

MELHORIAS EM UM MÉTODO HEURÍSTICO PARA A SOLUÇÃO DO PROBLEMA DE DESENHO DE REDE DE TRANSPORTE PÚBLICO URBANO

Lorena Mastrapa

lorenahmastrapa@gmail.com

José Eugenio Leal

jel@puc-rio.br

Maria Rita Pontes Assumpção

maria.assumpcao@unimep.br



Este trabalho trata do problema de Desenho de Redes de Transporte Público Urbano (PDRTPU). Este problema busca melhorias para minimizar tempo de viagem e de espera, custo de passagens entre outros. Apresenta-se uma revisão da literatura para este problema (PDRTPU), focando em um algoritmo heurístico baseado nos métodos de caminho mínimo. Baseado no Método de Aquino, foi proposto o algoritmo, obtendo-se com sua aplicação, uma solução que atende à demanda, minimizando o número de transbordos, com um número de linhas significativamente menor do que o encontrado na aplicação original.

Palavras-chave: Problema de Desenho de Redes de Ônibus Urbano, Métodos Heurísticos, Otimização de Redes de Transporte Público Urbano

MELHORIAS EM UM MÉTODO HEURÍSTICO PARA A SOLUÇÃO DO PROBLEMA DE DESENHO DE REDE DE TRANSPORTE PÚBLICO URBANO

Resumo

Este trabalho trata do problema de Desenho de Redes de Transporte Público Urbano (PDRTPU). Este problema busca melhorias para minimizar tempo de viagem e de espera, custo de passagens entre outros. Apresenta-se uma revisão da literatura para este problema (PDRTPU), focando em um algoritmo heurístico baseado nos métodos de caminho mínimo. Baseado no Método de Aquino, foi proposto o algoritmo, obtendo-se com sua aplicação, uma solução que atende à demanda, minimizando o número de transbordos, com um número de linhas significativamente menor do que o encontrado na aplicação original.

Palavras chave: Problema de Desenho de Redes de Ônibus Urbano, Métodos Heurísticos, Otimização de Redes de Transporte Público Urbano.

Abstract

This paper deals with the problem of Urban Public Transport Network Design (PDRTPU). This problem seeks improvements to minimize travel time and waiting times, ticket costs among others. We present a literature review for this problem (PDRTPU), focusing on a heuristic algorithm based on minimum path methods. The algorithm proposed was based on Aquino's Method and this application provided a solution that comply with the demand, minimizing the number of transfers, with a number of lines significantly smaller than the one found in the original application.

Keywords: *Transit Network Design Problem, Heuristic Method, Urban Public Transport Network Optimization.*

1. Introdução

O crescimento das cidades em termos de superfície e população mostra a relevância do problema de Desenho de Redes de Transporte Público Urbano (PDRTPU). Entre 1950 a 2050, enquanto a projeção de crescimento anual da população mundial em média é de 1,30%, a população urbana pode crescer em média 2,16% ao ano (Nações Unidas, 2014a). O Brasil está entre os quatro países com maior população urbana, depois de Índia, China e Japão (Nações Unidas, 2014b). Do ponto de vista da mobilidade urbana do Brasil, a ausência de rede metropolitana integrada de transportes que atenda à população dificulta o deslocamento de forma econômica, rápida, confortável e saudável.

O planejamento de rede de transporte público contribui na melhoria do atendimento da demanda, reduzindo o total de ônibus requerido com uso mais eficiente e econômico da frota, o congestionamento nas vias e a sobrecarga nos veículos nos horários de pico, com resultante aumento da segurança, conforto e velocidade.

O tempo total de viagem é uma das variáveis a considerar no planejamento de transporte urbano, podendo ser decomposto em tempo de acesso de às plataformas, das paradas, de espera, viagem e transbordos (Ortúzar e Willumsen, 2001; Raveau et al., 2011; Cepeda et al., 2006; Schmöcker et al., 2011). A otimização dos transbordos reduz a sobreposição das linhas dos percursos e a sobrecarga das vias pelo excesso de veículos. Por outro lado, contudo, um número elevado de transbordos reduz o nível de serviço pelo aumento de custo e tempo de viagem para o usuário (Barra, 2011).

