

# A Hybrid Genetic Algorithm for the Vehicle and Crew Scheduling in Mass Transit Systems

B. A. Prata

**Abstract**— The integrated vehicle and crew scheduling problem is a difficult and widely studied Combinatorial Optimization problem. Several studies have shown that exact approaches for this problem are not useful in practical situations due to the high computational costs involved. This paper describes a hybrid genetic algorithm for vehicle and crew scheduling, which is modeled as a maximal covering problem with multiples resources. In addition, an innovative mathematical formulation is presented. Computational results with real vehicle and crew scheduling problem instances are presented and discussed. These results indicate that the proposed approach has a considerable potential for achieving significant gains in terms of operation costs and planning times.

**Keywords**— Evolutionary Algorithms, GRASP, Maximal Covering Problem with Multiple Resources.

## I. INTRODUÇÃO

NO que concerne aos ônibus, o planejamento dos transportes urbanos pode ser decomposto nas seguintes principais etapas[1]: 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 [2,3].

Diversos pesquisadores têm relatado a forte interação entre os problemas de escalonamento de veículos e de motoristas e os ganhos em considerá-los conjuntamente [3,4,5]. O problema integrado é usualmente denominado *Vehicle and Crew Scheduling Problem* – VCSP. Diversos trabalhos também relatam a dificuldade em solucionar o VCSP de forma exata [4,5,6], fato que aconselha a adoção de abordagens baseadas em técnicas heurísticas.

Tradicionalmente, o escalonamento de veículos e motoristas é realizado de uma maneira sequencial. A programação de veículos (*Vehicle Scheduling Problem* – VSP) é realizada numa primeira etapa, sendo completada pela programação das tripulações (*Crew Scheduling Problem* – CSP). O CSP é, em geral, mais complexo que o VSP; portanto, não é uma estratégia adequada programar veículos sem ter em conta as tripulações, já que estas constituem o gargalo do processo. Em grande parte das empresas de transporte público, sobretudo nos países desenvolvidos, os custos com tripulações são os mais significativos. Assim, devem ser dirigidos esforços para minimizar tais custos, embora tendo sempre em conta o processo de alocação dos veículos.

Diversos trabalhos apresentam modelos matemáticos para o VCSP com um único depósito [4, 6, 7, 8]. Foram efetuados testes em instâncias reais e geradas aleatoriamente. Os

resultados apontam que estas formulações tendem a não permitir a obtenção de soluções exatas para problemas reais de grande porte.

Diversos trabalhos apresentam modelos matemáticos para o VCSP com múltiplos depósitos [5, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18] apresentam modelos matemáticos e métodos de resolução para o VCSP com múltiplos depósitos. Os resultados apresentados apontam para a complexidade de obtenção de soluções de elevada qualidade.

As abordagens baseadas em métodos exatos, tais como *Branch-and-Bound*, *Branch-and-Cut*, geração de colunas e relaxações lagrangeanas, em alguns casos se mostram inviáveis, devido aos tempos computacionais elevados. Outras limitações de tais abordagens consistem na elevada quantidade de parâmetros a ser ajustada. Entretanto, em termos de qualidade das soluções, tais abordagens vêm se mostrando as melhores reportadas na literatura.

Diante da complexidade de tratar o VCSP de forma exata, algoritmos heurísticos vêm sendo propostos na literatura [19, 20, 21]. 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 [22, 23, 24]. 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 apresentam meta-heurísticas para a resolução do problema em foco [25, 26, 27, 28, 29].

Uma meta-heurística *Simulated Annealing* é apresentada para solução do VCSP em uma aplicação real de uma empresa de transporte de limusines [25]. Tal problema tem elevada complexidade, pois os veículos são bastante diferenciados devido aos acessórios e os motoristas também são diferenciados de acordo com suas qualificações.

Três metaheurísticas são apresentadas para o problema de programação de veículos e de tripulações em uma empresa de transporte aéreo de cargas: *Simulated Annealing*, *Great Deluge Algorithm* e *Record-to-Record Travel* [26]. A complexidade do problema reside nas janelas de tempo, na legislação trabalhista para os motoristas e na alta heterogeneidade da frota.

Um algoritmo evolucionário híbrido é proposto para o problema de programação de veículos e motoristas envolvendo múltiplos depósitos. A abordagem híbrida combina as técnicas de relaxação lagrangeana, geração de colunas e Algoritmos Genéticos.

