ELISANGELA MARTINS DE SÁ

DESIGN OF HUB-AND-SPOKE NETWORKS APPLIED TO PUBLIC TRANSPORTATION SYSTEMS

Belo Horizonte
24 de fevereiro de 2015

ELISANGELA MARTINS DE SÁ
Orientador: Ricardo Saraiva de Camargo

DESIGN OF HUB-AND-SPOKE NETWORKS APPLIED TO PUBLIC TRANSPORTATION SYSTEMS

Tese apresentada ao Programa de Pós-Graduação em Engenharia de Produção da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Doutor em Engenharia de Produção.

Belo Horizonte
24 de fevereiro de 2015

ELISANGELA MARTINS DE SÁ
Advisor: Ricardo Saraiva de Camargo

DESIGN OF HUB-AND-SPOKE NETWORKS APPLIED TO PUBLIC TRANSPORTATION SYSTEMS

Thesis presented to the Graduate Program in Industrial Engineering of the Universidade Federal de Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Industrial Engineering.

Belo Horizonte
February 24, 2015

UNIVERSIDADE FEDERAL DE MINAS GERAIS

FOLHA DE APROVAÇÃO

Design of hub-and-spoke networks applied to public transportation systems

ELISANGELA MARTINS DE SÁ

Tese defendida e aprovada pela banca examinadora constituída por:

Ph. D. Ricardo Saraiva de Camargo – Orientador
Universidade Federal de Minas Gerais

Ph. D. Geraldo Robson Mateus
Universidade Federal de Minas Gerais

Ph. D. Marcelo Franco Porto
Universidade Federal de Minas Gerais

Ph. D. Leandro Callegari Coelho
Laval University

Ph. D. João Fernando Machry Sarubbi
Centro Federal de Educação Tecnológica

Ph. D. Luciana Pereira de Assis
Universidade Federal do Vales do Jequitinhonha e Mucuri

Belo Horizonte, 24 de fevereiro de 2015

This thesis is dedicated to my dear advisor Ricardo S. de Camargo and to our research partners Jean-François Cordeau, Ivan Contreras and Gilberto de Miranda for making all this possible.

Acknowledgments

I would like to express my special appreciation and thanks to my advisor Professor Dr. Ricardo Saraiva de Camargo, you have been my great mentor. I would like to thank you for encouraging my research and believe in my potential. Your advice on both research as well as in my career has been invaluable.

I would also like to thank Professor Dr. Gilberto de Miranda for all supporting throughout my graduate studies.

I would especially like to thank to Professor Dr. Gilbert Laporte, Professor Dr. Jean-Françoos Cordeau and Professor Dr. Ivan Contreras from Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) for your wonderful supervision through my inter-university exchange doctorate in Montreal - Canada.

I would like to thank my office friends of Industrial Engineering department and of CIRRELT. In special to my CIRRELT friend Leandro C. Coelho for your valuable helps, comments and suggestions, thanks to you.

I also want to thank to CAPES Foundation for their financial support granted through doctoral fellowship and to CNPQ Foundation for their financial support granted to my doctoral exchange.

Finally, I thank my family for encouraging me throughout this experience and God for letting me through all the difficulties.

Abstract

The growth of metropolitan areas steadily pushes governments to restructure and expand their public transport networks in order to improve urban mobility and lower traffic problems. In particular, reduce traffic congestion, energy consumption, air pollution, and vehicle accidents. Recently a new set of resources, based on the ideas of hub-and-spoke networks, has been cleverly incorporated into the design of public transportation systems. In hub-and-spoke systems, commodities from different origins are sent to intermediate facilities, known as hubs, which are responsible for the aggregation and distribution of the flows to multiple destinations. The use of hubs allows the connection of a large number of origin/destination (O/D) nodes with a small number of arcs. In this way, it is possible reduce the infrastructure and operational cost, besides enabling economies of scale to be applied to the transportation cost (or travel time) between hubs. In this work, different hub-and-spoke network design problems focused on public transportation system are proposed. Furthermore, mathematical programming formulations are presented to model the proposed problems while exact and heuristic algorithms are proposed to tackle them. Computational results obtained on benchmark instances confirm the efficiency of the proposed algorithms.

Resumo

O crescimento das grande áreas metropolitanas tem exigido dos governantes uma reestruturação e expansão de sua rede de transporte público com a finalidade de melhorar a mobilidade urbana e reduzir problemas no tráfico, tais como congestionamento, consumo de energia, poluição do ar e acidentes de veículos. Recentemente um novo conjunto de recursos, baseado na ideia de redes eixo-raio, tem sido inteligentemente incorporado ao projeto de sistemas de transporte público. Sistemas eixo-raio são frequentemente utilizados no desenho de redes de grande porte tais como aquelas encontradas no transporte de passageiros e cargas, serviços postais, telecomunicações, e sistemas de trânsito rápido. Nestas redes, fluxos de diferentes origens são enviadas a facilidade intermediárias, conhecidas como concentradores, que são responsáveis pela agregação e distribuição dos fluxos para múltiplos destinos. Isto permite a conexão entre um grande número de pares de nodos origem/destino (O/D) com um pequeno número de arcos, reduzindo os custos operacionais e de infraestrutura, além de possibilitar que economias de escalas sejam aplicadas no custo de transporte (ou tempo de viagem) entre concentradores. Neste trabalho, diferentes problemas de desenho de redes eixo-raio aplicado a sistema de transporte público são propostos. Para modelar os problemas propostos, formulações de programação matemática são apresentadas, enquanto algoritmos exatos e heurísticos são propostos para resolver os problemas. Resultados computacionais obtidos em instâncias padrão da literatura confirmam a eficiência dos algoritmos propostos.

Resumo estendido

Introdução

O crescimento das grandes áreas metropolitanas tem exigido dos governantes uma reestruturação e expansão de sua rede de transporte público com a finalidade de melhorar a mobilidade urbana e reduzir problemas no tráfego, tais como congestionamento, consumo de energia, poluição do ar e acidentes. Ao mesmo tempo, os usuários tem constantemente pressionado por melhores níveis de serviço e sistemas custo-eficientes (Gendreau et al., 1995; Bruno et al., 1998). Estas questões dão origem a um problema complexo que requer uma considerável quantia de recursos financeiros e um esforço significativo para gerenciá-la. Recentemente um novo conjunto de recursos, baseado na ideia de redes eixo-raio (do inglês, *hub-and-spoke networks*), tem sido inteligentemente incorporado ao projeto de sistemas de transporte público (Nickel et al., 2001; Gelareh and Nickel, 2011; Martins de Sá et al., 2013a).

Sistemas eixo-raio são frequentemente utilizados no desenho de redes de grande porte tais como aquelas encontradas no transporte de passageiros e de cargas, serviços postais, telecomunicações, e sistemas de trânsito rápido. Nessas redes, fluxos de diferentes origens são enviados a facilidades intermediárias, conhecidas como concentradores, que são responsáveis pela agregação e distribuição dos fluxos para múltiplos destinos. Isso permite a conexão entre um grande número de pares de nós origem/destino (O/D) com um pequeno número de arcos, reduzindo os custos operacionais e de infraestrutura (O'Kelly and Miller, 1994). Outra importante vantagem de redes eixo-raio é a possibilidade de conexão dos concentradores por vias altamente eficientes, possibilitando que economias de escalas sejam aplicadas no custo de transporte (ou tempo de viagem) entre concentradores.

Originalmente (O'Kelly, 1986, 1987), supunha-se que redes eixo-raio possuíam: uma conexão direta entre todos os pares de concentradores; nenhuma conexão direta entre qualquer par de nós não concentradores; e um caminho com no máximo dois concentradores para rotear o fluxo entre qualquer par de origem e destino. Além disso, duas diferentes estratégias para alocação de nós não concentradores a concentradores eram permitidas. Os nós não concentradores poderiam interagir com apenas um único

concentrador resultando em variantes com alocação simples; ou eles poderiam ser conectados a mais de um concentrador resultando em variantes com alocação múltipla.

Recentemente, suposições mais flexíveis foram propostas (Nickel et al., 2001; Labbé et al., 2004; Campbell et al., 2005a,b; Contreras et al., 2009; Alumur et al., 2009; Calık et al., 2009; Contreras et al., 2010) com a finalidade de ampliar a aplicabilidade de sistemas eixo-raio a determinadas áreas. Ao descartar a imposição de que todos os pares de concentradores estão conectados, e adaptar o desenho da rede a características da aplicação sendo abordada, diferentes problemas podem ser vistos como um caso especial de redes eixo-raio: (i) localização de concentradores em árvore (Contreras et al., 2009, 2010; Martins de Sá et al., 2013b), (ii) desenho de redes anel-estrela (Labbé et al., 2004) e (iii) redes incompletas de concentradores (Campbell et al., 2005a,b; Alumur et al., 2009), dentre outros. Uma revisão exaustiva das variantes de redes eixo-raio pode ser encontrada em Campbell et al. (2002), Alumur and Kara (2008), e Farahani et al. (2013).

Neste trabalho, diferentes problemas envolvendo o desenho de redes eixo-raio aplicadas a sistemas de transporte público são propostos. Estes problemas consideram o desenho de uma rede de concentradores ao selecionar um conjunto de nós para localizar os concentradores, ativar um conjunto de conexões, e rotear o fluxo através da rede enquanto otimizam uma função objetivo baseada em custos ou serviços. Um exemplo concreto de uma aplicação desse tipo de problema a sistemas de transporte público é a modificação de uma rede de transporte público já estabelecida. Planejadores da rede geralmente enfrentam problemas para expandirem uma rede existente em uma região metropolitana de forma a reduzir o tempo de viagem ou o custo total do sistema. Uma alternativa para resolver este problema é a instalação de redes de trânsito rápido, tais como, metrô, trem ou corredores exclusivos para ônibus. Concentradores correspondem a estações centrais, tais como, estações de ônibus ou de metrô, onde mudanças de modais de transporte geralmente estão disponíveis. Os nós não concentradores representam distritos urbanos, ponto de ônibus ou de táxi.

Os problemas propostos são modelados através de formulações de programação matemática adaptadas a partir de formulações tradicionais da área de localização de concentradores. Para resolver este problemas, algoritmos exatos e heurísticos são propostos. Resultados de experimentos computacionais obtidos utilizando instâncias padrão da literatura comprovam a eficiência dos algoritmos propostos ao comparar com o aplicativo comercial CPLEX.

Esta tese é basicamente constituída por três artigos científicos e está organizada da seguinte forma. Os Capítulos 2–4 apresentam cada um destes artigos, enquanto o Capítulo 5 apresenta uma conclusão geral deste trabalho, bem como, possíveis trabalhos futuros. Um resumo de cada um destes capítulos é apresentado nas próximas seções.

The hub line location problem

O Capítulo 2 apresenta o artigo intitulado *"The hub line location problem"*. Esse manuscrito introduz o problema de localização de concentradores em linha (HLLP) que consiste em localizar um conjunto de concentradores conectados por meio de um trajeto simples (ou linha). Potenciais aplicações surgem no desenho de sistemas de transporte público onde o custo de instalação da infraestrutura necessária domina consideravelmente os custos de roteamento e assim, a completa interconexão dos concentradores é irrealística. Considerando a otimização do tempo de serviço como o objetivo predominante neste tipo de aplicação, o problema considera a minimização do tempo total de viagem ponderado pela demanda entre os pares de nós de O/D, enquanto leva em conta o tempo gasto para acessar e deixar a rede de concentradores.

Este artigo tem três principais contribuições: *i)* introdução do HLLP, um novo problema na área de localização de concentradores, *ii)* formulação de programação inteira mista para o HLLP, e *iii)* o desenvolvimento de algoritmos de decomposição de Benders (Benders, 1962) baseados nesta formulação. A implementação básica do algoritmo é melhorada através da inclusão de vários recursos algorítmicos, tais como, estratégias de múltiplos cortes, um eficiente algoritmo para resolver o subproblema e obter cortes de otimalidade mais fortes, e uma variante Benders-branch-and-cut que requer a solução de um único problema mestre.

Dois conjuntos de experimentos usando instâncias padrão da literatura foram realizados. O primeiro conjunto de experimentos tem como objetivo analisar como o fator de economia de escala e o tempo de espera para acessar/deixar a rede de concentradores afetam as configurações da rede sendo projetada. De acordo com os experimentos quanto maior a economia de escala, bem como, quanto menor o tempo de espera para acessar a linha, maior é a utilização da redes de concentradores e menos fluxo é roteado via conexão direta. Além disso, a configuração da rede, concentradores e arcos instalados, é profundamente afetada por estes dois parâmetros, demonstrando a importância de se considerá-los durante o projeto da rede. O segundo conjunto de experimentos tem como objetivo avaliar a eficiência dos algoritmos propostos. Este conjunto de experimentos foi dividido em duas fases. A primeira fase tem como objetivo descobrir qual é a melhor variante do método de decomposição ao combinar diferentes estratégias para melhoria da performance do método. A segunda fase consiste em comparar a melhor variante encontrada com o aplicativo comercial CPLEX. Resultados experimentais obtidos em instâncias padrão da literatura com até 50 nós confirmam que a variante de Benders que apresenta o melhor desempenho é a versão que *i*) adiciona um corte para cada par origem e destino por rodada, ou seja, múltiplos cortes,; *ii*) adiciona cortes de Benders dentro da árvore de *branch-and-bound* a partir

de soluções fracionárias no nó raiz e de potenciais soluções incumbentes; e *iii*) filtra os cortes adicionando apenas cortes violados. Finalmente, resultados computacionais obtidos em instâncias com até 100 nós confirmam a eficiência desta variante do método de decomposição de Benders. Esta variante resolve as instâncias testadas consideravelmente mais rápido que o CPLEX, além de ser capaz de encontrar a solução ótima para a maioria das instâncias testadas.

Nesse artigo é apresentado um problema voltado ao projeto de sistemas de transporte público compostos por uma única linha, tais como, uma linha de metrô, ônibus e etc. Além disso, um eficiente algoritmo para resolver o problema é apresentado. Resultados experimentais confirmam a eficiência do algoritmo proposto ao resolver instâncias de grande porte para o problema. Apesar do potencial de aplicação deste problema, ele pode ser estendido a um problema um pouco mais complexo que consiste em localizar um conjunto de linhas de concentradores (apresentado no Capítulo 3) ao invés de se localizar uma única linha.

Exact and Heuristic Algorithms for the Design of Hub Networks with Multiple Lines

O Capítulo 3 apresenta o artigo intitulado *"Exact and Heuristic Algorithms for the Design of Hub Networks with Multiple Lines"*. Esse artigo introduz o problema de localização de q-linhas de concentradores (q-HLLP). Este problema é uma extensão do HLLP para o caso em que a rede de concentradores é composta por mais de uma linha. O q-HLLP consiste em localizar um conjunto de q linhas que minimize o tempo total de viagem entre os pares de origem e destino, enquanto satisfaz uma restrição orçamentária para instalação da rede, i.e. localizar os concentradores e instalar os arcos entre concentradores. Com o intuito de modelar o tempo total de viagem adequadamente, quando mais de uma linha é utilizada, um tempo de espera para fazer a transferência entre linhas também é levado em consideração.

Uma formulação de programação inteira mista que é usada em um algoritmo de decomposição de Benders é proposta. A versão de Benders usada é baseada na melhor variante de Benders encontrada para resolver o HLLP (vide seção anterior). Além disso, são desenvolvidas três heurísticas diferentes para fornecer soluções viáveis para instâncias grandes baseadas nas metaheurísticas: *i)* Método de Descida em Vizinhança Variável (*Variable Neighborhood Descent*, VND), *iii)* Procedimento de busca adaptativa gulosa e randômica (*greedy randomized adaptive search procedure*, GRASP) e, *ii)* uma busca em vizinhança de grande porte (*adaptive large neighborhood search*, ALNS).

Foram realizados um conjunto de experimentos considerando dois conjuntos de instâncias padrão da literatura com até 70 nós e até três linhas. Resultados dos exper-

imentos comparando a variante de Benders e o aplicativo comercial CPLEX mostram que ambos apresentam um comportamento bem similar quando aplicados na resolução de instâncias com 10 e 20 nós. Entretanto, a variante do método de decomposição de Benders se destaca ao resolver instâncias de 25 e 40 nós. Apesar deste algoritmo não ser capaz de encontrar a solução ótima para todas as instâncias testadas, ele fornece um limitante inferior para estas instâncias, além de uma solução viável para a maiorias delas. Por outro lado o CPLEX não é capaz de encontrar um limitante inferior e nem uma solução viável para nenhuma instância com 40 nós. Desta forma, a variante de Benders apresenta ser a melhor ferramenta para fornecer limitantes inferiores que podem ser utilizados para avaliar a qualidade de soluções viáveis para o problema.

Devido a dificuldade para resolver o problema de forma exata, as heurísticas propostas são aplicadas para encontrar boas soluções para instâncias com 10 a 70 nós. Inicialmente, estas heurísticas são aplicadas para resolver o HLLP, onde uma única linha deve ser instalada, a quantidade de concentradores a ser instalada é fixa e não existe uma limitação de capital para instalar a infraestrutura necessária. Resultados dos experimentos mostram que as heurísticas baseadas em GRASP e ALNS são capazes de encontrar a solução ótima para maioria das instâncias testadas apresentando um baixo *gap* de otimalidade quando a solução ótima não é encontrada. Resulstados dos experimentos aplicados na resolução de instâncias do q-HLLP, confirmam a eficiência das heurísticas baseadas em GRASP e ALNS que apresentam o menor desvio médio entre a solução encontrada pela heurística e a melhor solução encontrada durantes os experimentos.

Esse artigo apresenta uma extensão do HLLP que localiza múltiplas linhas de concentradores. Eficientes algoritmos para resolver o problema são propostos. Algoritmos exatos foram capazes de resolverem apenas instâncias pequenas para o problema. No entanto apesar da variante do método de decomposição de Benders não ser capaz de resolver instâncias grandes, ele fornece limitantes inferiores que são úteis para analisar soluções viáveis para o problema. Métods heurísticos são propostos para encontrar soluções para instâncias de grande porte. Resultados experimentais confirmam a eficiência destas heurítiscas ao encontrar boas soluções (ótimas ou próximas do ótimo) para instâncias grandes do problema.

Exact algorithms to solve the hub location problem under congestion

O Capítulo 4 apresenta o artigo intitulado *"Exact algorithms to solve the hub location problem under congestion'*. Este artigo aborda o desenho de uma rede incompleta de concentradores sob efeito de congestionamento (IHLPC). Este problema consiste

em desenhar uma rede eixo-raio em que os concentradores podem ser parcialmente interconectados, nós não concentradores devem ser alocados a um único concentrador e conexões diretas entre nós não concentradores não são permitidas. A rede é projetada visando minimizar o custo total que é gual a soma dos custos para transportar todas as demandas e instalar a infraestrutura necessária, i.e. localizar os concentradores e os arcos entre concentradores; além dos custos associados a congestionamentos na rede. Este problema tem um grande apelo no desenho de sistemas de transporte onde o custo de congestionamento tem um papel muito importante, tal como, em redes de transporte público.

Uma importante contribuição desse artigo é a abordagem do congestionamento associado a três diferentes situações frequentemente encontradas em redes de transporte público: entrada no sistema, embarque nos veículos de transporte e transferência entre estações (concentradores). Para modelar o custo de congestionamento, duas funções não-lineares convexas são utilizadas: função Kleinrock e função *power law*. Baseado nestas funções para o cálculo do custo de congestionamento e na formulação para a versão não congestionada do problema, proposta por Alumur et al. (2009), uma formulação não-linear inteira mista é apresentada para modelar o problema. Algoritmos exatos baseados no método de aproximação externa (OA) (*Outer approximation*) (Duran and Grossmann, 1986; Fletcher and Leyffer, 1994), decomposição de Benders generalizada (GBD) (*Generalized Benders decomposition*) Geoffrion (1972) e uma versão que hibridiza ambos os métodos (OA/GBD) são propostos para resolver o problema. Com intuito de melhorar a convergência dos métodos propostos vários mecanismos como adição de cortes dentro da árvore de *branch-and-bound*, bem como adição de corte GBD Pareto-ótimo são testados.

Dois conjuntos de experimentos foram realizados. O primeiro conjunto tem como objetivo analisar como a configuração da rede sendo projetada é afetada pelos principais parâmetros do problema: fator de congestionamento, custo de instalação dos arcos e fator de capacidade. De acordo com os resultados, ao ignorar o congestionamento associado a único serviço provido pelo sistema, as soluções ótimas resultam em redes eixo-raio que tendem a sobrecarregar este serviço. Enquanto, que ao considerar todos os tipos de congestionamento simultâneamente a rede ótima tende a balancear a utilização da capacidade associada a cada um destes serviços. Em geral, conforme o fator de custo de instalação dos arcos aumenta, arcos mais baratos e menos concentradores são instalados.

O segundo conjunto de experimentos tem como objetivos avaliar a performance dos algoritmos propostos. Estes experimentos foram realizados em duas fases. A primeira fase é baseada em testes computacionais usando instâncias pequenas, com 10 nós, e tem como objetivo encontrar as variantes dos métodos mais promissoras para

resolver o problema. Na segunda fase, as duas melhores variantes encontradas são comparadas com o aplicativo comercial CPLEX levando em consideração funções de custo de congestionamento quadráticas representadas por um *power law* para resolver instâncias com até 20 nodos de demanda. Em seguida a melhor variante é testada em instâncias com até 25 nós de demanda considerando dois tipos de função de custo de congestionamento: *power law* e Kleinrock.

Resultados experimentais mostram que as variantes que adicionam cortes dentro da árvore de *branch-and-bound* apresentam melhor performance que as variantes clássicas. Além disso, a adição de cortes GBD Pareto-ótimo melhoram a performance dos métodos baseados em GBD. Na primeira fase de testes as variantes que apresentam melhores performances são as variantes OA que adiciona cortes dentro da árvore de *branch-and-bound* e variante hibrida de OA/GBD que além de adicionar cortes na árvore de *branch-and-bound*, adiciona cortes GBD Pareto-ótimo. Ao comparar estas duas variantes com o aplicativo CPLEX, resultados experimentais mostram que os algoritmos OA e OA/GBD apresentam melhor desempenho que o aplicativo comercial CPLEX. Ao comparar as duas variantes de OA, percebe-se que o algoritmo híbrido se comporta melhor quando o congestionamento associado ao serviço de embarque ou ao serviço de transferência é ignorado. Estes serviços estão associados aos congestionamentos tratados pelo GBD no algoritmo. Enquanto, a variante baseada no OA apresenta melhor desempenho quando todos os congestionamentos são considerados. Uma vez que esta varianta apresenta um melhor desempenho em média, então ela é utilizada nos experimentos finais para resolver instâncias com até 25 nodes considerando funções de custo de congestionamento baseadas na *power law* e na Kleinrock. Resultados destes experimentos mostram que este algoritmo é capaz de resolver aproximadamente 90% das instância baseadas na formulação usando *power law* e 77% das instâncias baseadas na formulação usando Kleinrock. Ao comparar a formulação usando a função baseada *power law* e a formulação que utiliza a função de Kleinrock, esta última apresenta-se mais difícil de se resolver provavelmente devido a adaptação feita na *power law* que contabiliza os custos de congestionamento a partir de um determinado limiar da capacidade de serviço. Além disso, a função Kleinrock cresce exponencialmente conforme o fluxo se aproxima de sua capacidade.

Nesse trabalho é apresentado o problema de projetar um sistema eixo-raio levando em consideração efeitos de congestionamentos associado a diferentes serviços geralmente fornecidos por sistemas de transporte público. O efeito do congestionamento é abordado através da adição de um componente associado ao custo do congestionamento na função objetivo. Uma vez que o custo de congestionamento é modelado através de uma função não linear convexa, a adição desta nova parcela resulta em um problema não linear inteira mista. Eficientes algoritmos baseados em decomposição são propos-

tos para resolver o problema. Resultados de experimentos computacionais confirmam a performance destes algoritmos comparados com o aplicativo comercial CPLEX.

Considerações finais

Nesta tese foram apresentados vários problemas de desenho de redes eixo-raio aplicadas a projeto de sistemas de transporte público. Estes problemas abordam o desenho de uma rede eixo raio levando em consideração não apenas o custo para instalação de infraestrutura e custos (tempo) de transporte, mas também, os atrasos sofridos pelos usuários para acessar/deixar o sistema ou devido a congestionamentos na rede. Estes problemas são de considerável importância para a literatura de problemas de desenho de redes eixo-raio por apresentar um potencial de aplicação prática ao considerar diversas características de sistemas de transporte públicos reais.

Para cada problema proposto, uma formulação de programação matemática, bem como, métodos de resolução são propostos. Dentre os métodos propostos temos métodos exatos baseados em decomposição, método de decomposição de Benders, aproximação externa e decomposição de Benders generalizadas. Para melhorar a convergência destes métodos, algumas estratégias para este fim são implementadas. Experimentos computacionais comprovam o desempenho destes métodos comparados com o aplicativo comercial CPLEX. Dentre os mecanismos para melhoria da convergência destes métodos de decomposição, a adição de cortes (de Benders, OA ou GBD), gerados a partir de uma potencial solução incumbente, dentro da árvore de *branch-and-bound* se mostrou bastante eficiente para todos os métodos testados. A vantagem desta estratégia é que estes métodos convergem em uma única iteração sendo necessário, portanto, a resolução de um único problema mestre. Outro mecanismo eficaz para fortalecer os métodos de decomposição é a seleção dos cortes a serem adicionados no problema meste. Uma vez que os subproblemas responsáveis por gerar os cortes, em geral, são degenerados, então diferentes cortes podem ser gerados. Investir na seleção dos cortes a serem gerados dentre os cortes possíveis, buscando adicionar ao problema mestre cortes mais fortes resultam em melhores convergências. No caso do Benders aplicado na resolução do HLLP, um algoritmo para seleção de cortes é o proposto. Os algoritmos baseados no GBD resolvem um subproblema linear adicional para gerar cortes não dominados, ou seja, Pareto-ótimos.

Apesar do potencial de aplicação dos problemas propostos, algumas extensões destes problemas podem ser exploradas em trabalhos futuros. Dentre as possíveis pesquisas futuras associadas ao problema de localização de linhas de concentradores, pode-se citar a modelagem do comportamento dos usuários ao escolher a rota de deslocamento, ao invés de assumir que o usuário escolhe sempre a menor rota dentre todas as rotas

disponíveis. Uma pesquisa futura associada ao problema com congestionamento é o estudo da modelagem da rede de concentradores através de redes de filas generalizadas. Nesta caso ao invés de assumir que a cada serviço fornecido por um concentrador pode ser modelado usando um sistema de filas M/M/1, a modelagem é feita usando um sistema de fila cuja distribuição do tempo de atendimento e intervalos entre chegadas são genéricas. A partir desta modelagem os custos de congestionamento são computados.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The growth of metropolitan areas steadily pushes governments to restructure and expand their public transport networks in order to improve urban mobility and lower traffic problems. In particular, to reduce traffic congestion, energy consumption, air pollution, and vehicle accidents. At the same time, users constantly pressure for better service levels, while taxpayers request for more cost-efficient systems (Gendreau et al., 1995; Bruno et al., 1998). These give rise to a complex problem which requires considerable amounts of financial resources and a significant effort to manage it. Recently a new set of resources, based on the ideas of hub-and-spoke networks, has been cleverly incorporated into the design of public transportation systems (Nickel et al., 2001; Gelareh and Nickel, 2011; Martins de Sá et al., 2013a).

Hub-and-spoke architectures are often used in the design of large-scale networks such as those found in passenger and freight transportation, postal services, telecommunications, and rapid transit systems. In these networks, commodities from different origins are sent to intermediate facilities, known as hubs, which are responsible for the aggregation and distribution of the flows to multiple destinations. The use of hubs allows the connection of a large number of origin/destination (O/D) nodes with a small number of arcs, reducing the infrastructure and operational cost (O'Kelly and Miller, 1994). Another important advantage of hub-and-spoke networks is that hub facilities can be connected with highly efficient pathways, enabling economies of scale to be applied to the transportation cost (or travel time) between hubs.

Originally (O'Kelly, 1986, 1987), hub-and-spoke networks were assumed to have: an inter-hub connection between every hub pair; no direct link between any two non-hub nodes; and a path with one or at most two hubs for routing demand flows between all origin and destination pair. Further, two different schemes for allocating the non-hub nodes to hubs were allowed: the non-hub nodes could interact with a single hub only, i.e., be single allocated to a hub, or they could be connected to more than one hub or

be multiple allocated.

Recently, more flexible assumptions were proposed (Nickel et al., 2001; Labbé et al., 2004; Campbell et al., 2005a,b; Contreras et al., 2009; Alumur et al., 2009; Calık et al., 2009; Contreras et al., 2010) in order to broaden the applicability of hub-and-spoke networks to other areas. By disregarding the restriction that every pair of hubs has to be directly connected and by adapting the design of the network to the characteristics of the application being addressed, different problems can now be seen as special cases of hub-and-spoke networks: *(i)* tree-shaped facilities location (Contreras et al., 2009, 2010; Martins de Sá et al., 2013b), *(ii)* ring-star network designs (Labbé et al., 2004), *(ii)* lines (Martins de Sá et al., 2013a), *(v)* incomplete hub networks (Campbell et al., 2005a,b; Alumur et al., 2009). For exhaustive surveys on the variants of hub-and-spoke networks please refer to Campbell et al. (2002), Alumur and Kara (2008), and Farahani et al. (2013).

In this thesis, different hub-and-spoke network design problems applied to public transportation system are proposed. These problems consider the design of hub networks by selecting a set of nodes to locate hubs, activating a set of links, and routing commodities through the network while optimizing a cost-based (or service-based) objective function. A concrete example of an application of this kind of problem applied to public transportation system is the modification of already established public transportation networks. Network planners usually face the problem of expanding an existing network in a metropolitan region so as to reduce the users' travel times or costs by locating a rapid transit line, such as a subway, tram or light rail line, or an express bus lane. Hub facilities correspond to central stations such as subway or bus stations, where a change of mode of transportation is usually available. Non-hub nodes represent urban districts, bus stops or taxi stations. Furthermore, mathematical programming formulations are presented to model the proposed problems while exact and heuristic algorithms are proposed to tackle them.

The following section presents how this thesis is organized.

1.1 Thesis organization

This thesis is a collection of three articles, presented on Chapter 2-4, addressing hub-and-spoke network design to project public transportation systems and it is organized as following.