Este artigo estuda o planejamento de rede interurbana com várias linhas de ônibus buscando minimizar o número de transbordos. Apresenta-se um método heurístico melhorado para busca de solução do Problema de Desenho de Rede de Transporte Público Urbano (PDRTPU).

O trabalho, além dessa introdução, apresenta três seções. Em seguida, é feita revisão da literatura referente aos métodos de solução para o PDRTPU. A terceira seção apresenta as modificações propostas no método de Aquino (1980) e compara os resultados obtidos. O artigo é concluído com recomendações para futuras pesquisas.

2. Métodos de solução do PDRTPU

As formulações matemáticas para o projeto de rede de transporte, nos casos onde existe um grande número de rotas, são geralmente intratáveis por abordagens exatas, embora úteis para pequenos casos (Ibarra-Rojas, 2015; Wan e Lo, 2003), de projeto de linha única (Guan et al., 2003) e em simplificações que reduzem o tamanho da rede de transporte (Lownes e Machemehl, 2010). As heurísticas e meta-heurísticas são implementadas, na maioria das vezes, para resolver problemas de grande porte de PDRTPU. A formulação deste problema, algumas vezes, utiliza modelo de dois níveis para considerar decisões de alocação no transporte público (Ibarra –Rojas, et al., 2015).

2. 1 Análise e seleção do método heurístico de construção de solução para o PDRTPU

Foram analisadas várias formulações usadas para a construção da solução PDRTPU para cidades médias ou grandes. Em geral, trata-se de problemas multiobjetivos, complexos, não convexos que, viabiliza a busca de solução por heurísticas. Aquino (1980), como outros autores, usa como o primeiro passo de seu método heurístico, a geração de um conjunto de rotas candidatas. Ao invés de usar critério seguido por Lampkin e Saalmans (1967) e Silman, et al. (1974) de criar um esqueleto inicial de rotas, Aquino gera conjunto de rotas candidatas que representem os caminhos mínimos mais longos. Essa estratégia é semelhante à proposta por Dubois, et al. (1979) e também sugerido por Baaj e Mahmassani (1994).

Estes métodos dedicam pelo menos uma fase para validação das restrições (variadas em função da formulação do problema). Outro ponto em comum é uma fase para geração de rede de percursos para deslocamento entre qualquer par de zonas da área em estudo. Lampkin e Saalmans (1967), Silman et al. (1974) utilizam também uma abordagem que assegure a conexão entre todos os pontos de origem e destino, definindo um esqueleto inicial de rotas. No entanto isto leva a um grande número de transbordos para parte significativa dos usuários.

As vantagens do método de Aquino (1980), por isso escolhido para melhorias, são: i) eliminação da necessidade de definição antecipada de nós terminais; ii) adoção de critérios bem definidos para seleção de rotas candidatas; e iii) definição e avaliação de parâmetros para medir a rentabilidade das rotas candidatas.

3. Proposta da estratégia para a resolução do PDRTPU

O método proposto por Aquino (1980) foi escolhido, além das vantagens descritas, por ser aplicável a uma rede de grande porte, analisando a rentabilidade das rotas por vários parâmetros e gerando uma rede aceitável tanto para o usuário quanto para a empresa operadora. O método constrói uma rede a partir da conexão das rotas que minimizam o número de transbordos. Esta estratégia foi implementada na linguagem Fortran, na sua versão inicial e em C++ na sua versão com melhorias.

3.1 Definição da estratégia de solução do PDRTPU

Pela complexidade do problema ele foi dividido em subproblemas. A estratégia de solução está dividida em fases que resolvem progressivamente cada um destes subproblemas até a obtenção de solução para o todo PDRTPU. Denominamos rota para designar uma linha componente da rede de ônibus.