Metaheurísticas são propostas para a programação de

B.A. Prata, Universidade Federal do Ceará (UFC), Fortaleza, Ceará, Brasil, [haprata@ufc.br](mailto:haprata@ufc.br).

veículos e tripulações em transporte intermunicipal com tabelas de horário irregulares [27,28]. Os objetivos avaliados são custo e regularidade dos serviços, em instâncias aleatórias disponíveis para o problema de programação de veículos e motoristas com múltiplos depósitos. O autor salienta que tal abordagem pode prover o tomador de decisões de informação adicional, a qual pode enriquecer a qualidade das soluções geradas.

As abordagens heurísticas reportadas padecem de pouca generalidade, no sentido em que são muito específicas aos problemas para os quais foram desenvolvidas e de que não foram reportados experimentos em um grande número de instâncias.

O objetivo do trabalho reportado nesse artigo é elaborar um Algoritmo Genético (AG) híbrido para o VCSP. Para a modelagem do problema é proposta uma nova formulação baseada em um problema de cobertura com múltiplos recursos (*Maximal Covering Problem with Multiple Resources – MCPMR*).

O trabalho é composto por mais quatro seções. Na próxima, é apresentada a definição do problema. Na segunda seção, são apresentadas as novas formulações propostas para o VCSP. Na terceira seção, são apresentadas as abordagens heurísticas desenvolvidas para o VCSP. Na quarta seção, são apresentados os resultados computacionais obtidos, em instâncias adaptadas da literatura e em problemas reais. Por fim, apresentam-se algumas conclusões, e possíveis desenvolvimentos futuros para o trabalho.

## II. FORMULAÇÃO MATEMÁTICA PROPOSTA

Nesta seção são apresentadas duas formulações para o VCSP. Na primeira formulação, os tripulantes e veículos são considerados como um único recurso, visto que os *changeovers* são proibidos. Na segunda formulação, tripulantes e veículos são considerados como recursos distintos.

De acordo com a revisão bibliográfica realizada, constatou-se que o uso de modelos matemáticos para o VCSP é limitado, pois, mesmo para instâncias de pequeno porte, métodos exatos podem não obter as soluções ótimas em tempo computacional admissível. Nesse contexto, buscou-se a formulação de modelos que pudessem suplantam esta limitação, assumindo hipóteses que simplificam o problema, mas que preservam a utilidade dos modelos.

A seguir é apresentada a definição e a notação dos elementos para os modelos propostos para o VCSP, baseados no problema de máxima cobertura. Seja uma matriz, denotada por  $A$ , a qual representa o conjunto de serviços gerados. O número de linhas desta 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.

Conjuntos:  $M$ : conjunto de tramos; e  $N$ : conjunto de serviços. Parâmetros:  $d$ : número máximo de serviços permitido na solução; e  $c_{ik}$ : parâmetro igual à 1 se as viagens  $k$  e  $i$  são compatíveis, caso contrário  $c_{ik}=-\infty$ .

### i. Variáveis inerentes aos serviços das tripulações

Elas são variáveis binárias, tais como:

$$z_i = \begin{cases} 1, & \text{se o } i\text{-ésimo tramo não é coberto por um} \\ & \text{serviço presente na solução;} \\ 0, & \text{caso contrário;} \end{cases} \quad \forall i \in M.$$

$$x_j = \begin{cases} 1, & \text{se o } j\text{-ésimo serviço faz parte da solução;} \\ 0, & \text{caso contrário;} \end{cases} \quad \forall j \in N.$$

### ii. Variáveis inerentes aos veículos (ônibus)

Elas são variáveis binárias, tais como:

$$y_{ik} = \begin{cases} 1, & \text{se o veículo cobre a viagem } k \text{ exatamente} \\ & \text{após a viagem } i; \\ 0, & \text{caso contrário;} \end{cases} \quad \forall i, k \in M.$$

Na modelagem proposta, a inovação consiste no fato que os serviços são gerados diretamente sobre as viagens e não sobre blocos de viaturas definidos previamente, como ocorre na abordagem sequencial.

Com base em uma tabela de horários e no regime de trabalho dos tripulantes, são gerados serviços, de modo a conceber uma matriz de serviços (matriz  $A$ , conforme notação usada no artigo). Com base na mesma tabela de horários, serão programados os veículos para realizarem as viagens.

