

Dynamic Approach for a Demand Responsive Transport Service

Renan José dos Santos Viana
 Departamento de Informática
 Universidade Federal de Viçosa
 Campus da UFV, 36.570-900, Viçosa - MG - Brasil
 renan.viana@ufv.br

André Gustavo dos Santos
 Departamento de Informática
 Universidade Federal de Viçosa
 Campus da UFV, 36.570-900, Viçosa - MG - Brasil
 andre@dpi.ufv.br

Abstract- Demand Responsive Transport is a flexible transportation service that provides transport on demand by a fleet of vehicles, being especially useful in sparsely inhabited areas, which deal with a lack of transportation service. Users formulate requests specifying desired locations and times of pickup and delivery. Given a set of requests, the vehicle routes are to be planned and scheduled in order to minimize both the number of vehicles used and the journey durations, while respecting a set of constraints imposed by the service, the passengers and the vehicles. In a dynamic environment, requests arrive gradually along the day, and the routes must be adjusted in real time to accommodate the new demand. In this work we propose an approach that simulates a dynamic service using mixed linear programming models to produce good solutions. The obtained results for a set of instances are significant when compared to a static approach.

Index Terms—demand responsive transport, integer programming, dynamic service

I. INTRODUÇÃO

Todos os dias milhares de pessoas necessitam de transporte para se deslocarem até ao trabalho, escolas, supermercados, shoppings, aeroportos entre outros destinos. Este fato surge em decorrência das grandes distâncias a serem percorridas e da necessidade de reduzir o tempo gasto durante o traslado. Para isso, muitas destas pessoas necessitam dos serviços de transporte público.

Oferecer um serviço de transporte público com qualidade em regiões onde a demanda por transporte é baixa e variável é extremamente dispendioso. Geralmente operacionalizados por meio de rotas e horários fixos, sistemas de transporte tradicionais não podem adaptar-se economicamente e adequadamente às constantes variações da demanda [1].

Segundo Ambrosino *et al.* [2], nos últimos 50 anos ocorreu um declínio na utilização do transporte público em relação ao uso de veículos particulares. É cada vez maior a aquisição e o uso de veículos próprios. Além disso, a utilização de ônibus se tornou menos eficaz e menos eficiente como meio de transporte de pessoas [3]. Diante desta situação é interessante buscar serviços de transporte alternativos competitivos que atraem passageiros e contribuem para a redução do intenso tráfego, do ruído, da poluição (emissão de gases) e atrasos

causados pelo grande número de veículos particulares presente nas vias.

Transporte Reativo a Demanda, na língua inglesa *Demand Responsive Transport* (DRT), é um serviço de transporte operacionalizado sob demanda por meio de uma frota de veículos, os quais são agendados para coletar e entregar passageiros de acordo com suas necessidades [4]. Usuários formulam requisições nas quais determinam locais e horários de embarque e desembarque desejados [2]. Usualmente este tipo de serviço de transporte é compartilhado, ou seja, mais de um passageiro pode estar em um veículo ao mesmo tempo [5].

Um serviço DRT é extremamente útil em regiões que lidam com a falta de serviços de transporte tradicionais, dado que seus provedores não querem aceitar o custo de um serviço subutilizado. Por exemplo, em áreas esparsas com baixa densidade populacional ou horários com baixa procura pelo serviço [6] [1].

Em Ambrosino *et al.* [2], são citadas três das principais razões do crescimento em popularidade do DRT: deficiências do transporte público e dos serviços de táxi; deficiências de serviços especiais de transporte; e novos desenvolvimentos na comunidade do transporte. De acordo com Mageean e Nelson [4] há também um interesse em seu potencial no combate à exclusão social. Um serviço DRT é uma forma intermediária de transporte, situado entre o transporte público (transporte compartilhado) e os táxis (transporte personalizado) [2].

Em sua forma usual, um serviço DRT pode ser relacionado com o *Dial-a-Ride Problem* (DARP), problema de transporte bem conhecido na literatura científica [7]. Geralmente aplicado em serviços porta-a-porta destinado à idosos e deficientes, o DARP consiste em projetar rotas e definir horários de atendimento para um número de passageiros, os quais especificam requisições de coleta e entrega entre origens e destinos desejados. O planejamento das rotas busca atender o maior número possível de requisições minimizando os custos de operação e respeitando um conjunto de restrições [8].

O DARP em conjunto com o *Pickup and Delivery Problem* (PDP) compõem a classe de problemas em que bens e passageiros são transportados de pontos de coleta para pontos de entrega intitulada *Vehicle Routing Problems with Pickup and Delivery* (VRPPD). Ambos problemas são baseados em

requisições de transporte compostas por um par de pontos, os quais representam os locais de coleta e entrega dos passageiros. Entretanto, há uma importante diferença entre o PDP e o DARP. O primeiro lida com transporte de bens, enquanto o segundo foca o transporte de passageiros. Geralmente esta diferença é expressa por meio de restrições adicionais, ou por objetivos extras ou substituindo os objetivos relacionados ao custo, a fim de manter a qualidade do serviço [9] [10]. A forma como um DARP é modelado pode variar de acordo com a situação do mundo real para qual será aplicado.

A principal diferença entre um serviço DRT e o clássico DARP é a flexibilidade. O primeiro aceita atrasos nos atendimentos, os quais são limitados por restrições a fim de manter a qualidade do serviço. Outra diferença é o modo em que os usuários enviam requisições de transporte: no DARP, os usuários frequentemente enviam duas requisições durante o dia, uma para o transporte de um ponto de coleta para outro de entrega e outra para realizar a viagem de volta. Por exemplo, um deficiente que necessita se locomover de sua casa para o hospital para um tratamento e quando terminar, retornar para casa. Em um serviço DRT isto não ocorre necessariamente. Para mais informações sobre DARP e DRT, recomenda-se a leitura de [8] e [10].

Neste artigo, é apresentada uma modelagem matemática adaptada para um serviço DRT baseada no modelo DARP proposto em [11]. Também é introduzido um algoritmo de otimização que simula um ambiente dinâmico dado um conjunto de requisições, o qual otimiza dois objetivos relacionados ao custo final de operação do serviço DRT. Nosso objetivo é buscar soluções que necessitam de um baixo número de veículos e que atendem todas às requisições de forma a minimizar a duração das rotas.

O restante do artigo é organizado da seguinte forma: Na seção 2 é apresentada uma revisão literária que engloba alguns dos principais trabalhos sobre o problema abordado e trabalhos relacionados. Na seção 3 é introduzida uma definição formal do DRT abordado e são apresentados dois modelos de programação linear mista para o problema. Na seção 4 a abordagem exata proposta é detalhada. Finalmente, os resultados computacionais e a conclusão são apresentados nas seções 5 e 6, respectivamente.

II. REVISÃO LITERÁRIA

Na literatura, há diversas variações do DARP. A maioria dos trabalhos consideram a variante estática do problema, na qual todas as requisições são conhecidas antes do processo de planejamento das rotas. Cordeau [11] introduziu uma formulação matemática para o DARP e um algoritmo branch-and-cut. O algoritmo adota inequações válidas já utilizadas para tratar conhecidos problemas da literatura, como: *Traveling Salesman Problem* (TSP), *Vehicle Routing Problem* (VRP) e o *Pickup and Delivery Problem* (PDP), assim como propõem novas inequações válidas para o DARP.