Fase 1. Geração de um conjunto de rotas candidatas;

Fase 2. Geração de uma rede tal que seja possível o deslocamento por ônibus entre qualquer par origem-destino na área em estudo;

Fase 3. Avaliação dos parâmetros das rotas;

Fase 4. Definição de lista de rotas que minimizam o número de transbordos, inseridas na rede até zerar os transbordos.

```
INICIO PROGRAMA PRINCIPAL ORIGINAL
1. Inicializar os dados de entrada;
2. Gerar o conjunto G;
3. Gerar o conjunto G*;
4. SE existem rotas iguais em G* ENTÃO
    5. Eliminar as repetições;
    6. Atualizar conjunto G*;
7. FIM_SE
8. SE existem rotas inseridas em outras em G* ENTÃO
    9. Eliminar as rotas inseridas em outras;
    10. Atualizar conjunto G*;
11. FIM_SE
12. Gerar o conjunto  $\bar{G}$ ;
13. Gerar o conjunto  $F \leftarrow \bar{G} \cup G^*$ ;
14. Calcular os parâmetros das rotas de F;
15. Iniciar geração da rede;
    1. Escolher rota que apresente maior demanda potencial;
    2. Adicionar rota escolhida ao conjunto que conecta à rede;
    3. ENQUANTO a rede não está conectada
        4. Escolher rota que apresente maior demanda potencial;
        5. Adicionar rota escolhida ao conjunto que conecta à rede;
        6. Atualizar demandas potenciais das rotas restantes;
        7. Atualizar conjunto F;
    8. FIM_ENQUANTO
9. Atualizar F;
10. Calcular o número de transbordos;
11. ENQUANTO número de transbordos  $\neq 0$ 
    12. Escolher rota que reduza ao máximo o número de transbordos;
    13. Adicionar rota escolhida a lista de rotas que minimiza transbordos;
    14. Calcular número de transbordos resultantes;
    15. Atualizar conjunto F;
    16. Atualizar demandas potenciais das rotas restantes;
17. FIM_ENQUANTO
FIM PROGRAMA PRINCIPAL ORIGINAL
```

Figura 1: Pseudocódigo do programa principal do método de Aquino, 1980

No pseudocódigo, na Figura 3.1, é mostrado o método desenvolvido por Aquino (1980), com integração das fases referidas. Nele G significa o primeiro conjunto de rotas candidatas para a criação uma rede de rotas. As rotas são determinadas a partir do algoritmo de Floyd de caminhos mais curtos. G* é o segundo conjunto de rotas candidatas, rotas estas definidas como variantes das do conjunto G, considerando desvios dos caminhos mínimos. Os comprimentos destes caminhos serão menores ou iguais a $(1 + \mu)$ vezes o comprimento da rota original, sendo μ um parâmetro da estratégia de resolução a ser definido pelo planejador. O conjunto G* é criado com as rotas depuradas, qual seja, são eliminadas rotas contidas em outras e rotas com nós repetidos. O conjunto F é a união do conjunto G* e as rotas contidas

em \bar{G} ; que são aquelas contidas no conjunto G que não estão em G^* , ficando com os mais longos caminhos mínimos.

3. 2 Descrição do método e das adequações aplicadas

O método proposto neste artigo apresenta adequações, a seguir descritas, a fim de melhorar o desempenho do método de Aquino. Os resultados do método proposto são comparados ao do método de Aquino, por aplicação no mesmo problema. Os dados estão disponíveis em <https://drive.google.com/open?id=0B1JC2SPKtT9dQXFzSmhJNTlsZkk>.

O problema da aplicação é uma rede composta por 45 nós, entre eles: 18 são centroides e 27 são não centroides; e 51 arcos. Considerou-se uma matriz Origem-Destino simétrica. A demanda total na rede é de 963,4 unidades como 50 passageiros. A rede foi descrita com arcos bidirecionais com valores iguais de ida e volta para distâncias em quilômetros e os tempos de viagem em horas.

O programa original fazia uma representação anti-econômica da rede ao usar uma matriz que indicava para cada arco i,j o índice do arco (i,j) na lista sequencial lida como input no programa. É semelhante a uma matriz de incidência nó-a-nó, porém com o índice do arco no lugar de 0 ou 1.