Chapter 2 presents the article entitled *"The hub line location problem"*. This article introduces the hub line location problem (HLLP) in which the location of a set of hub facilities connected by means of a path (or line) is considered. Potential applica-

tions arise in the design of public transportation, where network design costs greatly dominate routing costs and thus, full interconnection of hub facilities is unrealistic. Assuming that service time is the predominant objective in these applications, the problem considers the minimization of the total weighted travel time between origin/destination nodes while taking into account the time spent to access and exit the hub line. The main contributions of this chapter are threefold: *(i)* the introduction of the HLLP, a new hub location problem, *(ii)* a mixed-integer programming formulation for the HLLP, and *(iii)* the development of a Benders decomposition algorithm based on this formulation to obtain optimal HLLP solutions. The basic implementation of the algorithm is enhanced through the inclusion of several algorithmic features such as a multi-cut strategy, an efficient algorithm to solve the subproblem and to obtain stronger optimality cuts, and a Benders-branch-and-cut scheme that requires the solution of only one master problem. Computational results obtained on benchmark instances with up to 100 nodes confirm the efficiency of the proposed algorithm, which is considerably faster and able to solve larger instances than a general purpose solver.

Chapter 3 presents the article entitled *"Exact and Heuristic Algorithms for the Design of Hub Networks with Multiple Lines"*. This paper generalizes the HLLP to the case in which the hub network is composed of more than one line by introducing the *q-line hub location problem* (q-HLLP). This problem consists of locating a set of q lines that minimizes the total travel time between O/D pairs, while satisfying a budget constraint on the total setup cost of the network associated with the location of hub nodes and hub arcs. In order to properly model the total travel time when using more than one hub line, the waiting time associated with transference between lines needs to be taken into account. To tackle the problem, exact and heuristic algorithms for designing hub line networks with multiple lines are proposed. In particular, a mixed-integer programming (MIP) formulation for the q-HLLP which is used in a Benders decomposition algorithm to obtain optimal solutions for small instances and to provide bounds for larger instances is presented. We also develop three different metaheuristics to provide feasible solutions to large instances: *i)* a variable neighborhood descent (VND), *ii)* a greedy randomized adaptive search procedure (GRASP) and, *iii)* an adaptive large neighborhood search (ALNS). Numerical results on two sets of benchmark instances with up to 70 nodes and three lines confirm the efficiency of the proposed solution algorithms.

Chapter 4 presents the article entitled *"Exact algorithms to solve the hub location problem under congestion"*. In this paper, the incomplete hub location problem under congestion (IHLPC) is addressed. This problem consists in designing a hub-and-spoke network in which the hub-level network can be partially interconnected and a non-hub node must be allocated to a single hub. The network is designed aiming to minimize

the total cost which is composed of the sum of (*i*) the total transportation costs which consider the economies of scale achieved by routing flows between hubs; (*ii*) the total infrastructure costs for locating hubs and hub arcs; and (*iii*) the total cost regarding network congestions. This problem has a great appeal in designing transportation system where congestion cost plays an important role, such as public transportation networks. An important contribution of this article is to consider congestion in three different services provided by hub-and-spoke system: entrance, boarding and transferring services. A mixed integer nonlinear formulation and exacts algorithms based on Outer Approximation (Duran and Grossmann, 1986; Fletcher and Leyffer, 1994) and Generalized Benders decomposition (Geoffrion, 1972) are proposed. Experiments on benchmark instances with up to 25 nodes confirm the efficiency of the proposed algorithms.

Finally, Chapter 5 presents a general conclusion of this thesis and possible future research.

Chapter 2

The Hub Line Location Problem

Chapter information

This chapter presents the article accepted for publication in Transportation Science: Martins de Sá, E., Contreras, I., Cordeau, J.-F., de Camargo, R.S., de Miranda,G., Forthcoming. The hub line location problem. Transportation Science. Forthcoming.

Abstract

This paper presents the hub line location problem in which the location of a set of hub facilities connected by means of a path (or line) is considered. Potential applications arise in the design of public transportation and rapid transit systems, where network design costs greatly dominate routing costs and thus, full interconnection of hub facilities is unrealistic. Given that service time is the predominant objective in these applications, the problem considers the minimization of the total weighted travel time between origin/destination nodes while taking into account the time spent to access and exit the hub line. An exact algorithm based on a Benders decomposition of a strong path-based formulation is proposed. The standard decomposition method is enhanced through the incorporation of several features such as a multi-cut strategy, an efficient algorithm to solve the subproblem and to obtain stronger optimality cuts, and a Benders-branch-and-cut scheme that requires the solution of only one master problem. Computational results obtained on benchmark instances with up to 100 nodes confirm the efficiency of the proposed algorithm, which is considerably faster and able to solve larger instances than a general purpose solver.

Keywords: Hub location, line networks, Benders decomposition method.

2.1 Introduction

Hub-and-spoke networks enable the routing of flows between many origin/destination pairs in a more efficient way than by directly connecting each O/D pair. In these systems, flows from different origins are sent to intermediate facilities, known as hubs, which are responsible for their aggregation and distribution. The main advantage of hub-and-spoke networks is their improved efficiency due to economies of scale by the bundling of flows at hubs. Moreover, network design costs can be considerably reduced since hub-and-spoke network topologies have fewer connections than fully connected networks (O'Kelly and Miller, 1994).

Hub location problems (HLPs) concern the design of hub-and-spoke networks by locating a set of hub facilities and selecting a set of links to route flows between O/D pairs. Objectives which are commonly considered include the minimization of the sum of set-up and/or transportation costs (minsum); the minimization of the maximum transportation cost or travel time (minimax); and coverage related objectives that may consider distance, time, cost or other attributes relevant to the application (see, for instance, Campbell, 1994). In addition, classical HLPs assume that hubs are fully interconnected and that direct connections between non-hub nodes are not allowed, i.e., O/D paths must contain at least one hub node. Although these classical variants of HLPs have attracted most of the attention in the literature since the seminal work of O'Kelly (1986), the study of real-world transportation/distribution systems giving rise to particular network topologies have recently become an important area of research. For recent surveys on hub location problems, please refer to Alumur and Kara (2008), Campbell and O'Kelly (2012), and Farahani et al. (2013).

Flexibility issues in hub network topologies were initially addressed by O'Kelly and Miller (1994) who proposed different protocols to classify hub networks according to: *(i)* whether non-hub nodes are singly allocated (non-hub nodes are assigned to exactly one hub) or multiply allocated (non-hub nodes may be assigned to more than one hub); *(ii)* whether hubs are fully or partially interconnected; and *(iii)* whether direct connections are allowed or not between pairs of non-hub nodes. The implications of the relaxation or imposition of these common assumptions has been addressed in some works. Nickel et al. (2001) address the design of public transportation systems by presenting HLPs in which the hubs are not fully interconnected and direct connections between pairs of non-hub nodes are allowed. Campbell et al. (2005a,b) and Contreras and Fernández (2013) study extensions of HLPs, referred to as hub arc location problems, in which additional network design decisions are incorporated to select a set of hub arcs whose end nodes are hubs which are not necessarily fully interconnected. Alumur et al. (2009) also relax the assumptions of full interconnection of hub nodes and do not assume any

topological structure.

Some authors have addressed the design of specific topological hub networks. Lee et al. (1993), Contreras et al. (2009) and Martins de Sá et al. (2013b), consider tree-star topologies in which the hub-level network is a tree and each non-hub node is assigned to exactly one hub. Lee et al. (1993) and Contreras et al. (2013b) study ring-star topologies, in which hubs are connected by means of a ring and each non-hub node is assigned to exactly one hub. Star-star topologies in which all hubs are connected to a central hub while non-hub nodes are singly allocated, resulting in a star configuration in both levels, are considered in Labbé and Yaman (2008) and Yaman (2008). Hub network topologies with more than two layers have also been introduced. Yaman (2009) studies the problem of designing a three-level hub network, where the top level consists of a complete network connecting the central hubs, and the second and third layers are unions of star networks connecting the remaining hubs to central hubs and the non-hub nodes to hubs or central hubs, respectively.

In this paper we introduce the *Hub Line Location Problem* (HLLP) which consists of locating a set of p hubs and of connecting them by means of a path (or line) using a set of $p-1$ hub arcs. We assume that each O/D node can be assigned to more than one hub node, i.e. a multiple allocation. Contrary to most p-hub median type problems considering a cost-based objective, the HLLP incorporates a service-based objective that focuses on the minimization of the total travel time between O/D pairs. Flows must be routed through the hub network via either a path containing a set of hub arcs (a segment of the hub line), or with a direct connection between origins and destinations, depending on whichever route provides the smallest travel time. Because of the use of a high-speed mode of transportation at the hub arcs, it is assumed that their associated travel speed is faster than on the other links of the network. Time savings are thus perceived when traveling on the hub line. However, in order to properly model the total travel time when using the hub line, other times incurred during the traveling process need to be taken into account. In particular, access and exit times may exist when using the hub line due to a change of mode of transportation or to waiting because of frequency or congestion related issues. The trade-off between the benefit of using a high-speed mode of transportation to efficiently travel on the network and the added time for interacting with it makes the design of this class of hub networks particularly challenging.

According to the classification scheme proposed by O'Kelly and Miller (1994), the HLLP fits in the H protocol, i.e., multiple-allocation hub location problems with incomplete hub-level network in which direct connections between non-hub nodes are allowed. The HLLP can also be seen as a q-hub arc location problem in which $q = p - 1$ hub arcs need to be located and connected by means of a path (see, Campbell et al., 2005a,

for details). Figure 2.1 illustrates a hub line network with six hub nodes and five hub arcs.

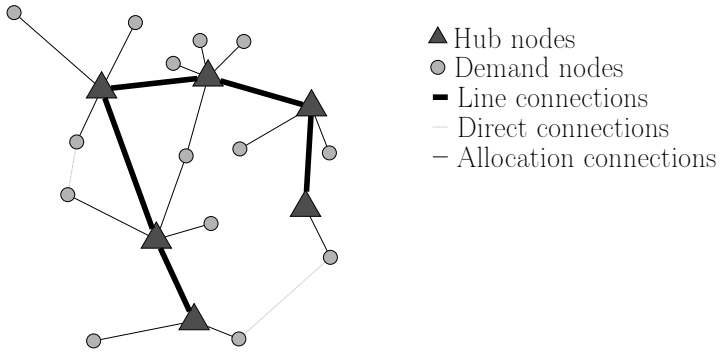


Figure 2.1: A hub line network with six hub nodes and five hub arcs.

Potential applications where the location of a hub line is required arise in public transportation planning, in particular in the design of rapid transit systems and highway networks. A concrete example of an application of the HLLP appears when modifying already established public transportation networks. Network planners usually consider the expansion of an existing (physical) network in a metropolitan region to improve its overall efficiency (users' travel times), by locating a rapid transit line, such as a subway, tram or light rail line, or an express bus lane with a fixed number of hub stations. This number is predetermined by considering budget restrictions or due to political reasons. Hub facilities correspond to different types of central stations such as subway, tram, bus and/or train stations, where a change of mode of transportation is usually possible. Non-hub nodes can be seen as bus stops, taxi stations or urban districts. The flows represent users traveling between O/D pairs who wish to arrive to their destinations in the smallest amount of time, that is, they want to minimize their travel (or commute) time. When the new rapid transit line is built, they will use the hub line if there is an improvement in their travel time or they will keep using the shortest route on the existing physical network. In order to more accurately represent the travel time of users, the time taken to access and exit the hub line needs to be incorporated. These times are usually observed when a change of transportation mode occurs. For instance, when arriving at a metro station by bus, the walking time between the bus terminal and the gates of the subway depend on the station and may be significant (i.e. 5 to 10 minutes). Moreover, when accessing the hub line the average time spent at the gate waiting for the next subway train to pass, which depends on the hub line frequency and congestion factors, could also be significant. These times might not compensate the reduction of travel time from using the hub line and thus, users may continue traveling as before. In the case of public transportation networks, the main goal is not the optimization of the transportation costs but rather the opti-

mization of the overall efficiency of the system, measured by the travel times between O/D pairs. Given that the number of users associated with different O/D pairs is usually substantially different, that minimization of the total weighted travel time of the system is an appropriate measure of its efficiency (see, for instance, Church and ReVelle, 1976; Current et al., 1987).

Additional applications for the location of hub lines appear in the design of road networks. In this case, network planners may be interested in studying the impact of extending current road networks in urban, suburban or rural regions when constructing a new path-shaped highway or express lane. Current travel times between O/D pairs may be improved by using the hub line, were time savings are observed due to higher speed limits associated with such lanes. Hub nodes can be seen as a set of interchanges between highways and other existing roads (see, Lari et al., 2008).

The design of path-shaped networks has been studied in the context of classical facility location and of network design problems. In the former case, Slater (1982) introduces facility location problems in which facilities are located on nodes that must constitute a path. Current et al. (1987) present the median shortest path problem which consists of designing a path-shaped facility such that the total weighted travel time to reach the line is minimized. Hakimi et al. (1993) analyze the complexity of several variants of the path-shaped facility location problem by taking into account four types of objective function: the minimization (maximization) of the maximum (minimum) distance, and the minimization (maximization) of the sum of the total distance to the line. Extensive facility location problems (see, Mesa and Brian Boffey, 1996) are generalizations of these problems that consider the location of facilities too large to be represented as a single point comparing its scale with its interaction environment. The interested reader is referred to Hakimi et al. (1993) and Labbé et al. (1998) for details on potential applications and modeling assumptions on path-shaped facilities.

In a context of rapid transit network design, Dufourd et al. (1996) study the problem of designing a line in a rapid transit network with the objective of maximizing the coverage of demand points. A demand point is covered by a line if it is within a given distance from a station on the line. Bruno et al. (1998) propose a bi-criterion model for designing a line for a rapid transit system to be integrated into a multi-modal topology composed by pedestrian and private networks. The objective is to minimize the total weighted travel costs and the construction costs, where the weights are related to an origin-destination demand matrix. Laporte et al. (2002) introduce the problem of maximizing the coverage of a continuous demand by installing stations on pre-defined alignments.

To the best of our knowledge, the design of hub line networks has not been previously addressed in the literature. Gelareh and Nickel (2011) study a related hub

location problem arising in the design of transportation networks, in particular in urban transport and liner shipping networks. In this problem, the full interconnection assumption is relaxed but no specific topology is required, other than connectivity, and direct connections between non-hub nodes are allowed. Their model rather focus on the minimization of the total set-up and transportation cost of the network and does not consider any access or exit time for interacting with the hub-level network. A Benders decomposition algorithm is used to obtain optimal solutions.

The main contributions of this paper are threefold: *i)* the introduction of the HLLP, a new hub location problem that considers the design of a hub line network topology, *ii)* a mixed-integer programming formulation for the HLLP, and *iii)* the development of a Benders decomposition algorithm based on this formulation to obtain optimal HLLP solutions. The basic implementation of the algorithm is enhanced through the inclusion of several algorithmic features such as a multi-cut strategy, an efficient algorithm to solve the subproblem and to obtain stronger optimality cuts, and a Benders-branch-and-cut scheme that requires the solution of only one master problem. In order to evaluate the efficiency and limitations of our algorithm, extensive computational experiments were performed on benchmark instances with up to 100 nodes.

The remainder of the paper is organized as follows. Section 2.2 provides a formal definition of the problem and introduces the MIP formulation. The standard Benders reformulation, the Benders decomposition algorithm, and several features that improve its convergence are described in Section 2.3. The results of extensive computational experiments are reported in Section 2.4. Conclusions follow in Section 2.5.

2.2 Definition and Formulation of the Problem

Let $G = (N, A)$ be a complete directed graph, where N and A are the sets of nodes and arcs, respectively. The demand to be routed from origin $i \in N$ to destination $j \in N$ is denoted by $w_{ij} > 0$, $(i \neq j)$. The travel time of arc $(i, j) \in A$, represented by $t_{ij} > 0$, is defined as the shortest time required to travel from i to j using one or more modes of transportation, other than the one associated with the hub line, on the original (physical) network. Without lost of generality, t_{ij} also incorporates any average transfer time required when changing modes of transportation from i to j. Note that this definition of travel times leads to t_{ij} values that satisfy the triangle inequality property. When a hub arc is located between hub nodes $i, j \in N$, the travel time between i and j is computed as $\alpha_{ij} t_{ij}$, where α_{ij} is a reduction factor that models the use of a faster transport technology to connect i and j. Depending on the considered application, this percent reduction time α_{ij} may be link dependent, as the shortest

path between node pairs may use different combinations of modes of transportation. The access and exit times to enter the first hub k and last hub node m of each O/D path are given by $\tilde{t}^a_k \geq 0$ and $\tilde{t}^e_m \geq 0$, respectively. The access time $\tilde{t}^a_k$ incorporates both the time required to change the mode of transportation between an access arc and a hub arc at node k and the average waiting time to access the first hub arc on the O/D path. The exit time $\tilde{t}^e_m$ only includes the time required to change the mode of transportation at last hub node m, as the waiting time associated with the access arc (if any) is already considered in t_{mj}.

The HLLP consists of locating p hub facilities connected by means of a line using a set of $p-1$ hub arcs, while allocating every non-hub node to at least one hub in such a way that the weighted sum of the total travel time is minimized. It is assumed that the demand from origin $i \in N$ to destination $j \in N$ (i.e. passengers) will use the fastest possible route on the solution network. That is, w_{ij} will travel either directly from i to j using the shortest path on the physical network, resulting in a travel time of t_{ij}, or it will use a combination of access arcs and hub arcs, in which the total travel time is equal to the sum of:

(i) the travel time from origin i to its closest hub k on the line,

(ii) the time to access the hub line at hub k,

(iii) the travel time between hubs k and m connected through a set of hub arcs,

(iv) the exit time to leave the hub line at hub m,

(v) the travel time from the closest hub m on the line to destination j.

Note that because of the triangle inequality property of travel times t_{ij}, a solution network of the HLLP will route demands w_{ij} either with a direct connection between i and j or with a path containing at most two access arcs and at least two hub nodes and one hub arc. That is, travel times associated with direct connections will be always smaller than or equal to any O/D path containing one hub node and no hub arcs.

In what follows, we introduce an MIP formulation for the HLLP based on the so-called path-based formulations commonly used in hub location research to model incomplete hub networks (see, Contreras et al., 2009; Gelareh and Nickel, 2011; Contreras and Fernández, 2012, and references therein). We define binary location variables z_k, $k \in N$, equal to 1 if and only if a hub is located at node k. We introduce binary hub arc variables y_{km}, $(k,m) \in A$, $k < m$, equal to 1 if and only if a hub arc is located between hubs k and m, enabling flows to be routed in both directions. We also define four sets of routing variables to model various structures of O/D paths arising in the HLLP. In particular, we introduce continuous variables $a_{ijk} \geq 0$ and $b_{ijm} \geq 0$ equal to

the fraction of the demand w_{ij} that enters and exits the hub line through hubs $k \in N$ and $m \in N$, respectively, while continuous variables $x_{ijkm} \geq 0$ denote the percentage of the demand w_{ij} routed on hub arc $(k, m) \in A$. Finally, we define continuous variables e_{ij}, $i, j \in N$, equal to the fraction of flow w_{ij} sent directly from i to j. To simplify the presentation, we assume $i, j, k, m \in N$ and $i \neq j$ henceforth. The HLLP can then be formulated as:

$$\text{minimize} \sum_i \sum_j w_{ij} \left[\sum_k \left(t_{ik} + \tilde{t}^a_k\right) a_{ijk} + \sum_k \sum_{m:m\neq k} \alpha_{km} t_{km} x_{ijkm} + \sum_m \left(t_{mj} + \tilde{t}^e_m\right) b_{ijm} + t_{ij} e_{ij} \right] \tag{2.1}$$

$$\text{subject to} \quad \sum_k a_{ijk} + e_{ij} = 1 \qquad \forall i, j \tag{2.2}$$

$$\sum_m b_{ijm} + e_{ij} = 1 \qquad \forall i, j \tag{2.3}$$

$$a_{ijk} + \sum_{\substack{m \\ m\neq k}} x_{ijmk} = b_{ijk} + \sum_{\substack{m \\ m\neq k}} x_{ijkm} \qquad \forall i, j, k \tag{2.4}$$

$$a_{ijk} \leq z_k \qquad \forall i, j, k \tag{2.5}$$

$$b_{ijm} \leq z_m \qquad \forall i, j, m \tag{2.6}$$

$$x_{ijkm} + x_{ijmk} \leq y_{km} \qquad \forall i, j, k, m : k < m \tag{2.7}$$

$$\sum_k z_k = p \tag{2.8}$$

$$\sum_k \sum_{m:m>k} y_{km} = p - 1 \tag{2.9}$$

$$\sum_{m:m>k} y_{km} + \sum_{m:m<k} y_{mk} \leq 2 z_k \qquad \forall k \tag{2.10}$$

$$\sum_{k\in S} \sum_{m\in S:m>k} y_{km} \leq \sum_{k\in S\setminus\{s\}} z_k \qquad \forall S \subseteq N, s \in S \tag{2.11}$$

$$y_{km}, z_k \in \{0, 1\} \qquad \forall k, m : k < m \tag{2.12}$$

$$x_{ijkm}, e_{ij}, a_{ijk}, b_{ijm} \geq 0 \qquad \forall\ i, j, k, m : k \neq m. \tag{2.13}$$

The objective function (2.1) minimizes the total weighted travel time. Constraints (2.2)-(2.4) are flow conservation constraints which ensure that the demand w_{ij} leaves node i and arrives at node j, and properly account for flows whenever a hub k is used, respectively. Constraints (2.5) only allow the demand between nodes i and j to enter the hub line through hub k if this hub is installed. Likewise, constraints (2.6) guarantee that this demand can only leave the hub line via hub m if this hub is located. Constraints (2.7) assure that inter-hub traffic can only flow through installed inter-hub connections. Constraints (2.8) and (2.9) state the exact number of hubs and inter-hub

links to be installed.

The hub line design is enforced by constraints (2.10) and (2.11). Constraints (2.10) only permit each hub to be linked to at most two other hubs, while constraints (2.11) are the well-known subtour elimination constraints (SECs) which assure that the hub line is always connected. Since pairs of demand nodes can be directly connected in this problem, constraints (2.2)-(2.10) are not sufficient to warrant the connectivity of the line. Therefore, SECs are required to do that. Further, as the number of SECs is exponential with respect to the number of nodes, they are not explicitly considered in the model, but are generated only as needed. Finally, constraints (2.12)-(2.13) show the domain of the decision variables.

Observing the constraint matrix of formulation (2.1)-(2.13), it is possible to see that it has a staircase structure, which makes the model amenable to solution by a Benders decomposition algorithm, the subject of the next section.

2.3 Benders Decomposition

The Benders decomposition method (Benders, 1962) is a partitioning procedure for solving mixed integer-linear and mixed integer non-linear programs with complicating variables. A given set of variables is considered to be complicating when, after temporarily setting these variables to some value, the obtained problem is easier to solve than the original one. For instance, variables z_k and y_{km} of the proposed formulation (2.1)-(2.13) can be considered as complicating variables since, after fixing them, it is possible to decompose the remaining problem into $n(n-1)$ shortest path problems, one for each OD pair.

The main idea of the method is to reformulate the problem by projecting out the set of non-complicating variables—which is assumed to have a larger cardinality than the set of complicating variables—with the objective of obtaining a problem with fewer variables but many more constraints. Since most of these constraints, known as Benders Cuts (BC), are not active in an optimal solution, all but a few of them are ignored in order to attain a relaxed version of the reformulation: the master problem (MP). By iteratively solving the MP, violated BCs are separated through the solution of a Benders subproblem (SP)—which is the original problem with the complicating variables temporarily held to values supplied by the MP—and added to the MP.

As the MP is a relaxation of the reformulation, it provides a lower bound (LB) for the original problem, while an upper bound (UB) is readily available through the conjunction of the solutions of the master problem and the subproblem. With the addition of BCs at each iteration, new tentative solutions are generated by the MP

and new cuts are produced until the convergence of the bounds is obtained, if an optimal solution exists.

In this section, efficient variants of the Benders decomposition method for the HLLP are presented.

2.3.1 Benders reformulation

By considering variables z and y as the complicating ones, it is possible to reformulate model (2.1)-(2.13) to obtain an equivalent problem. In order to achieve this, the non-complicating variables e, a, b and x are projected out through the parameterization of variables z and y, which results in the following primal linear SPs, one for each pair i, j:

$$\min\ w_{ij}\left[\sum_{k}\left(t_{ik}+\tilde{t}^{a}_{k}\right)a_{ijk}+\sum_{k}\sum_{m:m\neq k}\alpha_{km}t_{km}x_{ijkm}+\sum_{m}\left(t_{mj}+\tilde{t}^{e}_{m}\right)b_{ijm}+t_{ij}e_{ij}\right] \tag{2.14}$$

$$\text{s.t.:}\ \sum_{k}a_{ijk}+e_{ij}=1 \tag{2.15}$$

$$\sum_{m}b_{ijm}+e_{ij}=1 \tag{2.16}$$

$$a_{ijk}+\sum_{\substack{m\\ m\neq k}}x_{ijmk}-b_{ijk}-\sum_{\substack{m\\ m\neq k}}x_{ijkm}=0 \qquad \forall k \tag{2.17}$$

$$-a_{ijk}\geq -z^{h}_{k} \qquad \forall k \tag{2.18}$$

$$-b_{ijm}\geq -z^{h}_{m} \qquad \forall m \tag{2.19}$$

$$-x_{ijkm}-x_{ijmk}\geq -y^{h}_{km} \qquad \forall k,m:k<m \tag{2.20}$$

$$x_{ijkm},e_{ij},a_{ijk},b_{ijm}\geq 0 \qquad \forall\ k,m:k\neq m, \tag{2.21}$$

where z^h and y^h are fixed vectors for the complicating variables.

After associating the dual variables $\theta_{ij}\in\mathbb{R}$, $\Gamma_{ij}\in\mathbb{R}$, $\beta_{ijk}\in\mathbb{R}$, $u_{ijk}\geq 0$, $v_{ijm}\geq 0$ and $\delta_{ijkm}\geq 0$ to constraints (2.15)-(2.20), respectively, the dual linear Benders SPs for each i, j can be written as:

$$\max\ \theta_{ij}+\Gamma_{ij}-\sum_{k}z^{h}_{k}u_{ijk}-\sum_{m}z^{h}_{m}v_{ijm}-\sum_{k}\sum_{m:m>k}\delta_{ijkm}y^{h}_{km} \tag{2.22}$$

$$\text{s.t.:}\ \theta_{ij}+\Gamma_{ij}\leq w_{ij}t_{ij} \tag{2.23}$$

$$-\beta_{ijk}+\beta_{ijm}-\delta_{ijkm}\leq w_{ij}\alpha_{km}t_{km} \qquad \forall k,m:k<m \tag{2.24}$$

$$-\beta_{ijk}+\beta_{ijm}-\delta_{ijmk}\leq w_{ij}\alpha_{km}t_{km} \qquad \forall k,m:k>m \tag{2.25}$$

$$\theta_{ij} + \beta_{ijk} - u_{ijk} \leq w_{ij}\left(t_{ik} + \tilde{t}^a_k\right) \qquad \forall k \tag{2.26}$$

$$\Gamma_{ij} - \beta_{ijm} - v_{ijm} \leq w_{ij}\left(t_{mj} + \tilde{t}^e_m\right) \qquad \forall m \tag{2.27}$$

$$u_{ijk}, v_{ijk} \geq 0 \text{ and } \beta_{ijk} \in \mathbb{R} \qquad \forall\ k \tag{2.28}$$

$$\delta_{ijkm} \geq 0 \qquad \forall\ k, m : k < m \tag{2.29}$$

$$\Gamma_{ij}, \theta_{ij} \in \mathbb{R}. \tag{2.30}$$

It is worth noting that the dual SP (2.22)-(2.30) is always feasible and bounded.

Proposition 1. *For any fixed vectors z^h and y^h, the optimal value of the SP (2.22)-(2.30) is always bounded.*

Proof. Proof. Since direct connections are allowed to link pairs i, j of OD demand nodes for the HLLP, there is at least one path consisting of arc (i, j) to connect any pair i, j. Hence the primal SP (2.14)-(2.21) is always feasible and bounded. Then, by strong duality, the solution of the dual SP (2.22)-(2.30) is always feasible and bounded. □

Since the dual SP (2.22)-(2.30) is a linear program and from Proposition **??**, at least one extreme point of the polyhedron (2.23)-(2.30) corresponds to an optimal solution. As there is a finite number of such extreme points, it is possible to write the BC to be added to the MP as:

$$\eta \geq \sum_i \sum_j \left[\Gamma^g_{ij} + \theta^g_{ij} - \sum_k (u^g_{ijk} + v^g_{ijk}) z_k - \sum_k \sum_{\substack{m \\ k<m}} \delta^g_{ijkm} y_{km}\right] \quad \forall g \in \mathbb{G}, \tag{2.31}$$

where η is an under-estimator variable for the total weighted travel time and $\mathbb{G}$ is the set of extreme points of polyhedron (2.23)-(2.30).

The complicating variables, their respective constraints, the η variable, and the BC compose the Benders MP, which can be written as:

$$\min \quad \eta \tag{2.32}$$

$$\text{s.t.:} \quad \text{Constraints (2.8)} - \text{(2.11)} \tag{2.33}$$

$$\eta \geq \sum_i \sum_j [\Gamma^g_{ij} + \theta^g_{ij} - \sum_k (u^g_{ijk} + v^g_{ijk}) z_k - \sum_k \sum_{m:m>k} \delta^g_{ijkm} y_{km}] \quad \forall g \in \mathbb{G} \tag{2.34}$$

$$\eta \geq 0 \tag{2.35}$$

$$y_{km}, z_k \in \{0, 1\} \qquad \forall k, m : k < m. \tag{2.36}$$

The Benders MP (2.32)-(2.36) is equivalent to formulation (2.1)-(2.13) sharing therefore the same set of optimal solutions. On the one hand, the Benders MP has

fewer variables, but many more constraints (2.34) due to the possibly extremely large cardinality of set $\mathbb{G}$; on the other hand, in an optimal solution, just a few of these cuts will be active, allowing for a solution strategy based on relaxation, in which all but a few cuts are ignored, but added on the fly through an iterative procedure.

Furthermore, due to the presence of the SECs (2.11), the MP is solved by means of a branch-and-cut framework in which SECs are separated for every potential incumbent solution having a subtour, where by incumbent solution we mean the current best known feasible solution of the MP. The subtours are identified by detecting the connected components of the current line, and violated SECs are then added by considering all nodes of each connected component S. The Concorde callable library by Applegate et al. (2012) can be used to determine these connected components.

An outline of a classical Benders decomposition method for the HLLP is depicted in Algorithm 1, in which $\nu(MP)$ and $\nu(SP)$ denote the current optimal solution of the MP and SP, respectively; LB and UB are the current lower and upper bounds, respectively; and ϵ is a given tolerance. Furthermore, (z^h, y^h) and $(\Gamma^h, \theta^h, \beta^h, u^h, v^h, \delta^h)$ are the solutions supplied for the Benders MP and the dual SP at iteration h, respectively, and $\mathbb{G}^h$ is the restricted set of extreme points of $\mathbb{G}$ generated up to iteration h.

