

TRANSPORTE COMPARTILHADO COM ATENDIMENTO SUFICIENTEMENTE PRÓXIMO DE PASSAGEIROS: FORMALIZAÇÃO DO PROBLEMA E RESOLUÇÃO HEURÍSTICA

Allan Fernandes Balardino
Universidade Federal de Viçosa
Viçosa, MG - Brasil
allan.balardino@ufv.br

André Gustavo dos Santos
Universidade Federal de Viçosa
Viçosa, MG - Brasil
andre@dpi.ufv.br

RESUMO

O excesso de veículos nas ruas vem sendo uma dos principais adversidades nos grandes centros. Assim, o transporte compartilhado que se apresenta como uma maneira de amenizar o problema é um tema que tem sido bastante abordado na comunidade acadêmica. Uma revisão bibliográfica em diversos trabalhos na área de transporte compartilhado foi realizada e não foi encontrada nenhuma abordagem que fosse semelhante a proposta neste artigo. Nesta abordagem, uma pessoa que oferece carona não necessariamente precisa passar no ponto de origem de quem solicita a carona para o atender, mas sim em algum ponto próximo que seja mais conveniente para ela. Deste modo, temos um tratamento para o problema que mais se assemelha ao que acontece na prática. Foi feita uma formalização do problema, por meio de um modelo matemático de programação inteira e também uma proposta de heurística. Por fim, um algoritmo misto foi utilizado para encontrar soluções de boa qualidade. São apresentados resultados com diferentes números de motoristas e passageiros usando o mapa real de uma cidade de médio porte.

PALAVRAS CHAVE. Otimização combinatória, Transporte compartilhado, Metaheurística.

Área Principal: OC - Otimização Combinatória; MH - Metaheurísticas

ABSTRACT

The great number of vehicles in the streets is one of the biggest problem in big cities. Ridesharing, which has shown itself as a way to reduce the impact of this problem, is a subject widely discussed in the academic community nowadays. However, to the best of our knowledge, there is no paper in this subject including the characteristics we use in our work. In our approach, a person that offers a ride does not need to pass at the origin point of the person that request a ride but just at a point close enough point of it. This way, we have an approach more close to what happens in practice. In this paper, we propose a MILP formulation to the problem and a heuristic. Then we propose a mixed algorithm combining both approaches to reach good quality solutions. We show results for instances with different number of riders and drivers, using a map based on a real medium-size city.

PALAVRAS CHAVE. Combinatorial Optimization; Ridesharing; Metaheuristic.

Main area: OC - Combinatorial Optimization; MH - Metaheuristics

1. Introdução

O aumento da frota de veículos em circulação e a sobrecarga do sistema viário nas grandes cidades trazem diversos problemas como: desperdício de tempo e combustível ao longo do trajeto; subutilização do veículo, ou seja, lotação abaixo da capacidade; aumento da poluição ambiental e sonora; aumento dos gastos com manutenção, nível de stress, risco de acidentes, dentre outros.

Segundo o site Globo.com (2012) no ano de 2011 a venda de veículos por ano no Brasil atingiu o seu quinto recorde consecutivo, com uma média de 10 mil carros a mais por dia nas ruas. Com isso, temos uma tendência de cada vez mais veículos em circulação agravando os problemas da malha viária do país. Assim, é de grande necessidade mecanismos para o compartilhamento de transporte por meio de pessoas que já possuem seus veículos particulares e não utilizam sua capacidade máxima.

O uso compartilhado de um veículo por seu condutor e um ou mais passageiros é chamado *ridesharing*. Recentemente, há um grande interesse em *ridesharing* visto os grandes problemas causados pelo número excessivo de veículos no trânsito. Portanto, o tema tem sido abordado por muitos pesquisadores, já que se apresenta como mais uma solução para os problemas acima citados advindos do excesso de veículos no trânsito.

No trabalho de Herbawi & Weber (2012) cada participante, motorista ou passageiro, especifica uma origem e um destino de sua viagem. Também especifica uma janela de tempo que deseja viajar, ou seja, especifica um horário que gostaria de sair da sua origem e um horário que gostaria de chegar ao seu destino. Os motoristas definem o tempo máximo e a distância máxima que estão dispostos a percorrer, isso controla o quanto o motorista poderá desviar de seu caminho original para atender aos passageiros.

Mas na prática, quando uma pessoa solicita carona, por motivo de conveniência, o motorista a atende em um ponto que não altere muito sua rota e também não fique muito longe para a pessoa caminhar até ele. Assim, nossa abordagem será mais próxima do que acontece na realidade pois uma pessoa poderá ser atendida a uma certa distância de sua casa e não apenas com o carro tendo que passar exatamente em seu ponto de origem.

O objetivo deste trabalho é propor uma nova abordagem para o problema de *ridesharing*, combinando características dos problemas já propostos na literatura. Deste modo, será formalizado o problema que é mais semelhante às situações que são encontradas na realidade através de um modelo de programação inteira de pesquisa operacional, bem como sua função objetivo e suas restrições. Como será citado durante este artigo, o problema pertence à classe NP-Difícil. Assim, também será apresentada uma abordagem heurística para resolver instâncias onde há um grande volume de dados que será útil para auxiliar o modelo matemático a obter soluções de melhor qualidade.