Assim, a programação dos tripulantes não está atrelada à programação dos veículos, como ocorre na abordagem sequencial. Ambas as programações (veículos e tripulações) são relacionadas com a tabela de horários. O objetivo consiste em cobrir a demanda (viagens) pelos recursos disponíveis (veículos e tripulações). Tem-se, então, um *problema de cobertura com múltiplos recursos*.

Seja  $T$  um conjunto de viagens em uma tabela de horários, de modo que  $|T| = M$ . A relação  $i \alpha k$  representa a compatibilidade entre as viagens  $i$  e  $k$ , ou seja, a viagem  $k$  pode ser efetuada após a viagem  $i$ , pelo mesmo veículo ( $c_{ik}$  é o parâmetro que modela esta possibilidade). As variáveis do modelo são as seguintes:

[MCPMR]

$$\text{Maximizar} \quad \sum_{i=1}^m \sum_{k=1}^m c_{ik} y_{ik} - \sum_{i=1}^m z_i \quad (1)$$

Sujeito a:

$$\sum_{i=1}^m y_{ik} \leq 1 \quad \forall k \in M \quad (2)$$

$$\sum_{k=1}^m y_{ik} \leq 1 \quad \forall i \in M \quad (3)$$

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

$$\sum_{j=1}^n x_j \leq d \quad (5)$$

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

$$z_i \geq 0 \quad \forall i=1, \dots, m. \quad (7)$$

$$y_{ik} \geq 0 \quad \forall i,k \in M \quad (8)$$

com  $c_{ik} = 1$  se  $i \alpha k$ , caso contrário  $c_{ik} = -\infty$ .

A função objetivo do problema (1) é composta por duas parcelas: a primeira, que procura cobrir todas as viagens com a menor quantidade de veículos possível, e a segunda, que se destina a minimizar a quantidade de tramos descobertos por serviços de tripulação. Os conjuntos de restrições (2) e (3) impõem que um veículo cubra uma viagem apenas uma vez. Visto que a matriz de coeficientes das restrições de um problema de alocação é unimodular, a integralidade das variáveis  $y_{ik}$  pode ser relaxada. Caso o  $i$ -ésimo tramo não possa ser coberto, o conjunto de restrições do tipo (4) implica que  $z_i = 1$ . Em consequência desta restrição, tem-se que a integralidade das variáveis  $z_i$  pode ser relaxada. A restrição (5) impõe que um número máximo de  $d$  colunas da matriz  $A$  seja selecionado na solução. Os conjuntos de restrições (6), (7) e (8) dizem respeito à definição das variáveis do modelo.

Se as viagens  $i$  e  $k$  são compatíveis, isto é, se a viagem  $k$  pode ser realizada após a viagem  $i$ ,  $c_{ik} = 1$ ; caso contrário  $c_{ik} = -\infty$ . Como o MCPMR é um problema de maximização, quando  $c_{ik} = -\infty$  penaliza-se a realização de viagens incompatíveis e o modelo buscará a programação de viagens compatíveis na solução ótima. O problema consiste em uma dupla cobertura: cobertura de viagens por meio de veículos e cobertura de viagens por meio de serviços de tripulações.

### III. ALGORITMO GENÉTICO HÍBRIDO PROPOSTO

Foi desenvolvido um algoritmo híbrido GRASP/AG. O GRASP é aplicado à programação de veículos e o AG é aplicado na programação de tripulações. A integração do processo se dá na cobertura de viagens por múltiplos recursos (veículos e tripulações), diferentemente do que ocorre na abordagem tradicional.

A meta-heurística GRASP [29,30] é baseada em um algoritmo construtivo guloso e aleatorizado o qual é refinado por um processo de busca local. Para a programação de veículos, foi desenvolvida uma heurística GRASP, reportada no Quadro 1.

A fase de construção é composta por uma heurística construtiva. Uma função gulosa  $\Gamma$  é usada como critério de inserção de viagens a um veículo. Para cada veículo, calcula-se a diferença entre o tempo de término da última viagem alocada naquele veículo e as demais viagens. Após a solução inicial construída, tem-se uma programação de veículos. Essa programação não leva em conta os serviços de tripulantes, por isso, na prática, não seria exequível. A fase de melhoria da heurística proposta consiste em concatenar a programação de veículos com a programação de tripulações, a ser realizada pelo AG.