Algorithm 1 Classical Benders decomposition algorithm

Let $UB = +\infty$, $LB = -\infty$, $h = 1$, $G^h = \emptyset$
while $UB - LB > \epsilon$ **do**
  Solve the MP (2.32)-(2.36)
  $LB \leftarrow \nu(MP)$
  $z_k^h \leftarrow z_k$
  $y_{km}^h \leftarrow y_{km}$
  for all $(i, j) \in N \times N : i \neq j\}$ **do**
    Solve the SP (2.22)-(2.30)
  end for
  $\mathbb{G}^{h+1} \leftarrow \mathbb{G}^h \cup \{(\Gamma^h, \theta^h, \beta^h, u^h, v^h, \delta^h)\}$
  $UB \leftarrow \min\{UB\ ,\ \nu(SP)\}$
  $h \leftarrow h + 1$
end while

The algorithm stops when the UB and LB converge to the optimal solution value with a tolerance ϵ. In the next section, several strategies to improve this classical version of Benders decomposition are presented.

2.3.2 Benders decomposition enhancements

Although Benders decomposition can be successful even in its classical form (see, e.g., Geoffrion and Graves (1974)), several techniques to make the method perform better

have been proposed in the literature. Most of these techniques focus on one of the major bottlenecks of the method which is the need of solving several instances of the Benders MP. That is, the traditional way to design a Benders decomposition algorithm requires the solution of an integer (NP-hard) optimization problem at each iteration of the algorithm.

In order to improve the performance of the method, McDaniel and Devine (1977) proposed to relax the integrality constraints of the MP during the first iterations of the method. In this way, Benders cuts can be generated without solving an integer program. Geoffrion and Graves (1974) proposed to generate BCs without solving the master problem to optimality by stopping using a tolerance. More recently, Rei et al. (2009) integrated Benders decomposition with a local branching algorithm on the Benders MP to improve the upper bound and to find potential new cuts by solving several master problems in a restricted search space.

Magnanti and Wong (1981) proposed a strategy to improve the convergence of the method based on the generation of strong cuts at each iteration. The idea is to generate Pareto-optimal cuts, i.e., cuts that are not dominated by any other one. Another strategy that can reduce the number of iterations is to add multiple Benders cuts at each iteration (Birge and Louveaux, 1988). One can also generate cuts inside the branch-and-cut tree of a single Benders MP (Codato and Fischetti, 2006; Fortz and Poss, 2009; Naoum-Sawaya and Elhedhli, 2013) to avoid solving the MP from scratch at each iteration.

In this paper, three kinds of improvements are considered in order to speed up the convergence of the method: multiple cuts strategy, Benders decomposition within a branch-and-cut framework, and the use of a specialized algorithm to solve the dual subproblem.

2.3.2.1 Multiple cuts strategy:

The first enhancement is to add multiple Benders cuts to the MP at each iteration. Given that the Benders SP can be decomposed in $n(n-1)$ smaller subproblems, two strategies can be considered. The first consists of adding n Benders cuts at each iteration, i.e., one for each origin i. Therefore, for any optimal dual solution $(\Gamma^h, \theta^h, \beta^h, u^h, v^h, \delta^h)$ the following set of cuts can be defined:

$$\eta_i \geq \sum_j [\Gamma^g_{ij} + \theta^g_{ij} - \sum_k (u^g_{ijk} + v^g_{ijk}) z_k - \sum_k \sum_{m:m>k} \delta^g_{ijkm} y_{km}] \quad \forall i \in N,\ g \in \mathbb{G}. \tag{2.37}$$

In this case, the Benders MP objective function can be rewritten as min $\eta = \sum_i \eta_i$.

The second strategy is to add $n(n-1)$ cuts at each iteration, i.e., to add the

following set of cuts for each pair i, j:

$$\eta_{ij} \geq \Gamma^g_{ij} + \theta^g_{ij} - \sum_k (u^g_{ijk} + v^g_{ijk}) z_k - \sum_k \sum_{m:m>k} \delta^g_{ijkm} y_{km} \quad \forall i, j \in N,\ g \in \mathbb{G}, \tag{2.38}$$

where the Benders MP objective consists of minimizing $\eta = \sum_i \sum_j \eta_{ij}$.

2.3.2.2 Benders-branch-and-cut scheme:

The Benders-branch-and-cut (BBC) approach consists in adding Benders cuts within a standard branch-and-cut framework. This idea has been used by Codato and Fischetti (2006) who add combinatorial Benders cuts *(i)* before updating the incumbent solution; *(ii)* for all fractional solutions in nodes with a tree depth up to 10; and *(iii)* after each backtracking step. On a related note, Fortz and Poss (2009) and Gelareh and Nickel (2011) designed a Benders-branch-and-cut framework which add Benders cuts for each potential incumbent solution. Naoum-Sawaya and Elhedhli (2013) presented a Benders-branch-and-cut framework which add Benders cuts at every node of the tree. Adulyasak et al. (2012) proposed a Benders-branch-and-cut approach considering three cut generation strategies: adding Benders cuts at every node in the branch-and-bound tree, adding Benders cuts at the root node and for all potential incumbent solutions found and adding Benders cuts only when a potential incumbent solution is found. All of these strategies result in a single Benders iteration.

In order to solve the HLLP, the three cut generation strategies adopted by Adulyasak et al. (2012) are also tested in this paper. Let *ncomp* denote the number of connected components of the current line. Then, Algorithm 2 presents the modifications to the standard branch-and-cut framework needed to implement the Benders-branch-and-cut by adding cuts only from potential incumbent solutions. When adding Benders cuts from fractional solutions, however, the step of finding the connected components is ignored.

2.3.2.3 Algorithm to solve the subproblem:

Finally, to avoid using a general-purpose solver to find the dual SP solution, an algorithm to solve the dual SP is proposed. The purpose of such a procedure is twofold. The first one takes into account that the dual subproblem related to network flow problems is usually degenerated, i.e. it may have multiple optimal solutions, which allows for the generation of different cuts at each round (Magnanti and Wong, 1981). Hence, instead of using a general-purpose solver that finds an arbitrary optimal solution, the idea is to design a method that is able to select dual optimal values such that the resulted coefficients of the Benders cuts are closer to zero allowing then to obtain large values

Algorithm 2 Adaptation of a standard branch-and-cut framework for the BBC implementation

$h = 1$, $G^h = \emptyset$
for all Potential incumbent solution (z, y) **do**
  Find the connected components of the line by means of Concorde
  if $ncomp > 1$ **then**
    Add SECs to MP
  else
    $z_k^h \leftarrow z_k$
    $y_{km}^h \leftarrow y_{km}$
    for all $(i, j) \in N \times N : i \neq j$ **do**
      Solve the SP (**??**)-(**??**)
    end for
    $\mathbb{G}^{h+1} \leftarrow \mathbb{G}^h \cup \{(\Gamma^h, \theta^h, \beta^h, u^h, v^h, \delta^h)\}$
    $h \leftarrow h + 1$
  end if
end for

for η_{ij} variables. Furthermore, the other goal of designing a specialized algorithm is to solve the SP faster than by means of a solver.

The main idea of this algorithm is to take advantage of the fact that if the Benders MP solution is known, then the solution of the primal SP (2.14)-(2.21) consists in finding the shortest path between each pair i, j. After finding the solution of the primal SP, the dual SP can be solved by exploiting the complementary slackness conditions (CSCs).

let $H^1 = \{k : z_k^h = 1\}$ and $H^0 = \{k : z_k^h = 0\}$ be the set of open and closed hubs, respectively. Similarly, let $A^1 = \{(k, m) : y_{km}^h = 1\}$ and $A^0 = \{(k, m) : y_{km}^h = 0\}$ be the set of selected and not selected hub arcs, respectively. In addition, define $SPath[k, m]$ as the value of the travel time between the hubs k and m by means of the current hub line network, where this value is already considering the access time $\tilde{t}_k^a$ and exit time $\tilde{t}_m^e$. Let A_{km} denote the set of edges $(r, s) \in A^1$ that is in the path between hubs k and m by means of the current network. A procedure to find the primal subproblem solution is presented in Algorithm 3.

This algorithm consists in finding the shortest path between each pair of hubs in the line. After that, a complete enumeration of all pairs of hubs is done to determine the first and the last hub that is on the shortest path from i to j by using the line. Finally, the shortest path between the path that uses the line and the direct connection is chosen.

Let $(e_{ij}^h, a_{ijk}^h, b_{ijk}^h, x_{ijkm}^h)$ be the optimal solution of the primal SP. Furthermore, let H_{ij}^1 and A_{ij}^1 denote the sets of hubs and hub arcs, respectively, that form the path from i to j. According to the CSCs, the optimal solution of the dual subproblem at iteration

Algorithm 3 Solving the primal subproblem

$e_{ij} = a_{ijk} = b_{ijk} = x_{ijkm} = 0$
for all $(k,m) \in H^1 \times H^1 : k \neq m$ **do**
  Find $SPath[k,m]$
  $x_{kmrs} = 1,\ \forall (r,s) \in A_{km}$
end for
for all $(i,j) \in N \times N : i < j$ **do**
  $(k,m) = \arg_{(r,s)\in H^1\times H^1} \min(t_{ir} + SPath[r,s] + t_{sj})$
  if $t_{ij} < (t_{ik} + SPath[k,m] + t_{mj})$ **then**
    $e_{ij} = 1$
  else
    $a_{ijk} = b_{ijm} = 1$
    $x_{ijrs} = x_{kmrs}$ for $r,s \in N : r \neq s$
  end if
end for

h is a feasible dual solution $(\Gamma, \theta, \beta, u, v, \delta)$ that satisfies the following set of equations for each pair i, j:

$$
\begin{aligned}
&(-a^h_{ijk} + z^h_k)u_{ijk} = 0 && \forall k \in N\\
&(-b^h_{ijk} + z^h_k)v_{ijk} = 0 && \forall k \in N\\
&(-x^h_{ijkm} - x^h_{ijmk} + y^h_{km})\delta_{ijkm} = 0 && \forall k,m \in N : k < m\\
&(\theta_{ij} + \Gamma_{ij} - w_{ij}t_{ij})e^h_{ij} = 0 &&\\
&(\theta_{ij} + \beta_{ijk} - u_{ijk} - w_{ij}\left(t_{ik} + \tilde{t}^a_k\right))a^h_{ijk} = 0 && \forall k \in N\\
&(\Gamma_{ij} - \beta_{ijk} - v_{ijk} - w_{ij}\left(t_{kj} + \tilde{t}^e_k\right))b^h_{ijk} = 0 && \forall k \in N\\
&(-\beta_{ijk} + \beta_{ijm} - \delta_{ijkm} - w_{ij}\alpha_{km}t_{km})x^h_{ijkm} = 0 && \forall k,m \in N : k < m\\
&(-\beta_{ijk} + \beta_{ijm} - \delta_{ijmk} - w_{ij}\alpha_{km}t_{km})x^h_{ijkm} = 0 && \forall k,m \in N : k > m.
\end{aligned}
$$

These set of conditions imply that:

$$
\begin{aligned}
&u_{ijk} = 0 && \forall k \in H^1 : a_{ijk} = 0 && (2.39)\\
&v_{ijk} = 0 && \forall k \in H^1 : b_{ijk} = 0 && (2.40)\\
&\delta_{ijkm} = 0 && \forall (k,m) \in A^1 \setminus A^1_{ij} : k < m && (2.41)\\
&\theta_{ij} + \Gamma_{ij} = w_{ij}t_{ij} && \text{if } H^1_{ij} = \emptyset && (2.42)\\
&\theta_{ij} + \beta_{ijk} - u_{ijk} = w_{ij}\left(t_{ik} + \tilde{t}^a_k\right) && \forall k \in H^1_{ij} : a_{ijk} = 1 && (2.43)\\
&\Gamma_{ij} - \beta_{ijk} - v_{ijk} = w_{ij}\left(t_{kj} + \tilde{t}^e_k\right) && \forall k \in H^1_{ij} : b_{ijk} = 1 && (2.44)\\
&-\beta_{ijk} + \beta_{ijm} - \delta_{ijkm} = w_{ij}\alpha_{km}t_{km} && \forall (k,m) \in A^1_{ij} : k < m && (2.45)\\
&-\beta_{ijk} + \beta_{ijm} - \delta_{ijmk} = w_{ij}\alpha_{km}t_{km} && \forall (k,m) \in A^1_{ij} : m < k. && (2.46)
\end{aligned}
$$

Taking into account the feasibility requirement for the SP dual solutions, given by

constraints (2.24)-(2.30) and (2.39)-(2.41), the set of optimal dual solutions must also satisfies the following set of constraints:

$$\theta_{ij} + \beta_{ijk} \leq w_{ij}\left(t_{ik} + \tilde{t}^a_k\right) \qquad \forall k \in H^1 : a_{ijk} = 0 \tag{2.47}$$

$$\Gamma_{ij} - \beta_{ijk} \leq w_{ij}\left(t_{kj} + \tilde{t}^e_k\right) \qquad \forall k \in H^1 : b_{ijk} = 0 \tag{2.48}$$

$$-\beta_{ijk} + \beta_{ijm} \leq w_{ij}\alpha_{km}t_{km} \qquad \forall (k,m) \in A^1 \setminus A^1_{ij} : k < m \tag{2.49}$$

$$-\beta_{ijm} + \beta_{ijk} \leq w_{ij}\alpha_{km}t_{mk} \qquad \forall (m,k) \in A^1 \setminus A^1_{ij} : m < k. \tag{2.50}$$

Therefore, any optimal dual solution must satisfies constraints (2.24)-(2.30) and (2.39)-(2.50). An optimal dual solution can be found in three steps. The first step consists in finding a feasible solution for the system of equations (2.39)-(2.46). In this phase, we propose a natural solution for this system in which the variable Γ_{ij} is given as the weighted shortest travel time from i to j, the variables β_{ijk}, such that $k \in H^1_{ij}$, are set as the weighted shortest path from i to k in the optimal path from i to j and the others variables is fixed to zero. After setting the value of the variables in the first step, the second step is responsible for computing a proper value of variables β_{ijk} for all $k \in N \setminus H^1_{ij}$. As will be shown later, this phase is the most challenging part of the procedure because of the difficulty of finding a dual solution that satisfies the system of constraints (2.47)-(2.50) and the impact of these set of variables in the quality of the Benders cuts. Finally, the value of the other dual variables, u_{ijk}, v_{ijk} for all $k \in H^0$ and δ_{ijkm} for all $(k,m) \in A^0$, can be set such as to satisfies the set of dual solution given by constraint (2.24)-(2.29). This step is easier to perform since all of the others variables are already known.

Solving the system of equations (2.39)-(2.46). The solution of this system can be separated in two cases: the case where the flow from i to j is sent through a direct connection and the case where this flow uses the hub line. In the first case, no hub is used to route the flow from i to j, i.e. $A^1_{ij} = \emptyset$ and $H^1_{ij} = \emptyset$. Hence the problem reduces to finding a solution for equation (2.42). Possible values for the variables Γ_{ij} and θ_{ij} that satisfy this equation are:

$$\theta_{ij} = 0 \text{ and } \Gamma_{ij} = w_{ij}t_{ij}. \tag{2.51}$$

In the second case, where the flow is routed through the line, at least one hub is used. Let $L = \{r_1, r_2, ..., r_{p-1}, r_p\}$ denote an ordered set of open hubs, where the hubs are ordered according to the position in which they appear on the line. Let r_s and r_q be the first and the last hubs in the path between i and j. For the sake of simplicity, let $\widehat{\beta}_k = \beta_{ijk}$. By constraints (2.43)-(2.46), the value of the variables Γ_{ij}, θ_{ij}, u_{ijr_s}, v_{ijr_q},

$\widehat{\beta}_{r_l}$ for $s \leq l \leq q$ and $\delta_{ijkm} \in A_{ij}^1$ must satisfy the following equations:

$$\theta_{ij} + \widehat{\beta}_{r_s} - u_{ijr_s} = w_{ij}(t_{ir_s} + \tilde{t}^a_{r_s})$$
$$-\widehat{\beta}_{r_l} + \widehat{\beta}_{r_{l+1}} - \delta_{ijr_lr_{l+1}} = w_{ij}\alpha_{r_lr_{l+1}}t_{r_lr_{l+1}} \quad \text{where } s \leq l < q \text{ and } r_l < r_{l+1}$$
$$-\widehat{\beta}_{r_l} + \widehat{\beta}_{r_{l+1}} - \delta_{ijr_{l+1}r_l} = w_{ij}\alpha_{r_lr_{l+1}}t_{r_lr_{l+1}} \quad \text{where } s \leq l < q \text{ and } r_{l+1} < r_l$$
$$\Gamma_{ij} - \widehat{\beta}_{r_q} - v_{ijr_q} = w_{ij}(t_{r_qj} + \tilde{t}^e_{r_q}).$$

Letting $\theta_{ij} = u_{ijr_s} = v_{ijr_q} = 0$ and $\delta_{ijkm} = 0$ for all $(k,m) \in A_{ij}^1$, a solution for the above systems of equations can be found recursively by the following equations:

$$\widehat{\beta}_{r_s} = w_{ij}(t_{ir_s} + \tilde{t}^a_{r_s}) \tag{2.52}$$

$$\widehat{\beta}_{r_{l+1}} = \widehat{\beta}_{r_l} + w_{ij}\alpha_{r_lr_{l+1}}t_{r_lr_{l+1}}, \quad \text{for } s \leq l < q \tag{2.53}$$

$$\Gamma_{ij} = w_{ij}(t_{r_qj} + \tilde{t}^e_{r_q}) + \widehat{\beta}_{r_q} = w_{ij}(t_{r_qj} + \tilde{t}^e_{r_q}) + \sum_l w_{ij}\alpha_{r_{l-1}r_l}t_{r_{l-1}r_l} + w_{ij}(t_{ir_s} + \tilde{t}^a_{r_s}). \tag{2.54}$$

It is interesting note, that the variable $\widehat{\beta}_{r_l}$ is set as the weighted shortest travel time from origin i to hub r_l in the path from i to j. While, the value of the variable Γ_{ij} is set as the value of the weighted shortest travel time from i to j.

Solving the system of inequalities (2.47)-(2.50). This phase can also be separated in the case where the flow from i to j is sent through a direct connection and the case where this flow uses the hub line.

In the first case, the values of $\widehat{\beta}_k$ for $k \in H^1$ need to satisfy the bound constraints (2.47)-(2.48) and the set of constraints (2.49)-(2.50), where this set of constraints relates each variable $\widehat{\beta}_k$ to a $\widehat{\beta}_m$ such that m is adjacent to k on the hub line. One strategy to solve this system of equations is to give a convenient feasible value to $\widehat{\beta}_k$ associated with a k that is in one of the line ends. After that, giving a value to the variable $\widehat{\beta}_k$ associated with the hub that is next to the previous hub in the line can be done recursively until the other end of the line is reached. However, before setting a value to a variable $\widehat{\beta}_k$ it is necessary to ensure that this value belongs to a feasibility interval FI_k, where FI_k is an interval of possible values for $\widehat{\beta}_k$ such that the system resulting from fixing the value of this variable remains feasible.

Let Φ_l^i and Φ_l^j be the shortest travel time from i to hub r_l and from hub r_l to j, respectively, in the line segment $L_l = \{r_1, r_2, ..., r_{l-1}, r_l\}$. Φ_l^i can be computed

recursively by means of $\Phi_1^i = t_{ir_1} + \tilde{t}^a_{r_1}$ and $\Phi_l^i = \min\{\tilde{t}^a_{r_l} + t_{ir_l}, \Phi^i_{l-1} + \alpha_{r_{l-1}r_l} t_{r_{l-1}r_l}\}$, for all $1 < l \leq p$, which set Φ_l^i as the shortest travel time among the path that send the flow to r_l from the line segment L_{l-1} and the path that send the flow to r_l through the acess arc (i, r_l). In the same way, Φ_l^j can be set by means of $\Phi_1^j = t_{r_1 j} + \tilde{t}^e_{r_1}$ and $\Phi_l^j = \min\{t_{r_l j} + \tilde{t}^e_{r_l}, \alpha_{r_l r_{l-1}} t_{r_l r_{l-1}} + \Phi^j_{l-1}\}$, for all $1 < l \leq p$. The following proposition shows how to find the feasibility interval for $\widehat{\beta}_k$.

Proposition 2. *A feasibility interval for $\widehat{\beta}_k$, for $k \in L$, can be set as follows:*

$$LB_p \leq \widehat{\beta}_{r_p} \leq UB_p$$

$$\max\{LB_l, -w_{ij}\alpha_{r_l r_{l+1}} t_{r_l r_{l+1}} + \widehat{\beta}_{r_{l+1}}\} \leq \widehat{\beta}_{r_l} \leq \min\{UB_l, w_{ij}\alpha_{r_{l+1} r_l} t_{r_{l+1} r_l} + \widehat{\beta}_{r_{l+1}}\}, \ \forall l \in 1..(p-1),$$

where $UB_l = w_{ij}\Phi_l^i - \theta_{ij}$ and $LB_l = \Gamma_{ij} - w_{ij}\Phi_l^j$.

A proof of this proposition can be found in the Appendix A.

These results can also be used to find a feasibility interval for $\widehat{\beta}_k$, such that $k \in H^1 \setminus H^1_{ij}$, when the flow is routed through the line. In this situation, the line L can be partitioned into three line segments. The first line segment is from the hub at the extremity r_1 to r_s (Seg. 1), the second line segment is between hub r_s and r_q (Seg. 2) and the third one (Seg. 3) is from r_q to r_p. The solution of the system of inequalities (2.47)-(2.50) consists in finding feasible values for $\widehat{\beta}_k$ such that hub k is in Seg. 1 or Seg. 3. Taking into account that a line segment is also a line, then the previous proposition can be used to find the feasibility interval for these variables. However, before using these results it is necessary to prove that the value already given to $\widehat{\beta}_{r_q}$ and $\widehat{\beta}_{r_s}$ is in their feasibility interval. A proof of this statement can be found in the Appendix B.

Despite Proposition 2 presents a mechanism to find feasible values for $\widehat{\beta}_k$, setting arbitrary values for theses variables in their feasibility interval can result into weak Benders cuts. It is possible however to exploit the strategy of the first step to select good optimal values for the dual variables which allows for a more suitable Benders cut form. For instance, after fixing variables $\theta_{ij} = 0$, $u_{ijk} = v_{ijk} = 0$ for all $k \in H^1$ and $\Gamma_{ij} = \nu(SP)$, which is the value of the optimal solution of the SP, the Benders cut (2.38) can be rewritten as:

$$\eta_{ij} \geq \nu(SP) - \sum_{k \in H^0} (u_{ijk} + v_{ijk}) z_k - \sum_{(k,m) \in A^0} \delta_{ijkm} y_{km}. \tag{2.55}$$

We are only presenting here the multiple cuts version for the sake of simplicity, but the idea can be employed in all versions of the cut in a straightforward way. One can note that the coefficients of the master variables z_k and y_{km} will always be non-positive. So the closer these coefficients values are to zero the larger the values of variables η_{ij} may

be, resulting thus in potentially larger values for the lower bound of the MP.

Although these coefficients are directly dependent to the values of variables u_{ijk}, v_{ijk} for all $k \in H^0$ and δ_{ijkm} forall $(k,m) \in A^0$, these variables for their turn are also tied to the values of variables θ_{ij}, Γ_{ij} and $\widehat{\beta}_k$ by means of the dual SP constraints (2.24)-(2.30). That is, if the values of variables θ_{ij}, Γ_{ij} and $\widehat{\beta}_k$ are already known, then the largest values for the coefficients of z_k and y_{km} in the BC can be attained by:

$$u_{ijk} = \max\{0, \theta_{ij} + \widehat{\beta}_k - w_{ij}\left(t_{ik} + \tilde{t}^a_k\right)\} \qquad \forall k \in H^0 \tag{2.56}$$

$$v_{ijk} = \max\{0, \Gamma_{ij} - \widehat{\beta}_k - w_{ij}\left(t_{kj} + \tilde{t}^e_k\right)\} \qquad \forall k \in H^0 \tag{2.57}$$

$$\delta_{ijkm} = \max\{0, -\widehat{\beta}_k + \widehat{\beta}_m - w_{ij}\alpha_{km}t_{km}, \widehat{\beta}_k - \widehat{\beta}_m - w_{ij}\alpha_{km}t_{mk}\} \quad \forall (k,m) \in A^0 : k < m. \tag{2.58}$$

A proper value to $\widehat{\beta}_k$ can be chosen from the feasibility interval with the objective to make the values of the variables u, v and δ closer to zero. In order to obtain the smallest values for the variables u_{ijk} and v_{ijk}, which increase the coefficient of z_k, it is necessary that the second component of the maximization function of equations (2.56) and (2.57), respectively, to be non-positive. This is attained when $\widehat{\beta}_k$ satisfies

$$\widehat{\beta}_k \leq -\theta_{ij} + w_{ij}\left(t_{ik} + \tilde{t}^a_k\right) \quad \forall k \in H^0 \tag{2.59}$$

and

$$\widehat{\beta}_k \geq \Gamma_{ij} - w_{ij}\left(t_{kj} + \tilde{t}^e_k\right) \quad \forall k \in H^0, \tag{2.60}$$

respectively.

While the lowest possible value for δ_{ijkm}, which increases the coefficient of y_{km}, can be obtained when the second and the third components of the maximization function of equation (2.58) is non-positive, i.e., when $\widehat{\beta}_k$ satisfies the following inequalities:

$$\widehat{\beta}_m - w_{ij}\alpha_{km}t_{km} \leq \widehat{\beta}_k \leq \widehat{\beta}_m + w_{ij}\alpha_{mk}t_{mk}.$$

It is important to note that the value of $\widehat{\beta}_k$ affects both the coefficients of z_k and y_{km}. Since we can not guarantee that there is a valid $\widehat{\beta}_k$ that makes both set of coefficients to be zero, we can define a priority variable to tackle. After some empirical tests, we have chosen to prioritize the reduction of the coefficients of variables y_{km} when setting the values for the $\widehat{\beta}_k$. Furthermore, we can also observe that the values of a given $\widehat{\beta}_k$ can affect the values for δ_{km} for all m. Hence, we can set this value taking into account the impact on the set of variables δ_{km} for all $m < k$ such that $(k,m) \in A^0$ and for all $m > k$ such that $(m,k) \in A^0$.

Let

$$\beta^1_k = \max_{m \neq k}\{\bar{\beta}^1_m - w_{ij}\alpha_{km}t_{km} : \bar{\beta}^1_m - w_{ij}\alpha_{km}t_{km} \in FI_k\}, \tag{2.61}$$

and

$$\beta_k^2 = \min_{m \neq k}\{\bar{\beta}_m^2 + w_{ij}\alpha_{mk}t_{mk} : \bar{\beta}_m^2 + w_{ij}\alpha_{mk}t_{mk} \in FI_k\}, \tag{2.62}$$

be a possible value of $\widehat{\beta}_k$ that tries to reduce the second component and third component of the maximization function of equation (2.58), respectively, where the value of $\bar{\beta}_m^1$ and $\bar{\beta}_m^2$ is equal to $\widehat{\beta}_m$ when this value is already known or they are assumed to be equal to LB_m and UB_m, respectively, otherwise. Despite LB_m and UB_m for $m \in H^0$ to be originally defined as $LB_m = -\infty$ and $UB_m = \infty$ (remember that Proposition 2 is only applied to LB_m and UB_m for $m \in H^1$), we assume that $LB_m = \Gamma_{ij} - w_{ij}\left(t_{kj} + \tilde{t}_k^e\right)$ to satisfies at least constraints (2.60) in order to also increase the coefficients of z_m.

Define

$$R_k(\widehat{\beta}_k) = \sum_m \max\{0, -\widehat{\beta}_k + \bar{\beta}_m - w_{ij}\alpha_{km}t_{km}, -\bar{\beta}_m + \widehat{\beta}_k - w_{ij}\alpha_{mk}t_{mk} : (k,m) \in A^0\} \tag{2.63}$$

as an estimator of the sum of δ_{km} for a given value of $\widehat{\beta}_k$. A proper value for $\widehat{\beta}_k$ can be chosen among the values β_k^1 and β_k^2 that results in the smallest R_k. After fixing this variable, the other set of variable can be set by means of equations (2.56)-(2.58).

Algorithm 4 presents a general framework to find the optimal dual solutions. Where, $SEGSET$ is equal to the whole line, when the flow is sent by means of a direct connection and $SEGSET$ is composed by the segments Seg. 1 and Seg. 3, when the flow is sent by means of the line. In this case, step (6) and (7) can be ignored since the value of these variables have already been fixed in line 1.