Este trabalho está organizado da seguinte forma: A próxima seção diz respeito aos trabalhos relacionados a este que foram estudados. Na seção 3 é feita a formalização do problema, apresentando uma descrição detalhada e o modelo matemático proposto. As heurísticas propostas são detalhadas na seção 4. Na seção 5 é explicado como foram realizados os testes computacionais e os resultados obtidos. Por fim, na seção 6 é feita a conclusão do trabalho.

2. Revisão Bibliográfica

Foram estudados diversos trabalhos relacionados ao tema *ridesharing*. Segundo Furahata et al (2013), os primeiros sistemas de transporte compartilhado surgiram nos Estados Unidos na década de 70 tendo como grande motivação a crise do petróleo. Portanto, o tema já vem sendo tratado durante anos pela literatura e com diferentes abordagens. Serão citadas abaixo, as mais similares a esse trabalho.

Nos trabalhos de Herbawi & Weber (2011a, 2011b), a abordagem proposta foi elaborada para atender um único passageiro, com pontos de origem e destino já determinados e com sua rota final podendo incluir vários veículos com rotas pré-estabelecidas. Nessa abordagem os autores propuseram um algoritmo evolucionário e multiobjetivo.

Outra possível abordagem foi proposta em por Agatz et al (2011) onde são utilizados métodos avançados de otimização para melhorar o desempenho de sistemas de transporte compartilhado. Os autores simplificam o problema, determinando que cada motorista pode atender no máximo um passageiro. Desta forma, o problema fica muito mais tratável, mas situações como um motorista passar exatamente no caminho de um passageiro e não atendê-lo por já ter atendido outro passageiro podem acontecer. Assim, a subutilização de veículos aparece como um inconveniente desta abordagem.

Kamar & Horvitz (2011) consideraram um problema onde todos os participantes tinham seu próprio carro. A abordagem foi resolvida como um problema de cobertura de conjuntos. A proposta foi de agrupar os participantes em conjuntos cujo tamanho máximo era a capacidade máxima dos veículos e escolher um participante de cada conjunto para ser o motorista para todos os outros.

Dentre todos trabalhos analisados, Herbawi & Weber (2012) tratam o problema da forma mais semelhante a que foi proposta. Neste trabalho, há vários participantes (motoristas e passageiros) com pontos de origem e destino já definidos e janelas de tempo tanto de partida quando de chegada. O problema tem uma função multiobjetivo, buscando otimizar: distância total percorrida pelos motoristas, tempo total de viagem dos participantes e número de passageiros atendidos. Este problema também é NP-Difícil e assim os autores propuseram um algoritmo genético para obter soluções para instâncias que possuem um grande volume de dados.

A principal diferença da nossa abordagem em relação a de Herbawi & Weber (2012) é que para um passageiro ser atendido, o motorista não necessariamente deve passar em seu ponto de origem e destino, mas sim em pelo menos em algum ponto que esteja a no mínimo uma distância pré-estabelecida destes. Outras duas diferenças é que em nossa abordagem não tratamos casos com janelas de tempo e também o de múltiplos destinos para os participantes.

Hà et al (2012) tratam o problema de *arc routing* onde vários clientes devem ser atendidos por um fornecedor de serviço. Assim tem-se um grafo com os clientes sendo representados pelos vértices e as ruas sendo as arestas. O fornecedor de serviço deve sair da sua origem, atender todos os clientes e retornar ao seu ponto inicial. Nessa abordagem, para atender um cliente, o fornecedor de serviço deve passar em alguma aresta que esteja a pelo menos uma distância pré-determinada de seu cliente. O objetivo do problema é encontrar o conjunto de ruas (arestas) que o fornecedor de serviço deve passar de modo que todos os clientes sejam atendidos.

3. Formalização do problema

No problema proposto, cada participante (passageiro e motorista) tem um ponto de origem específico. O destino final é comum para todos os participantes. Assim, o objetivo do modelo é alocar passageiros a motoristas de modo a maximizar o número de passageiros atendidos e minimizar a soma das distâncias das rotas dos motoristas. Para um motorista poder atender um passageiro, deve necessariamente passar em algum ponto suficientemente próximo de sua origem, ter espaço físico para mais um passageiro e que sua rota não fique com uma distância maior que a permitida.

Nesta seção, também serão definidos os dados de entrada e também um modelo matemático de programação inteira do problema.

3.1. Exemplo de aplicação

Para exemplificar o problema, considere o caso típico da cidade de Viçosa, Minas Gerais onde diariamente uma grande quantidade de pessoas se desloca até a Universidade Federal de Viçosa (UFV) no mesmo horário. Várias dessas pessoas se deslocam de carro e podem oferecer carona as demais que se deslocam a pé. Apesar de se mostrarem prestativos, os motoristas não desejam efetuar um grande desvio em sua rota, sendo necessário que os passageiros se desloquem até um ponto conveniente para os motoristas.