Em [12], foram introduzidos dois novos algoritmos branch-and-cut para o clássico *Pickup and Delivery Problem with Time Windows* (PDPTW) e o DARP. Propõem também duas

novas formulações para o PDPTW as quais, ao contrário da formulação de [11], possui um número exponencial de restrições, mas conduz para algoritmos mais eficientes, porque contêm menos variáveis e fornecem limites mais apertados. Várias famílias de inequações válidas são propostas para fortalecer as duas formulações propostas. Os algoritmos foram testados em vários conjuntos de instâncias de teste do PDPTW e DARP. Ambos algoritmos obtiveram desempenho superior ao algoritmo branch-and-cut proposto por [11], o qual encontrou soluções ótimas para instâncias com no máximo 36 requisições. Rooke et al. [12] resolveram de forma ótima instâncias com até 96 requisições.

Parragh *et al.* [13] desenvolvam duas novas formulações matemáticas para tratar o DARP com usuários e veículos heterogêneos. Esta variante do problema surge em decorrência do transporte de idosos e deficientes. Ambas formulações são utilizadas para compor algoritmos branch-and-cut. Os algoritmos resolveram de forma ótima instâncias com no máximo 40 requisições.

Como uma alternativa aos métodos de otimização exatos, existem vários trabalhos na literatura que utilizam algoritmos aproximados. Em [14], uma heurística busca tabu é proposta para tratar o DARP estático, a qual minimiza o custo total de roteamento considerando restrições de janelas de tempo, tempo máximo de viagem do usuário e duração máxima das rotas.

Em [15] um serviço DRT é abordado como um caso especial do DARP no qual são adicionadas algumas especificidades do serviço e uma formulação multi-objetivo. Uma abordagem evolutiva foi proposta, assim como uma nova representação para as soluções e novos operadores de variação. Tais módulos foram integrados a três algoritmos estado da arte: *Non dominated Sorting Genetic Algorithm II* (NSGA-II) [16], *Strength Pareto Evolutionary Algorithm 2* (SPEA-2) [17], e *Indicator Based Evolutionary Algorithm* (IBEA) [18]. Para intensificar a busca no espaço de soluções, Chevrier *et al.* [7] trataram o DARP aplicado ao contexto do DRT por meio de uma abordagem evolutiva multi-objetivo híbrida baseada nos algoritmos utilizados em [15]. As rotas são otimizadas por meio de um algoritmo de busca local dentro do operador de mutação. Esta estratégia é baseada na meta-heurística *Iterated Local Search* (ILS) e contém o algoritmo 2-opt integrado.

Outra variante do DARP é a dinâmica, na qual o processo de roteamento é realizado em tempo real. Em outras palavras, novas requisições podem chegar durante o processo de planejamento e as rotas já existentes devem ser adaptadas para atendê-las. Métodos exatos não têm sido elaborados para tratar o DARP dinâmico. Segundo Parragh *et al.* [9] uma possível razão para tal fato, é que no contexto de roteamento dinâmico, o conceito de soluções ótimas torna-se discutível. Entretanto, mesmo que o ambiente do serviço seja dinâmico, pode-se aceitar como solução ótima a melhor solução para o momento atual. Em contrapartida existem vários trabalhos que utilizam métodos aproximados.

Colorni e Righini [19] abordam o DARP dinâmico dividindo-o em dois subproblemas: o primeiro agrupa os pas-

sageiros em subconjuntos e o segundo roteia cada subconjunto. Os dois subproblemas são solucionados alternadamente por meio de um algoritmo de duas-fases. Considerando somente as requisições mais urgentes, os subproblemas de roteamento são solucionados de forma ótima por meio de um algoritmo *branch-and-bound*. Em [20], a heurística busca tabu proposta por Cordeau e Laporte [14] foi adaptada por meio de paralelização para tratar o DARP dinâmico. Diferentes implementações paralelas foram testadas no trabalho desenvolvido. Coslovich *et al.* [21] desenvolveram um algoritmo de duas-fases baseado em perturbações em rotas. A primeira fase é realizada off-line e tem como objetivo a criação de uma vizinhança viável a partir da rota corrente. A segunda fase é executada em tempo real toda vez que uma nova requisição é recebida, sendo capaz de determinar de forma rápida a inclusão ou rejeição de uma requisição por meio de um simples processo de inserção. As perturbações são realizadas utilizando o algoritmo de troca de arestas 2-opt. Já Berbeglia *et al.* [22] introduziram uma abordagem híbrida que combina um algoritmo de programação por restrições exato e uma heurística busca tabu para o DARP dinâmico.

III. DEFINIÇÃO DO PROBLEMA

Em sua forma geral, um serviço DRT pode ser formulado como um *Demand Responsive Transport Problem* (DRTP) e é classificado como um problema de otimização combinatória multiobjetivo, ou na língua inglesa, *multi-objective combinatorial optimization problem* (MOCOP). Os modelos matemáticos propostos neste artigo são adaptações do modelo DARP apresentado em [11], nos quais é introduzido o conceito de flexibilidade (tolerância a atrasos). Neste trabalho, o problema é tratado considerando duas funções objetivo (bi-objetivo). Deseja-se minimizar tanto o número de veículos utilizados quanto a duração total das rotas. Dado que o problema abordado é uma generalização do DARP, o mesmo é considerado NP-Difícil dado a sua complexidade.

O DRTP pode ser definido por meio de um grafo completo direcionado $G = (V, A)$, onde $V = V^+ \cup V^- \cup \{0, 2n+1\}$ é o conjunto de todos vértices e A é o conjunto de todas as arestas. Os subconjuntos $V^+ = \{1, \dots, n\}$ e $V^- = \{n+1, \dots, 2n\}$ contém todos vértices de coleta e entrega, respectivamente. Um total de n usuários (ou requisições) a serem servidos consistem de um vértice de coleta i e um vértice de entrega $n+i$. Os vértices 0 e $2n+1$ representam os depósitos de origem e destino.

Seja K uma frota homogênea de veículos com k veículos idênticos, todos com capacidade Q . Para cada usuário $i = 1 \dots n$ é associado um número de passageiros q_i e uma duração de serviço d_i para entrada e saída de passageiros. Para os vértices de coleta e entrega define-se $q_i = -q_{n+i}$ ($i = 1, \dots, n$) e $d_i > 0$, e para os depósitos $q_0 = q_{2n+1} = 0$ e $d_0 = d_{2n+1} = 0$. Para cada aresta $(i, j) \in A$ é associado um tempo de viagem t_{ij} não nulo.

Cada usuário $i = 1, \dots, n$ define um horário de coleta desejado h_{i+} . A duração da janela de tempo w_{i+} em um ponto de coleta é proporcional a duração $t_{i, n+i}$ do percurso de i

para $n+i$, definida como: $w_{i+} = k_w \cdot t_{i, n+i}$, sendo k_w um coeficiente que determina a duração alocada para a janela de coleta. O horário máximo de coleta h'_{i+} é então definido como $h'_{i+} = h_{i+} + w_{i+}$. O horário teórico de chegada no ponto de entrega h_{i-} é a soma do horário de coleta desejado e a duração do percurso de i para $n+i$ ($h_{i+} + t_{i, n+i}$). O horário máximo de entrega h'_{i-} é definido como: $h'_{i-} = h_{i+} + (k_r \cdot t_{i, n+i})$ sendo k_r um coeficiente de relaxação.

Neste trabalho, o DRTP foi resolvido utilizando dois Modelos de Programação Mista. As principais variáveis de decisão são:

- x_{ij}^k : Variável binária definida como 1 se o veículo k percorre a aresta (i, j) , 0 caso contrário.
- v^k : Variável binária definida como 1 se o veículo k é utilizado, 0 caso contrário.