Algorithm 4 Algorithm to find the optimal dual solutions for a given pair i,j

1: Set $\Gamma_{ij}, \theta_{ij}, \beta_{ijk} \in H_{ij}^1, u_{ijk} \in H^1$ and $v_{ijk} \in H^1$ by means of (2.51) and (2.52)-(2.54).
2: **for all** $k \in r_1 \ldots r_p$ **do**
3: Find the feasibility interval $[UB_k, LB_k]$
4: **end for**
5: **for all** $L' = \{r_1, \ldots, r_{p'}\}' \in SEGSET$ **do**
6: Find $\widehat{\beta}_{r_{p'}}^1$ and $\widehat{\beta}_{r_{p'}}^2$ by means of (2.61) and (2.62)
7: Set $\widehat{\beta}_{r_{p'}}$ as the argument $\widehat{\beta}_k \in \{\beta_{r_{p'}}^1, \beta_{r_{p'}}^2\}$ that minimizes R_k given by Equation (2.63).
8: **for all** $k \in (p'-1) \ldots 1$ **do**
9: Update the feasibility interval to $\widehat{\beta}_{r_k}$ by using equation (A.13)
10: Find $\widehat{\beta}_{r_k}^1$ and $\widehat{\beta}_{r_k}^2$ by means of (2.61) and (2.62)
11: Set $\widehat{\beta}_{r_k}$ as the argument $\widehat{\beta}_s \in \{\beta_{r_k}^1, \beta_{r_k}^2\}$ that minimize R_k.
12: **end for**
13: **end for**
14: Find the value of variables u_{ijk}, v_{ijk} and δ_{ijkm} by means of equations (2.56)-(2.58)

2.4 Computational Experiments

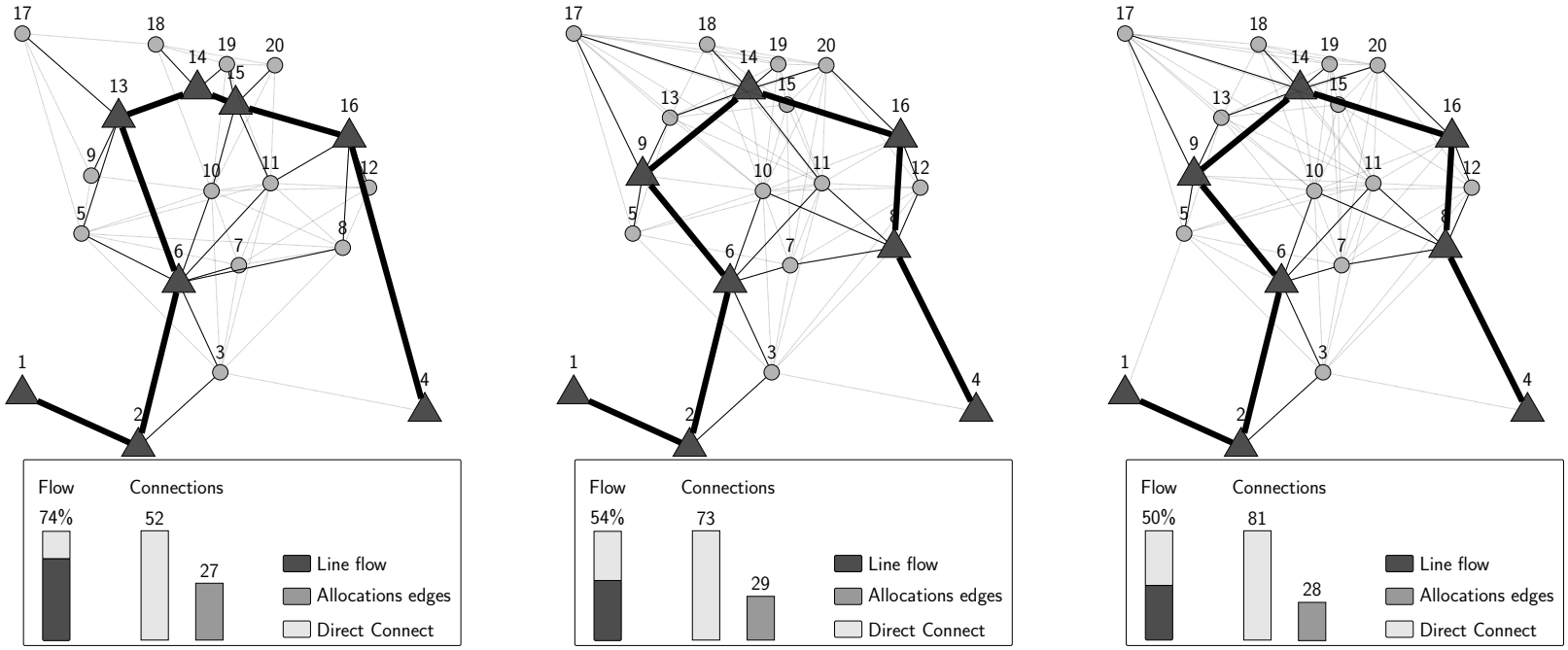
Computational experiments were performed by using two standard benchmark instances for hub location: the CAB data set of the US Civil Aeronautics Board and the Australian Post (AP) data set first used in Ernst and Krishnamoorthy (1996). CAB data set is a set of instances introduced by O'Kelly (1987) that has instances with 10, 15, 20, 25 node with a symmetric origin-destination demand matrix, while AP set of instances, first used in Ernst and Krishnamoorthy (1996), has instances ranging from 10 up to 200 nodes with an asymmetric origin-destination demand matrix. Tests were carried out by considering the following parameters: discount factor $\alpha_{ij} = \alpha = \{0.2, 0.5, 0.8\}$ and number of hubs $p = \{5, 8\}$. In all the test, we considered a access and exit times that does not depend on the node, i.e., $\tilde{t}^a_k = \tilde{t}^a$ and exit times $\tilde{t}^e_k = \tilde{t}^e$. In this particular case, using the hub line induces a total transfer time $\tilde{t}$ given as the sum of the access time and exit time, that is $\tilde{t} = \tilde{t}^a + \tilde{t}^e$. This transfer time is controlled by means of the parameter $\vartheta = \{0, 0.1, 0.25\}$ which sets this time interval as a proportion of the average travel time $\bar{t}$ computed as

$$\bar{t} = \frac{\sum_i \sum_j t_{ij}}{n(n-1)}.$$

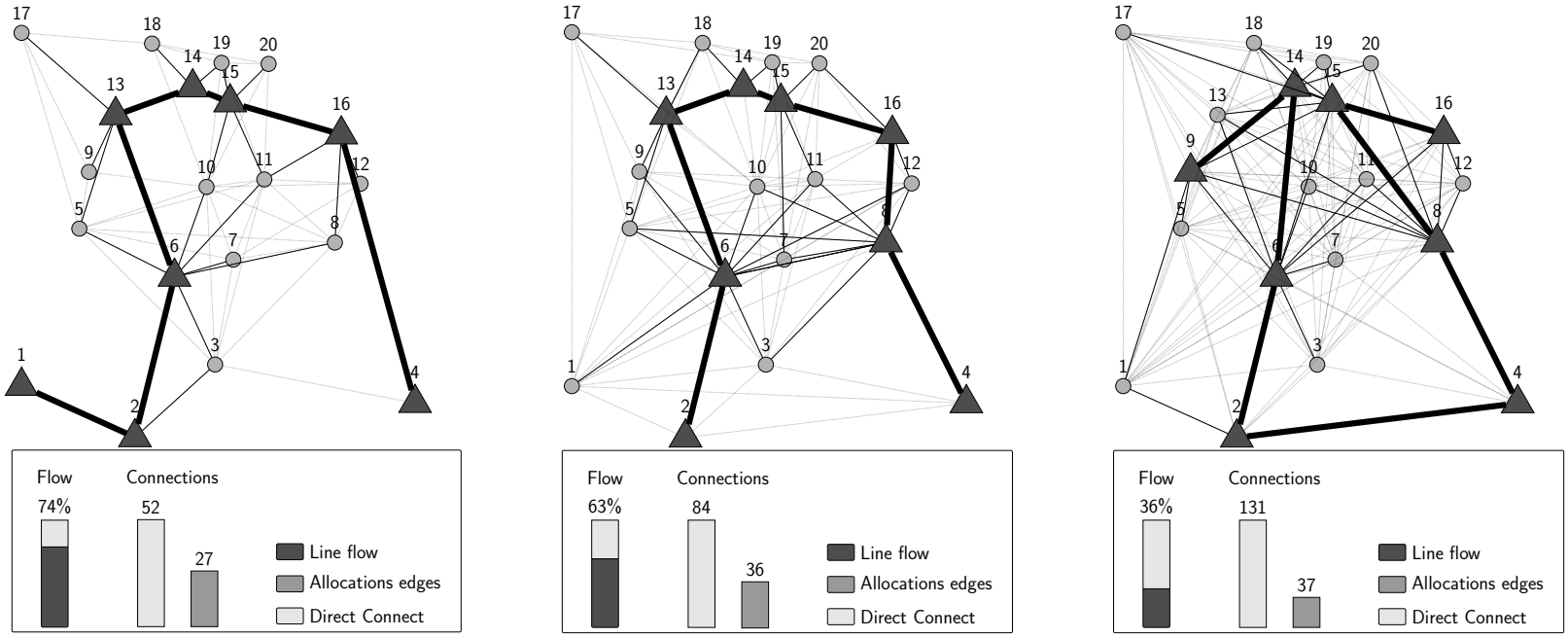
For example, setting $\vartheta = 0.1$ means that the transfer time is 10% of $\bar{t}$.

To analyze how the parameters α and ϑ affect the configuration of the hub line network, Figures 2.2 and 2.3 present different network configurations obtained by varying these parameters, considering a network with 20 demand nodes, using an AP instance, and 8 hubs. We also report some information about the designed system such as the percentage of flow that uses the line and the number of direct connections and allocations. Figure 2.2 presents the characteristics of the system when the economies of scale factor α is equal to 0.2. According to this figure the flow that uses the line decreases as the waiting time increases. Furthermore, the number of allocations of non-hub to hub nodes increases as the waiting time increases from $\vartheta = 0.0$ to $\vartheta = 0.1$. On the other hand, this number decreases for a large waiting time, likely because the use of the line may be less attractive. Figure 2.3 presents the network configurations by fixing the parameter of waiting time $\vartheta = 0$ and changing the value of α. In this case, the demand flow using the line decreases when α increases. Furthermore, the number of direct connections and the number of allocations increase as α increases.

Figure 2.2: Configurations of the hub line for $\alpha = 0.2$ and $p = 8$.



a) $\vartheta = 0.0$ b) $\vartheta = 0.1$ c) $\vartheta = 0.25$



a) $\alpha = 0.2$ b) $\alpha = 0.5$ c) $\alpha = 0.8$

Figure 2.3: Configurations of the hub line for $\vartheta = 0.0$ and $p = 8$.

Computational tests were performed in three steps. In the first step, several variants of the Benders decomposition method were tested using AP instances with 10 to 50 nodes. These first tests were performed in order to find the most promising method for solving the problem. After that, the best algorithm was compared with the use of the general-purpose solver CPLEX in an effort to solve AP instances with up 100 nodes and the CAB instances. In the last step, the best algorithm was used to solve a particular case of the problem in which one hub is known. This case is applied when a central hub is already given and we wish to design a hub line that pass for this hub.

All tests were performed on an 1260 Xeon Westmere 2.66 GHz computer with 24 GB of memory, running Linux. Furthermore, all variants tested were coded in C++ using the Concert Technology (CPLEX 12.5) to solve the Benders MPs and some SPs with a time limit of 24 hours (86,400 seconds). Since the path used to route the flow from i to j is the same (in the opposite direction) used to route the flow from j to i, only one direction (direction $i-j$ $(i<j)$) is taken into account. In this way, the number of subproblems solved by iteration can be reduced to $n(n-1)/2$.

To compare the single-cut variant of Benders decomposition method (BD-1) and multiple-cuts strategies, two variants were implemented: BD-n, which adds n cuts at each iteration, and BD-n^2, where $n(n-1)/2$ cuts are added, one for each pair i, j $(i<j)$. The three variants were implemented by using Algorithm 4 to find the value of the optimal dual variables. A summary of the results is presented in Table 2.1. This table reports the average CPU time in seconds to solve the problem, the average number of iterations (#iter) and the number of problems solved to optimality (#Opt). The instances are grouped by number of nodes and number of opened hubs. According to this table, the classical version that adds only one cut per iteration is able to solve only instances with 10 and 20 nodes. Furthermore, this variant spends more time to solve these instances than the other two variants. One can see that the variant that adds more cuts per iteration is able to solve more instances, spending smaller average CPU time. For this reason, the strategy with $n(n-1)/2$ cuts per iteration was adopted in the following tests.

The second set of tests aims to analyze the benefits of using a special algorithm to solve the Benders SP instead of using CPLEX for this purpose. Table 2.2 presents the results of tests comparing the variant using CPLEX to find the optimal dual solution, BD-cpx-n^2, and the variant using Algorithm 4 for this, BD-alg-n^2. This table presents an additional column to compare the CPU time spent by CPLEX and by the proposed algorithm to solve the Benders SP. Let τ_{cpx} and τ_{alg} denote the average time that CPLEX and the proposed algorithm, respectively, spent to solve the subproblem by iteration, which can be given as (the average total time spent solving subproblems)/(the average number of iterations), where the computation of the averages takes into account

Table 2.1: Comparison of single-cut and multiple-cuts versions.

n	p	BD-alg-1			BD-alg-n			BD-alg-n^2		
		#Opt	Time[s]	#iter	#Opt	Time[s]	#iter	#Opt	Time[s]	#iter
10	5	9/9	34.40	158.78	9/9	1.95	16.78	9/9	0.90	9.11
10	8	9/9	13343.35	782.22	9/9	61.62	45.00	9/9	4.38	13.78
20	5	3/9	64081.77	659.00	9/9	53.57	25.56	9/9	24.72	13.22
20	8	0/9	86400.00	—	6/9	35202.03	69.17	9/9	3732.17	28.56
25	5	0/9	86400.00	—	9/9	243.15	34.00	9/9	131.71	15.56
25	8	0/9	86400.00	—	3/9	59352.94	59.00	7/9	26737.37	28.00
40	5	0/9	86400.00	—	9/9	7450.89	45.11	9/9	3858.55	17.78
40	8	0/9	86400.00	—	0/9	86400.00	—	2/9	75272.74	46.00
50	5	0/9	86400.00	—	8/9	36056.43	56.38	9/9	34278.75	23.22
50	8	0/9	86400.00	—	0/9	86400.00	—	1/9	85410.28	45.00
Average		21/90	68225.95	533.33	62/90	31122.26	43.87	73/90	22945.16	24.02

– time limit exceeded for all instances.

only solved problems. Hence τ_{cpx}/τ_{alg} shows how much faster the proposed algorithm is compared to the CPLEX solver. According to the table, the variant BD-alg-n^2 solves more instances, spending less time and iterations to find the optimal solution. Furthermore, CPLEX spends on average more than twice (2.30) the time to solve the SP compared with the specialized algorithm. Therefore, the proposed algorithm was used to solve the SP in subsequent tests.

Table 2.2: Comparison of Benders decomposition method using the proposed algorithm to solve the SP and by using the CPLEX.

n	p	Bd-cpx-n^2			Bd-alg-n^2			τ_{cpx}/τ_{alg}
		#Opt	Time[s]	#iter	#Opt	Time[s]	#iter	
10	5	9/9	1.40	11.67	9/9	0.90	9.11	3.99
10	8	9/9	22.33	20.56	9/9	4.38	13.78	3.74
20	5	9/9	63.27	19.89	9/9	24.72	13.22	2.03
20	8	6/9	18495.91	39.89	9/9	3732.17	28.56	2.04
25	5	9/9	314.54	21.22	9/9	131.71	15.56	1.89
25	8	4/9	54760.36	32.75	7/9	26737.37	28.00	1.71
40	5	9/9	15113.93	21.78	9/9	3858.55	17.78	2.24
40	8	1/9	77920.90	69.00	2/9	75272.74	46.00	1.79
50	5	6/9	44694.39	24.33	9/9	34278.75	23.22	2.08
50	8	1/9	77190.60	43.00	1/9	85410.28	45.00	1.49
Average		61/90	28857.76	30.41	73/90	22945.16	24.02	2.30

A third set of experiments was performed in order to compare several Benders-branch-and-cut strategies. However, since a large number of Benders cuts are sometimes added at each branching node ($O(n^2)$), some tests were also performed to analyze the convenience of filtering the cuts before adding them to the MP, i.e., selecting which cuts will be added to the model. This test is based on the variant that adds Benders cuts only at the root node and every time a potential incumbent solution is found. Instead of adding all Benders cuts obtained from the SP solution, the algorithm retains

only those that are violated by the current MP solution. Two filtering strategies were analyzed: filter only the cuts from fractional solutions (BDCF1) and filter cuts from fractional solutions and from potential incumbents (BDCF2). These two strategies are compared with the strategy of not filtering (BDCF). Table 2.3 presents the results of the tests. Column #cut shows the average number of cuts added and column #No presents the average number of expanded B&B tree nodes, i.e., the number of nodes of the B&B tree that was examined. According to the table, filtering increases the average number of expanded nodes. However, the average CPU time and the average total number of cuts decrease. Since the variant BDCF2 spends less CPU time to solve the problem, this strategy is used in all Benders-branch-and-cut variants.

Table 2.3: Comparison of a variant without filtering of cuts and two strategies of cut filtering.

n	p	BDCF			BDCF1			BDCF2		
		Time[s]	#Cut	#No	Time[s]	#Cut	#No	Time[s]	#Cut	#No
10	5	0.20	775.00	1.89	0.18	644.67	2.00	0.19	707.78	1.89
10	8	0.55	1185.00	88.33	0.53	1002.44	96.11	0.50	1038.89	100.11
20	5	3.55	3757.78	3.22	3.23	3092.67	2.11	3.32	3808.44	82.11
20	8	78.16	6375.56	942.22	63.43	5202.67	1115.44	54.33	5554.67	991.00
25	5	11.82	6000.00	6.44	8.97	4673.67	9.56	9.07	5018.33	9.44
25	8	485.80	10766.67	1940.22	251.63	8235.56	1685.78	216.72	7246.44	1895.78
40	5	108.44	16640.00	10.44	87.39	14797.22	5.11	94.95	17123.67	11.33
40	8	4843.92	31893.33	1041.89	1768.72	22248.11	1008.11	1908.43	23110.78	1264.44
50	5	279.15	24636.11	3.89	277.05	20612.89	12.78	291.17	20659.67	14.44
50	8	*39164.77	45733.33	810.50	16518.12	37471.44	2543.00	12764.70	37749.33	2242.44
Average		4497.64	14776.28	484.91	1897.93	11798.13	648	1534.34	12201.80	661.30

* Only 6/9 instances are solved to optimality.

Using the filter strategy BDCF2, the three strategies for adding Benders cuts within a B&C tree were compared: adding BCs only for potential incumbent solutions (BDC); adding BCs for any potential incumbent and for any fractional solution at the root node (BDCFR); adding BCs for every solution (BDCFA). Table 2.4 reports the results. According to this table, the variant that adds Benders cuts only from integer MP solutions and in the root node is the most efficient and effective. This variant is able to solve all tested instances and the CPU time required to solve these instances is lower than for the other variants. Furthermore, the strategy of adding BCs at every node of the tree explores few nodes, but requires a lot of CPU time to solve the problem.

Table 2.4: Comparison of three strategies for adding cuts in the B&B tree: only for integer solutions (BDC), for all integer solutions and fractional solutions at the root node (BDCFR) and for all integer and fractional solutions (BDCFA).

n	p	BDC			BDCFR			BDCFA		
		Time[s]	#Cut	#No	Time[s]	#Cut	#No	Time[s]	#Cut	#No
10	5	0.16	1078.67	95.11	0.19	707.78	1.89	0.20	712.33	1.33
10	8	0.70	1575.78	591.22	0.50	1038.89	100.11	1.93	1614.11	34.89
20	5	4.56	8590.78	790.33	3.32	3808.44	82.11	3.37	2998.67	2.67
20	8	123.65	16772.44	7877.67	54.33	5554.67	991.00	1124.94	23174.89	138.89
25	5	19.64	15607.33	1683.22	9.07	5018.33	9.44	11.48	5195.11	3.22
25	8	985.20	31116.00	29189.44	216.72	7246.44	1895.78	1613.68	28343.44	180.44
40	5	410.71	55444.78	5109.56	94.95	17123.67	11.33	92.30	17569.00	1.56
40	8	22002.45	104146.33	89013.22	1908.43	23110.78	1264.44	14025.38	44254.44	214.78
50	5	1707.22	93310.11	11384.22	291.17	20659.67	14.44	271.34	20949.89	2.00
50	8	71795.79[a]	211062.00[a]	306365.33[a]	12764.70	37749.33	2242.44	40015.34[b]	29068.00[b]	41.60[b]
Average		9705.01	53870.42	45209.93	1534.34	12201.80	661.30	5716.00	17387.99	62.14

* a: Only 3/9 instances are solved to optimality.
* b: Only 5/9 instances are solved to optimality.

The second phase of the tests aims to compare the best Benders decomposition variant, BDCFR, and the CPLEX solver to tackle instances with up 100 nodes. Table 2.5 and Table 2.6 presents the results of the experiments for all AP and CAB instances, respectively. The smallest time to find the optimal solution is in boldface. The optimality gap for instances not solved to optimality is presented.

Table 2.5: Comparison of the best Benders decomposition variant and CPLEX using AP data set.

n	p	ϑ	$\alpha = 0.2$			$\alpha = 0.5$			$\alpha = 0.8$		
			CPLEX	BDCFR		CPLEX	BDCFR		CPLEX	BDCFR	
			Time[s]	Time[s]	#No	Time[s]	Time[s]	#No	Time[s]	Time[s]	#No
10	5	0	0.46	**0.27**	1	0.27	**0.13**	1	0.13	**0.1**	1
10	5	0.1	1.23	**0.31**	1	0.17	**0.15**	1	0.13	**0.12**	1
10	5	0.25	2.1	**0.33**	7	0.46	**0.24**	3	0.08	**0.1**	1
10	8	0	5.11	**1.06**	174	1.21	**0.28**	10	1.03	**0.2**	43
10	8	0.1	6.23	**0.69**	88	2.19	**0.38**	55	1.14	**0.21**	34
10	8	0.25	13.27	**0.99**	226	3.41	**0.51**	191	0.64	**0.16**	80
20	5	0	72.83	**4.25**	4	38.56	**2.53**	13	13.73	**1.43**	1
20	5	0.1	45.55	**4.05**	11	38.3	**2.35**	1	21.06	**1.26**	12
20	5	0.25	44.94	**4.31**	17	324.81	**6.34**	35	3.75	**3.34**	645
20	8	0	2279.1	**32.36**	319	898.06	**19.02**	500	56.25	**2.38**	6
20	8	0.1	4525.01	**200.57**	3122	910.55	**14.42**	218	34.43	**2.61**	5
20	8	0.25	6771.1	**193.91**	4131	660.18	**21.64**	610	12.69	**2.08**	8
25	5	0	609.74	**12.19**	1	338.04	**7.3**	6	78.26	**2.85**	6
25	5	0.1	456.68	**12.6**	1	422.78	**6.76**	1	107.51	**3.64**	12
25	5	0.25	686.39	**13.41**	3	1820.28	**20.64**	50	28.55	**2.21**	5
25	8	0	28829.4	**132.5**	715	2880.62	**31.39**	229	236.34	**6.47**	16
25	8	0.1	58465	**884.7**	5760	7253.66	**101.17**	1186	274.08	**8.09**	15
25	8	0.25	61911.6	**587.85**	6128	15787.7	**195.06**	2977	52.86	**3.23**	36
40	5	0	time	**237.31**	31	time	**91.03**	3	5770.48	**25.17**	1
40	5	0.1	33116.1	**160.98**	20	34651.4	**78.86**	43	4124.06	**37.7**	1
40	5	0.25	33347.1	**149.51**	1	14750.4	**50.33**	1	814.14	**23.62**	1
40	8	0	time	**2018.81**	590	time	**834.89**	688	23292.3	**95.71**	23
40	8	0.1	time	**898.95**	137	time	**1501.84**	1331	57783.5	**161.65**	193
40	8	0.25	time	**3711.55**	1827	time	**7843.22**	6117	7167.99	**109.22**	474
50	5	0	time	**657.96**	1	time	**322.53**	21	time	**78.42**	15
50	5	0.1	time	**550.45**	4	time	**154.74**	1	46729.1	**103.96**	19
50	5	0.25	time	**493.28**	1	time	**220.67**	5	6087.26	**38.54**	63
50	8	0	time	**3560.09**	104	time	**1996.59**	648	time	**142.98**	14
50	8	0.1	time	**32423.9**	2139	time	**16446.8**	4755	time	**402.82**	124
50	8	0.25	time	**45321.1**	5759	time	**14467.9**	6432	37545.8	**120.08**	207
75	5	0	mem	**34047.7**	1	time	**3673.9**	1	time	**729.05**	1
75	5	0.1	mem	**8843.2**	1	time	**1966.25**	1	time	**780.76**	1
75	5	0.25	time	**7779.53**	1	time	**1432.37**	1	time	**732.62**	105
75	8	0	time	**48390.2**	409	time	time (1.29%)	269	mem	**8950.62**	425
75	8	0.1	time	time (1.36%)	47	time	time(0.76%)	663	time	**15368.4**	891
75	8	0.25	time	time	0	time	time (0.19%)	690	time	**4049.04**	1225
100	5	0	mem	time	0	mem	**23542.6**	2	mem	**7646.86**	1
100	5	0.1	mem	time	0	mem	**32253.2**	1	mem	**7347.17**	1
100	5	0.25	mem	**65749.4**	1	mem	**15796.8**	1	mem	**4688.35**	10
100	8	0	mem	time	0	mem	time (2.64%)	0	mem	**23062.5**	1
100	8	0.1	mem	time (13%)	0	mem	time	0	mem	time(0.22%)	60
100	8	0.25	mem	time	0	mem	time	0	mem	time (0.02%)	1508

time: 24h time limit exceeded.
mem: 24Gb memory exceeded.

Table 2.6: Comparison of the best Benders decomposition variant and CPLEX using CAB data set.

			$\alpha = 0.2$			$\alpha = 0.5$			$\alpha = 0.8$		
n	p	ϑ	CPLEX	BDCFR		CPLEX	BDCFR		CPLEX	BDCFR	
			Time[s]	Time[s]	#No	Time[s]	Time[s]	#No	Time[s]	Time[s]	#No
10	5	0	0.36	**0.19**	1	1.34	**0.78**	1	739.22	**198.71**	291
10	5	0.1	0.36	**0.22**	1	0.77	**0.41**	1	308.97	**72.67**	33
10	5	0.25	0.4	**0.2**	1	0.6	**0.52**	1	130.95	**8.07**	2
10	8	0	0.36	**0.13**	1	1230.44	**157.94**	3491	31.38	**2.65**	14
10	8	0.1	1.44	**0.23**	1	1727.58	**464.54**	6630	10.02	**2.02**	1
10	8	0.25	1.76	**0.25**	11	1812	**631.99**	14241	118.06	**5.94**	216
15	5	0	1.06	**0.18**	14	57.5	**7.19**	177	1114.08	**89.7**	11
15	5	0.1	0.87	**0.19**	1	94.93	**9.74**	295	628.23	**55.02**	1
15	5	0.25	0.12	**0.09**	10	53.62	**3.76**	116	780.32	**80.96**	1
15	8	0	14.17	**2.11**	582	1.1	**0.42**	1	274.74	**23.21**	1
15	8	0.1	24.36	**5.01**	1140	0.81	**0.63**	1	339.94	**25.2**	1
15	8	0.25	28.8	**14.91**	2924	6.23	**1.08**	89	1436.21	**212.28**	51
20	5	0	6.02	**1**	335	34.13	**12.22**	4	75.27	**5.6**	1
20	5	0.1	11.69	**2.95**	1304	23.51	**5.98**	1	286.75	**76.62**	12
20	5	0.25	8.53	**2.76**	1087	20.03	**9.23**	1	213.17	**9.73**	207
20	8	0	2.12	**0.38**	73	32.77	**8.11**	1	3720.43	**579.69**	154
20	8	0.1	1.82	**0.3**	102	30.46	**2.86**	1	9730.08	**1379.46**	470
20	8	0.25	1	**0.25**	78	10.38	**2.22**	2	25523.4	**2574.82**	2187
25	5	0	9.7	**3.72**	1	7.06	**1.36**	1	8512.97	**859.7**	335
25	5	0.1	24.69	**17.71**	17	4.85	**1.65**	1	19396.7	**6217.05**	1849
25	5	0.25	31.3	**7.65**	21	30.33	**2.72**	5	13041	**2335.17**	1260
25	8	0	5.1	**1.28**	1	2454.24	**305.64**	995	1127	**117.08**	225
25	8	0.1	5.94	**1.54**	1	2938.52	**213.46**	1143	1918.12	**380.59**	1256
25	8	0.25	2.19	**0.83**	1	12144.3	**2479.78**	8358	2121.11	**195.91**	3208

According to Table 2.5, CPLEX is not able to solve some instances with 40 and 50 nodes, and all instances with more than 50 nodes exceed the time limit or the memory. On the other hand, the Benders variant is able to solve all instances with up 50 nodes, most instances with 75 nodes and eight instances with 100 nodes. In addition, nine of the fifteen instances that were not solved by this algorithm present an optimality gap lower than 1.36%. The variant BDCFR solves the problem faster than CPLEX for all CAB instances and for all AP instances solved.

By means of Table 2.5 and Table 2.6, it is possible to analyze how the parameters α and p affect the difficulty of each set of instances. The instances with $\alpha = 0.2$, with high economies of scale, seem to be more difficult than the instances with lower economies of scale (the majority of solved instance have $\alpha = 0.8$) for AP instances. On the other hand, CAB instances with high economies of scale presents to be easier to solve than instance with lower economies of scale. Another parameter that largely affects the difficulty of the problem is the number of open hubs. The solution of the problem is harder for instances AP that open eight hubs than for five hubs. Most instances with 100 nodes for $p = 5$ are solved, while for instances for $p = 8$ only one instance is solved. However, for CAB data set the impact of this parameter depends of the value of parameter α and the set of datas.

Finally, the last phase of the tests was performed to analyze the performance of BDCFR algorithm when the location of a hub is already known. Let OP denote the original problem in which the location of no hub is known beforehand. Define EP as the problem in which the location of an arbitrary hub that was in an extremity of the optimal line of OP is known, and denote MP as the problem in which the location of an arbitrary hub of the optimal line of OP that was not in an extremities is fixed. The results of the test using AP instance with 50 nodes and $p = 8$ is presented on Table 2.7, where the largest time to find the optimal solution is in boldface. According to the table the problems that the location of one hub is fixed is in average faster to solve than the original problem. However, for some instances the computational time required to solve the problems MP and EP is larger than the computational time spent to solve the original problem, see for example the instance $\alpha = 0.2$ and $\vartheta = 0.1$ or $\vartheta = 0.25$. The difference in the performance of the algorithm to solve the three problem may be the results of the high-sensitivity of tree search methods to initial conditions (Fischetti and Monaci, 2014).

Table 2.7: Performance of BDCFR variants when the location of one hub is known.

n	p	α	ϑ	Time [s]		
				OP	EP	MP
50	8	0.2	0.0	**3560.09**	3465.53	1251.42
50	8	0.2	0.1	32423.9	5612.54	**67760**
50	8	0.2	0.25	45321.1	**73180.4**	3740.06
50	8	0.5	0.0	**1996.59**	1843.38	1192.42
50	8	0.5	0.1	**16446.8**	14681.5	2543.75
50	8	0.5	0.25	**14467.9**	6760.05	6810.39
50	8	0.8	0.0	142.98	**180.24**	160.79
50	8	0.8	0.1	402.82	**484.18**	241.46
50	8	0.8	0.25	120.08	**172.15**	123.76
Average				**12764.69**	11819.99	9313.78

2.5 Conclusion

We have presented a new variant of the HLP in which the hubs are required to be connected by means of a line. The HLLP is suitable for some public transportation systems, in which a more flexible hub-and-spoke network allowing direct interactions between non-hub nodes and multiple assignments is desired. An exact algorithm based on Benders decomposition was proposed to optimally solve the problem. The basic Benders decomposition was enhanced through the incorporation of algorithmic features to improve its convergence and efficiency. Extensive computational experiments were performed to analyze the effectiveness of each of the proposed algorithmic refinements. The results show that the addition of multiple cuts per iteration is better than the traditional single-cut approach. Furthermore, the specialized algorithm to solve the dual problem and the integration of Benders decomposition in a B&C framework show a considerable improvement in the convergence of the method. The results confirm the efficiency of the algorithm, which is much faster than CPLEX and able to solve instances with up to 100 nodes.

Chapter 3

Exact and Heuristic Algorithms for the Design of Hub Networks with Multiple Lines

Chapter information

This chapter presents the article submitted for publication in European Journal of Operational Research: Martins de Sá, E., Contreras, I., Cordeau, J.-F. Exact and Heuristic Algorithms for the Design of Hub Networks with Multiple Lines. Submitted for publication.