Portanto, é na fase de melhoria do algoritmo de programação de veículos que reside a integração entre os dois

*schedulings*. Tal integração consiste em distribuir, de uma forma mais equilibrada, as viagens entre os veículos. Além disso, a fase de melhoria tenta reduzir os tempos que os veículos ficam a espera de viagens, evitando *changeovers* desnecessários.

A fase de melhoria é composta por dois tipos de movimentos: movimento de inserção caudal e movimento de inserção de blocos. O movimento de inserção caudal consiste na realocação das últimas viagens de um bloco para outro, de modo a equilibrar as viagens por veículo. O movimento de inserção de blocos tem como meta equilibrar as quantidades de viagens alocadas entre os veículos, por meio da inserção de subconjuntos de viagens (blocos) dos veículos mais utilizados para veículos menos utilizados.

Quadro 1 - Heurística GRASP para a programação de veículos.

```

Parâmetros: num_iter
Fase de construção
Movimento de inserção caudal
i ← 0
enquanto i ≤ num_iter faça
    Movimento de inserção de blocos
    i ← i + 1
fim-do-enquanto

```

Para a resolução do *crew scheduling*, foi desenvolvido um Algoritmo Genético híbrido, conforme ilustrado no Quadro 2. O AG proposto consiste em uma adaptação do algoritmo proposto por Beasley e Chu [31], adaptando à formulação de máxima cobertura.

Quadro 2 - Pseudocódigo do Algoritmo Genético proposto.

```

Passo 1: Gerar aleatoriamente uma população de
soluções viáveis.
Passo 2: Avaliar as soluções geradas.
Enquanto geração atual ≤ número máximo de gerações,
faça
    Passo 3: Efetuar seleção por torneio.
    Passo 4: Realizar o cruzamento uniforme com
pc=100%.
    Passo 5: Realizar mutação da solução com pm=100%.
    Passo 6: Aplicar os procedimentos DROP e ADD.
    Passo 7: Avaliar a aptidão da solução gerada.
    Passo 8: Substituir o pior indivíduo da população
pela solução gerada.
Fim-do-enquanto
Imprimir a melhor solução obtida.

```

A população inicial é gerada de forma aleatória, levando em conta a viabilidade das soluções geradas, de modo que são gerados apenas indivíduos viáveis (sem *overcovers*, ou seja, tramos cobertos por mais de uma tripulação simultaneamente).

A seleção dos pais a serem cruzados é efetuada por torneio binário: selecionam-se aleatoriamente 4 indivíduos da população e são realizados torneios com dois indivíduos. Os indivíduos mais aptos, isto é, aqueles com menores valores da função objetivo, serão escolhidos como pais.

A cada geração é gerada uma nova solução, por meio de cruzamento uniforme. Para cada gene da solução filho a ser gerada, é realizado um experimento, regido por uma distribuição de Bernoulli, no qual uma variável aleatória discreta  $b$  recebe um valor 0 ou 1. Se  $b$  for igual a 0, o filho receberá o valor correspondente ao gene do primeiro pai. Se  $b$

igual a 1, o filho receberá o valor correspondente ao gene do segundo pai.

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 randômico  $r$  entre  $[0,n]$  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).

Embora os indivíduos da população inicial sejam viáveis, após as operações de recombinação e mutação, esta viabilidade certamente será transgredida. Deste modo, é de fundamental importância a correção da inviabilidade da solução, bem como a melhoria da solução corrigida. Neste sentido, foram adaptados para o problema de máxima cobertura, os procedimentos de melhoria DROP e ADD [32].

Para mitigar o problema da convergência prematura da população, foi desenvolvido o seguinte procedimento para reposição das soluções. Se a solução gerada tiver melhor aptidão do que o indivíduo com pior aptidão na população, a solução ingressa na população no lugar do pior indivíduo. Caso a solução gerada não tenha uma melhor aptidão do que o indivíduo de pior aptidão na população, ocorrerá a substituição com uma probabilidade de 5%.

Caso só sejam aceitas soluções que incorram em melhoria, existe uma grande chance de o AG convergir para um ótimo local antes mesmo da execução de grande parte das gerações do algoritmo. Esse mecanismo permite a manutenção da diversidade da população ao longo das gerações do AG.