Desta forma, este trabalho considera um percentual máximo aceitável no desvio da rota do motorista, bem como um limite máximo de deslocamento por parte do passageiro até o seu ponto de atendimento. Assim, um passageiro só pode ser atendido se o motorista passar a uma distância suficientemente próxima dele.

3.2. Exemplo Gráfico

Na Figura 1, temos ilustrado um exemplo de pequena instância com 3 motoristas e 9 passageiros. A imagem ilustra as adjacências da universidade como parte do grafo que representa a cidade. Podemos notar o ponto em vermelho (vértice representando a universidade) como sendo o destino final de todos os participantes. Os pontos em verde simbolizam as origens dos 3 motoristas da instância. Já os pontos em cor marrom simbolizam as origens de cada passageiro.

O conceito de vértice suficientemente próximo é ilustrado na Figura 2. Em destaque temos um dos passageiros e um raio de 300 metros em relação ao seu ponto de origem. Todos os pontos que estão no interior do círculo (em vermelho) são considerados suficientemente próximos ao passageiro, ou seja, ele pode ser atendido em qualquer um destes pontos.



Figura 1: Exemplo de instância com 3 motoristas e 9 passageiros.



Figura 2: Vértices destacados são suficientemente próximos do passageiro 7.

Já na Figura 3, podemos notar a solução do exemplo acima. Com os motoristas iniciando suas rotas em seus respectivos pontos de origem, alguns atendendo determinados passageiros, e chegando ao destino final que é o ponto representando a universidade. Percebe-se que os passageiros 4 e 5 não estão sendo atendidos pois para incluí-los em suas rotas, qualquer motorista iria ter que fazer um desvio maior que as restrições permitem. Também podemos notar os passageiros 6, 7 e 9 sendo atendidos em pontos suficientemente próximos da rota do veículo 1 (azul). A mesma situação ocorre com os passageiros 2 e 3 em relação as rotas dos veículos 3 (marrom) e 2 (vermelha) respectivamente.

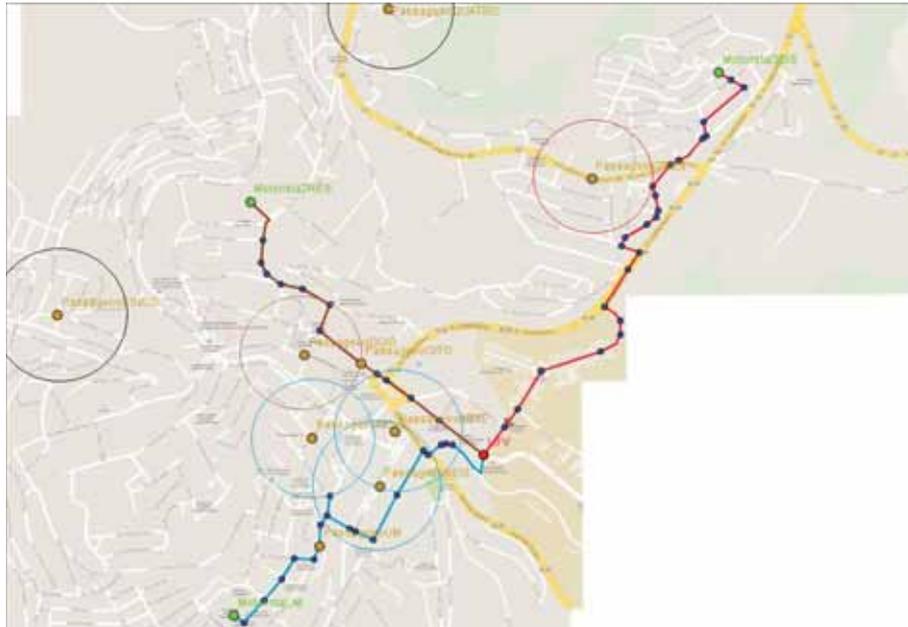


Figura 3: Solução: rotas dos motoristas e atendimento de passageiros.

3.3. Modelo Matemático

O modelo matemático espera os seguintes dados de entrada para qualquer instância do problema:

Dados:

- $G = (V, A)$ - grafo que define o mapa da cidade
- d_{ij} - distância direta entre os vértices i e j , $(i, j) \in A$
- K - conjunto de motoristas
- $O_k \in V$ - origem do motorista $k \in K$
- $D \in V$ - destino de todos os motoristas e passageiros
- $Dmax_k$ - distância máxima permitida ao motorista $k \in K$
- Q_k - capacidade de passageiros no veículo do motorista $k \in K$
- P - conjunto de passageiros
- L - “lucro” por atender um passageiro (valor grande, 10000)
- SP_i - conjunto de pontos suficientemente próximos do passageiro $i \in P$

Variáveis:

- y_i^k - variável binária: 1 se o veículo k atende o passageiro i , 0 caso contrário
- x_{ij}^k - variável inteira: quantas vezes o veículo k passa pela aresta (i, j)
- f_{ij}^k - variável inteira auxiliar, usada para evitar subciclos isolados
- p_i^k - variável binária: 1 se o veículo k passa pelo ponto i , 0 caso contrário

Formulação MILP proposta:

$$\max Z = \sum_{i \in P} \sum_{k \in K} Ly_i^k - \sum_{(i,j) \in A} \sum_{k \in K} d_{ij} x_{ij}^k \quad (1)$$

$$\sum_{j:(O_k,j) \in A} x_{O_k,j}^k - \sum_{j:(j,O_k) \in A} x_{j,O_k}^k = 1, \quad \forall k \in K \quad (2)$$

$$\sum_{j:(D,j) \in A} x_{D,j}^k - \sum_{j:(j,D) \in A} x_{j,D}^k = -1, \quad \forall k \in K \quad (3)$$

$$\sum_{j:(i,j) \in A} x_{ij}^k - \sum_{j:(j,i) \in A} x_{ji}^k = 0, \quad \forall i \in V \setminus \{O_k, D\}, k \in K \quad (4)$$

$$\sum_{(i,j) \in A} d_{ij} x_{ij}^k \leq Dmax_k, \quad \forall k \in K \quad (5)$$

$$\sum_{i \in P} y_i^k \leq Q_k, \quad \forall k \in K \quad (6)$$

$$\sum_{k \in K} y_i^k \leq 1, \quad \forall i \in P \quad (7)$$

$$y_i^k \leq \sum_{v \in SP_i} \sum_{j:(v,j) \in A} x_{vj}^k, \quad \forall i \in P, k \in K \quad (8)$$

$$p_i^k \leq \sum_{j:(i,j) \in A} x_{ij}^k, \quad \forall i \in V, k \in K \quad (9)$$

$$Mp_i^k \geq \sum_{j:(i,j) \in A} x_{ij}^k, \quad \forall i \in V, k \in K \quad (10)$$

$$\sum_{j:(i,j) \in A} f_{ij}^k \geq \sum_{j:(j,i) \in A} f_{ji}^k + 1 - M(1 - p_i^k), \quad \forall i \in V \setminus \{O_k, D\}, k \in K \quad (11)$$

$$\sum_{j:(O_k,j) \in A} f_{O_k,j}^k \geq \sum_{j:(j,O_k) \in A} f_{j,O_k}^k + 1, \quad \forall k \in K \quad (12)$$

$$f_{ij}^k \leq Mx_{ij}^k, \quad \forall (i,j) \in A, k \in K \quad (13)$$

$$x_{ij}^k \in \mathbb{N}, \quad \forall (i,j) \in A, k \in K \quad (14)$$

$$f_{ij}^k \in \mathbb{N}, \quad \forall (i,j) \in A, k \in K \quad (15)$$

$$y_i^k \in \{0, 1\}, \quad \forall i \in P, k \in K \quad (16)$$

$$p_i^k \in \{0, 1\}, \quad \forall i \in V, k \in K \quad (17)$$

A função objetivo (1) busca maximizar prioritariamente o número de passageiros atendidos e em seguida minimizar a distância total para atender esse máximo de passageiros. As restrições (2) e (3) garantem respectivamente que o motorista k sai de sua origem e chega ao seu destino, podendo passar outras vezes por esses pontos. A restrição (4) garante que nos demais nós o motorista sai a mesma quantidade de vezes que entra. As restrições (5) e (6) limitam respectivamente a distância máxima e a quantidade máxima de passageiros de cada motorista. A restrição (7) garante que no máximo um motorista atende um passageiro, e a restrição (8) que um motorista pode atender um passageiro somente se passar suficientemente próximo dele.

As restrições (9) e (10) garantem que um veículo visita um vértice se e somente se percorre alguma aresta incidente a ele. As restrições (11) e (12) impedem a formação de subciclos isolados na rota do motorista, ao mesmo tempo que permitem subciclos em sua rota, isto é, permite que o motorista repita vértices e arestas se necessário. Para isso utiliza valores de fluxo nas arestas, garantindo que, se um veículo passa por um vértice, o fluxo total que sai dele é maior que o fluxo total que entra; e (13) garante que só existe fluxo em aresta utilizada pelo veículo.

Por fim, as restrições (14)-(17) definem os tipos das variáveis de decisão.

3.4. Pré-processamento de dados

Devido à complexidade de resolução do modelo matemático para instâncias com grande volume de dados foi elaborado um algoritmo que realiza um pré-processamento sobre os mesmos, em duas etapas, contidos nas instâncias do problema com o objetivo de diminuir o espaço de busca do modelo e conseqüentemente atingir soluções melhores em um menor tempo de execução.

A primeira etapa do pré-processamento de dados é analisada a viabilidade em cada par (motorista, ponto). Também baseado na ideia de não ultrapassar a rota máxima permitida para cada motorista o algoritmo busca por pontos em que um motorista não pode chegar a partir da sua origem e depois sair desse ponto ir para o destino final para concluir sua rota sem que viole a restrição de distância máxima de rota.