Abstract

In this paper we study a hub location problem arising in the design of public transportation networks, where the hub-level network is composed by a set of lines. The objective is to minimize the total weighted travel time between pairs of nodes while taking into account a budget constraint on the total set-up cost of the hub network. A mathematical programming formulation, a Benders-branch-and-cut algorithm and several heuristic algorithms, based on variable neighborhood descent, greedy randomized adaptive search, and adaptive large neighborhood search, are presented and compared to solve the problem. Numerical results on two sets of benchmark instances with up to 70 nodes and three lines confirm the efficiency of the proposed solution algorithms.

Keywords: Hub location, hub-and-spoke networks, rapid transit networks.

3.1 Introduction

Hub-and-spoke architectures are often used in the design of large-scale networks such as those found in passenger and freight transportation, postal services, telecommunications, and rapid transit systems. In these networks, commodities from different origins are sent to intermediate facilities, known as hubs, which are responsible for the aggregation and distribution of the flows to multiple destinations. This allows the connection of a large number of origin/destination (O/D) nodes with a small number of arcs, reducing the infrastructure and operational cost (O'Kelly and Miller, 1994). Another important advantage of hub-and-spoke networks is that hub facilities can be connected with highly efficient pathways, enabling economies of scale to be applied on the transportation cost (or travel time) between hubs. *Hub location problems* (HLPs) consider the design of hub networks by selecting a set of nodes to locate hubs, activating a set of links, and routing commodities through the network while optimizing a cost-based (or service-based) objective function. We refer the reader to Alumur and Kara (2008), Campbell and O'Kelly (2012), and Farahani et al. (2013) for surveys on hub location.

Given the inherent difficulty of HLPs, most of the fundamental HLPs consider a fully interconnected hub-level network to simplify the network design decisions. However, it is known that this can be an oversimplification in applications where there is a considerable set-up cost associated with the inter-hub links (see O'Kelly and Miller, 1994). Several HLPs considering incomplete hub-level networks have thus been studied. These problems can be seen from a hub arc location perspective (see Campbell et al., 2005a,b; Contreras and Fernández, 2014), in which the location of a set of hub arcs and their associated hub nodes is considered. Motivated by specific applications, some of these models require the hub-level network to have a particular topological structure, such as cycles (Lee et al., 1993; Contreras et al., 2013a), stars (Labbé and Yaman, 2008), trees (Contreras et al., 2009, 2010; Martins de Sá et al., 2013b), or lines (Martins de Sá et al., 2013a). Some other models do not even require the hub arcs to define a single connected component (Campbell et al., 2005a; Contreras and Fernández, 2014).

The *hub line location problem* (HLLP), introduced in Martins de Sá et al. (2013a), consists of designing a hub network in which p hubs are located and connected by means of a single line. Contrary to most p-hub median models considering a cost-based objective, the HLLP uses a service-based objective that aims at minimizing the total weighted travel time between O/D pairs. It considers that a high-speed mode of transportation is available on the hub arcs and thus, their travel speed is faster than on the other links of the network. The total travel time when using the hub line takes into account the access and exit times that may exist when using the hub

line due to a change in mode of transportation or to waiting because of frequency or congestion related issues. The trade-off between the benefit of using a high-speed mode of transportation to efficiently travel and the added time for interacting with the hub line make the routing decisions more involved. Demand flow must be routed via either a path using a segment of the hub line or with a direct connection between origin and destination, depending on whichever route provides the smallest travel time.

Potential applications of hub line networks arise in public transportation planning, in particular in the design and modification of rapid transit systems and highway networks. A concrete example of an application of the HLLP is the modification of already established public transportation networks. Network planners usually face the problem of expanding an existing network in a metropolitan region so as to reduce the users' travel times by locating a rapid transit line, such as a subway, tram or light rail line, or an express bus lane with a fixed number of stations. Hub facilities correspond to central stations such as subway or bus stations, where a change of mode of transportation is usually available. Non-hub nodes represent urban districts, bus stops or taxi stations. Users will employ the hub line if there is a reduction in their travel time or they will keep using the shortest route on the existing network. For additional details and other applications of HLLP the reader is referred to Martins de Sá et al. (2013a).

One of the limiting aspects of the HLLP is that it is only applicable to situations in which the design of a hub network having exactly one line with a predetermined number of hubs is sought. In this paper we generalize the HLLP to the case in which the hub network is composed by more than one line. In particular, we introduce the *q-line hub location problem* (q-HLLP) which consists of locating a set of q lines that minimize the total travel time between O/D pairs, while satisfying a budget constraint on the total setup cost of the network associated with the location of hub nodes and hub arcs. As in the HLLP, we assume that O/D nodes can be assigned to more than one hub node, i.e. a multiple allocation pattern. However, instead of considering a predetermined number of hub nodes in a line, the q-HLLP considers as part of the decision process the determination of the number of hubs contained in each line, while respecting lower and upper limits on this number and the budget constraint for the total setup cost. In order to properly model the total travel time when using more than one hub line, a waiting time to transfer between lines needs to be taken into account. For instance, when transferring lines at a subway station, the average time spent walking between gates and waiting for the next subway train to pass, which depend on the size of the station and train frequency, could be significant. These transfer times might not compensate the reduction of travel time from using the subway, especially if transferring more than once, and thus users may continue traveling as before. These times make

the q-HLLP more general and thus, more challenging to formulate and solve.

Figure 3.1 illustrates two different 2-line hub networks that have the same topological structure, but actually represent different systems when transfer times are taken into account. For example, if transfer times are strictly positive, the travel time from hub node 1 to hub node 3 in hub network 1 is smaller than in hub network 2, since the latter implies a transfer from line 2 to line 1 at hub node 2.

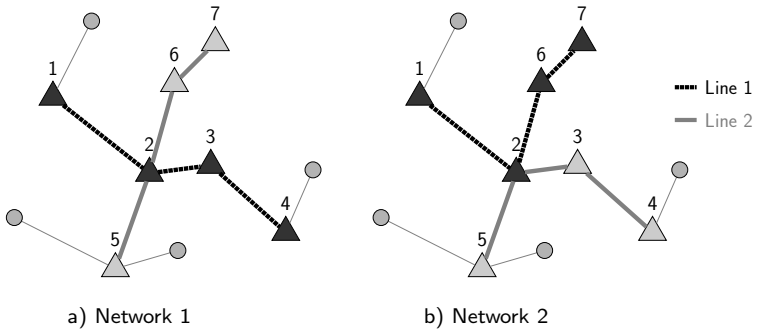


Figure 3.1: Ilustration of a 2-line hub network.

To the best of our knowledge, the design of transportation networks considering multiple lines has not been previously addressed from a hub location perspective. However, it has been considered in the context of extensive facility location and rapid transit systems design. An extensive facility is a facility considered to be too large for being represented as a single point when comparing its scale with its interaction environment (Mesa and Brian Boffey, 1996). A review of extensive facility location problems can be found in Mesa and Brian Boffey (1996), who also cover multiple path location problems.

In the context of the design of rapid transit systems, Bruno et al. (1998) address the design of a multi-modal rapid transit line. The problem consists of designing a bi-modal pedestrian-public network and considers a bi-objective function composed of the minimization of the construction costs and the minimization of the total weighted travel cost. The public network refers to a single rapid transit line whose extremities are given. It is assumed that the total travel cost associated with each O/D pair is equal to the minimum between the shortest path covered by means of the private system and the shortest path in the pedestrian-public network, which account for the total travel cost to transit between two nodes of the network and the costs associated to the waiting times to boarding and alighting in a station of the rapid transit line. García et al. (2006) address the design of a rapid transit system composed of multiple lines that maximizes the total weighted trip coverage by the system, where the extremities of each line are given. It is assumed that the total cost to satisfy the demand of each O/D pair by the system is equal to the sum of the total travel cost to move in the

transit vehicle and the costs associated with transferring from one line to another. In this case, a demand is covered by the system if the total travel cost is lower than the travel cost associated to a competitive private system.

Laporte et al. (2007) study a maximal coverage problem to select a set of potential stations that will be used to design a rapid transit system. They present a mathematical formulation for the problem of designing a multiple line network that maximizes the trip coverage in competition with a private mode taking into account a budget constraint. Marín (2007) presents an extension to the problem proposed by Laporte et al. (2007), where stations are not determined a priori and the number of lines is free but has an upper bound. The problem aims to maximize the public coverage (the main objective) and to minimize the routing cost. Marín and Jaramillo (2008) propose a long term planning model that aims to determine a network capacity expansion plan, i.e., to install additional lines or stations. Marín and Jaramillo (2009) present a Benders decomposition algorithm to solve the urban rapid transit design proposed by Marín (2007). It is important to note that all of these papers about rapid transit system design based on multiple line networks consider the maximization of demand coverage as the main component of the objective function. Consequently, the optimal solution using the proposed models may result in a multiple-line network that does not provide the minimum total weighted travel time, which is the objective of the q-HLLP.

As noted by Martins de Sá et al. (2013a), the design of hub line networks is a very challenging optimization problem, even for the case of a single line. The best Benders decomposition variant presented by the authors for the HLLP can consistently solve to optimality instances with up to 50 nodes and for some particular configurations of the parameters of the HLLP, it can solve instances with up to 100 nodes in one day of CPU time. In this paper, we present exact and heuristic algorithms for designing hub line networks with multiple lines. In particular, we present a mixed-integer programming (MIP) formulation for the q-HLLP which is used in a Benders decomposition algorithm to obtain optimal solutions for small instances and to provide bounds for larger instances. We also develop three different metaheuristics to provide feasible solutions to large instances: *i)* a variable neighborhood descent (VND), *ii)* a greedy randomized adaptive search procedure (GRASP) and, *iii)* an adaptive large neighborhood search (ALNS). In order to evaluate the efficiency and limitations of our algorithms, extensive computational experiments were performed on benchmark instances with up to 70 nodes and three lines.

The remainder of the paper is organized as follows. Section 3.2 provides a formal definition of the problem and introduces the MIP formulation. The Benders decomposition algorithm and the metaheuristics are presented in Sections 3.3 and 3.4, respectively. The results of the computational experiments are reported in Section 3.5.

Conclusions follow in Section 3.6.

3.2 Definition and Formulation of the Problem

Let $G = (N, A)$ be a complete digraph, where N is the set of nodes and A is the set of arcs. For each pair of nodes $i, j \in N$, we define w_{ij} as the amount of flow to be routed from the origin $i \in N$ to the destination $j \in N$. Let $t_{ij} \geq 0$ be the travel time defined as the shortest time required to travel from node i to j using one or more modes of transportation, other than the one associated with the hub lines, on the original network. Without loss of generality, t_{ij} also incorporates any average transfer time required when changing modes of transportation from i to j. Note that this definition of travel times ensures the t_{ij} values will satisfy the triangle inequality property. When a hub arc is used to connect hub nodes $i, j \in N$, the travel time is computed as $\alpha_{ij} t_{ij}$, where α_{ij} is a reduction factor that models the use of a faster transport technology to connect i and j. Let also $\tilde{t}^a_k \geq 0$ and $\tilde{t}^e_m \geq 0$ denote the access and exit times to enter a hub line at node k and leave a hub line at node m, respectively. The access time $\tilde{t}^a_k$ represents both the time required to change the mode of transportation between an access and a hub arc and the average waiting time to access the first hub line on the O/D path at hub k. The exit time $\tilde{t}^e_m$ only represents the time needed to change the mode of transportation at the last hub node m on the last hub line used on the O/D path. Note that in this case, the waiting time associated with the access arc (if any) is already considered in t_{mj}. Let $\tilde{t}^s_k \geq 0$ denote the transfer time at hub node k, which represents the average time to change from one line to another at hub node k. Furthermore, for $k, m \in N$, let f_k and c_{km} denote the fixed setup cost to locate a hub node and a hub arc, respectively. Let $\underline{p}$ and $\overline{p}$ be the bounds on the minimum and maximum number of hub nodes on each line, respectively, and let B denote the available budget to design the hub network.

The q-HLLP consists of locating q hub lines, each of them containing between $\underline{p}$ and $\overline{p}$ hub nodes, while allocating every non-hub node to at least one hub in such a way that the weighted sum of the total travel time is minimized and the budget constraint on the total network design cost is satisfied. It is assumed that the demand (i.e. passengers) from origin $i \in N$ to destination $j \in N$ will use the fastest possible route on the solution network. That is, w_{ij} will travel either directly from i to j using a shortest path on the original network resulting in a travel time of t_{ij}, or it will use a combination of access arcs and hub arcs associated with one or more hub nodes, in which case the total travel time is equal to the sum of:

(i) the travel time from origin i to the first visited hub k,

(ii) the time to access a hub line at hub k,

(iii) the travel time between the first hub k and the last hub m connected through a set of hub arcs associated with one or more hub lines,

(iv) the transfer time for changing lines at one or more hub nodes,

(v) the exit time to leave the hub line at the last visited hub m,

(vi) the travel time from hub m to destination j.

Because of the triangle inequality property of t_{ij}, a solution network of the q-HLLP will route w_{ij} either with a direct connection between i and j or with a path containing at most two access arcs and at least one hub arc and one hub line. That is, travel times associated with direct connections will always be smaller than or equal to any O/D path containing one hub node and no hub arcs.

We next introduce a MIP formulation for the q-HLLP based on the formulation proposed by Martins de Sá et al. (2013a) for the single line case. However, the characterization of O/D paths becomes more involved given that we now have to consider the possibility of transfers between two lines at the hub nodes. We define binary location variables z_k^l, $k \in N$, equal to 1 if and only if a hub is located at node k and is part of line l. We introduce binary hub arc variables y_{km}^l, $l = 1, \ldots, q$ and $(k, m) \in A$, $k < m$, equal to 1 if and only if a hub arc is located between hubs k and m and is part of line l, enabling flows to be routed in both directions. We also define five sets of continuous routing variables to model various structures of O/D paths arising in the q-HLLP. In particular, we introduce variables $a_{ijk} \geq 0$ and $b_{ijm} \geq 0$ equal to the fraction of the demand w_{ij} that enters and exits the q-line hub network through hubs $k \in N$ and $m \in N$, respectively, while variables $x_{ijkm}^l \geq 0$ denote the percentage of the demand w_{ij} routed on hub arc $(k, m) \in A$ of line l. We define variables τ_{ijk} equal to the fraction of demand w_{ij} changing hub lines at hub k, while variables e_{ij}, $i, j \in N$ are equal to the fraction of flow w_{ij} sent directly from i to j. To simplify the presentation, we assume $i, j, k, m \in N$, $i \neq j$, and $l = 1, \ldots, q$ henceforth. The q-HLLP can then be formulated as:

$$\text{minimize} \sum_i \sum_j w_{ij} \left[\sum_k (t_{ik} + \tilde{t}_k^a) a_{ijk} + \sum_l \sum_k \sum_{m:m \neq k} \alpha_{km} t_{km} x_{ijkm}^l + \sum_k \tilde{t}_k^s \tau_{ijk} + \sum_m (t_{mj} + \tilde{t}_m^e) b_{ijm} + t_{ij} e_{ij} \right] \tag{3.1}$$

$$\text{subject to} \quad \underline{p} \leq \sum_k z_k^l \leq \overline{p} \qquad \forall l \tag{3.2}$$

$$\sum_{k}\sum_{m:m>k} y^l_{km} = \sum_{k} z^l_k - 1 \qquad \forall l \tag{3.3}$$

$$\sum_{k}\sum_{m:m>k} y^l_{km} + \sum_{m:m<k} y^l_{mk} \leq 2z^l_k \qquad \forall k, l \tag{3.4}$$

$$\sum_{k\in S}\sum_{m\in S:m>k} y^l_{km} \leq \sum_{k\in S\setminus\{s\}} z^l_k \qquad \forall l, \forall S \subseteq N, s \in S \tag{3.5}$$

$$\sum_{l}\sum_{m:m>k} c^l_{km} y^l_{km} + \sum_{l}\sum_{k} f^l_k z^l_k \leq B \tag{3.6}$$

$$\sum_{k} a_{ijk} + e_{ij} = 1 \qquad \forall i, j \tag{3.7}$$

$$\sum_{m} b_{ijm} + e_{ij} = 1 \qquad \forall i, j \tag{3.8}$$

$$a_{ijk} + \sum_{l}\sum_{\substack{m\\ m\neq k}} x^l_{ijmk} = b_{ijk} + \sum_{l}\sum_{\substack{m\\ m\neq k}} x^l_{ijkm} \qquad \forall i, j, k \tag{3.9}$$

$$a_{ijk} \leq \sum_{l} z^l_k \qquad \forall i, j, k \tag{3.10}$$

$$b_{ijm} \leq \sum_{l} z^l_m \qquad \forall i, j, m \tag{3.11}$$

$$x^l_{ijkm} + x^l_{ijmk} \leq y^l_{km} \qquad \forall l, i, j, k < m \tag{3.12}$$

$$\sum_{m} x^l_{ijkm} - \sum_{m} x^l_{ijmk} \leq \tau_{ijk} + a_{ijk} \qquad \forall l, i, j, k \tag{3.13}$$

$$x^l_{ijkm}, e_{ij}, a_{ijk}, b_{ijm}, \tau_{ijk} \geq 0 \qquad \forall i, j, k, m \tag{3.14}$$

$$y^l_{km}, z^l_k \qquad \forall k, m, k < m. \tag{3.15}$$

The objective function (3.1) minimizes the total weighted travel time. Constraints (3.2) guarantee that at least $\underline{p}$ and at most $\overline{p}$ hubs are located on each line l. Constraints (3.3) guarantee that the number of hub arcs on each line is equal to the number of hubs minus one. Constraints (3.4) enforce a line topology for every l by allowing each hub to be connected to at most two others hubs. Constraints (3.5) are subtour elimination constraints which guarantee that each hub line does not contain cycles. The budget constraint (3.6) limits the total network construction cost. Constraints (3.7)-(3.9) are flow conservation constraints which ensure that all flow from i to j leaves node i, arrives at node j, and is properly accounted for whenever a hub k is used. Constraints (3.10) and (3.11) guarantee that the demand from i to j can only access or leave the hub lines through installed hubs. Constraints (3.12) ensure that only installed hub arcs can be used. Constraints (3.13) force the variables τ_{ijk} to take value 1 if the flow from i to j changes lines at node k. Finally, constraints (3.14)-(3.15) are the standard non-negativity and integrality constraints.

3.3 An Exact Algorithm

Benders decomposition is a partitioning procedure for solving mixed-integer linear and non-linear programs (Benders, 1962). The main idea is decompose the original problem into two simpler problems: an integer *master problem* (MP) and a linear *subproblem* (SP). In this section, we introduce a Benders reformulation of the q-HLLP, based on the formulation (3.1)-(3.15). We then describe a Benders decomposition algorithm to solve the reformulation. Martins de Sá et al. (2013a) present a comparison of several Benders decomposition variants for the single line problem based on a similar path-based MIP formulation. In particular, a multiple-cut strategy that adds a cut for each O/D pair showed better performance as compared to the single-cut variant in which only one cut is added per iteration. This version was embedded in a Benders-branch-and-cut scheme (BBC) in which Benders cuts are added within the branch-and-cut tree for every potential incumbent solution as well as at the root node, resulting in a single Benders iteration. Given that this BBC strategy provided the best results, we also adopt it for the q-HLLP.

By fixing the variables $z = z_h$ and $y = y_h$, we have the following primal linear SPs, one for each pair i, j:

$$\text{minimize } w_{ij}\left[\sum_k (t_{ik}+\tilde{t}^a_k)a_{ijk} + \sum_l\sum_k\sum_{m:m\neq k}\alpha_{km}t_{km}x^l_{ijkm} + \sum_k \tilde{t}^s_k\tau_{ijk} + \sum_m(t_{mj}+\tilde{t}^e_m)b_{ijm} + t_{ij}e_{ij}\right] \tag{3.16}$$

$$\text{subject to } \sum_k a_{ijk} + e_{ij} = 1 \tag{3.17}$$

$$\sum_m b_{ijm} + e_{ij} = 1 \tag{3.18}$$

$$a_{ijk} + \sum_l\sum_{\substack{m\\m\neq k}} x^l_{ijmk} - b_{ijk} - \sum_l\sum_{\substack{m\\m\neq k}} x^l_{ijkm} = 0 \qquad \forall k \tag{3.19}$$

$$-a_{ijk} \geq -\sum_l z^h_{kl} \qquad \forall k \tag{3.20}$$

$$-b_{ijm} \geq -\sum_l z^h_{ml} \qquad \forall m \tag{3.21}$$

$$-x^l_{ijkm} - x^l_{ijmk} \geq -y^h_{kml} \qquad \forall l, k<m \tag{3.22}$$

$$a_{ijk} + \sum_m x^l_{ijmk} - \sum_m x^l_{ijkm} + \tau_{ijk} \geq 0 \qquad \forall l,i,j,k \tag{3.23}$$

$$x^l_{ijkm}, e_{ij}, a_{ijk}, b_{ijm}, \tau_{ijk} \geq 0 \qquad \forall k,m. \tag{3.24}$$

After associating dual variables $\theta_{ij} \in \mathbb{R}$, $\Gamma_{ij} \in \mathbb{R}$, $\beta_{ijk} \in \mathbb{R}$, $u_{ijk} \geq 0$, $v_{ijk} \geq 0$, $\delta^l_{ijkm} \geq 0$ and $\pi^l_{ijk} \geq 0$ to constraints (3.17)-(3.23), respectively, the dual linear Benders SP for each i, j can be written as:

$$\begin{aligned}\text{maximize } \Gamma_{ij} + \theta_{ij} - \sum_k \sum_l z^h_{kl} u_{ijk} - \sum_m \sum_l z^h_{ml} v_{ijm} \\ - \sum_l \sum_k \sum_{m:m>k} y^h_{kml} \delta_{ijkml}\end{aligned} \tag{3.25}$$

$$\text{s.t. } \Gamma_{ij} + \theta_{ij} \leq w_{ij} t_{ij} \tag{3.26}$$

$$-\beta_{ijk} + \beta_{ijm} - \pi_{ijkl} + \pi_{ijml} - \delta_{ijkml} \leq \alpha_{km} w_{ij} t_{km} \quad \forall l, k, m : k < m \tag{3.27}$$

$$-\beta_{ijk} + \beta_{ijm} - \pi_{ijkl} + \pi_{ijml} - \delta_{ijmkl} \leq \alpha_{km} w_{ij} t_{km} \quad \forall l, k, m : k > m \tag{3.28}$$

$$\theta_{ij} + \beta_{ijk} - u_{ijk} + \sum_l \pi^l_{ijk} \leq w_{ij}(t_{ik} + \tilde{t}^a_k) \quad \forall k \tag{3.29}$$

$$\Gamma_{ij} - \beta_{ijm} - v_{ijm} \leq w_{ij}(t_{mj} + \tilde{t}^e_k) \quad \forall m \tag{3.30}$$

$$\sum_l \pi_{ijkl} \leq w_{ij} \tilde{t}^s_k \quad \forall k \tag{3.31}$$

$$u_{ijk}, v_{ijk} \geq 0 \quad \forall k \tag{3.32}$$

$$\delta_{ijkml} \geq 0 \quad \forall k, m. \tag{3.33}$$

It is worth noting that the primal SP is always feasible, since the direct connection between every pair i, j is a feasible solution. Therefore, the dual SP (3.25)-(3.32) is always feasible and bounded. Hence, an optimal solution can always be found at an extreme point of the polyhedron (3.26)-(3.32). As there are a finite number of such extreme points, it is possible to write the BC to be added to the MP, for each pair i, j, as:

$$\eta_{ij} \geq \Gamma^g_{ij} + \theta^g_{ij} - \sum_k \sum_l \left(u^g_{ijk} + v^g_{ijk}\right) z^l_k - \sum_k \sum_{\substack{m \\ k<m}} \sum_l \delta^g_{ijkml} y^l_{km} \quad \forall g \in \mathbb{G}_{ij}, \tag{3.34}$$

where η_{ij} is an under-estimator variable for the weighted travel time from origin i to destination j and $\mathbb{G}_{ij}$ is the set of extreme points of polyhedron (3.26)-(3.32) associated with the pair i, j.

The binary variables and their respective constraints, the η_{ij} variables, and the BC compose the Benders MP, which can be written as:

$$\text{maximize } \sum_i \sum_j \eta_{ij}$$

$$\text{subject to } (3.2) - (3.6)$$

$$\eta_{ij} \geq \Gamma^g_{ij} + \theta^g_{ij} - \sum_k \sum_l (u^g_{ijk} + v^g_{ijk}) z^l_k - \sum_k \sum_{\substack{m \\ k<m}} \sum_l \delta^g_{ijkml} y^l_{km} \qquad \forall g \in \mathbb{G}_{ij} \qquad (3.35)$$

$$\eta_{ij} \geq 0 \qquad \forall i, j \qquad (3.36)$$

$$y_{km}, z_k \in \{0, 1\} \qquad \forall k, m : k < m. \qquad (3.37)$$

The MP can be solved by means of a BBC framework in which SECs (3.5) are separated for every potential incumbent solution having a subtour. Subtours are identified by detecting the connected components of the current line, and violated SECs are then added by considering all nodes of each connected component. The Concorde callable library by Applegate et al. (2012) can be used to determine these connected components. Furthermore, the Benders cuts (3.35) are separated for every potential incumbent solution and for every fractional solution at the root node. The BBC algorithm is outlined in Algorithm 5, where $\mathcal{C}$ is the number of connected components in the current network.

Algorithm 5 Branch-and-cut framework for the BBC implementation

```
h = 1, G^h_ij = ∅
for all Incumbent solution or fractional solution at root node (z, y) do
    flag ← false
    for l = 1, ..., q do
        Find the connected components of l
        if C > 1 then
            Add SECs to MP and flag ← true
        end if
    end for
    if flag = false then
        z^h ← z and y^h ← y
        for all (i, j) ∈ N × N : i ≠ j do
            Solve the SP (3.25)-(3.32)
            G^{h+1}_ij ← G^h_ij ∪ {(Γ^h, θ^h, β^h, u^h, v^h, δ^h, π^h)}
            h ← h + 1
        end for
    end if
end for
```

3.4 Heuristic Algorithms

We next present three different metaheuristic algorithms to obtain feasible solutions for the q-HLLP, especially for large-size instances: *i)* a variable neighborhood decent (VND), *ii)* a greedy randomized adaptive search procedure (GRASP), and *iii)* an adaptive large neighborhood search (ALNS). Before describing these algorithms, we introduce a deterministic constructive heuristic to design an initial q-line hub network that satisfies the design constraints and an efficient combinatorial algorithm to solve the routing subproblem. These two algorithms play and important role in the efficiency and effectiveness of the proposed metaheuristics.

3.4.1 A Constructive Procedure

A constructive heuristic is an iterative procedure that aims to construct an initial feasible solution. The main idea is to start with an empty solution, and iteratively add new elements to the current solution until a feasible solution is obtained. We propose a deterministic constructive method based on an insertion criterion that takes into account the design constraints, i.e. the budget constraint on the total setup cost and the lower and upper limits on the number of hub nodes in each line, to obtain an initial feasible q-line hub network. It is an iterative greedy-type procedure in which at each iteration a cost-benefit ratio function is used to determine which hub node, and associated hub arc(s), should be added to the current solution.

Let s be the current solution and s'_{kl} the solution obtained by adding hub node k in line l to solution s. Also, let $C(s)$ and $f(s)$ be the setup cost and the objective function value associated with solution s, respectively. The cost-benefit of an insertion is measured as the ratio of the increase in the setup cost, $C(s'_{kl})-C(s)$, and the decrease in the objective function value, $f(s) - f(s'_{kl})$, associated with s and s'_{kl} when adding hub k in the best feasible position in line l, i.e. the one that minimizes this ratio and does not exceed the budget constraint. Therefore, the insertion cost of hub k in the best position of line l of solution s can be given by:

$$IC(k,l,s) = \frac{C(s'_{kl}) - C(s)}{f(s) - f(s'_{kl})}. \tag{3.38}$$

Let $\mathcal{F}(s)$ denote the set of feasible pairs (k,l) that are candidates to be inserted in the partial solution s, i.e., node k does not belong to line l, the current number of hubs in line l (denoted by p_l) is strictly lower than the upper bound $\overline{p}$, and the setup cost after adding k in the best position in line l does not exceed the budget. In the first part of the constructive heuristic, we iteratively add hubs to the network until we

have a feasible q-line hub network satisfying the design constraints. In the second part of the heuristic, we iteratively try to add additional hubs to some lines as long as we are able to keep improving the objective function or until there are no candidates left. The constructive procedure is outlined in Algorithm 6.

Algorithm 6 A deterministic constructive heuristic.

while $p_l < \underline{p}$ for any l and $\mathcal{F}(s) \neq \emptyset$ **do**
  $(k', l') \leftarrow \underset{\{(k,l) \in \mathcal{F}(s):\ p_l < \underline{p}\}}{\operatorname{argmin}} IC(k, l, s)$
  $s \leftarrow$ Insert (k', l') in s
end while
while $\mathcal{F}(s) \neq \emptyset$ **do**
  $(k', l') \leftarrow \underset{\{(k,l) \in \mathcal{F}(s)\}}{\operatorname{argmin}} IC(k, l, s)$
  $s \leftarrow$ Insert (k', l') in s
end while

If Algorithm 6 does not result in a feasible q-line hub network because of the budget constraint, then the insertion cost function (3.38) is replaced by

$$IC'(k, l, s) = C(s'_{kl}) - C(s),$$

which accounts only for the infrastructure cost and the algorithm is repeated.

3.4.2 Solving the Routing Subproblem

Once an initial q-line hub network is built, we still need to solve the routing subproblem to determine the optimal paths of O/D pairs that minimize the total weighted travel time. This subproblem can be seen as an extension of the all-pairs shortest path problem where the transfer time between hub lines and the existence of four types of arcs between nodes (access, hub and bridge arcs, and direct connections) needs to be taken into account. In Section 3.3 we showed how this problem can be formulated and solved as a linear program. However, it can actually be solved more efficiently by using an adaptation of the well-know Floyd-Warshall (FW) algorithm. This adaptation, it first computes the shortest path between all pairs of hub nodes, taking into account the transfer time between lines, and then computes the shortest path between all pairs of nodes using this information.

Let d_{ij}^k denote the shortest distance between the pair i, j using only nodes from $\{1, \ldots, k\}$ as intermediate nodes. In the case of the FW algorithm, the shortest path from i to j using all nodes in a set N, d_{ij}^n, can be found by means of the recursive equation $d_{ij}^k = \min\{d_{ij}^{k-1}, d_{ik}^{k-1} + d_{kj}^{k-1}\}$. That is, the shortest path from i to j using

nodes $\{1, \ldots, k\}$ is equal to the minimum between the shortest path that has k as an internal node and the one that does not have k as an intermediate node. Moreover, the length of the shortest path from i to j having k as an internal node is equal to the concatenation of the shortest path from i to k and the shortest path from k to j.