Outras variáveis de decisão são utilizadas para modelar as restrições e os objetivos:

- t^k : Tempo total de viagem do veículo k se o mesmo é utilizado, 0 caso contrário.
- D_i : Atraso na entrega do usuário i
- H_i^k : Horário de chegada do veículo k no nó i , 0 se não é servido pelo veículo.
- Q_i^k : Número de passageiros a bordo do veículo k após visitar o nó i , 0 se não é servido pelo veículo.

A. Modelo 1

O primeiro modelo matemático proposto tem como objetivo minimizar o número de veículos necessários para servir um conjunto de requisições respeitando um grupo de restrições impostas pelo problema. A quantidade de veículos disponíveis k é igual ao número de requisições a serem atendidas e o número mínimo de veículos necessários k_{min} é a solução ótima para este conjunto de requisições. No pior caso será utilizado um veículo para servir cada requisição ($k_{min} = n$). O DRTP otimizando o número de veículos utilizados pode ser formulado por meio do seguinte modelo de programação linear mista:

A função objetivo (1) minimiza o número de veículos utilizados. As restrições (2) e (3) garantem que cada requisição seja servida exatamente uma vez e que o mesmo veículo visite os nós de origem e entrega. As restrições (4), (5) e (6) asseguram que se o veículo k é utilizado, o início da rota seja no depósito de origem e o término no depósito de destino e que se o veículo visita um nó, ele deve deixar este nó. A restrição (7) certifica que o veículo será definido como utilizado se o mesmo serve alguma requisição. As restrições (8) e (9) garantem a consistência dos horários de visita e ocupação dos veículos. As restrições (10) e (11) definem a duração da rota de cada veículo e o atraso da chegada em cada nó de entrega. A restrição (12) remove a simetria na utilização dos veículos. Finalmente, o conjunto de restrições (13) define as variáveis binárias e os limites de cada uma das outras variáveis do modelo, assegurando que cada usuário seja servido dentro de sua respectiva janela de tempo e que a capacidade dos veículos seja respeitada a cada visita.

Função Objetivo

$$F_1 = \min \sum_{k \in K} v^k \quad (1)$$

Sujeito a:

$$\sum_{k \in K} \sum_{j \in V} x_{ij}^k = 1, \forall i \in V^+ \quad (2)$$

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{n+i,j}^k = 0, \forall i \in V^+, k \in K \quad (3)$$

$$\sum_{j \in V} x_{0j}^k = v^k, \forall k \in K \quad (4)$$

$$\sum_{j \in V} x_{ji}^k - \sum_{j \in V} x_{ij}^k = 0, \forall i \in V^+ \cup V^-, k \in K \quad (5)$$

$$\sum_{i \in V} x_{i,2n+1}^k = v^k, \forall k \in K \quad (6)$$

$$\sum_{j \in V} x_{ij}^k \leq v^k, \forall i \in V^+, k \in K \quad (7)$$

$$H_j^k \geq H_i^k + d_i + t_{ij} - M_{ij}^k(1 - x_{ij}^k), \quad (8)$$

$$\forall i \in V, j \in V, k \in K$$

$$\text{where } M_{ij}^k \geq \max\{0, h_{i-} + d_i + t_{ij} - h_{j+}\}$$

$$Q_j^k \geq Q_i^k + q_j - W_{ij}^k(1 - x_{ij}^k), \quad (9)$$

$$\forall i \in V, j \in V, k \in K$$

$$\text{where } W_{ij}^k \geq \min\{Q, Q + q_i\}$$

$$t^k = H_{2n+1}^k - H_0^k, \forall k \in K \quad (10)$$

$$D_i \geq \max\{0, H_i^k - h_{i-}\}, \forall i \in V^-, k \in K \quad (11)$$

$$v^{k-1} \geq v^k, \forall k \in K \setminus 1 \quad (12)$$

$$v^k \in \{0, 1\}, \forall k \in K \quad (13)$$

$$x_{ij}^k \in \{0, 1\}, \forall i \in V, j \in V, k \in K$$

$$t^k \geq 0, \forall k \in K$$

$$D_i \geq 0, \forall i \in V^-$$

$$h_{i+} \leq H_i^k \leq h_{i+} + w_{i+}, \forall i \in V^+, k \in K$$

$$h_{i-} \leq H_i^k \leq h'_{i-}, \forall i \in V^-, k \in K$$

$$\max\{0, q_i\} \leq Q_i^k \leq \min\{Q, Q + q_i\}, \forall i \in V, k \in K$$

B. Modelo 2

O segundo modelo matemático tem como objetivo a minimização da duração total das rotas. Para isso, o tamanho da frota de veículos disponíveis é fixado e assume o valor K_{min} obtido por meio da resolução do primeiro modelo, sendo assim um parâmetro de entrada. Além de uma função objetivo diferente, o segundo modelo carece de algumas modificações em relação as restrições vistas no modelo anterior. As restrições (4) e (6) são alteradas e as restrições (7) e (12) são removidas do modelo. Com o intuito de facilitar a compreensão e permitir uma visão mais clara do modelo 2, mesmo as restrições que não foram alteradas serão repetidas. O segundo modelo de programação linear mista pode ser definido como:

Função Objetivo

$$F_2 = \min \sum_{k \in K_{min}} t^k \quad (14)$$

Sujeito a:

$$\sum_{k \in K_{min}} \sum_{j \in V} x_{ij}^k = 1, \forall i \in V^+ \quad (15)$$

$$\sum_{j \in V} x_{ij}^k - \sum_{j \in V} x_{n+i,j}^k = 0, \forall i \in V^+, k \in K_{min} \quad (16)$$

$$\sum_{j \in V} x_{0j}^k = 1, \forall k \in K_{min} \quad (17)$$

$$\sum_{j \in V} x_{ji}^k - \sum_{j \in V} x_{ij}^k = 0, \forall i \in V^+ \cup V^-, k \in K_{min} \quad (18)$$

$$\sum_{i \in V} x_{i,2n+1}^k = 1, \forall k \in K_{min} \quad (19)$$

$$H_j^k \geq H_i^k + d_i + t_{ij} - M_{ij}^k(1 - x_{ij}^k), \quad (20)$$

$$\forall i \in V, j \in V, k \in K_{min}$$

$$\text{where } M_{ij}^k \geq \max\{0, h_{i-} + d_i + t_{ij} - h_{j+}\}$$

$$Q_j^k \geq Q_i^k + q_j - W_{ij}^k(1 - x_{ij}^k), \quad (21)$$

$$\forall i \in V, j \in V, k \in K_{min}$$

$$\text{where } W_{ij}^k \geq \min\{Q, Q + q_i\}$$

$$t^k = H_{2n+1}^k - H_0^k, \forall k \in K_{min} \quad (22)$$

$$d_i \geq \max\{0, H_i^k - h_{i-}\}, \forall i \in V^-, k \in K_{min} \quad (23)$$

$$x_{ij}^k \in \{0, 1\}, \forall i \in V, j \in V, k \in K_{min} \quad (24)$$

$$t^k \geq 0, \forall k \in K_{min}$$

$$D_i \geq 0, \forall i \in V^-$$

$$h_{i+} \leq H_i^k \leq h_{i+} + w_{i+}, \forall i \in V^+, k \in K_{min}$$

$$h_{i-} \leq H_i^k \leq h'_{i-}, \forall i \in V^-, k \in K_{min}$$

$$\max\{0, q_i\} \leq Q_i^k \leq \min\{Q, Q + q_i\}, \forall i \in V, k \in K_{min}$$

A função objetivo (14) minimiza a duração total das rotas. As restrições (15) e (16) garantem que cada requisição seja servida exatamente uma vez e que o mesmo veículo visite os nós de origem e entrega. As restrições (17), (18) e (19) asseguram que o início da rota seja no depósito de origem e o término no depósito de destino e que se o veículo visita um nó, ele deve deixar este nó. As restrições (20) e (21) garantem a consistência dos horários de visita e ocupação dos veículos. As restrições (22) e (23) definem a duração da rota de cada veículo e o atraso da chegada em cada nó de entrega. O conjunto de restrições (24) define as variáveis binárias e os limites de cada uma das outras variáveis do modelo, assegurando que cada usuário seja servido dentro de sua respectiva janela de tempo e que a capacidade dos veículos seja respeitada a cada visita.