3.2.1 Geração do conjunto de rotas candidatas

O objetivo da primeira fase, correspondente à linha 2 do pseudocódigo do programa descrito na Figura 3.1, é gerar um conjunto de rotas que minimizem o tempo de trajeto, o número de transbordos e sejam rentáveis. Para isto é preciso gerar rotas não muito curtas, ou seja, suficientemente longas como para atender a vários centróides da rede. Este conjunto de rotas é capaz de cobrir toda a rede, havendo um serviço direto entre qualquer par de zonas pelo caminho mínimo, atendendo a cada par origem-destino da demanda.

A geração do conjunto G é determinada no programa original pelo Algoritmo de caminhos mínimos de Floyd- Warshall (1962). Ele trabalha com duas matrizes. Uma guarda o valor do caminho mínimo entre i e j , neste caso, a matriz de tipo *float*, $tmin(i,j)$. A outra matriz guarda

para cada nó i de origem o primeiro nó sucessor no caminho de i para cada j , refere-se à matriz de tipo inteira $ic(i,j)$. Logo outra matriz de sucessores é gerada, agora contendo apenas os centroides presentes na rota. O algoritmo de Floyd tem o inconveniente de buscar a rota mínima entre todos os pares de nós da rede, enquanto apenas as rotas mínimas entre pares de centroides são de interesse.

Este algoritmo foi substituído pelo algoritmo de Dijkstra (1959). Esta adequação visa possibilitar a aplicação em redes de grande porte, dado Dijkstra tem melhor desempenho em redes com maior número de arestas e vértices, com tempo computacional de $O([E + V] \log V)$, sendo E , o número de arestas e V o número de vértices. Este tempo é menor que o do algoritmo de Floyd-Warshall igual a $O(V^3)$.

Na rotina original só é tomado o sucessor no caminho de i para j , o que não contraria o princípio da árvore de caminho mínimo. Nesta árvore um nó pode ter mais de um sucessor, mas desde uma origem, para um destino específico, só pode ter um sucessor. Tendo isto em mente, o Algoritmo Dijkstra foi adaptado para ser aplicado de cada destino para todas as origens.

Para obter a matriz de sucessores $icu(i,j)$, na aplicação do algoritmo de Dijkstra, é conveniente fazer a rota mínima dos destinos para cada origem, lembrando que é de interesse apenas os caminhos mínimos entre cada par de centroides. Para fazer a rota mínima do destino para cada origem trabalhou-se com a estrutura de lista de predecessores (*backward star*). Para criar esta estrutura toma-se a lista de todos os arcos ordenados pelos nós de destino. Faz-se uma rotina para criar os ponteiros de cada nó j que indicarão a primeira posição de arco tendo o nó j como sucessor. O trabalho será feito com a lista de ponteiros e depois com as listas de nós de origem de cada arco e de custos nos arcos. Para fazer a rota mínima entre todos os pares de centroides, cria-se, além de um conjunto de nós temporários, um conjunto de centroides temporários. A cada vez que um centroide é selecionado como o menor temporário e passa a permanente ele é eliminado da lista de centroides temporários. Quando ela estiver vazia, o procedimento termina. Essa é uma clara vantagem do algoritmo de Dijkstra frente ao de Floyd.

No entanto o conjunto G obtido inicialmente no método conduz geralmente ao uso de uma frota muito grande, com a possibilidade de criar rotas muito curtas, contrariando o objetivo de rentabilidade dessas rotas. É por isso que é adotada a seguinte estratégia. Para cada um dos caminhos pertencentes ao conjunto G , são investigados os caminhos vizinhos que possuam os mesmos nós terminais. Seus comprimentos serão menores ou iguais a $(1 + \mu)$ vezes o comprimento da rota original, sendo μ um parâmetro da estratégia de resolução a ser determinado pelo planejador.

Estas novas rotas são obtidas inserindo em cada rota $\lambda \in G$, nós centroides k ainda não contidos na rota e que a distância entre cada par de nós centroides da rota original via k atenda ao limite estabelecido. Cria-se um novo conjunto G^* , como referenciado na linha 3 do pseudocódigo do programa principal na Figura 3.1. Esse novo conjunto é depurado mediante a eliminação das rotas: repetidas, inseridas em outras e que retrocedem sobre elas mesmas.