Our adaptation of the FW algorithm needs to take into account that in the shortest path from a hub i to a hub j: *a)* only hub nodes are allowed to be internal nodes, *b)* the shortest path length depends on whether an internal hub node k is a transfer node for this path, and *c)* hub arcs, with a reduced discount factor, can only be considered. Let H^1 and A^1 denote the set of open hub nodes and open hub arcs in all the lines, respectively, of the current solution network. For every $i, j \in N$ we denote as P^0_{ij} the shortest path from i to j where no node is used as intermediate node. If $(i, j) \in A^1$, then P^0_{ij} is equal to $\alpha_{ij} t_{ij}$, otherwise it is equal to ∞. The algorithm then adds one node $k \in H^1$ at a time as candidate internal node and finishes after $|H^1|$ iterations with the shortest paths between all hub nodes. The shortest path in the q-line hub network between every node pair $(i, j) \in N \times N$, denoted as SP_{ij}, is then computed by selecting the minimum of either the direct connection between i and j, or a path that uses a combination of access and bridge arcs and one or more hub lines. Access and bridge arcs can only be used as the first or last leg of an OD path. The modified FW algorithm is outlined in Algorithm 7.

Algorithm 7 A modified FW algorithm

$r = 0$
for all $k \in H^1$ **do**
  $r = r + 1$
  for all $(i, j) \in H^1 \times H^1$ **do**
    if a transfer is required between paths $[i, k]$ and $[k, j]$ **then**
      $P^r_{ij} = \min\{P^{r-1}_{ij}, P^{r-1}_{ik} + P^{r-1}_{kj} + \tilde{t}^s_k\}$
    else
      $P^r_{ij} = \min\{P^{r-1}_{ij}, P^{r-1}_{ik} + P^{r-1}_{kj}\}$
    end if
  end for
end for
for all $(i, j) \in N \times N$ **do**
  $SP_{ij} = \min\left\{t_{ij}, \min\left\{t_{ik} + \tilde{t}^a_k + P^r_{km} + \tilde{t}^e_m + t_{mj} : k, m \in H^1, k \neq m\right\}\right\}$
end for

3.4.3 A Variable Neighborhood Descent Method

Local search techniques are iterative procedures based on the improvement of a given solution s by finding a new solution s' in a neighborhood $\mathcal{N}(s)$ of s with a lower objective function value (for a minimization problem). This procedure stops when

no such solution can be found, in which case the current solution is called a local optimum. We now propose a local search procedure based on a VND method for the q-HLLP. The VND paradigm was proposed by Hansen and Mladenović (2001) and is based on a systematic local search in a set of m neighborhoods, $\mathcal{N}_1, \mathcal{N}_2, \ldots, \mathcal{N}_m$. The main idea is to perform a local search in a neighborhood $\mathcal{N}_1$ until a local optimum is found. After that, the search switches to neighborhoods $\mathcal{N}_2, \ldots, \mathcal{N}_m$, sequentially, until an improved solution is found. When an improvement is achieved, the search restarts using the neighborhood N_1. Our implementation of the VND algortihm considers the following three neighborhoods:

- $\mathcal{N}_1$: The set of feasible solutions that can be reached by either closing one hub or opening one hub. Closing a hub consists of removing the hub from its associated line and removing the arcs connecting it to the other hubs on that line. If the closed hub is not an extremity of the line, the two hubs that were connected to it will then be connected to each other. Opening a hub consists of inserting a hub in any position of a line;
- $\mathcal{N}_2$: The set of feasible solutions that can be reached by swapping two hubs in the network. Swapping consists of selecting two hubs in the same line or in different lines, and exchanging their position;
- $\mathcal{N}_3$: The set of feasible solutions that can be reached by simultaneously closing and opening one hub in the network. The new hub can be placed in any position of a line.

All these neighborhoods consider only feasible movements with respect to the design constraints. Moreover, since closing a hub usually results in a worse solution, the local search in $\mathcal{N}_1$ allows closing a hub if the cost of the new solution is within $\gamma\%$ of the best known solution value, where $\gamma \geq 0$ is a parameter of the algorithm.

3.4.4 A Greedy Randomized Adaptive Search Procedure

GRASP is a multi-start metaheuristic proposed by Feo and Resende (1989), in which each iteration consists of two phases: a constructive phase and a local search phase. In the constructive phase, an initial solutions is built, iteratively, by adding at random an element from a restricted candidate list (RCL) until a feasible solution is obtained. A local search phase is later used to improve the initial solution by exploring different neighborhoods but always considering feasible solutions. As before, let s denote the current partial solution.

In our GRASP algorithm, we use the cost-benefit ratio function (3.38) presented in Section 3.4.1 as the greedy function to construct the RCL. At each iteration, one element is randomly selected, according to a discrete uniform probability distribution, from the RCL to become a hub in a given line. The RCL is updated at each iteration of the constructive phase and contains the best candidate elements $\mathcal{F}(s)$ with respect to function (3.38). Let $c_{min} = \min\{IC(k,l,s) : (k,l) \in \mathcal{F}(s)\}$ and $c_{max} = \max\{IC(k,l,s) : (k,l) \in \mathcal{F}(s)\}$, then

$$RCL = \{(k,l) : IC(s,k,l) \leq c_{min} + \nu(c_{max} - c_{min})\},$$

where $0 \leq \nu \leq 1$ is a parameter that controls how greedy or randomized the heuristic is. For instance, when $\nu = 0$ the algorithm is completely greedy, while when $\nu = 1$ the algorithm is completely random. Given that the GRASP algorithm is sensitive to this parameter, we use a *reactive* GRASP method, proposed by Prais and Ribeiro (1999), to achieve a good trade-off between the quality and diversity of the constructed solutions. This method consists of choosing at random a value for ν from a set of potential values $\{\nu_1, \nu_2, \ldots, \nu_m\}$. Let z^* be the value of the incumbent solution and A_i^t be the average value of all solutions found so far using $\nu = \nu_i$ at iteration t. The probability of choosing a value ν_i is then equal to $w_i / \sum_{j=1}^{m} w_j$, where $w_i = z^*/A_i^t$.

Once an initial feasible solution is obtained, we use the VND algorithm presented in Section 3.4.3 in the local search phase to improve this solution.

3.4.5 An Adaptive Large Neighborhood Search Method

The ALNS method, proposed by Ropke and Pisinger (2006), is a metaheuristic based on performing a search in a large neighborhood by partially destroying the current solution and reconstructing it by applying some heuristic rules. Given a set of removal operators and a set of insertion operators, the destroy and repair phases consist of choosing at random a removal and an insertion operator, respectively. Let w_i be the weight associated to operator i. The probability of choosing operator i is given by $w_i / \sum_j w_j$. The adaptive part of the algorithm is given by the dynamic updating of these weights. The update of the weight of each operator is done according to the performance of each operator in a time interval (segment) by means of scores, where the score measures the contribution of the operator to the improvement in the objective function. The score of each operator is initially set to zero, in the beginning of the segment, and can be increased at each iteration by: *a)* σ_1 if the removal-insertion operation pair results in a new best solution, *b)* σ_2 if the removal-insertion operation pair results in a new solution worse than the minimum, but better than the current

one, and *c)* σ_3 if the removal-insertion operation pair results in a solution that is worse than the current one, but satisfies the acceptance criterion. In practice, $\sigma_1 \geq \sigma_2 \geq \sigma_3$. The weight associated to operator i can be updated through

$$w_i = w_i(1 - r) + r\frac{s_i}{o_i},$$

where s_i is the score associated to operator i, o_i is the number of times that operator i was used in the last segment, and r is the reaction factor parameter that indicates how fast the weights change according to the last segment's performance.

In our ALNS algorithm, we consider the following removal operators:

(a) **R**andom removal operator. This operator consists of removing m hubs chosen randomly according to a discrete uniform probability distribution.

(b) **M**inimum deterioration removal operator. This operator consists of removing, iteratively, m hub nodes that yield the smallest increase in the objective function value.

(c) **C**ost-benefit removal operator. This operator is similar to the previous one, but it also takes into account the infrastructure cost associated to each node. It consists of removing m hub nodes with the worst (i.e., largest) cost-benefit ratio. The idea behind considering the infrastructure cost in the destroy phase aims to make the repair phase more flexible in relation to the budget.

(d) **P**roximity removal operator. The idea is to remove m hub nodes that are geographically close to each other. The first hub node is chosen at random and after that we remove, iteratively, $m - 1$ hubs that are the closest to the set of hubs already removed. The distance between a hub node and a set of nodes is measured as the distance between this hub and the closest node of the set.

(e) **C**ost-benefit relatedness operator. This operator consists in removing a segment of a line with m hub nodes. The first hub node is chosen randomly and the other $m-1$ are chosen among the hub nodes that were connected to the previously removed hub and that have the worst cost-benefit ratio. If the chosen line has fewer than m hub nodes, the other hub nodes are removed by selecting a new initial hub node randomly.

Since most of these operators are deterministic, we introduce some randomness in the algorithm by including a parameter $\rho > 1$ in the removal operators (b)-(e), as suggested by Ropke and Pisinger (2006). Let H be a sorted set of hub nodes ordered according to the removal operator's main criterion and let $\mathcal{R}$ be a random number from

the interval $[0, 1)$. The idea is to remove the $[\mathcal{R}^\rho|H|]-$th element of H, instead of always removing the first one. Observe that the parameter ρ controls how much randomness is added to these operators, where small values of ρ result in more randomness.

In order to reconstruct the network so as to obtain a feasible solution, we use the following two insertion operators:

(a) **S**avings insertion operator. Insert, iteratively, hub nodes that will result in the largest improvement in the objective function value.

(b) **C**ost-benefit insertion operator. Insert iteratively hubs with the lowest cost-benefit ratio (3.38).

Finally, a solution s' is accepted according to a simulated annealing acceptance criterion, i.e. we accept a new solution s' with probability $e^{f(s')-f(s))/T}$, where $T > 0$ is the temperature. The temperature starts with a value $T = T_0$ and decreases periodically at a cooling rate c.

3.5 Computational Experiments

We next present the results of extensive computational experiments performed to assess the behaviour of our exact and heuristic algorithms. In the first part of the experiments, we compare the Benders-branch-and-cut method with a general-purpose solver for instances with up to 40 nodes. The second part of the experiments focuses on the comparison of the proposed heuristic methods. We first apply the metaheuristics to solve the case of $q = 1$, i.e. the single-line HLLP introduced by Martins de Sá et al. (2013a). We then use the metaheuristics to solve the q-HLLP. In the third part of the experiments, we analyze how the discount factor, the transfer time and the budget parameters affect the topology of q-line hub networks. All experiments were performed on a 1260 Xeon Westmere 2.66 GHz computer with 24 GB of memory and running Linux. The BBC was coded in C++ using the Concert Technology of CPLEX 12.5 to solve the Benders MPs and SPs.

We have used two standard benchmark instances for hub location in our computational experiments: the Australian Post (AP) data set introduced by Ernst and Krishnamoorthy (1996) and the CAB data set of the US Civil Aeronautics Board first used in O'Kelly (1987). Both data sets provide an OD demand matrix and coordinates for each OD node. The AP instances also provide a fixed cost associated to the installation of each potential hub node. Since CAB instances do not provide these data, we use the fixed setup cost generated by Camargo et al. (2008). Furthermore, the cost for opening a hub arc (k, m), which is not provided in the two data sets, is assumed to be

equal to 2000 t_{km} for AP instances and to 500 t_{km} for CAB instances. As in other hub location problems, the travel time between a pair of nodes is assumed to be equal to the Euclidean distance between them.

Tests were carried out by considering the discount factor values of $\alpha_{ij} = \alpha$ in $\{0.2, 0.4, 0.6, 0.8\}$. Moreover, in all experiments we consider that the access, exit and transfer times are the same for every node $k \in N$, and controlled by means of the parameter $\vartheta = \{0, 0.1, 0.25\}$, which is used to set the transfer time as $\tilde{t}^s_k = \vartheta \bar{t}_{ij}$, where $\bar{t}_{ij}$ is the average travel time between all node pairs computed as $\bar{t}_{ij} = \sum_i \sum_j t_{ij}/n(n-1)$. The access and exit times are assumed to be equal to 90% and 10% of the transfer time, respectively. Furthermore, we consider the budget constraint to be proportional to the total design cost of the network in which each line has $p = \lfloor(\overline{p} + \underline{p})/2\rfloor$ hubs, where the hub node and hub arc setup costs are assumed to be equal to the average hub node setup cost $\bar{f}$ and the average hub arc installation cost $\bar{c}$, respectively. A proportionality factor $\beta = \{0.6, 1.0, 2.0\}$ is used to vary the budget as $B = \beta q \left(p\bar{f} + (p-1)\bar{c}\right)$, where $p\bar{f}$ is the average cost to install the hubs and $(p-1)\bar{c}$ is the average cost to install the hub arcs.

3.5.1 Benders-branch-and-cut Performance

Tables 3.1 and 3.2 present the computational results comparing the BBC and CPLEX for instances with $n = \{10, 20\}$, $\vartheta = \{0.0,\ 0.1,\ 0.2\}$ and $\beta = \{0.6, 1.0, 2.0\}$ for both the AP and CAB data sets. Each line of these tables presents the average CPU time, the average optimality gap and the number of solved instances aggregated by the ϑ and β parameters. Table 3.3 presents the computational results for instances with $n = \{25, 40\}$, $\vartheta = 0.1$ and $\beta = 1.0$ for AP instances. We consider instances with $q = \{2, 3\}$, $\underline{p} = 2$ and $\overline{p} = 6$ for $n = 10$, and $\underline{p} = 3$ and $\overline{p} = 8$ for the other values of n, and $\alpha = \{0.2, 0.4, 0.6, 0.8\}$. In all the experiments, we use a CPU time limit of 24 hours.

According to Tables 3.1 and 3.2, BBC solves the problem slightly faster on average. Furthermore, CPLEX presents a better average optimality gap for AP instances with up to 20 nodes, while BBC presents a better average optimality gap for CAB instances. It is important to mention that for instances with up to 20 nodes, CPLEX cannot obtain a feasible solution for three CAB instances and BBC cannot obtain a feasible solution for one CAB instance. For the case of larger instances with 25 and 40 nodes, Table 3.3 shows that only one of these instances is solved to optimality by both CPLEX and BBC. However, BBC is able to provide a lower bound for all considered instances whereas CPLEX fails to even solve LP relaxation after 24 hours for the 40 node instances. In the case of 25-node instances, the BBC presents better optimality gap than CPLEX. These

Table 3.1: Comparison of the Benders-branch-and-cut algorithm with CPLEX for AP instances with 10 and 20 nodes.

n	q	α	CPLEX			BBC		
			Time	gap	# Opt	Time	gap	# Opt
10	2	0.2	249.02	0.00%	9	241.17	0.00%	9
		0.4	59.78	0.00%	9	29.92	0.00%	9
		0.6	21.15	0.00%	9	5.28	0.00%	9
		0.8	6.49	0.00%	9	1.61	0.00%	9
	3	0.2	13173.96	0.00%	9	36157.91	0.63%	7
		0.4	2337.32	0.00%	9	11428.68	0.06%	8
		0.6	387.87	0.00%	9	393.56	0.00%	9
		0.8	111.25	0.00%	9	51.82	0.00%	9
20	2	0.2	86400.00	9.73%	0	80391.28	7.90%	1
		0.4	78006.76	2.39%	4	51139.57	1.30%	4
		0.6	34377.73	0.17%	7	20953.81	0.14%	7
		0.8	2731.06	0.00%	9	376.92	0.00%	9
	3	0.2	86400.00	27.51%	0	86400.00	28.17%	0
		0.4	86400.00	8.94%	0	86400.00	14.49%	0
		0.6	80356.67	1.68%	1	86400.00	6.56%	0
		0.8	46651.40	0.02%	5	28551.60	0.03%	7
Avg/Sum			32354.40	3.15%	98	30557.70	3.70%	97

Table 3.2: Comparison of the Benders-branch-and-cut algorithm with CPLEX for CAB instances with 10 and 20 nodes.

n	q	α	CPLEX			BBC		
			Time[s]	gap	# Opt	Time[s]	gap	# Opt
10	2	0.2	561.83	0.00%	9	3004.45	0.00%	9
		0.4	90.43	0.00%	9	81.72	0.00%	9
		0.6	25.12	0.00%	9	15.20	0.00%	9
		0.8	11.33	0.00%	9	2.85	0.00%	9
	3	0.2	26284.55	0.21%	7	39997.25	1.60%	5
		0.4	8189.68	0.00%	9	22574.43	0.12%	8
		0.6	596.70	0.00%	9	1123.05	0.00%	9
		0.8	134.66	0.00%	9	64.80	0.00%	9
20	2	0.2	86400.00	9.59%	0	86400.00	12.37%	0
		0.4	75938.97	2.36%	2	60226.78	7.84%	3
		0.6	53972.09	0.32%	5	33287.15	0.48%	7
		0.8	15610.31	0.00%	8	12978.63	0.01%	8
	3	0.2	86400.00	43.48%	0	86400.00	29.26%	0
		0.4	86400.00	30.41%	0	86400.00	18.20%	0
		0.6	86400.00	2.32%	0	86400.00	6.42%	0
		0.8	73179.74	0.16%	2	71354.21	0.80%	2
Avg/Sum			37512.21	5.55%	87	36894.41	4.82%	87

Table 3.3: Comparison of the Benders-branch-and-cut algorithm with CPLEX for instances with 25 and 40 nodes.

n	L	α	CPLEX gap	BBC gap
25	2	0.2	31.97%	8.14%
		0.4	5.18%	2.28%
		0.6	1.27%	0.15%
		0.8	0.00%	0.00%
	3	0.2	–	–
		0.4	–	13.09%
		0.6	12.07%	3.02%
		0.8	0.43%	0.34%
40	2	0.2	*	–
		0.4	*	15.89%
		0.6	*	0.93%
		0.8	*	0.10%
	3	0.2	*	–
		0.4	*	–
		0.6	*	–
		0.8	*	0.56%
Avg			8.49%	3.71%

– No feasible solution was found.
* No feasible solution neither a LB was found.

results provide an indication of the increased complexity of solving to optimality the q-HLLP as compared to the single line case. We recall that the Benders decomposition presented in Martins de Sá et al. (2013a) for the HLLP can solve instances to optimality with up to 100 nodes. Our BBC algorithm for the q-HLLP can still be used for larger instances to obtain lower bounds on the optimal solution value to evaluate the performance of the proposed heuristic algorithms.

3.5.2 A Comparison of Metaheuristics

We now present a comparison between the three metaheuristic algorithms presented in Section 3.4. After some preliminary computational experiments, we set the values of the parameters used in the algorithms as:

VND: The value of γ is equal to 2, which allows the selected solution from $\mathcal{N}_1$ to be at most 2% worse than the current one.

GRASP: The value for ν is chosen at random from $\{0.01, 0.02, \ldots, 0.1\}$ and the weights w_j, for $j = 1, \ldots, 10$, associated with each of these values are updated every 10 iterations as described in Section 3.4.4. Furthermore, the termination criteria

used are the maximum number of iterations equal to 40 or a time limit of 3,600 seconds.

ALNS: The values for the σ_i parameters are $\sigma_1 = 10$, $\sigma_2 = 7$, $\sigma_3 = 5$, $\rho = 3$, and $r = 0.3$. Furthermore, we set the initial temperature $T_0 = 10{,}000$ and the cooling rate $c = 0.990822$, so as to obtain a maximum number of iterations of 1000. Finally, the parameter m was chosen randomly considering values between $[1, (\overline{p} + \underline{p})/2]$, where the weight associated with each value is updated adaptively. Finally, the termination criteria used are the temperature T becoming smaller than the threshold $\epsilon = 1$ or a time limit of 3,600 seconds.

We first focus on analyzing the performance of our algorithms for the single line case, i.e. the HLLP. We use the Benders-branch-and-cut algorithm presented in Martins de Sá et al. (2013a) to provide the optimal solutions or lower bounds for the considered instances. Since the HLLP has a fixed value for the number of hubs on the line and no budget constraint, the tests were performed considering $\underline{p} = \overline{p} \in \{5, 8\}$, $f_i^1 = 0$, for $i \in N$ and $c_{km}^1 = 0$, for $k, m \in N$. Furthermore, the tests consider the following parameter values: $\alpha = \{0.2, 0.5, 0.8\}$, $\vartheta = \{0.0,\ 0.1,\ 0.2\}$ and $n = \{10, 20, 25, 40, 50, 75, 100\}$. Table 3.4 presents the average gap and number of instances solved for each metaheuristic. For comparative purposes, we also present the results from the constructive heuristic from Section 3.4.1. Each row aggregates the results by α and ϑ parameters.

Table 3.4: Comparison between the proposed heuristics for the HLLP.

n	p	BBC	Constructive		VND		GRASP		ALNS	
		#Opt	gap	#Opt	gap	#Opt	gap	#Opt	gap	#Opt
10	5	9/9	2.22%	3/9	0.06%	8/9	0.00%	9/9	0.00%	9/9
10	8	9/9	0.08%	6/9	0.05%	7/9	0.00%	9/9	0.00%	9/9
20	5	9/9	0.81%	2/9	0.18%	5/9	0.00%	9/9	0.00%	9/9
20	8	9/9	0.99%	0/9	0.23%	4/9	0.00%	9/9	0.00%	9/9
25	5	9/9	1.53%	0/9	0.14%	7/9	0.00%	9/9	0.00%	9/9
25	8	9/9	2.00%	0/9	0.67%	2/9	0.00%	9/9	0.03%	5/9
40	5	9/9	1.37%	0/9	0.00%	8/9	0.00%	9/9	0.00%	9/9
40	8	9/9	1.10%	0/9	0.01%	7/9	0.00%	9/9	0.01%	8/9
50	5	9/9	1.82%	0/9	0.00%	9/9	0.00%	9/9	0.00%	9/9
50	8	9/9	1.98%	0/9	0.37%	4/9	0.00%	9/9	0.01%	8/9
75	5	9/9	1.43%	0/9	0.05%	5/9	0.00%	9/9	0.00%	8/9
75	8	4/9	1.80%	0/9	0.31%	0/9	0.24%	3/9	0.24%	2/9
100	5	7/9	2.05%	0/9	0.23%	8/9	0.23%	8/9	0.23%	8/9
100	8	1/9	3.70%	0/9	2.11%	1/9	1.82%	1/9	1.88%	0/9
		111/126	1.63%	11/126	0.31%	75/126	0.16%	111/126	0.17%	102/126

According to Table 3.4, the GRASP and ALNS algorithms present the best performance. The GRASP is able to obtain the optimal solution in 111 out of 114 instances in which the optimal solution is known, whereas the ALNS obtains 102 optimal solutions.

Furthermore, the solutions provided by both metaheuristics present a small gap with the optimal solution (or best lower bound) of less than 0.17% deviation, confirming that these heuristics can successfully solve the HLLP.

The comparison of the proposed heuristics to solve the q-HLLP is presented in Tables 3.5, 3.6 and 3.7. The results presented in these tables use the same set of instances as in Tables 3.1, 3.2 and 3.3, respectively. Table 3.7 also presents results for instances with $n = 50$ and $n = 70$. In order to better compare the solutions produced by each algorithm, we take into account two different performance measures: gap is the deviation between the optimal solution (or the best lower bound found by CPLEX and BBC) and the best solution found in any algorithm; $\%Dev$ is the percent deviation between the solution found by each algorithm and the best solution found in any algorithm. To compute the average $\%Dev$ for BBC and CPLEX, we consider only the instances for which the method obtain a feasible solution.

Table 3.5: Comparison of exact and heuristic algorithms for the q-HLLP for AP instances with 10 and 20 nodes.

n	q	α	gap	# Opt	CPLEX		BBC		Const		VND		GRASP		ALNS	
					$\%Dev$	#	$\%Dev$	#	$\%Dev$	#	$\%Dev$	#	$\%Dev$	#	$\%Dev$	#
10	2	0.2	0.00%	9	0.00%	9	0.00%	9	7.37%	0	1.69%	4	0.33%	8	0.33%	8
		0.4	0.00%	9	0.00%	9	0.00%	9	3.84%	0	1.42%	2	0.20%	8	0.41%	7
		0.6	0.00%	9	0.00%	9	0.00%	9	1.39%	1	0.84%	1	0.00%	9	0.03%	7
		0.8	0.00%	9	0.00%	9	0.00%	9	0.40%	3	0.26%	3	0.00%	8	0.01%	7
	3	0.2	0.00%	9	0.05%	8	0.00%	9	5.47%	0	2.51%	0	1.46%	5	0.49%	6
		0.4	0.00%	9	0.00%	9	0.03%	8	3.24%	0	1.50%	1	0.17%	4	0.16%	6
		0.6	0.00%	9	0.00%	9	0.00%	9	1.12%	0	0.71%	0	0.09%	7	0.11%	7
		0.8	0.00%	9	0.00%	9	0.00%	9	0.54%	1	0.29%	1	0.01%	8	0.00%	8
20	2	0.2	6.72%	1	2.72%	1	3.21%	1	4.41%	0	3.00%	0	0.00%	1	0.52%	0
		0.4	1.30%	4	0.70%	4	0.22%	4	2.77%	1	1.72%	1	0.01%	4	0.16%	4
		0.6	0.09%	7	0.00%	7	0.06%	7	0.98%	1	0.34%	3	0.00%	6	0.03%	6
		0.8	0.00%	9	0.00%	9	0.00%	9	0.31%	0	0.24%	0	0.04%	4	0.03%	7
	3	0.2	24.18%	0	17.47%	0	18.25%	0	5.35%	0	3.74%	0	0.15%	0	0.12%	0
		0.4	8.90%	0	3.78%	0	9.62%	0	1.96%	0	1.24%	0	0.20%	0	0.00%	0
		0.6	1.50%	1	0.46%	1	5.42%	0	0.62%	0	0.30%	0	0.10%	1	0.00%	1
		0.8	0.01%	7	0.00%	7	0.02%	7	0.47%	1	0.30%	1	0.05%	1	0.05%	4
Avg			2.67%	101	1.57%	101	2.30%	98	2.52%	8	1.26%	17	0.18%	74	0.15%	78

Once more, from Tables 3.5, 3.6 and 3.7 we observe that the GRASP and ALNS algorithms present better performance than the constructive heuristic and the VND procedure. The average gap between the best solution found and the lower bound is 1.38% for CAB instances, 2.67% for AP instances with up to 20 nodes and 6.38% for the larger AP instances. Furthermore, for instances where we do not know the optimal solution, the solution of GRASP and ALNS is, on average, better than the best incumbent found by both CPLEX and BBC. By comparing the ALNS and GRASP algorithms, GRASP presents a better performance than ALNS with the CAB instances by solving more instances and presenting a better gap. On the other hand, ALNS presents a better performance than GRASP for the experiments using the AP instances.

Table 3.6: Comparison between heuristic algorithms to solve the q-HLLP for CAB instances with 10 and 20 nodes.

n	q	α	gap	# Opt	*C*PLEX		*B*BC		Const		VND		GRASP		ALNS	
					%Dev	#	*%Dev*	#	*%Dev*	#	*%Dev*	#	*%Dev*	#	*%Dev*	#
10	2	0.2	0.00%	9	0.00%	9	0.00%	9	5.89%	0	1.54%	1	0.00%	9	0.76%	2
		0.4	0.00%	9	0.00%	9	0.00%	9	2.84%	2	1.67%	3	0.00%	9	0.83%	4
		0.6	0.00%	9	0.00%	9	0.00%	9	1.04%	3	0.66%	3	0.00%	9	0.12%	6
		0.8	0.00%	9	0.00%	9	0.00%	9	0.56%	1	0.32%	2	0.00%	9	0.06%	4
	3	0.2	0.21%	7	0.00%	7	0.37%	5	7.22%	0	2.48%	1	0.11%	6	1.37%	1
		0.4	0.00%	9	0.00%	9	0.03%	8	2.36%	0	1.47%	0	0.01%	7	0.07%	7
		0.6	0.00%	9	0.00%	9	0.00%	9	1.28%	0	0.77%	0	0.04%	5	0.17%	5
		0.8	0.00%	9	0.00%	9	0.00%	9	0.33%	0	0.24%	0	0.00%	8	0.08%	2
20	2	0.2	4.85%	0	3.49%	0	7.97%	0	6.35%	0	3.38%	0	0.00%	0	0.38%	0
		0.4	1.12%	3	0.54%	3	6.83%	3	4.09%	0	2.34%	0	0.00%	3	0.48%	2
		0.6	0.11%	7	0.06%	5	0.37%	7	2.48%	0	0.75%	0	0.07%	4	0.19%	3
		0.8	0.00%	8	0.00%	8	0.00%	8	0.84%	0	0.34%	0	0.04%	3	0.07%	1
	3	0.2	9.90%	0	28.86%	0	21.40%	0	8.17%	0	4.09%	0	0.17%	0	0.50%	0
		0.4	4.43%	0	6.00%	0	8.91%	0	4.42%	0	2.07%	0	0.15%	0	0.29%	0
		0.6	1.27%	0	1.05%	0	5.08%	0	2.00%	0	1.08%	0	0.09%	0	0.11%	0
		0.8	0.13%	3	0.00%	3	0.61%	2	1.45%	0	0.55%	0	0.04%	1	0.08%	1
Average			1.38%	91	2.50%	89	3.22%	87	3.21%	6	1.49%	10	0.05%	73	0.35%	38

In particular for instances with 25 up to 70 nodes, ALNS provides in most cases the best known solution.

3.5.3 Analyzing Network Configurations

To analyze how the parameters α, ϑ and β affect the configuration of the hub line network, Figures 3.2, 3.3 and 3.4 present different network configurations obtained by varying these parameters, considering a network with 25 nodes and 3 lines having between 3 and 8 hubs each. We also report information about the designed system such as the percentage of flow that uses the line and the number of direct connections and access edges. Since we do not have an optimal solution for all of these instances, we are using the solution provided by the ALNS algorithm.

Figure 3.2 presents the system configuration when the economies of scale factor α is set to 0.2 and the budget factor β is set to 1.0, while we vary the value for ϑ.

According to Figure 3.2, these parameters have a substantial impact on the number of transfer hub nodes, i.e., nodes that are at the intersection of two or more lines. For small transfer times, the network has two transfer hub nodes, while for medium and large transfer times we have just one transfer hub node. Furthermore, the number of access edges increases as the transfer time increases.

Figure 3.3 presents the network configurations by fixing the transfer time parameter $\vartheta = 0.1$ and the budget parameter $\beta = 1.0$, and changing the value of α. In this case, the parameter has a great impact on the number of access edges, i.e., the number of access edges increases as the economies of scale factor increases.

Table 3.7: Comparison between the heuristic algorithms to solve the q-HLLP for AP instances with 25, 40, 50 and 70 nodes.

