados na metaheurística evolucionária PESA-II.

No que concerne ao estudo de caso, pode-se salientar que a abordagem baseada em otimização multiobjetivo incorreu em soluções de melhor qualidade do que as implementadas na prática corrente, sob o prisma dos critérios analisados.

O automatismo na obtenção de soluções também é um grande benefício das abordagens propostas, visto que, em Fortaleza, apesar do porte do sistema de transportes coletivos desta cidade, este ainda é planejado manualmente, exigindo um grande esforço o qual pode ser significativamente reduzido pela adoção de um procedimento automático.

Em conclusão, pode-se afirmar que uma abordagem multiobjetivo para o VCSP, tal como a desenvolvida neste trabalho, deverá permitir um planejamento de melhor qualidade, assim como mais flexível, no sentido de fornecer ao decisor a possibilidade de escolher soluções entre um conjunto de alternativas, sem a necessidade de arbitrar pesos para a ponderação dos critérios em análise.

REFERÊNCIAS

- Baita, F.; Pesenti, R.; Ukovich, W. e Favaretto, D. (2000) A Comparison of Different Solution Approaches to the Vehicle Scheduling Problem in a Practical Case. *Computers & Operations Research*, v. 27, p. 1249–1269. DOI: 10.1016/S0305-0548(99)00073-8
- Ball, M., Bodin, L. e Dial, R. (1983) A Matching Based Heuristic for Scheduling Mass Transit Crews and Vehicles. *Transportation Science*, v. 17, p. 4–31.
- Bartodziej, P., Derigs, U., Malcherek, D. e Vogel U. (2007) Models and Algorithms for Solving Combined Vehicle and Crew Scheduling Problems With Rest Constraints: an Application to Road Feeder Service Planning in Air Cargo Transportation. *OR Spectrum*, v. 31, p. 405–429. Disponível em: <<http://link.springer.com/article/10.1007%2F00291-007-0110-7>> Acesso em: 12 ago. 2015.
- Beasley, J. E. e Chu, P.C. (1998) Constraint Handling in Genetic Algorithms: the Set Partitioning Problem. *Journal of Heuristics*, v. 11, p. 323–357. DOI: 10.1023/A:1008668508685
- Ceder, A. (2002) Urban transit scheduling: framework, review and examples. *Journal of Urban Planning and Development*, v. 128, p. 225–244. DOI: 10.1061/(ASCE)0733-9488(2002)128:4(225)
- Corne, D. W.; Knowles, J. D.; Oates, M. J. (2000) The Pareto Envelope-based selection algorithm for multiobjective optimization. *Proceedings of sixth International Conference on parallel problem solving from Nature*, Paris, 2000. Disponível em: <http://link.springer.com/chapter/10.1007%2F3-540-45356-3_82> Acesso em: 12 ago. 2015.
- Corne, D. W.; Jerram, N. R.; Knowles, J. D.; Oates, M. J. (2001) PESA-II: Region-based selection in evolutionary multiobjective optimization. *Proceedings of the Genetic and Evolutionary Computation Conference (GECCO-2001)*, San Francisco. Disponível em: <<https://www.macs.hw.ac.uk/~dwcorne/pesaII.pdf>> Acesso em: 12 ago. 2015.
- Daduna, J. R. e Paixão, J. M. P. (1995) Vehicle Scheduling for public mass transit – an Overview. In: Daduna, J. R., Branco, I. e Paixão, J. M. P. (Eds.) *Computer-Aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*, Springer, v. 430, p. 76–90.
- Deb, K. *Multi-objective optimization using evolutionary algorithms*. Chichester: John Wiley & Sons, 2001.
- Dias, M. T. G. (2005) A new approach to the bus driver scheduling problem using multiobjective genetic algorithms. *Ph.D. Thesis (Doctorate in Engineering Sciences)*. Faculty of Engineering, University of Porto, Porto. Disponível em: <<https://repositorio-aberto.up.pt/bitstream/10216/11426/2/Texto%20integral.pdf>> Acesso em: 12 ago. 2015.
- Falkner, J. C. e Ryan, D. M. (1992) Express: Set Partitioning for Bus Crew Scheduling in Christchurch. In: Desrochers, M. e Rosseau, J. M. (Eds.) *Computer-Aided Transit Scheduling: Proceedings of the Fifth International Workshop*, Springer, p. 359–378.
- Fischetti, M., Lodi, A., Martello, S. e Toth, P. (2001) A Polyhedral Approach to Simplified Crew Scheduling and Vehicle Scheduling Problems. *Management Science*, v. 47, p. 833–850. DOI: 10.1287/mnsc.47.6.833.9810
- Fleurent, C. e Rosseau, J. M. (2007) Integrated Vehicle and Crew Scheduling in Practice. In: *Technical Report GIRO*, Montreal.
- Freling, R., Wagelmans, A. P. M. e Paixão, J. M. P. (1999) An Overview of Models and Techniques for Integrating Vehicle and Crew Scheduling. In: Wilson, N. H. M. (ed.) *Computer-aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*, Springer, v. 471, p. 441–460, Springer.
- Freling, R., Huisman, D. e Wagelmans, A.P.M. (2001) Applying an Integrated Approach to Vehicle and Crew Scheduling in Practice. In: Voß, S. e Daduna, J. R. (eds.) *Computer-aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*, Springer, v. 505, p. 73–90.
- Freling, R., Huisman, D. e Wagelmans, A. P. M. (2003) Models and Algorithms for Integration of Vehicle and Crew Scheduling. *Journal of Scheduling*, v. 6, p. 63–85. Disponível em: <<http://link.springer.com/article/10.1023%2FA%3A1022287504028>> Acesso em: 12 ago. 2015.
- Friberg, C. e Haase, K. (1999) An Exact Branch and Cut Algorithm for the Vehicle and Crew Scheduling Problem. In: Wilson, N. H. M. (ed.) *Computer-aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*, Springer, v. 471, p. 63–80.