Uma outra modificação foi introduzida com o intuito de criar rotas ainda mais longas e que ainda atendam a condição de caminho mínimo. Após a geração inicial das rotas de G^* , foi adicionada uma nova função que permitisse juntar aquelas linhas do conjunto G^* que tivessem o nó destino em comum com o nó de origem da outra linha, mas apenas esses nós em comum entre elas.

Para a eliminação das rotas inseridas em outras e as que retrocedem sobre elas mesmas, como referenciado na linha 8 do pseudocódigo do programa na Figura 3.1, foram declaradas variáveis tipo *strings*, ao invés de variáveis inteiras, para armazenar as rotas do conjunto G^* e sua forma inversa. Isso simplificou o procedimento de eliminação de rotas repetidas e daquelas inseridas dentro de outras. Rotinas simples permitem comparar *strings* sem ter que correr explicitamente cada elemento da lista.

Tendo as rotas resultantes do conjunto G^* , eliminam-se do conjunto G as rotas inseridas em G^* e o conjunto resultante define-se como \bar{G} . Gera-se com isso o conjunto final de rotas candidatas (F), como descrito nas linhas 12 e 13 do pseudocódigo da Figura 3.1. O conjunto F é analisado para eliminar repetições de rotas e de rotas inclusas em outras.

3.2.2 Geração de uma rede inicial conectada

Uma vez obtidas rotas candidatas, agrupadas no conjunto F, é inicializado o procedimento para formar uma rede conectada, como descrito na linha 15 da Figura 3.1. O objetivo principal é conectar a menor quantidade de rotas que consiga atender à demanda total ainda que com transbordos.

É feito o cálculo da demanda potencial (U_{λ}) de cada rota λ , pela somatória das demandas entre os pares de origem e destino i e j dentro da linha λ , dado pela fórmula (3.1);

$$U_{\lambda} = \sum_{ij \in \lambda} D_{ij} \quad (3.1)$$

Inicialmente é selecionada a rota que apresente maior demanda potencial, para formar parte do conjunto de rotas que definem a rede conectada. A cada inclusão de rota na rede atualizam-se as demandas potenciais das rotas restantes do conjunto de rotas candidatas.

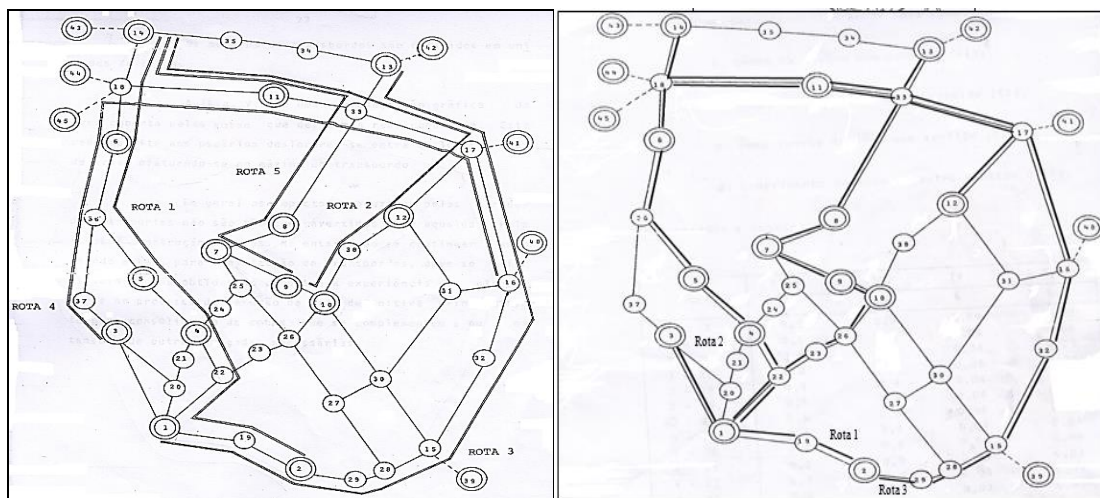
Outra modificação no método refere-se ao critério de escolha da próxima rota a ser incluída na rede conectada. No método original continuava-se incluindo as rotas em ordem de maior demanda. Na versão modificada mantém-se o objetivo do método que é criar uma rede conectada com os mais longos caminhos mínimos. No entanto, a partir da primeira rota com maior demanda, a próxima rota a inserir seria aquela que tenha pelo menos um nó centroide em comum com a rede já criada e o máximo de centroides ainda não conectados. Isso garante a criação de uma rede conectada de forma mais rápida, atendendo, com este critério de seleção, tanto aos interesses das empresas quanto dos clientes, pois seriam rotas de maior extensão e nível de rentabilidade. A rentabilidade da rota é dada pela relação *demanda total na rota/comprimento da rota*.