Na segunda etapa, consiste em analisar a viabilidade de cada par (motorista, passageiro) em associá-los para a carona. Neste caso, se um motorista não puder passar em nenhum ponto suficientemente próximo da origem de um determinado passageiro e chegar no destino final sem que ultrapasse a restrição de distância máxima de sua rota esse par é indicado como inviável.

As informações de inviabilidade serão passadas para o modelo matemático que não irá utilizar as variáveis de decisão que determinam a associação entre os pares (motorista, passageiro) e (motorista, ponto).

4. Heurísticas

4.1. Heurística construtiva

Para se obter soluções iniciais para a Metaheurística, foi projetada e implementada uma heurística construtiva gulosa, visando fornecer soluções de boa qualidade. A heurística analisa cada par (motorista, passageiro) a cada iteração e encontra qual deles fornece o menor desvio de rota para que o motorista passe em algum ponto que atenda o passageiro. Chamamos esse desvio na rota de impacto. Então a heurística constrói uma solução gulosa para o problema, procurando a cada momento qual a combinação de motorista e passageiro que proporciona o menor impacto e (se não violar nenhuma restrição) a realiza. O algoritmo para quando não houver maneira de inserir mais nenhum passageiro. O Algoritmo 1 mostra pseudocódigo da heurística construtiva.

4.2. Heurística ILS proposta

Para aperfeiçoar as soluções encontradas na heurística construtiva, foi aplicada a Metaheurística ILS, apresentada por Lourenço et al (2003), ao problema abordado. O ILS trabalha durante toda a execução com apenas uma solução, buscando a aperfeiçoá-la a cada iteração. O primeiro passo é aplicar uma busca local na solução obtida pela heurística construtiva. A busca local visa encontrar um ótimo local baseado em alguma estrutura de vizinhança, ou seja, uma solução que tenha o melhor valor de função do que todos seus vizinhos. Após a busca local, o algoritmo perturba

Algoritmo 1 Heurística Construtiva

```
1: faça
2:   Troca ← FALSE
3:   menorImpacto ← ∞
4:   para(i ← 1 to |K|)
5:     para(j ← 1 to |P|)
6:       se(calculaImpacto(i,j) ≤ menorImpacto)
7:         m ← i
8:         p ← j
9:         menorImpacto ← calculaImpacto(i,j)
10:        Troca ← TRUE
11:   se(Troca)
12:     inserePassageiro(p,m)
13: enquanto(Troca)
```

(mantém a maioria das características e altera uma pequena parte) o ótimo local encontrado e aplica novamente a busca local nesta solução já perturbada. No caso de a busca local sobre a solução perturbada encontrar um melhor valor de função objetivo do que a melhor solução até o momento, essa passará a ser a solução corrente do método. Foi estipulado que o processo de perturbação e busca local se repete até que o algoritmo atinja 100 iterações sem encontrar uma solução melhor que a solução corrente. Os algoritmos 2 e 3 mostram o pseudocódigo da busca local e da heurística ILS proposta, respectivamente.

Algoritmo 2 Busca Local (BL)

```
1: Repita
2:   Troca ← FALSE
3:   melhorAvaliacao ← ∞
4:   para(i ← 1 to N)
5:     para(j ← 1 to N)
6:       se(avaliaTroca(i,j) ≤ melhorAvaliacao) então
7:         Troca ← TRUE
8:         P1 ← i
9:         P2 ← j
10:   se(Troca)
11:     trocaPassageiros(P1,P2)
12: Até (Troca = TRUE)
```

4.2.1. Estrutura de vizinhança

Foi considerada para a estrutura de vizinhança a operação de troca entre dois passageiros. Essa operação pode ocorrer em duas situações diferentes: ambos passageiros já são atendidos ou um passageiro é atendido e o outro não. No primeiro caso, a operação remove os dois passageiros das rotas dos motoristas que já os atendem e os insere na rota do motorista que atende ao outro passageiro. Já no segundo, o passageiro que está sendo atendido é retirado da rota do motorista que o atende e o outro passageiro é inserido na rota do mesmo motorista, passando a ser atendido. Antes de qualquer operação de troca, é verificada sua viabilidade, no caso a distância máxima da rota.