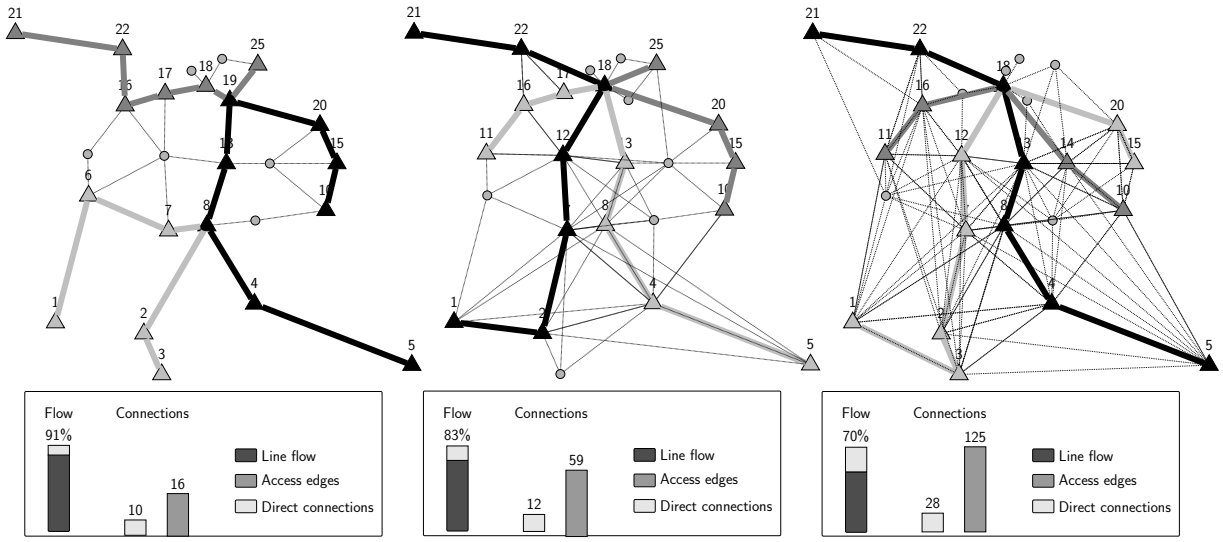
n	q	α	LB	gap	BBC	Constructive		VNS		GRASP		ALNS	
					%*Dev*	Time	%*Dev*	Time	%*Dev*	Time	%*Dev*	Time	%*Dev*
25	2	0.2	33051.4	6.21%	2.05%	3.26	12.07%	3.26	1.56%	589.61	0.51%	362.05	0.00%
		0.4	43909.3	1.54%	0.76%	2.13	6.75%	2.13	2.33%	624.66	0.00%	419.73	0.00%
		0.6	51980.5	0.15%	0.00%	4.71	2.10%	4.71	0.06%	650.12	0.00%	432.64	0.00%
		0.8	56801.4	0.00%	0.00%	2.35	0.48%	2.35	0.05%	514.64	0.00%	316.88	0.00%
	3	0.2	24812.2	15.94%	–	8.1	14.72%	8.1	1.10%	3601.67	0.00%	1021.41	0.07%
		0.4	38410.5	6.84%	6.70%	7.64	6.72%	7.64	0.70%	3610.91	0.69%	779.73	0.00%
		0.6	49259.1	1.88%	1.17%	13.12	3.15%	13.12	0.42%	2402.13	0.31%	806.55	0.00%
		0.8	56279.0	0.18%	0.15%	7.11	0.54%	7.11	0.00%	1676.89	0.00%	807.74	0.00%
40	2	0.2	37877.1	5.40%	–	9.03	8.15%	9.03	2.28%	2279.88	0.11%	1334.06	0.00%
		0.4	47308.4	2.27%	13.94%	8.23	4.37%	8.23	2.23%	2467.76	0.05%	1618.1	0.00%
		0.6	54070.9	0.65%	0.28%	25.23	2.33%	25.23	0.00%	2450.97	0.00%	1594.98	0.00%
		0.8	58453.9	0.09%	0.01%	9.16	0.59%	9.16	0.13%	2277.7	0.00%	1345.31	0.00%
	3	0.2	31643.6	10.82%	–	44.89	10.12%	44.89	4.35%	3669.51	1.20%	3007.86	0.00%
		0.4	43155.7	4.93%	–	30.07	5.91%	30.07	1.93%	3607.29	0.00%	3510.89	0.00%
		0.6	52108.3	1.68%	–	66.66	2.63%	66.66	0.14%	3604.67	0.09%	3602.27	0.00%
		0.8	57930.5	0.37%	0.19%	37.45	0.85%	37.45	0.04%	3642.73	0.06%	3028.83	0.00%
50	2	0.2	38667.5	5.55%	–	24.39	8.11%	24.39	0.23%	3627.87	0.00%	2752.6	0.00%
		0.4	47884.5	1.95%	16.77%	18.1	4.82%	18.1	0.88%	3614.51	0.01%	3034.72	0.00%
		0.6	54383.9	0.45%	8.32%	16.22	2.28%	16.22	0.31%	3660.83	0.00%	2898.02	0.02%
		0.8	58459.2	0.10%	0.36%	30.92	0.42%	30.92	0.03%	3639.07	0.03%	2646.5	0.00%
	3	0.2	32907.0	9.77%	–	50.94	10.30%	50.94	0.82%	3731.04	2.14%	3605.93	0.00%
		0.4	44047.6	4.03%	–	71.86	6.24%	71.86	1.89%	3656	1.16%	3602.83	0.00%
		0.6	52581.5	1.33%	–	148.87	2.53%	148.87	0.18%	3738.08	0.15%	3601.6	0.00%
		0.8	58035.7	0.28%	–	61.79	0.48%	61.79	0.19%	3657.11	0.00%	3606.98	0.00%
70	2	0.2	35689.6	11.61%	–	45.14	7.66%	45.14	3.48%	3734.38	1.53%	3605.26	0.00%
		0.4	46060.8	5.74%	–	63.43	4.00%	63.43	0.17%	3716.78	1.05%	3611.33	0.00%
		0.6	54611.2	0.96%	–	97.61	2.30%	97.61	0.08%	3631.24	0.39%	3605.38	0.00%
		0.8	58899.4	0.14%	1.96%	68.48	0.33%	68.48	0.02%	3736.81	0.01%	3609.03	0.00%
	3	0.2	14925.0	59.59%	–	133.61	10.54%	133.61	1.04%	3757.14	0.71%	3621.76	0.00%
		0.4	30860.7	33.80%	–	214.07	5.85%	214.07	0.56%	4399.35	2.71%	3607.46	0.00%
		0.6	48760.8	9.34%	–	464.78	2.32%	464.78	0.43%	3720.12	0.76%	3634.07	0.00%
		0.8	58278.7	0.72%	–	207.59	0.45%	207.59	0.02%	3696.67	0.15%	3602.22	0.00%
Average				6.38%	3.51%	62.40	4.83%	62.40	0.89%	3043.38	0.44%	2457.34	0.00%

– No feasible solution was found.

Figure 3.4 presents the network configuration when we fix the waiting time parameter $\vartheta = 0.1$ and the economies of scale factor $\alpha = 0.2$ and just vary the budget parameter. The figures confirm that this parameter has a great impact on the network topology. The number of installed hubs and the amount of flow using the multiple hub line network is directly proportional to the budget available to design the network. Furthermore, the number of direct connections decreases and the number of access edges increases as the budget increases.

3.6 Conclusion

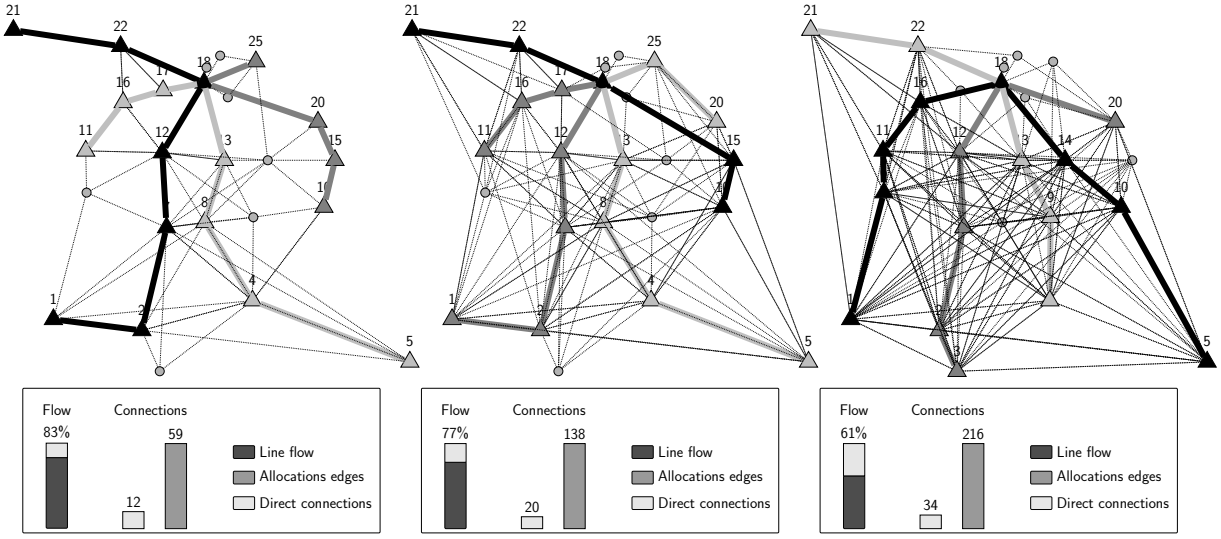
In this paper we have studied the q-line hub location problem in which the hub-level network is now composed by a set of q lines. A mathematical formulation and an algorithm based on Benders decomposition were proposed. Although the BBC algo-



a) $\vartheta = 0.0$ b) $\vartheta = 0.1$ c) $\vartheta = 0.2$

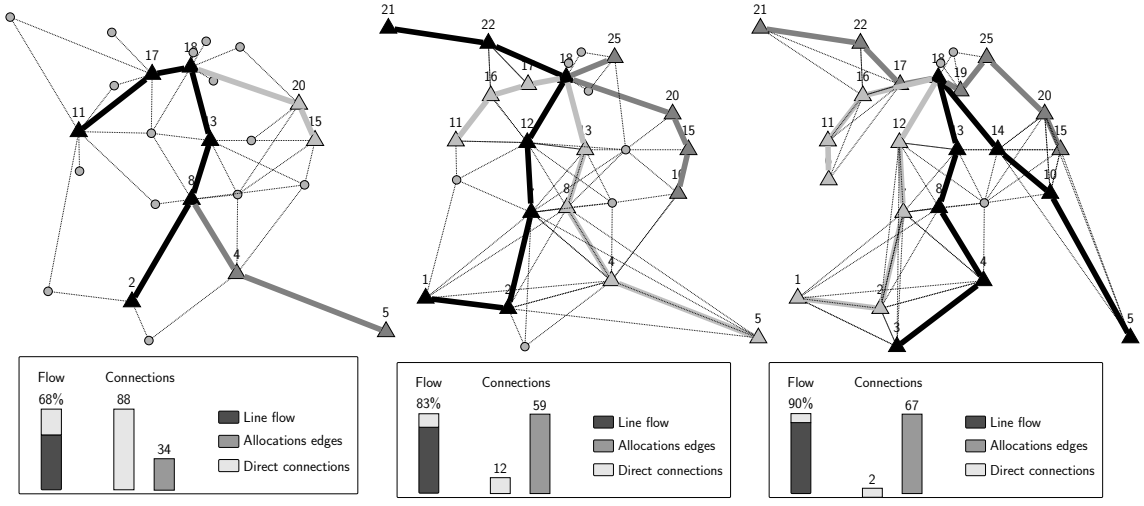
Figure 3.2: Configurations of the hub lines for $\alpha = 0.2$ and $\beta = 1.0$.

rithm can optimally solve instances with up to 20 nodes, this method is able to provide good lower bounds for larger size instances. Three metaheuristc algorithms were also introduced to obtain feasible solutions for the HLLP and the q-HLLP for larger-size instances. The results from the computational experiments show that, for the considered instances, the ALNS and GRASP algorithms are able to find high quality solutions for the HLLP and the q-HLLP in reasonable CPU times.



a) $\alpha = 0.2$ b) $\alpha = 0.4$ c) $\alpha = 0.6$

Figure 3.3: Configurations of the hub lines for $\vartheta = 0.1$ and $\beta = 1.0$.



a) $\beta = 0.6$ b) $\beta = 1.0$ c) $\beta = 2.0$

Figure 3.4: Configurations of the hub lines for $\vartheta = 0.1$ and $\alpha = 0.2$.

Chapter 4

Exact algorithms to solve the hub location problem under congestion

Chapter information

This chapter presents the article: Martins de Sá, E. and de Camargo, R.S. Exact algorithms to solve the hub location problem under congestion. This article is in formatting stage for future submission in a journal.

Abstract This paper addresses the single allocation incomplete hub location problem under congestion. This problem consists in designing a hub-and-spoke network in which the hub-level network can be partially interconnected, and a non-hub node must be allocated to a single hub. The network is designed aiming to minimize the total cost which is composed of the sum of the transportation cost; the fixed cost for locating hubs and hub arcs; and the total cost regarding network congestions. This problem has a great appeal in designing transportation system where congestion cost plays an important role, such as public transportation networks. An important contribution of this paper is to consider congestion in three different services provided by hub-and-spoke systems: entrance, boarding and transferring services. A mixed integer nonlinear formulation and exacts algorithms based on outer approximation framework, and generalized Benders decomposition are proposed. Experiments on benchmark instances with up to 25 nodes confirm the efficiency of the proposed solution algorithms.

Keywords: Hub Location Problem, Generalized Benders Decomposition, Outer Approximation, Congestion cost.

4.1 Introduction

The growth of metropolitan areas steadily pushes governments to restructure and expand their public transport networks in order to improve urban mobility and lower traffic problems, such as traffic congestion, energy consumption, air pollution, and vehicle accidents. At the same time, users constantly pressure for better service levels, while taxpayers request for more cost-efficient systems (Gendreau et al., 1995; Bruno et al., 1998). These give rise to a complex problem which requires considerable amounts of financial resources and a significant effort to manage it.

Over the years, the Operational Research community has provided tools to tackle different aspects of this complex problem. Mathematical programming formulations, specialized solution methods, and decision support systems were developed to facilitate the network design process; the combination of topological and transport technology configurations; the determination of frequencies and timetabling; and the daily operations management such as vehicle and crew scheduling (Guihaire and Hao, 2008; Schöbel, 2012). Recently a new set of resources, based on the ideas of hub-and-spoke systems, has been cleverly incorporated into the design of public transportation networks (Nickel et al., 2001; Gelareh and Nickel, 2011; Martins de Sá et al., 2013a).

Hub-and-spoke systems have attracted the interest of many researchers due to their wide and successful applicability in different economic areas, which require the transportation of goods or persons from many origins to many destinations or a many-to-many distribution system (Campbell, 1992, 1994; Campbell et al., 2002; Alumur and Kara, 2008). Instead of connecting each pair of origin and destination directly, which is usually prohibitively expensive and, in most times, it is not even technological feasible, intermediate facilities are used for aggregating, routing and disseminating the traffic in a hub-and-spoke network. These intermediate facilities are known as hub nodes or just hubs and, together with their respective inter-hub connections or hub arcs, they form the hub-level network. Further, the non-hub nodes or spoke nodes or just spokes, once allocated to the hubs, constitute the local access network. Hub-and-spoke networks are then a hierarchical network with the hub and the local networks at the top and bottom levels, respectively, and having a distinctive economical appeal: As consolidated flows are transported by larger, more efficient, higher volume carriers in the top level (hub-level network), lower unitary transportation costs can be achieved, allowing thus the exploitation of scale economies (O'Kelly, 1998).

Originally (O'Kelly, 1986, 1987), hub-and-spoke networks were assumed to have: an inter-hub connection between every hub pair; no direct link between any two non-hub nodes; and a path with one or at most two hubs for routing demand flows between all origin and destination pair. Further, two different schemes for allocating the non-hub

nodes to hubs were allowed: the non-hub nodes could interact with a single hub only, i.e., be single allocated to a hub, or they could be connected to more than one hub or be multiple allocated.

Recently, more flexible assumptions were proposed (Nickel et al., 2001; Labbé et al., 2004; Campbell et al., 2005a,b; Contreras et al., 2009; Alumur et al., 2009; Calık et al., 2009; Contreras et al., 2010) in order to broaden the applicability of hub-and-spoke networks to other areas. By disregarding the imposing restriction that every pair of hub has to be directly connected and by adapting the design of the network to the characteristics of the application being addressed, different problems can now be seen as special cases of hub-and-spoke networks: (i) tree-shaped facilities location (Contreras et al., 2009, 2010; Martins de Sá et al., 2013b), (ii) ring-star network designs (Labbé et al., 2004), (ii) Lines (Martins de Sá et al., 2013a), (v) incomplete hub networks (Campbell et al., 2005a,b; Alumur et al., 2009). For exhaustive surveys on the variants of hub-and- spoke networks please refer to Campbell et al. (2002) and Alumur and Kara (2008).

This paper addresses the incomplete hub location problem under congestions (IHLPC). This problem consists in selecting a set of nodes to install hubs and installing a set of hub arcs, where a hub-level network partially interconnected are allowed, and allocating each non-hub node to a single hub. The network is designed aiming to minimize the total cost which is composed of the sum of (*i*) the total transportation costs which consider the economies of scale achieved by routing flows between hubs; (*ii*) the total infrastructure costs for locating hubs and hub arcs; and (*iii*) the total cost regarding network congestions. This problem has a great appeal in designing transportation system where congestion cost plays an important role, such as public transportation networks. When designing this kind of systems, it is important to take into account other aspects, as congestion costs, than just transportation, operational and installation costs since these costs can be very conflicting. When only theses costs are observed, networks with flow overloaded hubs may be induced, which may implicate in network users experiencing congestion effects. On a daily basis, users may then be discouraged to use the public transportation system and may rely on private means of transport, which may produce an increase in the cities traffic, and consequently worsening urban mobility. Hence, congestion effects have to be addressed during the modeling of public transportation networks. The design of an incomplete hub-and-spoke network has already been applied to public transportation system by Nickel et al. (2001) and Gelareh and Nickel (2011), but congestion effects are not taken into account.

The effects of congestion in hub and spoke systems have already been addressed by several authors, where the majority address a standard hub system that assumes a hub level network fully interconnected. O'Kelly (1986) presents a heuristic approach

to analyzing hub networks taking into account transportation costs, the total usage of hubs and the variability of the hubs usage. He shows that minimizing only transportation costs can result in high total usage of hubs and high variability of the hub usage. Guldmann and Shen (1997) proposes a general hub network design problem that install hubs and hub arcs assigning capacities to them. The objective of this network design problem is to minimize the sum of infrastructure costs to locate hubs and activate arcs, the total capacities installation costs, and the total operational costs of hubs and arcs, where operational costs relates to system congestions. They model the congestion effects by means of a convex nonlinear function of the flow. To model the proposed problem, they present a mixed-integer nonlinear programming (MINLP) formulation. Marianov and Serra (2003) address congestion in air passenger transportation systems, where the airports are modeled as a $M/D/c$ queue. Assuming that the airport have a fixed number of runways, they propose a congested model that extends the formulation for a classical uncapacited multiple allocation problem (Campbell, 1994). The congestion is addressed by adding a set of probabilistic constraints that bound the probability that the number of airplane waiting, in a queue, to land in a given hub is lower than a given limit. A linearization of this model replacing probabilistic constraints with deterministic capacity constraints on the arrival rate, to each hub, is proposed. A tabu search procedure is proposed to find good solutions for the problem. Elhedhli and Hu (2005) propose a hub location problem considering congestion effects by addressing the congestion cost in the objective function. Modeling the congestion cost as a convex nonlinear function, they present a MINLP formulation. A linearization of the model and a Lagrangean heuristic are proposed to tackle the problem. The proposed heuristic can solve instances with up to 25 nodes with an average optimality gap of less than 1%. Comparing the congestion model to the classical one, they show that congestion model achieves a balanced distribution of flows. Camargo et al. (2009) address the uncapacited multiple allocations hub-and-spoke network design under hub congestion problem. To solve the nonlinear problem, they propose a generalized Benders decomposition approach. By considering congestion costs represented by a power law function, computational experiments using a standard data set of hub location problems, they confirm the efficiency of the proposed algorithm which is able to solve instances with up to 81 nodes.

Recent papers had addressed more realistic characteristics of congestion. Elhedhli and Wu (2010) presents a problem that takes into account the relationship between the routing decisions, capacity and congestion of the network. Camargo et al. (2011) address a single allocation hub location problem under congestion considering, only, congestion associated with flows coming from the local network. In order to solve the problem, they propose a hybrid algorithm that integrate a Benders decomposition

to solve linear MIP problems and an outer approximation (OA) algorithm to solve nonlinear MIP problems. Computational experiments verify the performance of the proposed algorithm which can solve instances with up to 200 nodes. Miranda Junior et al. (2011) present a single allocation hub location problem under congestion and demand uncertainty problem. This model is applied to propose a redesign of the Brazilian Air Transportation Network. Camargo and Miranda (2012) address a single allocation hub location under congestion considering two different perspective: the network owner who aims a system with the least cost and the network user who is willing to accept the minimum of congestion effect at a reasonable cost.

This paper extends the formulation for the incomplete hub location problem (IHLP) proposed by Alumur et al. (2009) to properly account for the congestion effects. The main contribution of this paper is twofold. The congestion costs are considered when designing an incomplete hub-and-spoke system and congestion associated with three different situations found transportation system are taken into account: access the system, boarding in a vehicle and transferring demand flows between hubs. Further, efficient exact methods, based on Outer Approximation (Duran and Grossmann, 1986; Fletcher and Leyffer, 1994) and Generalized Benders decomposition Geoffrion (1972) are developed to tackle the proposed congested problem. The proposed approaches is capable of solving to optimality instances ranging from 10 to 25 nodes in a reasonable time.

The paper is organized as follows. In section 4.2, the definitions and the notation used for the IHLPC are presented. In Section 4.3, an outer approximation and generalized Benders decomposition cut selection schemes are explained. Computational analysis are shown in Section 4.4. Finally, final remarks and possible future research lines are presented in section 4.5.

4.2 Notation and definitions

In order to model the IHLPC, a mathematical formulation based on the formulation for the IHLP, introduced by Alumur et al. (2009), are proposed. Before introducing the formulation for the congested version, the formulation for the IHLP is presented. Both formulations requires the following definitions: Let $G = (N, A)$ be a complete graph, where N is the set of demand node, and A is the set of arcs. For each node $k \in N$, let H_k be the fixed cost of locating a hub at node $k \in N$. Furthermore, let A_{km} denote the fixed cost of locating a hub arc (k, m) between hubs k and m.

For each origin/destination pairs $i, j \in N$, let w_{ij} represents the demand that has to be routed from origin node i to destination node j. Let also $O_i = \sum_{j \in N} w_{ij}$ and

$D_i = \sum_{j \in N} w_{ji}$ be the total demand that is originated from and destined to node $i \in N$, respectively. Each segment arc $(u, v) \in A$ of a path has an unitary transportation cost $c_{uv} > 0$ associated to it. When u and v are hubs, $u = k$ and $v = m$, then a discount factor $0 \leq \alpha \leq 1$, representing the scale economies, is applied, resulting in αc_{km}.

The IHLP consists in selecting some nodes to become hub, installing some connection between hub nodes, and allocating each non-hub to a single hub such that the total cost to install the infrastructure, i.e. locating hubs and hub arcs, and the total transportation cost to routing the demand flows between all origin and destination pair are minimized. The hub-level network can be partially interconnected, i.e., there are some hub that is not interconnected. Further, direct connection between non-hub nodes are not allowed.

The mixed integer linear programming (MILP) formulation to model the IHLP uses the following set of variables. Flow variables $f_{ikm} \geq 0$ to represent the total demand originated at node i that is routed through the hub arc $(k, m) \in A$. Binary variables $z_{ik} \in \{0, 1\}$, for $i, k \in N$, to indicate if a node $i \in N$ is allocated to a hub $k \in N$ ($z_{ik} = 1$) or not ($z_{ik} = 0$). When a hub is located at node $k \in N$, then $z_{kk} = 1$; otherwise $z_{kk} = 0$. Binary variables $y_{km} \in \{0, 1\}$ to indicate if the hub arc $(k, m) \in A$ is selected to link the hubs k and m ($y_{km} = 1$) or not ($y_{km} = 1$), respectively. The formulation for the IHLPC with fixed cost can be formulated as:

$$\min \sum_k H_k z_{kk} + \sum_k \sum_{m:m>k} A_{km} y_{km} + \sum_i \sum_{\substack{k \\ k \neq i}} (O_i + D_i) c_{ik} z_{ik} + \sum_i \sum_k \sum_{\substack{m \\ m \neq k}} \alpha c_{km} f_{ikm} \tag{4.1}$$

s.t.:

$$z_{ik} \leq z_{kk} \qquad \forall i, k \in N : i \neq k \tag{4.2}$$

$$\sum_k z_{ik} = 1 \qquad \forall i \in N \tag{4.3}$$

$$z_{mk} + y_{km} \leq z_{kk} \qquad \forall k, m \in N : k < m \tag{4.4}$$

$$z_{mk} + y_{mk} \leq z_{kk} \qquad \forall k, m \in N : k > m \tag{4.5}$$

$$\sum_{\substack{k \\ k \neq m}} f_{ikm} + o_i z_{im} = \sum_{\substack{k \\ k \neq m}} f_{imk} + \sum_{\substack{j \\ i \neq j}} w_{ij} z_{jm} \qquad \forall i,\ m \in N \tag{4.6}$$

$$f_{ikm} + f_{imk} \leq o_i y_{km} \qquad \forall i, k, m \in N :\ k < m \tag{4.7}$$

$$f_{ikm} \geq 0 \qquad \forall\ i, k, m \in N : i, k \neq m \tag{4.8}$$

$$y_{km} \in \{0, 1\} \qquad \forall k, m \in N : k < m \tag{4.9}$$

$$z_{ik} \in \{0,1\} \qquad \forall\, i,k \in N. \tag{4.10}$$

The objective function (4.1) minimizes the sum of the total cost to install the hubs, the total cost to install the hub arcs and the total transportation costs. Constraints (4.2) ensure that a node i can be allocated to a hub k only if k is installed. Constraints (4.3) guarantee that each node must be assigned to a single hub. Constraints (4.4) and (4.5) ensure that demand nodes and hub nodes can be linked by means of access arcs and hub arcs, respectively, only to installed hubs. Constraints (4.48) are flow balancing constraints. Constraints (4.7) guarantee that the flow originated at node $i \in N$ can use a hub arc $(k,m) \in A$ only if (k,m) is installed. Finally, constraints (4.8)-(4.10) are the standard non-negativity and integrality constraints.

4.2.1 Assessing the congestion effects

One of the most common measure of congestion impacts on transportation systems is the cost associated with users delay, i.e. the waiting time imposed on users due to congestion in a service. The users delay was already used by Guldmann and Shen (1997) and Elhedhli and Hu (2005) to compute congestion costs in hub systems by charging congestion delays. A similar approach will be addressed. Assuming that customers arrive at a given service center according to a Poisson process with the arrival rate of x customers per time unit. Each hub service can be seen as a single server with exponential service times and service rate of Γ customers per time unit. Hence, the hub-and-spoke system can be modeled as $M/M/1$ network queue. From the $M/M/1$ queueing theory, the average waiting time in a service center can be given as

$$\frac{1}{\Gamma - x}.$$

Hence, the congestion cost can be computed by means of the Kleinrock function (Kleinrock, 1964) given as

$$kl(x) = a_k \frac{x}{\Gamma - x}, \tag{4.11}$$

where the arrival rate x can be given as the total flow enter in a given process center and the service rate Γ can be given as the service capacity. In this case, the parameter a_k is the cost charged per unit of waiting time. It's important to note that this function is very useful to measure a congestion since it can represent the explosive growth of congestion cost, where the congestion costs become larger as the flow is closer to service capacity.

Another very common function used to model congestion cost in hub-and-spoke systems is the power law function. This function was introduced by Gillen and Levinson

(1999) to model congestion effects in air transportation systems and was first used, to model congestion costs, in an hub-and-spoke system by Elhedhli and Hu (2005) and Camargo et al. (2009). It can be defined as following. Let x be the total flow, the power law function can be given as: $pl(x) = ax^b$, where a and b are positive parameters ($b \geq 1$).

In order to model the connection between congestion in given service center and its capacity, an adaptation of the power law function, as proposed by Camargo et al. (2011), is used. This new function takes into account that congestion effects begin to deteriorate the level of service when the flow reaches a given threshold of $m\%$ of the system capacity. The congestion cost based on the adapted power law function can be given as

$$apl(x) = \max\{0, a(x - \frac{m}{100}\Gamma)^b\}. \tag{4.12}$$

In this paper, we consider a quadratic $apl(x)$, i.e. $b = 2$, and a threshold of 80% of the service center capacity. Further, as proposed by Camargo et al. (2011), the parameter a are set by means of a curve fitting procedure based on a least squares approach to find a good fit of the $apl(x)$ function into a $kl(x)$ function. Please, refer to Figure 4.1 to see an illustration of an approximation of an adapted power law function and a Kleinrock function.

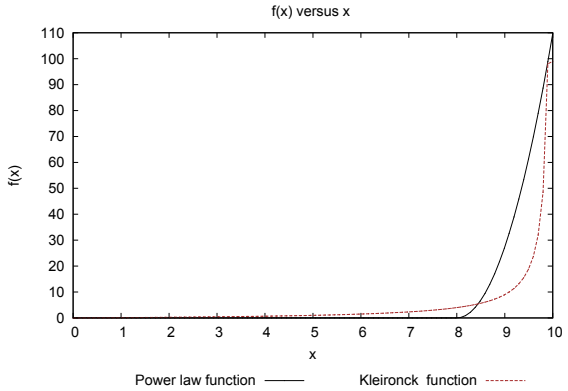


Figure 4.1: A graph of an adapted Power law function of a Kleinrock function.

Another important issue to address when dealing with network design under congestion is which flows impact on system congestion. Depending on the application being dealt with, different schemes are applied for computing the value of x. Some authors (Ernst and Krishnamoorthy, 1999; Marianov and Serra, 2003; Elhedhli and Hu, 2005; Elhedhli and Wu, 2010) only account for flows originated from the local access network; while others (Aykin, 1994; Labbé et al., 2005) consider these flows

(a) Purchase of ticket congestion

(b) Congestion to enter into a station

(c) Vehicle boarding congestion

Figure 4.2: Illustration of different kinds of congestion that can be found in hub-and-spoke networks applied to public transport systems.

plus the traffic incoming from the inter-connections. Further, Guldmann and Shen (1997) also account the flow that is routed through a hub arc. The former is more common in postal services in which sorting activities are the most demanding tasks in a hub, and the others can be found in air and freight transportation systems, and telecommunication networks.