A seguir, são apresentados os parâmetros utilizados para as heurísticas propostas. Heurística para programação de veículos:  $num\_iter: \lfloor nv/2 \rfloor$ . Algoritmo Genético: tamanho da população ( $maxpop$ ): 100; número de gerações ( $maxgen$ ): 10000; e probabilidade de mutação ( $p_{mut}$ ): 100%.

#### IV. EXPERIMENTOS COMPUTACIONAIS

No que se refere à formulação proposta para o VCSP com *changeovers*, foram efetuados experimentos computacionais em instâncias reais, advindas do sistema de transporte público de Fortaleza, situada no Nordeste do Brasil. No sistema de transportes coletivos por ônibus de Fortaleza, *changeovers* não são permitidos. Contudo, tal restrição foi relaxada para uma avaliação teórica do modelo proposto.

A seguir, na Tabela I, são apresentados os resultados dos experimentos computacionais para o caso em que *changeovers* são permitidos e os veículos são alojados em um único depósito. A coluna 1 consiste na identificação da instância, a qual é representada pelo número da linha. A coluna 2 apresenta o número de viagens da linha. A coluna 3 apresenta o número de serviços gerados. Na coluna 4, é ilustrado o tempo de ciclo da linha, o qual consiste no tempo decorrido entre o início e o término de uma viagem. Deve-se ressaltar que, no sistema de transporte público por ônibus de Fortaleza, o tempo de ciclo varia ao longo do dia. Portanto, é apresentada a moda da variável tempo de ciclo. Na coluna 5, é apresentada a densidade da matriz de serviços gerados. Na coluna 6 é apresentada a quantidade máxima de colunas a comporem a solução. Na coluna 7, apresenta-se o número de colunas utilizado na

solução. Na coluna 8, apresenta-se o número de veículos obtido na solução ótima do modelo. Na coluna 9, consta o número de tramos descobertos. Na coluna 10, é apresentado o tempo de processamento para obtenção da solução ótima. Na coluna 11, consta o número de nós do *Branch-and-Bound* (B&B).

TABELA I. CARACTERÍSTICAS DAS INSTÂNCIAS E RESULTADOS DO SOLVER – INSTÂNCIAS COM *CHANGEOVERS*.

1. Instância	2. $m$	3. $n$	4. Ciclo (min)	5. $\rho$ (%)	6. $d$	7. $ x $	8. # veículos	9. $z$	10. $t(s)$	11. # nós (B&B)
RW406	85	712	110	4,5	21	21	12	5	2	8
RW504	64	446	48	13,9	7	6	4	10	5	84
RW905	49	382	72	12,0	8	7	4	7	4	647
RW013	75	530	55	10,6	9	8	5	11	45	11217
RW833	80	344	84	6,1	15	14	8	10	5	252
RW015	111	882	56	7,1	14	12	7	15	137	12489
RW501	64	450	48	13,9	7	6	4	10	9	144
RW407	67	608	94	5,8	14	14	10	11	3	189
RW466	74	515	52	10,7	9	8	5	11	4	24
RW609	69	513	90	6,2	14	14	9	9	1	0
RW907	72	550	72	8,2	12	11	6	7	4	49
RW810	64	435	48	14,0	7	6	4	10	6	31
RW605/606	86	766	100	4,4	20	19	14	11	7	133
RW081	59	392	48	15,2	6	5	4	14	2	16
RW070	83	744	110	4,6	21	20	11	7	1	0
RW102	104	741	50	7,6	12	12	6	9	2	0
RW201	73	505	62	9,5	10	9	6	10	3	26
RW411	84	620	62	8,2	12	10	8	15	7	232
RW316	55	421	91	7,4	12	12	7	7	1	0
RW913	80	554	42	12,4	8	7	4	10	21	1969

As heurísticas foram implementadas no MATLAB. Os experimentos foram realizados em um processador Genuine Intel 1.86 GHz com 1GB de memória RAM.

Tendo em vista que os algoritmos propostos consistem em processos de amostragem aleatória, optou-se por rodá-los 10 vezes, para avaliar o comportamento médio dos mesmos, em termos de qualidade de solução obtida e de tempo de processamento. Deve-se observar que a heurística GRASP proposta para a programação de veículos obteve o número ótimo de veículos em todas as replicações do algoritmo.