IV. ABORDAGEM EXATA

Esta seção descreve a estrutura do algoritmo exato proposto para tratar o DRTP. Inicialmente é explicado como foi realizada a simulação de um serviço DRT dinâmico dado um

conjunto de requisições. Em seguida são descritas as etapas da execução do algoritmo.

A. Simulação do serviço DRT Dinâmico

Como já explicado anteriormente, um serviço DRT que opera de forma dinâmica deve ser capaz de lidar com novas requisições que chegam gradualmente ao longo do dia. Rotas que ainda não foram iniciadas ou que já estão em processo de atendimento devem ser adaptadas em tempo real para atender novas demandas de transporte. Caso o conjunto de rotas existentes não for capaz de absorver as novas requisições, são alocados mais veículos para atendê-las.

Dado um conjunto com n requisições, é construída uma lista com todas as requisições, a qual é ordenada em ordem crescente de acordo com o horário desejado de coleta. Para simular um serviço dinâmico o algoritmo exato é executado várias vezes em sequência. O processo de otimização inicia com a execução do algoritmo para as m (parâmetro de entrada) primeiras requisições da lista ordenada. Na segunda execução o algoritmo é executado para as $2m$ primeiras requisições. Na terceira para $3m$ primeiras requisições e assim por diante até atribuir todas as n requisições à uma rota. A inclusão das novas requisições é sempre realizada de forma acumulativa, onde a execução corrente lida com todas as requisições disponíveis nas execuções anteriores somadas às novas requisições. A cada execução o algoritmo retorna a melhor solução para as requisições disponibilizadas até o momento. Durante a simulação do ambiente dinâmico, as requisições já disponíveis podem ser divididas em 3 status, apresentados abaixo:

- 1) Requisição não atendida.
- 2) Requisição atendida mas não finalizada (passageiro a bordo do veículo).
- 3) Requisição atendida e finalizada (passageiro entregue).

Na primeira execução do algoritmo todas as requisições disponíveis são classificadas como não atendidas. De forma a reproduzir um serviço DRT real, este conjunto de requisições são solicitações de transporte realizadas antes do início do processo de planejamento das rotas. As demais requisições, que se tornam disponíveis nas execuções seguintes simulam a chegada de novas demandas e a necessidade de planejamento em tempo real.

B. Abordagem Exata Proposta

O algoritmo exato proposto para tratar o DRT dinâmico executa os dois modelos de programação linear mista já apresentados em sequência. Inicialmente o primeiro modelo é executado com o intuito de encontrar o número mínimo de veículos necessários para atender o conjunto de requisições disponíveis no momento e logo em seguida, dado o número de veículos obtido na primeira etapa, o segundo modelo é executado minimizando a duração total das rotas. Como entrada do algoritmo, são informados o conjunto de requisições R , a cardinalidade do conjunto $|R|$ (n) e o número m de novas requisições que irão surgir em cada etapa do planejamento das rotas. O algoritmo abaixo, descreve as etapas da abordagem proposta.

Algoritmo 1: ALGORITMOEXATO

Entrada: R, n, m

Saída: Solução Final

```

1 inicio
2    $nReq = m$ 
3    $sol = null$ 
4   enquanto  $nReq \leq n$  faça
5      $v = \text{modeloDRT1}(R, nReq, sol)$ 
6      $sol = \text{modeloDRT2}(R, nReq, sol, v)$ 
7      $nReq = nReq + m$ 
8   fim
9 fin
10 retorna  $sol$ 

```

Inicialmente, os dois modelos são executados para determinar o número de veículos necessários para atender as m requisições disponíveis e para este número mínimo de veículos definir a duração total mínima possível para atender à todas as requisições. Na segunda iteração, o número de requisições disponíveis é incrementado em m unidades. Para atualizar o status das requisições disponíveis na iteração anterior, o horário desejado de coleta mais tarde dentre todas as requisições disponíveis no momento é definido como o horário de chegada das m novas requisições e a solução obtida por meio da execução do segundo modelo na iteração anterior é utilizada para verificação dos horários de visita definidos. Assim, todas as requisições em que a coleta já ocorreu mas que ainda não foram entregues são definidas com o status (2) e todas que a coleta e entrega já foram efetuadas são definidas com o status (3). As demais, permanecem com o status (1). A partir da segunda iteração, à medida que as requisições passarem do status (1) para os status (2) ou (3), as visitas e seus respectivos horários serão fixados, não podendo mais serem alterados nas iterações posteriores.

C. Exemplo de Execução

Para exemplificar como a abordagem proposta simula um serviço DRT dinâmico é utilizado como exemplo um pequeno conjunto de requisições. A tabela I apresenta um conjunto R com $n = 20$ requisições ordenadas de acordo com os respectivos h_{i+} . Das vinte requisições, dez estarão disponíveis antes do início do processo de planejamento das rotas e as outras dez se tornarão disponíveis posteriormente, sendo assim, $m = 10$. O h_{i+} da m -ésima requisição (hachurado na tabela I) é definido como o horário da chegada das m novas requisições.

Na primeira iteração, após a execução em sequência dos dois modelos, é obtida uma solução na qual os horários de visita de cada uma das m primeiras requisições estão definidos. Utilizando o horário de chegada da nova demanda como base, é possível atualizar o status destas requisições. A tabela II apresenta os horários de visita obtidos na primeira iteração sendo que os horários hachurados são aqueles que aconteceram antes da chegada da nova demanda, consequentemente fixados

Tabela I
EXEMPLO DE UM CONJUNTO DE REQUISIÇÕES, ORDENADAS POR
HORÁRIO DE COLETA; EM DESTAQUE O HORÁRIO DE CHEGADA DAS 10
REQUISIÇÕES SEGUINTES.

ID	Origem	Destino	q_i	h_{i+}	h'_{i+}	h_{i-}	h'_{i-}
2	132	47	1	29160	29309,6	29908	30282
19	1109918	1109789	1	29160	29221,2	29466	29619
17	1110044	1109944	1	29280	29380,0	29780	30030
15	1109885	141	1	30000	30102,0	30510	30765
11	1110116	1110039	1	30180	30284,8	30704	30966
1	131	46	1	30240	30370,0	30890	31215
10	1110116	1110023	1	30300	30364,4	30622	30783
0	129	1110532	1	30360	30482,8	30974	31281
9	1109807	1109789	1	30480	30589,6	31028	31302
14	1110646	1109789	1	30540	30635,2	31016	31254
13	153	1110033	1	30660	30766,8	31194	31461
8	1110057	1109962	1	30660	30741,6	31068	31272
5	137	1110033	1	30960	31085,6	31588	31902
4	1109799	1110040	1	31020	31052,0	31180	31260
3	132	1110600	1	31080	31145,6	31408	31572
6	1109793	1109993	1	31260	31355,6	31738	31977
7	1109807	1109944	1	31440	31525,2	31866	32079
18	2344054	1109814	1	31680	31736,0	31960	32100
16	155	1109960	1	32160	32266,4	32692	32958
12	2263832	1109944	1	32280	32369,6	32728	32952

e mantidos inalterados nas próximas iterações. Os horários não hachurados representam as visitas (coleta ou entrega) que ainda não foram realizadas até o momento da chegada da nova demanda, sendo assim suscetíveis a alterações durante os planejamentos posteriores.