- Gaffi, A. e Nonato, M. (1999) An Integrated Approach to Ex-urban Crew and Vehicle Scheduling Problem. In: Wilson, N. H. M. (ed.) *Computer-aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*, Springer, v. 471, p. 103–128.
- Groot, S. W. e Huisman, D. (2008) Vehicle and Crew Scheduling: Solving Large Real-World Instances with an Integrated Approach. In: Hickman, M. Mirchandani e Voß, S. (eds.) *Computer-aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*, Springer, v. 600, p. 43–56. Disponível em: <http://link.springer.com/chapter/10.1007%2F978-3-540-73312-6_3> Acesso em: 12 ago. 2015.
- Haase, K., Desaulniers, G. e Desrosiers, J. (2001) Simultaneous Vehicle and Crew Scheduling in Urban Mass Transit Systems. *Transportation Science*, v. 35, p. 286–303. DOI: 10.1287/trsc.35.3.286.10153
- Huisman, D. (2004) Integrated and Dynamic Vehicle and Crew Scheduling. *Tese (Doutorado), Tinbergen Institute, Erasmus University Rotterdam*, Rotterdam. Disponível em: <http://repub.eur.nl/pub/6779/few_huisman_20040220.pdf> Acesso em: 12 ago. 2015.
- Huisman, D., Freling, R. e Wagelmans, A.P.M. (2005) Multiple-depot Integrated Vehicle and Crew Scheduling. *Transportation Science*, v. 39, p. 491–5025. Disponível em: <<http://pubsonline.informs.org/doi/abs/10.1287/trsc.1040.0104>> Acesso em: 12 ago. 2015.
- Klabjan, D., Johnson, E. L., Nemhauser, G. L., Gelman, E. e Ramanaswamy, S. (2001) Solving Large Airline Crew Scheduling Problems: Random Pairing Generation and Strong Branching. *Computational Optimization and Applications*, v. 20, p. 73–91. DOI: 10.1023/A:1011223523191
- Kliwer, N., Amberg, B., e Amberg, B. (2012) Multiple Depot Vehicle and Crew Scheduling with Time Windows for Scheduled Trips. *Public Transport*, v. 3, p. 213–244. DOI: 10.1007/s12469-011-0049-6
- Konak, A.; Coit, D. W. e Smith, A. E. (2006) Multiobjective Optimization Using Genetic Algorithms: a Tutorial. *Reliability Engineering and System Safety*, v. 91, p. 992–1007. DOI: 10.1016/j.res.2005.11.018
- Laurent, B. e Hao, J.K. (2007) Simultaneous Vehicle and Driver Scheduling: a Case Study in a Limousine Rental Company. *Computers & Industrial Engineering*, v. 53, p. 542–558. DOI: 10.1016/j.cie.2007.05.011
- Laurent, B. e Hao, J. K. (2008) Simultaneous Vehicle and Crew Scheduling for Extra Urban Transports. In: *XXI International Conference on Industrial Engineering & Other Applications of Applied Intelligent Systems*, Wroclaw. DOI: 10.1007/978-3-540-69052-8_49
- Lourenço, H. R.; J. P. Paixão; Portugal, R. (2000) Multiobjective Metaheuristics for the Bus-driver Scheduling Problem. *Transportation Science*, v. 35, p. 331–342. DOI: 10.1287/trsc.35.3.331.10147
- Mesquita, M. e Paias, A. (2008) Set Partitioning/covering-based Approach for the Integrated Vehicle and Crew Scheduling Problem. *Computers & Operations Research*, v. 35, p. 1562–1575. DOI: 10.1016/j.cor.2006.09.001
- Mesquita M, Paias A, Respicio A (2009) Branching Approaches for Integrated Vehicle and Crew Scheduling. *Public Transport*, v. 1, p. 21–37. DOI: 10.1007/s12469-008-0005-2