No que concerne ao AG desenvolvido para a programação de tripulantes, a heurística supracitada obteve a solução ótima em 19 das 20 instâncias analisadas (a solução ótima do problema 70 não foi encontrada pelo AG).

Na Tabela II é apresentada uma comparação entre as soluções obtidas pelo AG e as soluções ótimas obtidas pelo B&B. Para as 200 execuções do AG, este obteve a solução ótima em 163 rodadas, tendo um aproveitamento de 81,5%. Em 14 das 20 instâncias analisadas, o AG conseguiu obter a solução ótima em todas as 10 execuções do algoritmo, apontando para a eficácia do mesmo.

A seguir, na Tabela III, é feita uma análise comparativa entre o desempenho computacional (expresso em termos do tempo de processamento) do B&B e das heurísticas propostas. Com relação aos tempos de processamento, com base nos dados da Tabela III, pode-se constatar que as heurísticas apresentaram desempenho superior ao B&B, em 13 das 20 instâncias analisadas.

TABELA II. COMPARAÇÃO ENTRE O MÉTODO EXATO E O ALGORITMO GENÉTICO PROPOSTO.

Problema	Percentual de cobertura (c)				Gap (%)			Soluções ótimas em 10 execuções do GA
	zIP	Me-lhor	Pior	Mé-dia	Me-lhor	Pior	Mé-dia	
RW406	94,1	94,1	91,8	92,0	0,0	2,4	2,1	1
RW504	84,4	84,4	84,4	84,4	0,0	0,0	0,0	10
RW905	85,7	85,7	85,7	85,7	0,0	0,0	0,0	10
RW13	85,3	85,3	85,3	85,3	0,0	0,0	0,0	10
RW833	87,5	87,5	87,5	87,5	0,0	0,0	0,0	10
RW015	86,5	86,5	86,5	86,5	0,0	0,0	0,0	10
RW501	84,4	84,4	84,4	84,4	0,0	0,0	0,0	10
RW407	83,6	83,6	83,6	83,6	0,0	0,0	0,0	10
RW466	85,1	85,1	75,7	84,2	0,0	9,5	0,9	10
RW609	87,0	87,0	81,2	82,9	0,0	5,8	4,1	3
RW907	90,3	90,3	83,3	88,8	0,0	6,9	1,5	3
RW810	84,4	84,4	84,4	84,4	0,0	0,0	0,0	10
RW605	87,2	87,2	86,0	86,9	0,0	1,2	0,3	8
RW081	76,3	76,3	76,3	76,3	0,0	0,0	0,0	10
RW070	91,6	89,2	88,0	88,3	2,4	3,6	3,3	0
RW102	91,3	91,3	83,7	90,5	0,0	7,7	0,9	8
RW201	86,3	86,3	86,3	86,3	0,0	0,0	0,0	10
RW411	82,1	82,1	82,1	82,1	0,0	0,0	0,0	10
RW316	87,3	87,3	87,3	87,3	0,0	0,0	0,0	10
RW913	87,5	87,5	87,5	87,5	0,0	0,0	0,0	10

TABELA III. COMPARAÇÃO DO DESEMPENHO ENTRE O *BRANCH-AND-BOUND* E AS ABORDAGENS PROPOSTAS.

Instância	$t_{zIP}(s)$	$t_{GA}(s)$			Gap (%)		
		Melhor	Pior	Média	Melhor	Pior	Média
RW406	2	61	378	322	2950,0	18800,0	16000,0
RW504	5	1	2	1	-80,0	-60,0	-80,0
RW905	4	1	1	1	-75,0	-75,0	-75,0
RW013	45	1	3	1	-97,8	-93,3	-97,8
RW833	5	5	10	7	0,0	100,0	40,0
RW015	137	16	47	30	-88,3	-65,7	-78,1
RW501	9	1	1	1	-88,9	-88,9	-88,9
RW407	3	2	25	12	-33,3	733,3	300,0
RW466	4	3	116	44	-25,0	2800,0	1000,0
RW609	1	14	159	123	1300,0	15800,0	12200,0
RW907	4	6	153	101	50,0	3725,0	2425,0
RW810	6	1	1	1	-83,3	-83,3	-83,3
RW605	7	20	375	139	185,7	5257,1	1885,7
RW081	2	1	11	3	-50,0	450,0	50,0
RW070	1	353	372	363	35200,0	37100,0	36200,0
RW102	2	12	297	98	500,0	14750,0	4800,0
RW201	3	1	5	3	-66,7	66,7	0,0
RW411	7	3	166	23	-57,1	2271,4	228,6
RW316	1	3	77	24	200,0	7600,0	2300,0
RW913	21	1	96	27	-95,2	357,1	28,6