Tabela II
HORÁRIOS DE COLETA E ENTREGA DEFINIDOS PELA 1ª ITERAÇÃO DO
ALGORITMO; HORÁRIOS HACHURADOS SÃO FIXOS (INALTERÁVEIS) NA 2ª
ITERAÇÃO.

ID	Coleta	Entrega
2	29160,0	29911,0
19	29221,2	29530,2
17	29353,0	29856,0
15	30000,0	30513,0
11	30284,8	30946,0
1	30301,0	31007,0
10	30300,0	30625,0
0	30360,0	31130,0
9	30480,0	31031,0
14	30540,0	31019,0

V. EXPERIMENTOS COMPUTACIONAIS

Nesta seção, serão discutidos os resultados obtidos por meio da aplicação da abordagem dinâmica proposta e da abordagem criada para comparação em dois conjuntos de instância de teste disponibilizados na literatura. Tanto a abordagem proposta quanto a abordagem utilizada para comparação foram implementadas utilizando a biblioteca Concert C++ e executadas por meio do CPLEX 12.6. Os testes computacionais foram executados em um computador 3,40 GHz Intel Core i5, com 16 GB de RAM no sistema operacional Windows Seven.

A. Conjunto de Instâncias de Teste

Para simulação de um serviço DRT dinâmico, foram utilizados dois conjuntos de instâncias de teste introduzidos por

[7]. O primeiro conjunto, chamado “Rnd100” é composto por 10 instâncias com uma distribuição quase homogênea de clientes, as quais contêm 100 pedidos gerados aleatoriamente. O segundo conjunto, denotado por “Gravit100” contém 10 instâncias com uma distribuição heterogênea de clientes, que têm 100 pedidos gerados usando um modelo geográfico de pessoas ou de fluxos de mercadorias.

B. Dados de Entrada

Para a execução dos modelos, foram definidos alguns parâmetros necessários: todos os veículos terão capacidade máxima para quatro pessoas ($Q = 4$), a duração do serviço em todos os pontos de visita será de 3 minutos ($d_i = 3, \forall i \in V^+ \cup V^-$) e as constantes de janela de tempo e de relaxação são estabelecidas em 0.2 e 1.5, respectivamente ($k_w = 0.2$ e $K_r = 1.5$).

Para um serviço DRT dinâmico ser considerado eficiente, além de gerar boas soluções, também é importante que execute o planejamento de forma rápida, pois é operacionalizado em tempo real. Portanto, os experimentos computacionais foram realizados impondo um limite de tempo execução, de 2 minutos para as instâncias do conjunto “Gravit100” e um tempo limite de 4 minutos para as instâncias “Rnd100”. Note-se, que devido à imposição deste limite de tempo, existe a chance da solução gerada não ser a ótima.

Nos experimentos computacionais, foi definido $m = 10$, sendo assim, em cada instância de teste houve 10 iterações. Não foram usados valores maiores devido a dificuldade de se solucionar o modelo na otimalidade com um maior número de requisições disponíveis por iteração.

C. Método de Avaliação de Desempenho Utilizado

Para validar o desempenho da abordagem proposta, foi desenvolvido um algoritmo que executa em sequência os dois modelos apresentados separadamente para cada um dos subconjuntos com m requisições utilizados para simular o serviço DRT dinâmico em cada instância, sem contudo realizar o aproveitamento das rotas já construídas para atender requisições dos conjuntos anteriores. Ao final das execuções, todas as m soluções foram concatenadas com o intuito de formar uma única solução composta por todas as n requisições do conjunto R . O propósito da construção desta solução é simular parcialmente um serviço DRT dinâmico, no qual requisições chegam ao longo do planejamento das rotas, mas não é realizado o aproveitamento de rotas já construídas para atender requisições de conjuntos anteriores, ou seja, a otimização é processada de forma estática. Desta forma pode ser medido o ganho proporcionado pela abordagem dinâmica proposta, que considera aproveitar e redefinir as rotas em andamento durante o processo de planejamento, tanto em termos de quantidade de veículos utilizados quanto em duração total das rotas.

D. Resultados

As tabelas III e IV apresentam os resultados obtidos por meio da abordagem proposta para os conjuntos

“Gravit100” e “Rnd100”, respectivamente. Para cada instância são detalhados os valores: k_{min} (F1), duração total das rotas (F2), além dos GAP’s fornecidos pelo CPLEX e tempos de execução de cada modelo em segundos.

Nessas tabelas, quando o valor de GAP não é zero, a solução ótima não foi encontrada ou não pode ser provada dentro do tempo limite. Um “-” nos valores de F e GAP indica que nenhuma solução viável foi encontrada dentro do tempo de execução limite do modelo em questão. Quando isso acontece não são mostrados os resultados das demais iterações, porque cada iteração depende de uma solução viável da iteração anterior. Além disso, quando esta situação acontece no primeiro modelo o segundo não é executado, porque depende da solução encontrada pelo primeiro. Isto aconteceu apenas em algumas instâncias “Rnd100”.

De acordo com a tabela III, na grande maioria das execuções, tanto do primeiro quanto do segundo modelo, o tempo de resolução foi da ordem de poucos segundos. Nota-se que em apenas uma iteração da instância Gravit_5 dentre todas as iterações apresentadas na tabela não foi possível achar a solução ótima dentro dos 2 minutos de execução. Em todas as instâncias o número mínimo de veículos encontrado está abaixo de 32, o que mostra que a abordagem dinâmica satisfaz bem o seu propósito, agrupando o maior número possível de requisições em uma mesma rota de forma viável. Entretanto, as mesmas observações não podem ser feitas em relação aos resultados apresentados pela tabela IV. Devido a distribuição homogênea dos clientes nas instâncias do conjunto “Rnd100” o algoritmo não obteve soluções ótimas na mesma frequência que encontrou para as instâncias do conjunto “Gravit100” e para todas as instâncias o número mínimo de veículos foi próximo do número total de requisições. Para quatro dessas instâncias o primeiro modelo não conseguiu encontrar solução viável dentro do prazo máximo estipulado de 4 minutos.

Com o aumento do número de requisições a serem processadas simultaneamente, não foi possível minimizar o número de veículos. De toda forma, mesmo quando o primeiro modelo não consegue retornar uma solução viável em tempo hábil, é sempre possível utilizar um veículo para cada requisição que ainda não foi atribuída a uma rota. Essa quantidade máxima pode ser usada pelo segundo modelo, que pode ainda diminuir essa quantidade de veículos, embora não seja seu objetivo, quando por exemplo acomoda duas requisições próximas num mesmo veículo, o que diminui a distância total para atender às duas requisições.

A tabela V mostra a quantidade de veículos necessários nas soluções encontradas por cada abordagem, a cada conjunto de $m = 10$ requisições. Para cada instância são mostrados na primeira linha o número de veículos usados na solução da abordagem dinâmica aqui proposta, e na segunda linha o número de veículos usados obtido por uma abordagem estática, que resolve separadamente cada conjunto de m requisições.