Algoritmo 3 Iterated Local Search (ILS)

```
1: Inicialize  $it \leftarrow 0$ ,  $ITMAX \leftarrow 100$ 
2: copiaSolucao(melhorSolucao,solucaoCorrente)
3: Repita
4: perturba(solucaoCorrente)
5: BuscaLocal(solucaoCorrente)
6: se(avaliaFuncaoObjetivo(solucaoCorrente) $\geq$ avaliaFuncaoObjetivo(melhorSolucao))
7:   melhorSolucao  $\leftarrow$  solucaoCorrente
8:    $it \leftarrow 0$ 
9: senão
10:   solucaoCorrente  $\leftarrow$  melhorSolucao
11:    $it++$ 
12: Até ( $it \leq ITMAX$ )
```

4.2.2. Perturbação

A perturbação tem o objetivo de realizar uma alteração na solução de modo que esta não seja vizinha da própria solução perturbada evitando que a solução corrente volte para ela pela busca local. Também não se pode realizar uma alteração muito grande na solução de modo que ela perca completamente suas características de ótimo local. Assim, foi buscado um certo equilíbrio na perturbação utilizada, utilizando também a operação de troca. Neste caso, são considerados 3 passageiros diferentes, escolhidos aleatoriamente e são feitas tentativas de troca (pois não há garantia de viabilidade) entre os pares de passageiros: (P1,P2), (P2,P3) e (P3,P1). Pode-se notar que, neste caso, diferentemente da busca local, uma troca não precisa fazer com que a qualidade da solução melhore para ser feita mas apenas apresentar viabilidade. Em 4 temos o pseudocódigo da perturbação.

Algoritmo 4 Perturbação ILS

```
1: Inicialize  $P1 \leftarrow rand(1, \dots, N)$ ,  $P2 \leftarrow rand(1, \dots, N)$ ,  $P3 \leftarrow rand(1, \dots, N)$ 
2: se(podeRealizarTroca( $P1, P2, P3$ )) então
3:   trocaPassageiros( $P1, P2, P3$ )
```

5. Testes e Resultados

Como não foi encontrado nenhum trabalho na literatura que abordasse um problema de ridesharing desta maneira, não haviam instâncias já existentes para se validar os algoritmos propostos. Assim, foi necessário gerar os dados para os testes. Para testar os algoritmos propostos, foi tomado como base o contexto da cidade de Viçosa-MG que tem aproximadamente 80.000 habitantes. Foi obtido o grafo que representa a rede de ruas e cruzamentos da cidade utilizado por Mendes et al (2014), e aplicado um algoritmo de caminho mínimo para gerar duas matrizes de entrada para o problema: menor distância e caminho mínimo entre todo par de vértices. Em seguida, foram geradas aleatoriamente 50 instâncias com variadas proporções entre o número de motoristas e passageiros.

O destino final de todos os participantes foi fixado como o campus da Universidade Federal de Viçosa (UFV). Deste modo, o tema, com algumas adaptações, por exemplo o estabelecimento de horários de chegada, poderia ser utilizado na implementação de um sistema de carona entre a comunidade acadêmica contribuindo para reduzir o trânsito excessivo que é uma adversidade nesta cidade.

Por fim, foi definido como a taxa de desvio máximo da rota o valor de 1.5. Ou seja, um determinado motorista pode ter um desvio máximo de 50 por cento em sua rota original (distância entre sua origem e o ponto da universidade) para atender eventuais passageiros. Para gerar os

vértices suficientemente próximos dos passageiros, foi utilizado um raio de 300 metros de sua origem. Para todas as instâncias, os veículos tinham capacidade de atender no máximo 4 passageiros.

Cinquenta instâncias separadas em cinco diferentes conjuntos foram testadas, com números de motoristas variando entre 5 e 50 e passageiros variando entre 5 e 250, dependendo do conjunto. Cada conjunto é nomeado como $MxkP$, onde k varia de 1 a 5 e é a quantidade de passageiros que existem para um motorista, em média, em suas instâncias. Todas instâncias foram criadas aleatoriamente, ou seja, para cada motorista e passageiro foi sorteado um vértice para ser sua origem.

Todos os modelos e métodos foram implementados e executados no software Microsoft Visual Studio. A linguagem adotada para a programação desses modelos foi C++. Para programar e resolver os modelos na linguagem C++ era necessário a integração com um solver (software especializado na solução de modelos matemáticos). Foi utilizado o software CPLEX versão 12.4, da IBM para resolver o modelo matemático, juntamente com a biblioteca Concert, fornecida pelo CPLEX, que permite a tradução dos modelos matemáticos e sua solução.

Todos os testes foram executados em um computador Intel Core i7, 3.07 GHz com 10 GB de memória (RAM).

A princípio foi fixado um tempo limite de execução de uma hora tanto para a execução da heurística, quanto do modelo matemático. Testes preliminares mostraram que a heurística ILS apresenta bons resultados enquanto o modelo matemático é capaz de encontrar soluções viáveis apenas para instâncias pequenas, mesmo com 1 hora de execução. Assim, foi testada uma abordagem mista, que combina a heurística e o modelo, a fim de obter soluções de melhor qualidade. Esta abordagem consiste em resolver o modelo considerando como solução inicial a solução encontrada pela heurística. Para isso foram utilizadas funções da biblioteca Concert que permitem instanciar valores iniciais para as variáveis de decisão do modelo. A solução encontrada pela heurística é um ótimo local viável, portanto pode ser inserida como solução inicial.