Outra melhoria no programa é da função que conecta a rede. No programa original havia muitas repetições no código, o que tornava a rotina muito longa e difícil de interpretar. Aquelas partes repetidas, foram identificadas e definidas como funções, sendo chamadas quando necessário e consequentemente fazendo esta rotina mais curta e inteligível. Como resultados das modificações obteve-se uma rede conectada com apenas três linhas ao invés

das 5 da solução original (Tabela 3.1), no problema original. A Figura 3.2 representa a trajetória das rotas que definem a rede conectada para o método original e para o modificado.

Tabela 1: Conjunto de rotas que definem a rede conectada. Segundo o método original e o modificado

No.	Conjunto de rotas que definem a rede conectada (algoritmo original)	Número de transbordos	Conjunto de rotas que definem a rede conectada (algoritmo modificado)	Número de transbordos
1	2-19-1-22-4-5-36-6-18-14	0	2-19-1-22-4-5-36-6-18-11-33-17-12	0
2	9-10-38-12-17-33-11-18-14	0	3-1-22-23-26-10-9-7-8-33-13	0
3	1-19-2-29-28-15-32-16-17-33-13	0	2-29-28-15-32-16-17-33-11-18-14	547
4	3-37-36-6-18-11-33-17-16	0		
5	9-7-8-33-11-18-14	419		



a)

b)

Figura 2: Trajetória das rotas que definem a rede conectada pelo método original (a) e pelo método modificado (b)

3.2.3 Definição de um conjunto de rotas em ordem de minimização de transbordos

Uma vez determinadas as rotas que definem a rede conectada, calcula-se o número total de transbordos seguindo a rotina descrita na linha 9 da Figura 3.1. Então seleciona-se um

conjunto de rotas dentre as candidatas que reduzam o número de transbordos (NT) na rede, até que $NT=0$.

Calcula-se o número de transbordos resultantes como:

$$NT = \sum D_{ij} - \sum U_{\lambda}^* ; ij \in N^2 \text{ e } \lambda^* \in \text{conjunto de rotas já inseridas na rede} \quad (3.2)$$

Onde:

N: Conjunto de nós da rede;

NT: Número de transbordos;

D_{ij} : Demanda entre ij ;

U_{λ}^* : Demanda potencial atendida dentro de cada linha λ ;

Aqui também foi feita uma melhoria importante. Segundo o método original seriam escolhidas como rotas candidatas, as rotas que reduzem ao máximo o número de transbordos, conforme as linhas 10 e 11 da Figura 3.1. Para atingir este objetivo foram declaradas, na versão aqui proposta, duas novas funções que permitem identificar todos os pares O-D atendidos apenas pelos transbordos. Isto tem o intuito de identificar que rotas ainda não inseridas na rede atendem ao maior número de pares O-D ainda não servidos diretamente. Ou seja, só serão incluídas as rotas que contribuem para reduzir gradativamente os transbordos até chegar a zero. Com isto as rotas candidatas são incluídas para redução de transbordos, como apresentado no lado direito da Tabela 3.2, até zerar esse número de transbordos, atendendo assim ao segundo objetivo do problema. Para este conjunto foram geradas um menor número de rotas (19) pelo método modificado, em comparação com as 24 do método original.

Tabela 2: Conjunto de rotas candidatas em ordem de minimização de transbordos. Segundo o método original e o modificado.