Note que para as instâncias “Gravit100” houve uma grande economia de veículos. Para algumas instâncias o número de veículos utilizados se estabilizou após algumas iterações. Por exemplo, na instância Gravit100_8, en-

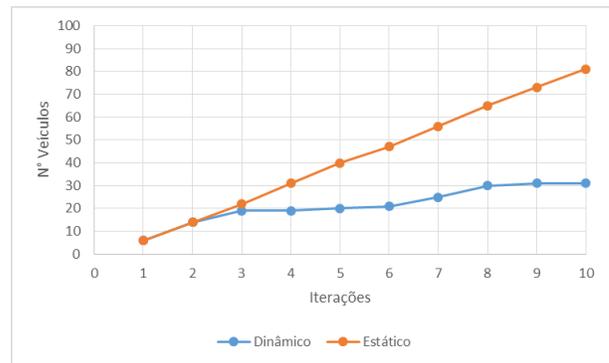


Figura 1. Comparação do aumento do número de veículos com a chegada de novas requisições para a instância Gravit100_0

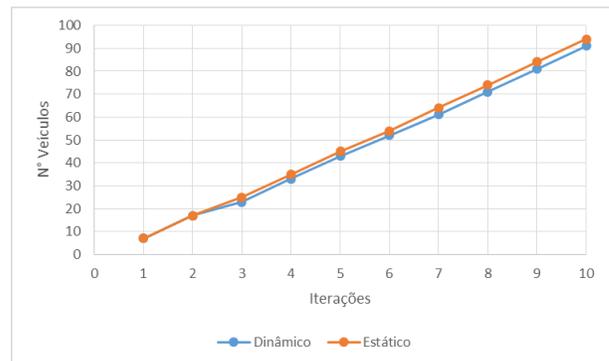


Figura 2. Comparação do aumento do número de veículos com a chegada de novas requisições para a instância Rnd100_0

quanto a abordagem estática usa 28 veículos na 3ª iteração, essa quantidade foi atingida na abordagem dinâmica apenas da 4ª iteração, quando já existiam 40 requisições. E essa quantidade, 28 veículos, se manteve até o final da simulação, contemplando todas as 100 requisições. O gráfico da Figura 1 ilustra o aumento do número de veículos a cada 10 requisições na instância Gravit100_0. Percebe-se que a abordagem dinâmica aqui proposta consegue acomodar muitas das novas requisições nos veículos que já estão atendendo as requisições anteriores, justificando o uso da abordagem dinâmica.

Entretanto, para as instâncias “Rnd100” a economia foi mínima. O gráfico da Figura 2 mostra o resultado para a instância Rnd100_0. Nessas instâncias as requisições possuem horários bastante homogêneos, o que dificulta o compartilhamento de veículos. Na instância Rnd_4, por exemplo, o número de veículos foi exatamente o mesmo.

De forma semelhante, a tabela VI compara as duas abordagens, mas desta vez em relação a duração total das rotas em cada iteração. É possível perceber que também para este objetivo a abordagem dinâmica atingiu resultados com qualidade superior à abordagem estática. Em ambos conjuntos de instância as durações totais das rotas quando há o aproveitamento de rotas já construídas obtiveram valores inferiores, provando a efetividade da abordagem também para este objetivo.

2015 XLI Latin American Computing Conference (CLEI)

Tabela III
 RESULTADOS OBTIDOS PELA ABORDAGEM PROPOSTA PARA CADA INSTÂNCIA DO CONJUNTO "GRAVIT100"

ITERAÇÕES	1	2	3	4	5	6	7	8	9	10
Gravit100_0										
F1	6	14	19	19	20	21	25	30	31	31
F2	10301,6	22455,6	29698	36068	43772	50363	58651	69326	75986	82810
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	6,3	2,0	0,9	1,3	2,5	4,3	6,3	9,0	18,1	36,7
Tempo2	1,2	1,1	0,8	0,6	0,9	1,3	2,4	3,2	4,2	10,9
Gravit100_1										
F1	8	14	15	21	26	26	26	26	28	30
F2	9091,2	18341,2	27367,2	38721,8	49645,8	56922,8	64941,8	74127,0	82847,0	92156,0
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	0,3	0,7	1,2	1,5	2,7	4,6	6,8	8,1	16,8	35,5
Tempo2	0,5	0,8	0,5	1,0	1,9	1,9	3,0	2,5	4,6	10,5
Gravit100_2										
F1	9	17	22	23	23	26	26	26	26	26
F2	10724	20614	32080	40954,4	49127,4	56110	64491	72424	80783	86824
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	0,4	0,7	1,1	2,1	2,8	10,2	7,6	9,2	17,1	37,2
Tempo2	0,5	0,9	1,4	1,1	1,5	2,5	5,0	3,4	4,0	5,8
Gravit100_3										
F1	10	16	19	22	23	25	26	27	27	27
F2	11590	22120	32070	42581,6	51875,6	60627,6	66595,6	75845,6	85381,6	93697,6
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	0,7	1,0	1,0	1,7	2,5	3,2	6,0	8,7	15,0	36,3
Tempo2	84,4	1,0	0,7	1,3	1,2	1,5	1,9	2,9	5,9	4,4
Gravit100_4										
F1	8	15	22	22	23	23	23	27	27	27
F2	11144	20481	30285	37803,8	43846,8	53195,8	61224,4	68980	75510	82337
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	0,5	0,9	1,0	1,5	2,4	4,0	5,6	9,3	16,0	29,2
Tempo2	0,7	2,8	1,0	0,6	1,4	1,7	1,7	2,7	5,5	5,3
Gravit100_5										
F1	9	16	18	18	19	22	23	23	24	26
F2	10344	19774,6	27725,4	35966,4	48929,4	59335,4	67632,4	76035,4	84585,4	92559,4
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	24	0	0	0	0	0	0	0	0	0
Tempo1	15,3	1,1	1,9	3,7	4,6	6,2	7,9	10,0	43,3	52,5
Tempo2	120,1	1,8	1,1	9,4	9,6	1,9	2,3	2,9	6,8	9,0
Gravit100_6										
F1	10	15	23	27	27	27	27	27	27	27
F2	11338	18892	28700	38712	45120	52435	61007	71638	82590	95836
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	0,2	2,4	1,7	4,7	2,6	4,4	6,5	9,2	21,8	39,4
Tempo2	0,4	7,5	2,8	1,8	1,5	4,0	4,1	4,6	5,4	5,8
Gravit100_7										
F1	9	16	20	20	21	21	24	27	28	28
F2	10214	21257	30340	37448	47576,6	58254,6	68428,6	75216,6	82082,6	88292,6
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	0,9	0,7	1,1	1,3	3,0	5,1	6,7	8,9	20,2	19,2
Tempo2	1,3	1,0	0,9	0,9	1,2	1,5	1,9	2,8	3,5	5,0
Gravit100_8										
F1	10	18	24	28	28	28	28	28	28	28
F2	16250	25290	33606,6	43178,2	49750,2	57311,2	67835,2	76089,2	84219,2	91181,2
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	0,3	1,4	2,3	1,5	3,5	4,7	5,9	8,1	26,4	27,0
Tempo2	0,4	2,0	1,2	1,4	2,9	8,6	5,8	3,0	14,9	6,4
Gravit100_9										
F1	9	15	19	20	21	21	21	24	29	29
F2	11488	21028,6	29682,2	35686,8	44918	52283,6	60058,6	70728,6	79504,6	86302,6
GAP1	0	0	0	0	0	0	0	0	0	0
GAP2	0	0	0	0	0	0	0	0	0	0
Tempo1	1,0	0,6	1,0	2,3	2,8	4,0	6,1	9,8	21,3	21,0
Tempo2	15,2	1,6	0,8	0,9	1,1	1,7	1,8	2,3	3,7	4,5