A Tabela 1 mostra o resultado das execuções da heurística ILS e do modelo resolvido pelo CPLEX para as 50 instâncias, usando a solução inicial da heurística. Na primeira coluna encontra-se o nome da instância com seus respectivos números de motoristas e passageiros. As segunda e terceira colunas representam o valor da função objetivo encontrado pela heurística e depois pelo modelo matemático, após a inserção da solução inicial encontrada pela heurística. A quarta coluna mostra a melhora sobre a solução heurística proporcionada pela execução do modelo matemático. Na quinta coluna temos o valor do *gap* de integralidade (diferença entre melhor solução inteira encontrada e o limite superior encontrado para o modelo relaxado) para a solução final encontrada. As sexta e sétima colunas mostram o número de passageiros atendidos na solução encontrada pela heurística e pelo modelo. Por fim, a oitava e nona colunas indicam de quanto foi e qual tipo de melhora que a execução do modelo matemático proporcionou sobre a solução heurística: em número de passageiros ou nas distâncias percorridas pelos motoristas.

Pode-se observar que em todas as instâncias houve alguma melhora na solução encontrada pela heurística e passada como inicial para o modelo. O que significa que atendeu-se mais passageiros ou a distância total percorrida pelos motoristas diminuiu. Foi obtido um valor de *gap* menor que 16% para 44 instâncias e menor que 1% para 11 instâncias das 50 testadas, o que indica uma boa qualidade das soluções obtidas pelo método.

Note que a solução ótima foi obtida em algumas instâncias e que a heurística encontra soluções quase ótimas em alguns casos (por exemplo, 10M-10P, 30M-30P, 35M-35P, 5M-10P). Para os dois conjuntos de instâncias com maiores proporções de passageiros para motoristas, a melhora do modelo sobre a solução encontrada pela heurística sempre foi em relação ao número de passageiros, sugerindo que esses casos se apresentam como mais difíceis para a heurística.

6. Conclusão

Neste projeto, foi elaborada uma nova abordagem de problema na área de transporte compartilhado. O contexto proposto neste trabalho visa uma aproximação do que acontece na realidade, onde para facilitar a rota do motorista que oferece carona, o passageiro que solicita a carona se

Tabela 1: Resultados da execução para as 50 instâncias

Instância	Função Objetivo		Melhora (%)		Passageiros Atendidos		Melhoria	
	Heurística	Modelo	Melhora (%)	Gap (%)	Heurística	Modelo	Pass	Dist (Km)
5M-5P	28.455	48.307	69,77	0,4	3	5	2	-
10M-10P	57.702	57.751	0,08	0,0	6	6	-	0,49
15M-15P	145.224	145.445	0,15	0,1	15	15	-	2,21
20M-20P	173.609	183.740	5,84	5,7	18	19	1	-
25M-25P	192.561	222.269	15,43	0,2	20	23	3	-
30M-30P	291.726	291.825	0,03	0,1	30	30	-	0,99
35M-35P	338.722	338.983	0,08	0,2	35	35	-	2,61
40M-40P	377.539	387.801	2,72	0,2	39	40	1	-
45M-45P	435.942	435.960	0,01	0,2	45	45	-	0,18
50M-50P	424.726	444.924	4,76	6,8	44	46	2	-
5M-10P	28.310	28.386	0,27	0,0	3	3	-	0,76
10M-20P	176.933	177.187	0,14	5,7	18	18	-	2,54
15M-30P	274.645	284.645	3,64	3,7	28	29	1	-
20M-40P	352.957	363.046	2,86	8,4	36	37	1	-
25M-50P	402.200	402.278	0,02	17,5	41	41	-	0,78
30M-60P	520.020	570.010	9,61	0,2	53	58	5	-
35M-70P	648.749	658.890	1,56	4,7	66	67	1	-
40M-80P	747.280	767.314	2,68	2,7	76	78	2	-
45M-90P	854.003	854.270	0,03	3,6	87	87	-	2,67
50M-100P	963.851	963.852	0,01	2,1	98	98	-	0,01
5M-15P	108.043	108.070	0,02	37,1	11	11	-	0,27
10M-30P	177.398	207.297	16,85	14,6	18	21	3	-
15M-45P	325.219	365.255	12,31	11,1	33	37	4	-
20M-60P	523.064	523.248	0,04	9,6	53	53	-	1,84
25M-75P	531.310	551.291	3,76	21,9	54	56	2	-
30M-90P	670.837	690.734	2,97	8,8	68	70	2	-
35M-105P	869.420	909.767	4,64	9,9	88	92	4	-
40M-120P	1.006.580	1.066.980	6,00	10,4	102	108	6	-
45M-135P	1.146.370	1.196.440	4,37	11,8	116	121	5	-
50M-150P	1.246.140	1.255.710	0,77	15,2	126	127	1	-
5M-20P	49.054	58.978	20,23	0,0	5	6	1	-
10M-40P	247.244	257.408	4,11	19,4	25	26	1	-
15M-60P	325.062	355.159	9,26	39,5	33	36	3	-
20M-80P	603.099	652.818	8,24	15,4	61	66	5	-
25M-100P	671.317	751.062	11,88	24,0	68	76	8	-
30M-120P	841.359	971.350	15,45	10,4	85	98	13	-
35M-140P	1.029.020	1.079.140	4,87	13,1	104	109	5	-
40M-160P	1.296.530	1.376.670	6,18	8,1	131	139	8	-
45M-180P	1.395.991	1.445.580	3,55	10,5	141	146	5	-
50M-200P	1.634.710	1.714.940	4,91	13,4	165	173	8	-
5M-25P	137.954	147.958	7,25	13,8	14	15	1	-
10M-50P	236.942	266.994	12,68	11,3	24	27	3	-
15M-75P	494.385	514.506	4,07	15,6	50	52	2	-
20M-100P	613.925	693.925	13,03	14,5	62	70	8	-
25M-125P	703.087	762.988	8,52	9,2	71	77	6	-
30M-150P	890.798	920.864	3,38	13,1	90	93	3	-
35M-175P	1.099.470	1.239.150	12,70	12,2	111	125	14	-
40M-200P	1.316.820	1.427.180	8,38	11,3	133	144	11	-
45M-225P	1.377.770	1.487.510	7,97	12,1	139	150	11	-
50M-250P	1.664.689	1.774.650	6,61	11,9	168	179	11	-