No.	Conjunto de rotas candidatas em ordem de minimização de transbordos (algoritmo original)	Número de transbordos	Conjunto de rotas candidatas em ordem de minimização de transbordos (algoritmo modificado)	Número de transbordos
1	5-4-24-25-7-8-33-17-16-32-15	337	6-36-5-4-22-23-26-10-9-7-8-33-13	485
2	3-37-36-6-18-11-33-13	286	2-19-1-22-23-26-10-9-7-8-33-17-12	445
3	9-10-38-12-17-33-13-34-35-14	247	10-9-7-8-33-11-18-14	336
4	4-24-25-7-8-33-14	214	3-20-21-4-5-36-6-18-11-33-17-12	285
5	2-19-1-3-37-36-6-18-14	183	2-19-1-22-4-24-25-7-9-10-30-15	239
6	5-36-6-18-11-33-13	157	5-36-6-18-11-33-17-16-32-15	218
7	1-22-23-26-10-38-12-17-16	132	2-29-28-15-32-16-17-33-13	214
8	3-20-21-4-24-25-7-9-10-38-12	109	2-19-1-3-37-36-6-18-14	152
9	2-29-28-15-30-31-12-38-10-9	86	4-24-25-7-8-33-17-16	145
10	2-29-28-15-32-16-17-33-11-18-14	66	13-33-11-18-14	129
11	3-1-19-2-29-28-15	50	4-5-36-6-18-14	73
12	2-19-1-22-4-24-25-7-8	39	12-17-16-32-15	71
13	5-36-6-18-11-33-17-12	25	9-10-30-31-16	57
14	1-22-23-26-10-38-12-17-33-11	21	3-37-36-6-18-11-33-17-16	39
15	4-24-25-7-8-33-11	15	1-22-23-26-27-30-31-12-17-33-13	37
16	6-18-11-33-17-12-38-10-9	11	1-19-2-29-28-15-32-16	21
17	5-4-24-25-7-9-10-30-15	9	3-1-19-2-29-28-15	5
18	1-22-23-26-10-9-7-8	7	12-17-33-11-18-14	4
19	9-10-38-12-17-16	5	2-29-28-15-32-16-17-33-8	0
20	3-20-21-4-24-25-7-8	3		
21	3-20-21-4-5	2		
22	6-18-11-33-8-7-9	0,8		
23	8-33-17-12	0,4		
24	5-36-6-18-11-33-17-16-32-15	0		

3.3.3 Avaliação das rotas que minimizam os transbordos na rede

Outra modificação foi avaliar o desempenho do conjunto final de rotas que minimizam o número de transbordos a partir do cálculo de diferentes indicadores. À diferença do método original que calcula estes indicadores para o conjunto de rotas candidatas F, esta mudança foi feita como o intuito de fornecer mais informações ao planejador sobre o conjunto de rotas que minimizam o número de transbordos, permitindo, com isto, avaliar o desempenho das rotas finais que respondem aos objetivos do problema.

Considerações finais

O trabalho apresenta o problema PDRTPU e métodos para sua solução, focando no método que utiliza o Algoritmo de Caminhos Mínimos de Floyd-Warshall (1962). Apresenta-se um algoritmo baseado no Algoritmo de Dijkstra (1959) com outras mudanças nos critérios de escolha das rotas para construção da Rede final, com o objetivo de diminuição de transbordos. As modificações feitas no método de Aquino (1980) apresenta melhoria de desempenho no

tempo computacional na busca de solução, além de solução melhor, comparados com os apresentados pelo método original. Obteve-se menor número de rotas, zerando o número de transbordos. Considera-se, no entanto, que zerar o número de transbordos que eleva o nível de serviço pode ser antieconômico. Nesse caso o planejador pode optar por criar uma rede que atenda a uma dada porcentagem de viagens com transbordos.

Espera-se com essa proposta uma contribuição no processo de planejamento de PDRTPU, com redução dos custos para as empresas prestadoras deste serviço e melhoria para os usuários. O método proposto pode ser adaptado a casos diferentes, tais como redes de transporte urbano de tamanho real e redes de transporte interurbano de passageiros. A análise e inserção de variáveis como tarifa, tempo de espera, programação da frota, entre outras, pode ampliar a aplicabilidade para o planejamento. A alocação da demanda sobre as linhas da rede é uma lacuna a ser preenchida assim como melhorar a heurística por meio de buscas locais a partir das soluções iniciais obtidas.