## V. CONCLUSÕES

Neste trabalho foram apresentadas novas formulações para a programação integrada de veículos e de tripulações em sistemas de transporte público, baseadas no *Maximum Covering Problem* – MCP. No melhor conhecimento do autor, não existe nenhum outro trabalho que reporte uma formulação para o *Vehicle and Crew Scheduling Problem* baseada no MCP.

Nas 20 instâncias analisadas, pode-se constatar que o tempo computacional médio requerido pelo B&B, para a obtenção da solução ótima, é de apenas 13,5 segundos. Deste modo, a nova formulação proposta representa um ganho do ponto de vista

prático, pois pode subsidiar o planejamento operacional em empresas de transporte público. O GRASP/GA desenvolvido para o VCSP com *changeovers* apresentou um desempenho satisfatório, visto que foi obtido um *gap* médio de 0,7% para as 20 instâncias analisadas.

A abordagem proposta se mostrou eficiente para a resolução prática do problema em estudo, podendo ser aplicadas no planejamento operacional de sistemas de transporte público, incorrendo em reduções de custos e de tempos de planejamento.

Embora o algoritmo GRASP/GA tenha apresentado um bom desempenho em comparação ao B&B, não foi possível compará-lo com outras abordagens presentes na literatura. Isso se deve ao fato de que a abordagem proposta no presente artigo é a primeira baseada em um MCPMR. Deste modo, não se pode ter uma comparação justa entre os resultados obtidos por outras formulações. O desenvolvimento de uma medida de comparação entre resultados obtidos pela abordagem proposta neste artigo e outras abordagens é objeto de investigação para estudos futuros.

Este artigo reporta apenas uma parte de um projeto de pesquisa mais amplo. Tendo em vista a natureza multiobjetivo de problemas de programação de veículos e tripulações em sistemas de transporte público, os autores estão desenvolvendo modelos e algoritmos que levem em consideração a natureza multicritério dos problemas reais neste domínio.

## REFERÊNCIAS

- [1] Ceder, A. "Urban transit scheduling: framework, review and examples". *Journal of Urban Planning and Development*, v. 128, pp. 225 – 244, 2002.
- [2] Daduna, J. R. e Paixão, J. M. P. "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, vol. 430, pp. 76–90, 1995.
- [3] Wren, A. e Rosseau, J. M. "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, vol. 430, pp. 173–183, 1995.
- [4] Freling, R., Wagelmans, A. P. M. e Paixão, J. M. P. "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, pp. 441–460, Springer, 1999.
- [5] Gaffi, A. e Nonato, M. "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, pp. 103–128, 1999.
- [6] Friberg, C. e Haase, K. "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, pp. 63–80, 1999.
- [7] Haase, K., Desaulniers, G. e Desrosiers, J. "Simultaneous vehicle and crew scheduling in urban mass transit systems". *Transportation Science*, vol. 35, pp. 286–303, 2001.
- [8] Huisman, D. "Integrated and dynamic vehicle and crew scheduling". Tese de Doutorado, Tinbergen Institute, Erasmus University Rotterdam, Rotterdam, 2004.
- [9] Fischetti, M., Lodi, A., Martello, S. e Toth, P. "A polyhedral approach to simplified crew scheduling and vehicle scheduling problems". *Management Science*, vol. 47, n. 6, pp. 833–850, 2001.
- [10] Freling, R., Huisman, D. e Wagelmans, A.P.M. "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, pp. 73–90, 2001.