Tabela V
COMPARAÇÃO DO NÚMERO DE VEÍCULOS OBTIDOS EM CADA
ABORDAGEM

Instância	Iteração									
	1	2	3	4	5	6	7	8	9	10
Gravit100_0	6	14	19	19	20	21	25	30	31	31
Gravit100_1	8	14	15	21	26	26	26	26	28	30
	8	16	26	35	44	53	61	70	80	89
Gravit100_2	9	17	22	23	23	26	26	26	26	26
	9	19	28	38	47	56	64	73	82	91
Gravit100_3	10	16	19	22	23	25	26	27	27	27
	10	18	28	36	46	56	65	74	83	93
Gravit100_4	8	15	22	22	23	23	23	27	27	27
	8	16	26	36	45	55	65	75	84	93
Gravit100_5	9	16	18	18	19	22	23	23	24	26
	9	17	25	33	42	51	61	69	78	88
Gravit100_6	10	15	23	27	27	27	27	27	27	27
	10	19	28	37	45	53	62	71	81	90
Gravit100_7	9	16	20	20	21	21	24	27	28	28
	9	18	26	35	44	53	63	72	81	89
Gravit100_8	10	18	24	28	28	28	28	28	28	28
	10	19	28	38	46	55	64	74	82	91
Gravit100_9	9	15	19	20	21	21	21	24	29	29
	9	17	26	33	43	52	62	71	79	87
Rnd100_0	7	17	23	33	43	52	61	71	81	91
	7	17	25	35	45	54	64	74	84	94
Rnd100_1	9	17	23	32	41	50	59	68	78	87
	9	17	24	33	43	52	62	71	81	90
Rnd100_2	8	13	20	-	-	-	-	-	-	-
	8	17	26	36	45	54	64	74	84	93
Rnd100_3	9	19	28	37	46	56	65	75	85	95
	9	19	28	38	48	58	68	78	88	98
Rnd100_4	8	18	26	36	45	54	64	73	83	93
	8	18	26	36	45	54	64	73	83	93
Rnd100_5	9	17	26	35	43	53	62	72	82	92
	9	17	26	35	44	54	64	74	84	94
Rnd100_6	8	-	-	-	-	-	-	-	-	-
	8	18	27	37	46	56	66	76	86	96
Rnd100_7	8	17	26	34	43	52	62	71	81	91
	8	18	26	35	45	55	65	74	84	94
Rnd100_8	9	16	24	34	44	-	-	-	-	-
	9	17	26	36	46	55	65	75	85	95
Rnd100_9	8	16	-	-	-	-	-	-	-	-
	8	17	27	37	47	57	67	76	86	96

VI. CONCLUSÃO

Serviços de transporte reativos a demanda são necessários para cobrir lacunas deixadas pelos serviços de transporte público e de táxis. Sendo assim, é importante que sejam competitivos e eficientes. Operacionalizado com intuito de suprir demandas marginalizadas por outras opções de transporte, é importante que este tipo de serviço seja sustentável a ponto de manejar todas as demandas com o mínimo possível de veículos e trafegar de forma econômica e inteligente.

Neste trabalho foi proposto um algoritmo que, dada uma demanda, simula um ambiente dinâmico e busca soluções otimizadas. Esta abordagem resolve dois modelos de programação linear mista de forma que os objetivos de interesse em um serviço DRT fossem otimizados independentemente. Primeiro se busca o número mínimo de veículos necessários para servir toda a demanda e posteriormente busca-se a duração total mínima das rotas, dado o número de veículos encontrados.

Apesar da abordagem ser baseada em métodos exatos, consegue encontrar soluções em poucos segundos, característica desejável em um serviço dinâmico como o tratado neste trabalho. A abordagem pode inclusive ser utilizada em serviços com requisições estáticas, quando não for possível resolver o modelo com todas as requisições em uma única vez. Elas podem ser divididas em subconjuntos menores e serem acrescentadas aos poucos.

Comparada com a abordagem que trata os subconjuntos de requisições de forma estática, sem aproveitar rotas já existentes, a abordagem dinâmica proposta obtém resultados bem melhores que a concorrente em relação ao número de veículos necessários para suprir a demanda, principalmente quando as requisições não seguem um padrão homogêneo. Isto fica evidente analisando-se a quantidade de veículos necessários à medida que chegam novos subconjuntos de requisições. A abordagem dinâmica consegue aproveitar rotas que ainda estão em processo de atendimento de modo a reduzir o número de veículos que deverão ser acrescentados para servir a nova demanda e reduz também o tempo total de duração das rotas.

Como proposta de trabalhos futuros, serão exploradas técnicas para melhorar os modelos de programação linear mista, por exemplo melhoria de limites inferiores, para que consigam trabalhar com subconjuntos de requisições maiores e cheguem ao ótimo mais rápido. Além disso, será investigada a utilização de abordagens heurísticas que consigam tirar proveito das características do problema e obterem boas soluções demandando tempos computacionais baixos. Pode ser investigada ainda a versão multiobjetivo do problema, onde busca-se otimizar os dois objetivos simultaneamente, gerando um conjunto pareto de soluções.

AGRADECIMENTOS

Os autores agradecem à CAPES, CNPq e FAPEMIG pelo financiamento parcial do projeto. O primeiro autor trabalhou neste projeto com financiamento da CAPES.

REFERÊNCIAS

- [1] T. G. Crainic, F. Errico, F. Malucelli, and M. Nonato, "Designing the master schedule for demand-adaptive transit systems," *Annals of Operations Research*, vol. 194, no. 1, pp. 151–166, 2012.
- [2] G. Ambrosino, J. Nelson, and M. Romanazzo, *Demand responsive transport services: Towards the flexible mobility agency*. ENEA, Italian National Agency for New Technologies, Energy and the Environment, 2004.
- [3] G. M. P. T. Executive, M. Enoch, S. Potter, G. Parkhurst, and M. Smith, "Intermode: Innovations in demand responsive transport," 2004.
- [4] J. Mageean and J. D. Nelson, "The evaluation of demand responsive transport services in europe," *Journal of Transport Geography*, vol. 11, no. 4, pp. 255–270, 2003.
- [5] J.-F. Cordeau, G. Laporte, J.-Y. Potvin, and M. W. Savelsbergh, "Transportation on demand," *Transportation, handbooks in operations research and management science*, vol. 14, pp. 429–466, 2007.
- [6] R. Chevrier, "Optimisation de transport à la demande dans des territoires polarisés," Ph.D. dissertation, Université d'Avignon et des Pays de Vaucluse, 2008.
- [7] R. Chevrier, A. Liefooghe, L. Jourdan, and C. Dhaenens, "Solving a dial-a-ride problem with a hybrid evolutionary multi-objective approach: Application to demand responsive transport," *Applied Soft Computing*, vol. 12, no. 4, pp. 1247–1258, 2012.