REFERÊNCIAS BIBLIOGRÁFICAS

AQUINO, W. J. S. (1980) *Uma abordagem do problema de definição de rede interurbana de rotas de ônibus*. Dissertação em opção ao título de mestre em Engenharia de Produção. Coordenação do Programa de Pós-Graduação de Engenharia. Universidade Federal do Rio de Janeiro, Rio de Janeiro, Outubro, pp.1-88.

BAAJ, M.; MAHMASSANI, H. (1995) *A hybrid route generation heuristic algorithm for the design of transit networks*. Transportation Research Part C Vol.3, pp. 31–50.

BARRA, R. A. (2011) *O impacto do transbordo em sistemas integrados de transporte coletivo por ônibus: uma análise quantitativa e qualitativa no município de Belo Horizonte*. Dissertação apresentada ao curso de Mestrado em Geotécnica e Transportes da Universidade Federal de Minas Gerais, como requisito parcial à obtenção do título de Mestre em Geotécnica e Transportes. Minas Gerais, pp.1-149.

CEPEDA, M.; COMINETTI, R.; FLORIAN, M. (2006) *A frequency-based assignment model for congested transit networks with strict capacity constraints: characterization and computation of equilibrio*. Transportation Research Part B 40, pp. 437–459.

DIJKSTRA, E.W. (1959). *A note on two problems in connexion with graphs*. Numerische Mathematik, Vol. 1, pp 269–271. December, 1959.

- DUBOIS, D.; BEL, G.; LLIBRE, M. (1979) *A set of methods in transportation network synthesis and analysis*. Journal of Operational Research Society, 39[9], pp.797-808.
- FLOYD, R. W. (1962) *Algorithm 97: Shortest Path*. Communications of the ACM 5 (6): 345. June, 1962.
- GUAN, J., YANG, H., WIRASINGHE, S.C. (2003) *Simultaneous optimization of transit line configuration and passenger line assignment*. Transportation Research Part B Vol.40, pp. 885–902.
- IBARRA-ROJAS, O.; LÓPEZ-IRARRAGORRI, F.; RIOS-SOLIS, Y. (2015) *Multiperiod synchronization bus timetabling*. Transportation Research Part B Vol. 77, Abril, 2015, pp. 38–75.
- LAMPKIN, W.; SAALMANS, P. D. (1967) *The design of routes, service frequency and schedules for a municipal bus undertaking: a case study*. Operational Research Quarterly, Vol. 18[4], Dec., 1967, pp.375-397.
- LOWNES, N., MACHEMEHL, R. (2010) *Exact and heuristic methods for public transit circulator design*. Transportation Research Part B Vol.44, pp.309–318.
- ORTÚZAR, J., WILLUMSEN, L. (2001) *Modelling Transport*. John Wiley and Sons Ltd., Chichester, England.
- RAVEAU, S.; MUÑOZ, J.; DE GRANGE, L. (2011) *A topological route choice model for metro*. Transportation Research Part A, Vol. 45, pp.138–147.
- SCHMÖCKER, J.; FONZONE, A.; SHIMAMOTO, H.; KURAUCHI, F.; BELL, M. (2011) *Frequency-based transit assignment considering seat capacities*. Transportation Research Part B, Vol.45, pp. 392–408.
- SILMAN, L. A.; BARZILY, Z.; PASSY, U. Planning the route system for urban buses. Computers & Operations Research 1(2), 1974, pp.201–211.
- UNITED NATIONS 2014a, *World Urbanization Prospects: The 2014 Revision Population Database*. United Nations. Available: <http://esa.un.org/unup/>. Consultado em: 20, Fevereiro de 2017.
- UNITED NATIONS 2014b, *World Urbanization Prospects: The 2014 Revision*, United Nations. New York, NY, USA.
- WAN, Q.; LO, H. (2003) *A mixed integer formulation for multiple-route transit network design*. Journal of Mathematical Modelling and Algorithms, Vol.2, pp. 299–308.