desloca a pé até um determinado ponto que é conveniente ao motorista, dentro de um limite estabelecido por ele.

O problema foi formalizado, definindo um modelo matemático de programação inteira contendo informações sobre qual seu objetivo e suas restrições. Devido à complexidade do problema, o modelo matemático se apresentou impraticável na busca por soluções em instâncias que apresentavam um número razoável de motoristas e passageiros como dados de entrada, sem que fosse fornecida uma solução inicial. Assim, foi proposta uma heurística ILS para encontrar soluções que sirvam como iniciais para o modelo, que poderia assim começar a busca pela melhora em sua qualidade.

Pelos resultados apresentados, com o modelo matemático melhorando todas as soluções da heurística e fornecendo bons valores de *gap*, podemos concluir que para a abordagem proposta a técnica mista, ou seja, combinar a heurística ILS proposta com o modelo matemático a fim de obter resultados de melhor qualidade se apresenta como a mais eficiente.

Para trabalhos futuros, temos como objetivo estudar uma abordagem mais geral, em que os participantes, além de apenas a origem, também tenham destinos diferentes. Assim, poderíamos encaixar esse problema em qualquer contexto da área e não apenas em casos particulares como o que foi abordado neste trabalho.

Agradecimentos: Agradecemos às instituições CAPES e FAPEMIG por proporcionarem os recursos necessários para a realização deste trabalho.

Referências Bibliográficas

Agatz, N. A., Erera, A. L., Savelsbergh, M. W. e Wang, X. (2011), Dynamic ride-sharing: A simulation study in metro atlanta, *Transportation Research Part B: Methodological*, vol. 45, no. 9, pp. 1450 – 1464.

Furuhata M., Dessouky M., Ordóñez F., Brunet M., Wang X. e Koenig S. (2013), Ridesharing: The state-of-art and future directions”, *Transportation Research Part B*, pp. 28-29.

Globo.com (2012), Venda de veículos bate novo recorde em 2011, segundo Fenabrave. Disponível em <http://g1.globo.com/carros/noticia/2012/01/venda-de-veiculos-bate-novo-recorde-em-2011-segundo-fenabrave.html> Acessado em: 13 de novembro de 2014.

Hà M., Bostel N., Langevin A., Rousseau L. (2014), Solving the Close-Enough Arc Routing Problem, *NETWORKS an International Journal*, pp. 107-118.

Herbawi, W. e Weber, M. (2011a), Ant colony vs. genetic multiobjective route planning in dynamic multi-hop ridesharing, *Tools with Artificial Intelligence (ICTAI)*, IEEE Conference, 2011.

Herbawi, W. e Weber, M. (2011b), Comparison of multiobjective evolutionary algorithms for solving the multiobjective route planning in dynamic multi-hop ridesharing, *Evolutionary Computation (CEC)*, IEEE Congress on, june 2011, pp. 2099-2106.

Herbawi, W. e Weber, M. (2012), The ridematching problem with time windows in dynamic ridesharing: A model and a genetic algorithm, *Evolutionary Computation (CEC)*, IEEE Congress on, june 2012, pp. 1 – 8.

Kamar, E. e Horvitz, E. (2009), Collaboration and shared plans in the open world: studies of ridesharing, *International Joint Conference on Artificial Intellingence (IJCAI)*, San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2009, pp 187 – 194.

Lourenço, H.R., Martin, O. e Stützle, T. (2003), Iterated Local Search, *Handbook of Metaheuristics*, Kluwer Academic Publishers, 57, pp. 321–353.

Mendes, N.F.M., Santos, A.G., Gonçalves, L.B. (2014), Métodos para o problema de posicionamento de unidades policiais, *XLVI SBPO - Simpósio Brasileiro de Pesquisa Operacional*, pp. 639-650.