- [11] Freling, R., Huisman, D. e Wagelmans, A. P. M. "Models and algorithms for integration of vehicle and crew scheduling". *Journal of Scheduling*, v. 6, pp. 63 – 85, 2003.
- [12] Huisman, D., Freling, R. e Wagelmans, A.P.M. "Multiple-depot integrated vehicle and crew scheduling". *Transportation Science*, vol. 39, pp. 491–5025, 2005.
- [13] Weider, S. "Integration of vehicle and duty scheduling in public transport". Tese de Doutorado, Universidade Técnica de Berlim, Berlim, 2007.
- [14] Groot, S. W. e Huisman, D. "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, pp. 43 – 56, 2008.
- [15] Mesquita, M. e Paias, A. "Set partitioning/covering-based approach for the integrated vehicle and crew scheduling problem". *Computers & Operations Research*, vol. 35, pp. 1562–1575, 2008.
- [16] Mesquita M, Paias A, Respicio A "Branching approaches for integrated vehicle and crew scheduling". *Public Transport*, vol. 1, pp. 21–37, 2009.
- [17] Steinzen, I.; Ginter, V.; Suhl, L.; Kliwer, N. "A Time-Space Network Approach for the Integrated Vehicle and Crew Scheduling Problem with Multiple Depots". *Transportation Science*, vol. 44, pp. 367–382, 2010.
- [18] Kliwer, N., Amberg, B., Amberg, B. "Multiple depot vehicle and crew scheduling with time windows for scheduled trips". *Public Transport*, vol. 3, pp. 213–244, 2012.
- [19] Ball, M., Bodin, L. e Dial, R. "A matching based heuristic for scheduling mass transit crews and vehicles". *Transportation Science*, vol. 17, pp. 4–31, 1983.
- [20] Patrikalakis, G. e Xerokostas, D. "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, pp. 407–425, 1992.
- [21] Valouxis, C. e Housos, E. "Combined bus and driver scheduling". *Computers and Operations Research*, v. 170, pp. 8443–862, 2002.
- [22] Falkner, J. C. e Ryan, D. M. "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, pp. 359–378, 1992.
- [23] Wren, A. e Gualda, N. D. F. "Integrated Scheduling of Buses and Drivers". In: Wilson, N. H. M. (Ed.) *Computer-aided Transit Scheduling, Lecture Notes in Economics and Mathematical Systems*, Springer, vol. 471, pp. 155–176, 1999.
- [24] Rodrigues, M. K., Souza, C.C. e Moura, A.V. "Vehicle and crew scheduling for urban bus lines". *European Journal of Operational Research*, vol. 39, pp. 491–5025, 2006.
- [25] Laurent, B. e Hao, J.K. "Simultaneous vehicle and driver scheduling: a case study in a limousine rental company". *Computers & Industrial Engineering*, vol. 53, pp. 542–558, 2007.
- [26] Bartodziej, P., Derigs, U., Malcherek, D. e Vogel U. "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". *Operation Research - OR Spectrum*, 2007.
- [27] Steinzen, I. "Topics in integrated vehicle and crew scheduling in public transport". Tese de Doutorado, Universidade de Paderborn, Paderborn, 2007.
- [28] Steinzen, I., Becker, M. e Suhl, L. "A hybrid evolutionary algorithm for the vehicle and crew scheduling problem in public transit". In: 2007 *IEEE Congress on Evolutionary Computation – CEC*, Singapore, 2007.
- [29] Feo, T.A., e Resende, M.G.C. "A probabilistic heuristic for a computationally difficult set covering problem". *Operations Research Letters*, 8, 67–71, 1989.
- [30] Feo, T.A., e Resende, M.G.C. "Greedy randomized adaptive search procedures". *Journal of Global Optimization*, 6, 109–133, 1995.
- [31] Beasley, J. E. e Chu, P.C. "Constraint handling in genetic algorithms: the set partitioning problem". *Journal of Heuristics*, vol. 11, pp. 323–357, 1998.



**Bruno de Athayde Prata** é graduado em Engenharia Civil pela Universidade Federal do Ceará (UFC), Fortaleza, Ceará, Brasil, em 2006. Obteve o título de mestre em Logística e Pesquisa Operacional pela Universidade Federal do Ceará (UFC), Fortaleza, Ceará, Brasil, em 207 e de Doutor, em Engenharia Industrial e Gestão pela Universidade do Porto (UP), Porto, Portugal, em 2011. É professor Adjunto da Universidade Federal do Ceará (UFC). Atualmente suas pesquisas se concentram na área de problemas de roteamento e sequenciamento, com aplicações em Logística e Gestão de Operações.