- Patrikalakis, G. e Xerokostas, D. (1992) Experimentation with a New Decomposition Scheme of the Urban Public Transport Scheduling. In: Desrochers, M. e Rosseau, J. M. (eds.) *Computer-Aided Transit Scheduling: Proceedings of the Fifth International Workshop*, Springer, p. 407–425.
- Rodrigues, M. K., Souza, C.C. e Moura, A.V. (2006) Vehicle and Crew Scheduling for Urban Bus Lines. *European Journal of Operational Research*, v. 39, p. 491–5025. DOI: 10.1016/j.ejor.2004.06.035
- Silva, G. P.; Prates e R. F. C. (2014) Otimização da Escala Mensal de Motoristas de Ônibus Urbano Utilizand a Heurística Variable Neighborhood Search, *Transportes*, v. 22, p. 31–43. DOI: 10.14295/transportes.v22i1.698
- Silva, T. A. e Silva, G. P. (2015) O Uso da Metaheurística Guided Local Search para Resolver o Problema de Escala de Ônibus Urbano. *Transportes*, v. 23, p. 105–116. DOI: 10.14295/transportes.v23i2.856
- Silva, G.P. e Cunha, C.B. (2010) O Uso da Técnica de Busca em Vizinhaça de Grande Porte para a Programação da Escala de Motoristas de Ônibus Urbano. *Transportes*, v.18, p. 37–45. DOI: 10.14295/transportes.v18i2.422.
- Steinzen, I. (2007) *Topics in Integrated Vehicle and Crew Scheduling in Public Transport*. Tese (Doutorado em Ciências da Computação em Negócios), Universidade de Paderborn, Paderborn. Disponível em: <<http://d-nb.info/987434659/34>> Acesso em: 12 ago. 2015.
- Steinzen, I., Becker, M. e Suhl, L. (2007) A Hybrid Evolutionary Algorithm for the Vehicle and Crew Scheduling Problem in Public Transit. In: *2007 IEEE Congress on Evolutionary Computation – CEC*, Singapore. DOI: 10.1109/CEC.2007.4424963
- Steinzen, I.; Ginter, V.; Suhl, L. e Kliwer, N. (2010) A Time-Space Network Approach for the Integrated Vehicle and Crew Scheduling Problem with Multiple Depots. *Transportation Science*, v. 44, p. 367–382. DOI: 10.1287/trsc.1090.0304
- Valouxis, C. e Housos, E. (2002) Combined Bus and Driver Scheduling. *Computers and Operations Research*, v. 170, p. 8443–862. DOI: 10.1016/S0305-0548(00)00067-8
- Weider, S. (2007) *Integration of Vehicle and Duty Scheduling in Public Transport*. Tese (Doutorado em Ciências Naturais), Universidade Técnica de Berlim, Berlim. Disponível em: <<https://www.deutsche-digitale-bibliothek.de/binary/TLQVJPFLN7RTLDCM6BQMXWSJ6T6KZGS2/full/1.pdf>> Acesso em: 12 ago. 2015.

Wren, A. e Gualda, N. D. F. (1999) Integrated Scheduling of Buses and Drivers. In: Wilson, N. H. M. (ed.) *Computer-aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*, Springer, v. 471, p. 155–176.

Wren, A. e Rosseau, J. M. (1995) Bus Driver Scheduling – an Overview. In: Daduna, J. R., Branco, I. e Paixão, J. M. P. (eds.) *Computer-aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*, Springer, v. 430, p. 173–183.

Zitzler, E. e Thiele, L. (1999) Multiobjective Evolutionary Algorithms - A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, v. 3, p. 257–271. DOI: 10.1109/4235.797969

Zitzler, E.; Laumanns, M. e Thiele, L. (2001) SPEA 2: Improving the Strength Pareto Evolutionary Algorithm. In: *Technical Report, Swiss Federal Institute Technology, Zurich*. Disponível em: <<http://e-collection.library.ethz.ch/eserv/eth:24689/eth-24689-01.pdf>> Acesso em: 12 ago. 2015.