- [8] J.-F. Cordeau and G. Laporte, "The dial-a-ride problem: models and algorithms," *Annals of Operations Research*, vol. 153, no. 1, pp. 29–46, 2007.
- [9] S. N. Parragh, K. F. Doerner, and R. F. Hartl, "A survey on pickup and delivery problems. part ii: transportation between pickup and delivery locations," *Journal für Betriebswirtschaft*, vol. 58, no. 1, pp. 81–117, 2008.
- [10] —, "Demand responsive transportation," *Wiley Encyclopedia of Operations Research and Management Science*, 2010.
- [11] J.-F. Cordeau, "A branch-and-cut algorithm for the dial-a-ride problem," *Operations Research*, vol. 54, no. 3, pp. 573–586, 2006.
- [12] S. Ropke, J.-F. Cordeau, and G. Laporte, "Models and branch-and-cut algorithms for pickup and delivery problems with time windows," *Networks*, vol. 49, no. 4, pp. 258–272, 2007.
- [13] S. N. Parragh, J.-F. Cordeau, K. F. Doerner, and R. F. Hartl, "Models and algorithms for the heterogeneous dial-a-ride problem with driver-related constraints," *OR spectrum*, vol. 34, no. 3, pp. 593–633, 2012.
- [14] J.-F. Cordeau and G. Laporte, "A tabu search heuristic for the static multi-vehicle dial-a-ride problem," *Transportation Research Part B: Methodological*, vol. 37, no. 6, pp. 579–594, 2003.
- [15] R. Chevrier, A. Liefvooghe, L. Jourdan, and C. Dhaenens, "On optimizing a demand responsive transport with an evolutionary multi-objective approach," in *Intelligent Transportation Systems (ITSC), 2010 13th International IEEE Conference on*. IEEE, 2010, pp. 575–580.
- [16] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *Evolutionary Computation, IEEE Transactions on*, vol. 6, no. 2, pp. 182–197, 2002.
- [17] E. Zitzler, M. Laumanns, L. Thiele, E. Zitzler, E. Zitzler, L. Thiele, and L. Thiele, "Spea2: Improving the strength pareto evolutionary algorithm," 2001.
- [18] E. Zitzler and S. Künzli, "Indicator-based selection in multiobjective search," in *Parallel Problem Solving from Nature-PPSN VIII*. Springer, 2004, pp. 832–842.
- [19] A. Colomi and G. Righini, "Modeling and optimizing dynamic dial-a-ride problems," *International transactions in operational research*, vol. 8, no. 2, pp. 155–166, 2001.
- [20] A. Attanasio, J.-F. Cordeau, G. Ghiani, and G. Laporte, "Parallel tabu search heuristics for the dynamic multi-vehicle dial-a-ride problem," *Parallel Computing*, vol. 30, no. 3, pp. 377–387, 2004.
- [21] L. Coslovich, R. Pesenti, and W. Ukovich, "A two-phase insertion technique of unexpected customers for a dynamic dial-a-ride problem," *European Journal of Operational Research*, vol. 175, no. 3, pp. 1605–1615, 2006.
- [22] G. Berbeglia, J.-F. Cordeau, and G. Laporte, "A hybrid tabu search and constraint programming algorithm for the dynamic dial-a-ride problem," *INFORMS Journal on Computing*, vol. 24, no. 3, pp. 343–355, 2012.

Tabela VI
COMPARAÇÃO DA DURAÇÃO TOTAL DAS ROTAS OBTIDAS EM CADA ABORDAGEM

Instância	Execução									
	1	2	3	4	5	6	7	8	9	10
Gravit100_0	10301,6	22455,6	29698,0	36068,0	43772,0	50363,0	58651,0	69326,0	75986,0	82810,0
	10301,6	22455,6	31484,8	41560,6	54678,6	62896,4	73528,4	85896,4	94853,8	103808,0
Gravit100_1	9091,2	18341,2	27367,2	38721,8	49645,8	56922,8	64941,8	74127,0	82847,0	92156,0
	9091,2	19457,2	32475,2	44935,8	57143,8	67459,8	78463,8	90439,8	101722,0	114819,0
Gravit100_2	10724,0	20614,0	32080,0	40954,4	49127,4	56110,0	64491,0	72424,0	80783,0	86824,0
	10724,0	21728,0	34978,0	48930,0	59293,4	69956,0	80351,2	91823,2	102117,0	113674,0
Gravit100_3	11590,0	22120,0	32070,0	42581,6	51875,6	60627,6	66595,6	75845,6	85381,6	93697,6
	11590,0	22656,0	36388,0	48898,0	61422,0	72472,0	84296,0	96608,0	109538,0	119740,0
Gravit100_4	11144,0	20481,0	30285,0	37803,8	43846,8	53195,8	61224,4	68980,0	75510,0	82337,0
	11144,0	20952,6	32292,6	44324,6	54564,6	67714,6	78524,6	88338,6	98954,6	109807,0
Gravit100_5	10344,0	19774,6	27725,4	35966,4	48929,4	59335,4	67632,4	76035,4	84585,4	92559,4
	10344,0	20298,6	30437,0	41217,4	52969,4	65421,4	77759,4	88553,4	101131,0	113219,0
Gravit100_6	11338,0	18892,0	28700,0	38712,0	45120,0	52435,0	61007,0	71638,0	82590,0	95836,0
	11338,0	21060,0	31220,0	42892,0	53044,0	63596,0	76102,0	88384,0	100796,0	111936,0
Gravit100_7	10214,0	21257,0	30340,0	37448,0	47576,6	58254,6	68428,6	75216,6	82082,6	88292,6
	10214,0	22829,0	33710,0	45534,0	59028,0	71754,0	84126,0	94124,0	104896,0	115070,0
Gravit100_8	16250,0	25290,0	33606,6	43178,2	49750,2	57311,2	67835,2	76089,2	84219,2	91181,2
	16250,0	25872,0	35218,6	47306,6	58784,4	67762,4	79356,4	91074,4	100470,0	113232,0
Gravit100_9	11488,0	21028,6	29682,2	35686,8	44918,0	52283,6	60058,6	70728,6	79504,6	86302,6
	11488,0	22058,6	33172,6	42401,6	55645,6	66793,6	78433,6	90287,6	100036,0	109764,0
Rnd_0	25914,0	54244,0	69806,0	91598,0	112518,0	128106,0	150006,0	167044,0	179286,0	192722,0
	25914,0	54506,0	74804,0	96596,0	117516,0	134404,0	156304,0	173342,0	185584,0	199020,0
Rnd_1	31534,0	55304,0	70836,0	93850,0	117184,0	138072,0	155074,0	172190,0	189038,0	201970,0
	31534,0	55304,0	76076,0	99090,0	122424,0	143312,0	162556,0	179672,0	196520,0	209452,0
Rnd_2	27588,0	43476,0	-	-	-	-	-	-	-	-
	27588,0	53786,0	78248,0	101614,0	122472,0	139746,0	159220,0	177092,0	191954,0	203552,0
Rnd_3	29690,6	56832,6	79950,6	100715,0	120045,0	140237,0	156619,0	176621,0	193497,0	208907,0
	29690,6	56832,6	79950,6	102909,0	124153,0	144345,0	162445,0	182447,0	199323,0	214733,0
Rnd_4	27898,0	54662,0	75614,0	100006,0	119188,0	139108,0	156388,0	171462,0	189198,0	203536,0
	27898,0	54662,0	75614,0	100006,0	119188,0	139108,0	156388,0	171462,0	189198,0	203536,0
Rnd_5	31268,0	57054,0	81724,0	104428,0	123084,0	142270,0	157364,0	175196,0	189134,0	202070,0
	31268,0	57054,0	81724,0	104428,0	124724,0	143910,0	160232,0	178064,0	192002,0	204938,0
Rnd_6	27254,0	-	-	-	-	-	-	-	-	-
	27254,0	56594,0	79192,0	102370,0	122738,0	143226,0	162968,0	179310,0	193156,0	208404,0
Rnd_7	26786,0	53400,0	76158,0	98016,0	119434,0	140520,0	159854,0	177676,0	192328,0	204728,0
	26786,0	55916,0	78674,0	100532,0	123510,0	146222,0	165556,0	183378,0	198030,0	210430,0
Rnd_8	29352,0	52704,0	73466,0	98338,0	120024,0	-	-	-	-	-
	29352,0	55382,0	78332,0	103204,0	124890,0	143702,0	162880,0	178114,0	193986,0	207728,0
Rnd_9	28716,0	52410,0	-	-	-	-	-	-	-	-
	28716,0	54430,0	79634,0	102650,0	124670,0	143278,0	162158,0	177904,0	195278,0	210634,0