Nevertheless, for public transportation networks, one must observe that the congestion effects may be caused by three distinct situations that are present in a hub: Waiting lines for the users to enter the station (hub) due to congestion on system access service (Figure 4.2(a) and 4.2(b)), and, once inside, the users difficulty or easiness in embarking and disembarking at the platform due to a crowded or empty vehicle regarding congestion on boarding service (Figure 4.2(c)). Finally, the users may be faced with congestion on transfer service associated with roads due to the amount of vehicle using the same path.

These cases suggest then the consideration of three different points for assessing the congestion effects in a hub. In other words, three different variables are required to account for the total flow at a hub: One $g_k \geq 0$ to represent the total flow that

comes from the local network (users accessing the station); $\tilde{q_k} \geq 0$ to represent the flow leaving the hub by means of any inter-hub connections (the users difficulty or easiness to board) and, $q_{km} \geq 0$, to characterize the total flow using a particular inter-hub connections (the users difficulty or easiness to be transfered between to hubs). In the same manner, three different congestion cost functions $\tau(g_k)$, $\tilde{\upsilon}(\tilde{q_k})$ and $\upsilon(q_{km})$ are needed.

The IHLPC is a extension of the IHLP which design a incomplete hub-and-spoke network such that minimize the total cost to install infrastructure, total transportation costs and total cost associated to congestion cost, where the congestion cost is equal the sum of the congestion associated to system access services, boarding services and transfer services. Since the congestion cost are compute through nonlinear functions a mixed integer nonlinear programming (MINLP) formulation is used to model the IHLPC. The formulation for the IHLPC considering the congestion effects is given as:

$$\begin{aligned} \min & \sum_k H_k z_{kk} + \sum_k \sum_{m:m>k} A_{km} y_{km} + \sum_k \tau(g_k) + \sum_k \sum_{m \neq k} \upsilon(q_{km}) \\ & + \sum_k \tilde{\upsilon}(\tilde{q_k}) + \sum_i \sum_{\substack{k \\ k \neq i}} (O_i + D_i) c_{ik} z_{ik} + \sum_i \sum_k \sum_{\substack{m \\ m \neq k}} \alpha c_{km} f_{ikm} \end{aligned} \tag{4.13}$$

$$\text{s.t.:} \quad \text{Constraints (4.2)} - \text{(4.10)} \tag{4.14}$$

$$g_k \geq \sum_i O_i z_{ik} \qquad \forall k \in N \tag{4.15}$$

$$q_{km} \geq \sum_i f_{ikm} \qquad \forall k, m \in N : k \neq m \tag{4.16}$$

$$\tilde{q_k} \geq \sum_i \sum_{m:m \neq k} f_{ikm} \qquad \forall k \in N \tag{4.17}$$

The objective function (4.13) is the same as the previous one (4.1), but with three additional terms accounting for the costs of the congestion effects. Constraints (4.15)-(4.17) properly account the total flow coming from a hub local network, traveling in a hub arc and boarding in a given hub, respectively. This formulation is a huge MINLP model and very hard to solve, even for small size instances. Hence, a specialized tailored method is then required to tackle it. Two procedures to solve the problem are described in the following section.

4.3 Exact algorithm

This paper address two classical methods available to address convex MINLPs, the Generalized Benders Decomposition (GBD) (Geoffrion, 1972) and the Outer-Approximation (OA) technique (Duran and Grossmann, 1986; Fletcher and Leyffer, 1996), that have

been successfully applied to solve hub location problem modeled as MINLPs (Camargo et al., 2009, 2011; Miranda Junior et al., 2011; Camargo and Miranda, 2012). Both approaches are based on a decomposition procedure which consists in solving iteratively a relaxed master problem (RMP) and a subproblem (SP). The former is a relaxed version of a mixed integer program (MIP) reformulation of the original problem with an exponential number of additional constraints, known as cut, where the most of them are relaxed. The SP is a nonlinear problem (NLP) from which violated cuts can be generated to be added to the RMP at each iteration. The stopping criterion normally used for both methods is the convergence of the lower and upper bounds to an optimal solution, if one exists.

When comparing both techniques, the OA approach has the advantage of provides the same or better lower bound than the ones obtained by the GBD method if the same set of subproblems is considered (Grossmann, 2002), requiring fewer iterations to converge to the optimal solution. On the other hand, this advantage related to convergence of OA methods may be achieved at a price of dealing with an RMP larger than the one of the GBD. In this way, it can restrict the size of problems that can be solved by an OA based algorithm. For these reasons, exact algorithms based on OA and GBD are developed to tackle the proposed problem.

4.3.1 Outer approximation method

The OA method is an exact technique proposed by Duran and Grossmann (1986); Fletcher and Leyffer (1996) to solve mixed integer nonlinear problem (MINLP) and can be formalized as following. Let $f : \mathbb{R}^q \mapsto \mathbb{R}$ e $g : \mathbb{R}^q \mapsto \mathbb{R}^m$ be two convex and continuously differentiable functions. Further, let $\mathbf{X} \subseteq \mathbb{R}^q$ and $\mathbf{Y} \subseteq \mathbb{R}^r$ be convex sets. Consider, the following MINLP:

$$\min\ c^T y + f(x) \tag{4.18}$$

$$\text{s.t.:}\quad By + g(x) \leq 0 \tag{4.19}$$

$$x \in \mathbf{X}, y \in \mathbf{Y}. \tag{4.20}$$

Let $\bar{y}$ be a fixed value of variables $y \in \mathbf{Y}$, then the problem can be reduced to a pure nonlinear problem (NLP), the OA subproblem (OASP), given as:

$$\min\ c^T \bar{y} + f(x) \tag{4.21}$$

$$\text{s.t.:}\quad B\bar{y} + g(x) \leq 0 \tag{4.22}$$

$$x \in X. \tag{4.23}$$

Let x^h be the solution for the OASP to a given $\bar{y}$ at a given iteration h. Further, let $\nabla g_j(x^h)$ and $\nabla f(x^h)$ denote the gradient of the functions $g_j(x)$ and $f(x)$, respectively, at x^h. Hence, an OA master problem (OAMP) associated to formulation (4.18)-(4.20) can be given as:

$$\min\ c^T y + \eta \tag{4.24}$$

$$\text{s.t.:}\quad \eta \geq f(x^h) + \nabla f(x^h)^T \left(x - x^h\right) \qquad \forall h \in \mathbf{H} \tag{4.25}$$

$$0 \geq By + g(x^h) + \nabla g(x^h)^T \left(x - x^h\right) \qquad \forall h \in \mathbf{H} \tag{4.26}$$

$$x \in \mathbf{X}, y \in \mathbf{Y}. \tag{4.27}$$

Here $\mathbf{H}$ is the number of iterations and η is a variable that underestimate the value of the nonlinear part of the objective function by means of constraints (4.25). While, constraints (4.26) are used to represents the feasibility set, respectively.

Let $\nu(OAMP)$ and $\nu(OASP)$ denote the optimal value of the objective function for OAMP and OASP, respectively. Hence, a classical OA framework is presented on Algorithm 8.

Algorithm 8 Classical outer approximation algorithm

```
Let UB ← +∞, LB ← −∞ h ← 1
while UB − LB > ϵ do
  Solve the OAMP
  LB ← ν(OAMP)
  ȳ ← y
  Solve the OASP
  Add OA cuts
  UB ← min{UB, ν(OASP)}
  h ← h + 1
end while
```

As mentioned by Grossmann (2002), an OA approach can be embedded in a branch-and-cut scheme in which OA cuts are added within the branch-and-cut tree of the OAMP for every potential incumbent solution. The advantage of this strategy is that by adding OA cut for every potential incumbent solution the OA algorithm converges in only one iteration which can improve the OA algorithm convergence. This approach referred as OA-branch-and-cut is outlined in Algorithm 9.

Since the nonlinearity of the proposed formulation (4.13)-(4.17) refers only to objective function, a reformulation based on outer approximation gives rise to the following

Algorithm 9 Branch-and-cut framework for an OA algorithm implementation.

for all incumbent solutions (z, y) at the OAMP branch-and-cut tree **do**
  $\bar{y} \leftarrow y$
  Solve the OASP
  Add OA cuts
end for

OAMP:

$$\min \sum_k H_k z_{kk} + \sum_k \sum_{m:m>k} A_{km} y_{km} + \sum_k [\xi_k^1 + \tilde{\xi}_k^2] + \sum_k \sum_{m:m\neq k} \xi_{km}^2 + \sum_i \sum_{\substack{k \\ k\neq i}} (O_i + D_i) c_{ik} z_{ik} + \sum_i \sum_k \sum_{\substack{m \\ m\neq k}} \alpha c_{km} f_{ikm} \tag{4.28}$$

s.t.: Constraints (4.2) − (4.10), (4.15) − (4.17) (4.29)

$$\xi_k^1 \geq \tau(g_k^h) + \nabla\tau^T(g_k^h)(g_k^h - g_k) \qquad \forall k \in N, h \in H \tag{4.30}$$

$$\xi_{km}^2 \geq \upsilon(q_{km}^h) + \nabla\upsilon^T(q_{km}^h)(q_{km}^h - q_{km}) \qquad \forall k, m \in N, h \in H : k \neq m \tag{4.31}$$

$$\tilde{\xi}_k^2 \geq \tilde{\upsilon}(\tilde{q_k}^h) + \nabla\tilde{\upsilon}^T(\tilde{q_k}^h)(\tilde{q_k}^h - \tilde{q_k}) \qquad \forall k \in N, h \in H \tag{4.32}$$

$$\xi_k^1, \tilde{\xi}_k^2 \geq 0 \qquad \forall k \in N \tag{4.33}$$

$$\xi_{km}^2 \geq 0 \qquad \forall k, m \in N : k \neq m. \tag{4.34}$$

By fixing the variables $\bar{z} = z$ and $\bar{f} = f$, the following OASP is obtained.

$$\min \sum_k \tau(g_k) + \sum_k \sum_{m\neq k} \upsilon(q_{km}) + \sum_k \tilde{\upsilon}(\tilde{q}_k) \tag{4.35}$$

s.t.: $$g_k \geq \sum_i O_i \bar{z}_{ik} \qquad \forall k \in N \tag{4.36}$$

$$q_{km} \geq \sum_i \bar{f}_{ikm}^h \qquad \forall k, m \in N : k \neq m \tag{4.37}$$

$$\tilde{q}_k \geq \sum_i \sum_{m:m\neq k} \bar{f}_{ikm} \qquad \forall k \in N \tag{4.38}$$

$$g_k, q_{km}, \tilde{q}_k \geq 0 \qquad \forall\ k \in N. \tag{4.39}$$

Observe that this subproblem is trivial to solve when the values of variables z_{ik} and f_{ikm} are known.

4.3.2 Generalized Benders decomposition (GBD)

The generalized Benders decomposition is an extension of a classical method to solve mixed integer linear problem, the Benders decomposition method (Benders, 1962), to be applied to solve MINLP. The basic idea behind Benders decomposition is to project the original problem on the space of complicating variables, usually integer variables, resulting in a formulation with fewer variables by with an exponential number of additional constraints, known as Benders cuts. Since, the most of these constraints are not active in the optimal solution, the most of them are relaxed and added to the Benders master problem only when violated. To find the violated constraints a subproblem with only noncomplicated variables are solved.

The idea behind the GBD is very similar to OA. The method consists in reformulation of the problem (4.18)-(4.20) yielding a linear formulation with an exponential number of additional constraints. The main difference is that the GBD is based on a linear reformulation of the original MINLP that are projected on the space of variables y which results in a RMP with fewer variables. Depending on the size of the vector of variables x, the RMP of GBD is smaller than the one of OA. The reformulation based on GBD give rise the following the GBD master problem (GBDMP):

$$\min\ c^T y + \eta \tag{4.40}$$

$$\text{s.t.:}\quad \eta \geq f(x^h) + (\mu^h)^T[By + g(x^h)] \qquad \forall h \in \mathbf{H} \tag{4.41}$$

$$0 \geq (\mu^h)^T[By + g(x^h)] \qquad \forall h \in \mathbf{G} \tag{4.42}$$

$$y \in Y, \tag{4.43}$$

where x^h and μ^h are the primal and dual optimal variables associated to the NLP (4.21)-(4.23).

Let $\nu(GBDMP)$ and $\nu(GBDSP)$ denote the optimal value of the objective function for GBD MP and GBD SP, respectively. Algorithm 10 presents a framework to solve the problem by means of GBD considering the previously presented formulations. As presented in the previous section, a GBD-branch-and-cut version can be devised by adding GBD cuts inside the branch-and-cut tree of the GBD MP resulting in a convergence with only one iteration.

The GBD framework for the formulation (4.13)-(4.17) can be developed by keeping the set of fractional variables f_{ikm} and their respective objective function component and constraints in the nonlinear subproblem. Further, let f^h_{ikm}, g^h_k, q^h_{km} and $\tilde{q}^h_k$ to be the optimal value of the primal variables of the nonlinear subproblem at iteration h. Define β^h_{im}, δ^h_{ikm}, ω^h_k, ψ^h_{km} and $\tilde{\psi}^h_k$ as the optimal dual variables at a given iteration h.

Algorithm 10 Classical generalized Benders decomposition algorithm.

Let $UB \leftarrow +\infty$, $LB \leftarrow -\infty$ $h \leftarrow 1$
while $UB - LB > \epsilon$ **do**
  Solve the GBD MP
  $LB \leftarrow \nu(GBDMP)$
  Let $\bar{y} \leftarrow y$
  Solve the NLP
  Set values for x^h and μ_h variables
  Add GBD cuts
  $UB \leftarrow \min\{UB,\ \nu(GBDSP)\}$
  $h \leftarrow h + 1$
end while

Hence, the GBD master problem associated to formulation (4.13)-(4.17) can be given as:

$$\min \sum_k H_k z_{kk} + \sum_k \sum_{m:m>k} A_{km} y_{km} + \sum_i \sum_{\substack{k \\ k \neq i}} (O_i + D_i) c_{ik} z_{ik} + \eta \tag{4.44}$$

$$\text{s.t.: Constraints (4.2)} - \text{(4.5)}, \text{(4.9)} - \text{(4.10)} \tag{4.45}$$

$$\eta \geq \nu^h + \sum_i \sum_m (\sum_{\substack{j \\ i \neq j}} w_{ij} z_{jm} - o_i z_{im}) \beta^h_{im} - \sum_i \sum_k \sum_{m:k<m} y_{km} \delta^h_{ikm}$$
$$+ \sum_k \omega_k (\sum_i O_i z_{ik} - g^h_k) + \sum_k \sum_m \phi^h_{km} (\sum_i o_i f^h_{ikm} - q^h_{km}) \tag{4.46}$$
$$+ \sum_k \tilde{\phi}^h_k (\sum_{m:m \neq k} \sum_i o_i f^h_{ikm} - \tilde{q}^h_k) \qquad \forall h \in H$$

$$\sum_{k \in S} \sum_{m \in N \setminus S} y_{km} \geq z_{ss} + z_{rr} - 1 \qquad \forall S \subseteq N, s \in S, r \in N \setminus S \tag{4.47}$$

$$\sum_k \sum_{\substack{m \\ m>k}} y_{km} \geq \sum_m z_{mm} - 1, \tag{4.48}$$

where H is the maximum number of iterations and ν^h is the optimal value of the SP objective function at iteration h. Constraints (4.46) refer to GBD cuts. Constraints (4.47) are cut set constraints (CSCs) which guarantee that the GBD MP solution refers to a connected network. Since, there are a exponential number of these constraints, the most of them are ignored and added to the GBD MP only when they are violated by means of a branch-and-cut framework. These constraints are identified by usign the Concorde callable library by Applegate et al. (2012) to determine the connected components. Finally, constraint (4.48) are added to the GBD MP to ensure that the solutions have enough hub arcs to be connected.

Observe that since f_{ikm} variables are kept in the nonlinear subproblem, GBD algorithm deals with a MP smaller than the one dealt with the OA framework. Considering a given $(\bar{z}, \bar{y})$, the primal variables of the GBD SP can be found by solving the primal nonlinear SP, while the dual variables can be obtained by dualizing the constraints of the GBD SP related to g_k, q_{km} and $\tilde{q}_k$ variables by associating Lagrangean multiplies vectors ω_k^h, ψ_{km}^h and $\tilde{\psi}_k^h$, respectively. The resulting problem can be separated into a linear SP having only f_{ikm} variables, and three nonlinear SPs associated with g_k, q_{km} and $\tilde{q}_k$ variables, respectively. By differential calculus, the optimal dual variables value for the nonlinear SPs at iteration h can be given as

$$\omega_k^h = \tau'(g_k^h), \psi_{km}^h = \upsilon'(q_{km}^h) \text{ and } \tilde{\psi}_k^h = \tilde{\upsilon}'(\tilde{q}_k^h), \tag{4.49}$$

respectively. While, the dual variables β_{im} and δ_{ikm} associated to the linear SP for a given origin i can be obtained by means of the following linear dual SP:

$$\max \sum_m (\sum_{\substack{j \\ i \neq j}} w_{ij} \bar{z}_{jm} - o_i \bar{z}_{im}) \beta_{im} - \sum_k \sum_{m:k<m} o_i \bar{y}_{km} \delta_{ikm} \tag{4.50}$$

$$\text{s. t: } \beta_{im} - \beta_{ik} - \delta_{ikm} \leq o_i(\alpha c_{km} + \psi_{km}^h + \tilde{\psi}_k^h)) \qquad \forall k, m \in N : k < m \tag{4.51}$$

$$\beta_{im} - \beta_{ik} - \delta_{imk} \leq o_i(\alpha c_{mk} + \psi_{mk}^h + \tilde{\psi}_m^h) \qquad \forall k, m \in N : k > m \tag{4.52}$$

$$\beta_{im} \in \mathbb{R} \qquad \forall m \in N \tag{4.53}$$

$$\delta_{ikm} \geq 0 \qquad \forall k, m \in N : k < m. \tag{4.54}$$

As mentioned by Camargo and Miranda Jr (2012), the performance of a GBD algorithm mostly depends on the total number of iterations to find the optimal solution. On the other hand, this amount is directly related to cuts quality, where good cuts results in better convergence. Magnanti and Wong (1981) propose a cut selection scheme to accelerate the convergence of Benders decomposition by adding in each iteration Pareto-optimal cuts, i.e. cuts that are not dominated by any other cut. Aiming to improve the convergence of the GBD algorithm, the idea of adding Pareto-optimal GBD cuts in GBDMP will be embedded to the algorithm. These auxiliary cuts can be generated by means of the idea proposed by Papadakos (2008) for Benders decomposition which consists in solving an auxiliary subproblem, known as independent Magnanti-Wong problem. This subproblem is very similar to the subproblem but instead to be parameterized by a fixed MP solution $(\bar{z}, \bar{y})$, it is parameterized by a core point (z^0, y^0) which is a point that belongs to the relative interior of the convex hull of master problem feasibility space. The optimal dual solution associated with the following independent Magnanti-Wong problem is used to generated the Pareto-optimal

cuts for the GBD algorithm:

$$\max \sum_{i} \sum_{m} (\sum_{\substack{j \\ i \neq j}} w_{ij} z^0_{jm} - o_i z^0_{im}) \beta_{im} - \sum_{i} \sum_{k} \sum_{m:k<m} o_i y^0_{km} \delta_{ikm} \tag{4.55}$$

$$\text{s. t: } (4.51) - (4.54). \tag{4.56}$$

Observe that to generate different cuts in each iteration different core points are necessary. A set of core points can be generated as following. Assuming that a valid initial core point is available, Papadakos (2008) proposes to use the convex combination of the current core point and the current master solution to generate a new valid core point. In this way, the most challenge task, to generate a set of core points, is to find a valid initial core point. In this paper, the core point proposed by Martins de Sá et al. (2013b) for the *three of hub location problem* is used as an initial core point. A three of hubs is a hub-and-spoke network where the hubs are connected by means of a tree. Hence, a tree of hubs is a particular case of an incomplete hub-level network. Thus, the proposed initial core point is a valid one. This core point is given as

$$z^0_{kk} = 1/2 \qquad \forall\ k \tag{4.57}$$

$$z^0_{ik} = 1/(2\ n - 2) \qquad \forall\ i \neq k \tag{4.58}$$

$$y^0_{km} = (n-2)/(n^2 - n) \qquad \forall\ k < m. \tag{4.59}$$

Algorithm 11 outline a framework of a generalized Benders decomposition algorithm with addition of Pareto-optimal cuts.

4.3.3 Hybrid outer approximation/generalized Benders decomposition strategy

As previously mentioned, outer approximation and generalized Benders decomposition are among the strategies most commonly used to solve MINLPs. As described in preceding sections, each one of these methodologies has advantage and disadvantage when tackling this kind of problems. If for one hand, OA methods provides tighter lower bounds than GBD, on the other hand, a high price for dealing with a large master problem, i.e. a formulation that keeps all f_{ikm} fractional variables, must to be paid. An idea to improve the algorithms proposed in previous sections is by devising a new algorithm that hybridize both approaches.

This hybrid strategy is based on the following observation. The congested problem

Algorithm 11 Generalized Benders decomposition algorithm with addition Pareto-optimal cuts.

Let $UB \leftarrow +\infty$, $LB \leftarrow -\infty$, $h \leftarrow 1$, $y^0 \leftarrow$ valid initial core point
while $UB - LB > \epsilon$ **do**
  Solve the GBD MP
  $LB \leftarrow \nu(GBDMP)$
  Let $\bar{y} \leftarrow y$
  Solve the NLP
  Set values for x^h and μ_h variables
  Add a GBD cut
  $UB \leftarrow \{UB,\ \nu(GBDSP)\}$
  Let $\bar{y} \leftarrow y^0$
  Solve the NLP
  Set values for x^h and μ_h variables
  Add a GBD Pareto-optimal cut
  Update the core point: $y^0 \leftarrow 0.5y^0 + 0.5y$
  $h \leftarrow h + 1$
end while

has three set of flow variables, g_k, q_{km} and $\tilde{q_k}$, where each one is associated with three different nonlinear congestion cost functions $\tau(g_k)$, $\tilde{v}(\tilde{q_k})$ and $v(q_{km})$. By the previously developed algorithms, when OA aproach tackle all these nonlinearities, it is necessary to keep all this flow variables and the f_{ikm} flow variables in the master problem which can results in heavy master problem. A smaller master problem can be obtained by using OA approach to deal with the nonlinearities of $\tau(g_k)$, while GBD method are used to tackle the nonlinearities regarding to $\tilde{v}(\tilde{q_k})$ and $v(q_{km})$ congestion functions. The advantage of this approach is to keep good bounds. Most likely this bound are not better than the one provide by OA method, however probably it will be better than the one of GBD approaches. Further, this approach will deal with smaller master problem than OA algorithm.

A similar approach, the OA/BD framework, has been proposed by Camargo et al. (2011) to solve the single allocation hub location problem under congestion. This problem refers to a single allocation hub location problem, where the hub network is fully interconnected. The hybrid method proposed integrates the OA algorithm with Benders decomposition (BD). The main idea is to apply the OA approach in the original formulation which results in reformulation with more variables and constraints than the original one. Since, the flow variables associated with the transportation component was not directly linked to flow variables related to congestion component, they project out the former set of flow variables by applying BD. In this case, the OA Framework handles with congestion component while BD decomposition manages the transportation component which has fractional variables with four indexes. This

approach is not suitable to tackle the IHLPC because the total transportation cost depends on congestion costs related to boarding and transfer service, i.e. transportation cost may vary according to the congestion level associated with these services. For this reason, variables related to transportation costs and variables related to boarding and transfer congestion service have to be dealt with the same approach, the GBD. This is the major contribution of the OA/GBD approach.

By integrating OA MP reformulation and GBD reformulation, the OA-GBD master problem reformulation can be given as:

$$\min \sum_k H_k z_{kk} + \sum_k \sum_{m:m>k} A_{km} y_{km} + \sum_i \sum_{\substack{k \\ k \neq i}} (O_i + D_i) c_{ik} z_{ik} + \sum_k \xi_k^1 + \eta \tag{4.60}$$

$$\text{s.t.: Constraints (4.2)} - \text{(4.10) and (4.15)} \tag{4.61}$$

$$\xi_k^1 \geq \tau(g_k^h) + \nabla\tau^T(g_k^h)(g_k^h - g_k) \qquad \forall k \in N, h \in H \tag{4.62}$$

$$\eta \geq \nu^h + \sum_i \sum_m (\sum_{\substack{j \\ i \neq j}} w_{ij} z_{jm} - o_i z_{im}) \beta_{im}^h - \sum_i \sum_k \sum_{m:k<m} y_{km} \delta_{ikm}^h + \sum_k \left(\sum_m \phi_{km} (\sum_i o_i f_{ikm} - q_{km}) + \tilde{\phi}_k (\sum_{m:m \neq k} \sum_i o_i f_{ikm} - \tilde{q}_k) \right) \qquad \forall h \in H \tag{4.63}$$

$$\xi_k^1 \geq 0 \qquad \forall k. \tag{4.64}$$

Here H is the maximum number of iteration and ν^h is the optimal value of subproblem objective function at iteration h. The dual variables β_{im} and δ_{ikm} for a given $(\bar{z}, \bar{y})$ can be obtained by means of the linear dual subproblem (4.50)-(4.54), where the optimal dual variables value for the variables ψ_{km}^h and $\tilde{\psi}^h{}_m$ at iteration h can be given as

$$\psi_{km}^h = \upsilon'(q_{km}^h) \text{ and } \tilde{\psi}_k^h = \tilde{\upsilon}'(\tilde{q}_k^h),$$

respectively.

The hybrid OA/GBD approach is outlined in Algorithm 12. Using the same idea of the algorithms proposed in the previous sections, an OA/GBD-branch-and-cut approach and an OA/GBD approach with Pareto-optimal cuts can be devised.

In the next section, the results of computational experiments are presented.

Algorithm 12 Hybrid OA/GBD algorithm.

Let $UB \leftarrow +\infty$, $LB \leftarrow -\infty$, $h \leftarrow 1$
while $UB - LB > \epsilon$ **do**
  Solve the OA/GBD MP
  $LB \leftarrow \nu(GBDMP)$
  Let $\bar{y} \leftarrow y$
  Add OA cut for ξ^1
  Solve the NLP
  Set values for the optimal primal and dual variables
  Add a GBD cut
  $UB \leftarrow \{UB,\ \nu(GBDSP) + \nu(OASP)\}$
  $h \leftarrow h + 1$
end while

4.4 Computational results

Two types of experiments were carried out. The first set of experiments aims to analyze the system configuration considering different levels of congestion. The second set of experiments aims to evaluate the efficiency of the proposed algorithms. The tests were performed using a standard set of instances from the literature: AP data set from Australian postal service introduced by Ernst and Krishnamoorthy (1996). This data set provide a flow matrix with the demand for each origin and destination pair and the Euclidean distances between each pair of nodes. It also provides the set of fixed cost to install a hub. The capacity of incoming flows was generated as been the sum of the total flow originated at that node and a random fraction, between 15% and 50%, of the total demand. Let $\overline{w}$ denote the total demand flow. Hence, the capacity associated to a potential hub k is generated as

$$\Gamma_k = O_k + \text{Unif}[0.15, 0.50]\overline{w}.$$

The same idea are used to generate the capacity associated with boarding service. The capacity associated the hub arcs are assumed to be equal 50% of the average capacity of the boarding capacity. In order to analyze how the capacities affect the design of the system, we consider two level of capacities by multiplying the capacities previously defined by a parameter: $\beta \in \{0.7, 1.0\}$. In addition, since AP data set does not provide a fixed cost to install a hub arc, these data was generated in the following way. The cost of opening a hub arc between hubs k and m are assumed to be proportional to the Euclidean distance between k and m. On the experiments a proportional factor $\vartheta \in \{0, 1, 4\}$ is considered, where $\vartheta = 0$ means no arc installation cost. Different congestion level are achieved by varying the congestion cost parameter described on

Section 4.2.1. Let $a_c \in \{5, 5000\}$, $a_b \in \{5, 5000\}$ and $a_t \in \{5, 5000\}$ be the system access, boarding and transfer service congestion cost parameter, respectively. Finally, the experiments consider discount factor $\alpha \in \{0.2, 0.5, 0.8\}$.

The first set of experiments aims to analyze how the system being designed is affected by the parameters. Since the total congestion cost is affected by the capacities of the system, congestion cost level, and the network topology, these experiments take into account the variation of parameters associated with capacity represented by β, congestion costs that is influenced by the delay cost given by parameters a_c, a_b and a_t and the cost to install each hub arc that is controlled by the parameters ϑ. All the tests were performed considering the adapted power law function presented on Section 4.2.1 to measure congestion costs. The characteristics to be observed in tests related to: (i) the network topology; (ii) the cost distribution concerning to transportation costs, infrastructure costs and congestion costs; and (iii) the percentage of flow related to the available capacity associated with the solution. The results of the experiments are presented on Figure 4.3-4.5 and on Table 4.1.

Figure 4.3 presents the networks configuration considering different combinations of congestion level for system access service, boarding service, and transfer service. This figure presents different system configuration obtained by ignoring the congestion related to only one kind of service Figure 4.3-a, Figure 4.3-b and Figure 4.3-c and the system configuration when all congestion service are taken into account. According to figures, when only congestion related to access service are ignored (Figure 4.3-a), the optimal network has only one hub installed which have to provide access to all flows, while no boarding or transference between hubs is possible. When only transference congestion is disregarded (Figure 4.3-b), two hubs are located and the allocation to this hub is made to balance the use of capacities associated with the others hub services, but since there is only one hub arc, all flow pass trough this arc overloading it. When only boarding congestion is disregarded (Figure 4.3-c), more hubs and hub arcs are installed to reduce hub arc traffic. When the congestion related to all services are accounted, the optimal network has three hub-arcs which reduce the hub traffic and the most nodes are allocated to node 10 that is the node with more capacities associated with boarding and access services (Figure 4.3-d).

To realize the service load related to the previous figure (Figure 4.3), Table 4.1 shows how loaded the hubs are regarding the capacity of each of these services. This table is presented as following. Column C^*_{5000}/C^* presents the ratio between the total cost associated with the optimal configuration found when considering aggressive congestion costs for all service provided by the hubs and the congestion level used to solve the problem. Hence, this column gives a measure of the solution quality when exists congestion in a given service, but it is not taken into account when designing the net-

work. In this way, we can confirm the importance of considering all kind of congestion to design the transportation system. The columns associated to *Capacity utilization*, *Local*, *Board* and *Arc*, present the average capacity utilization associated to system access, boarding and transfer service, respectively. According to these columns, when the congestion function associated with a given service are not accounted and the congestion associated to the other two are more aggressive, the optimal solution overload the first service. However, when the congestion associated with the three services are accounted, the optimal solution tends to balance the use of capacity in the three services provided by hub. Showing again how important is to address the three kind of congestion.

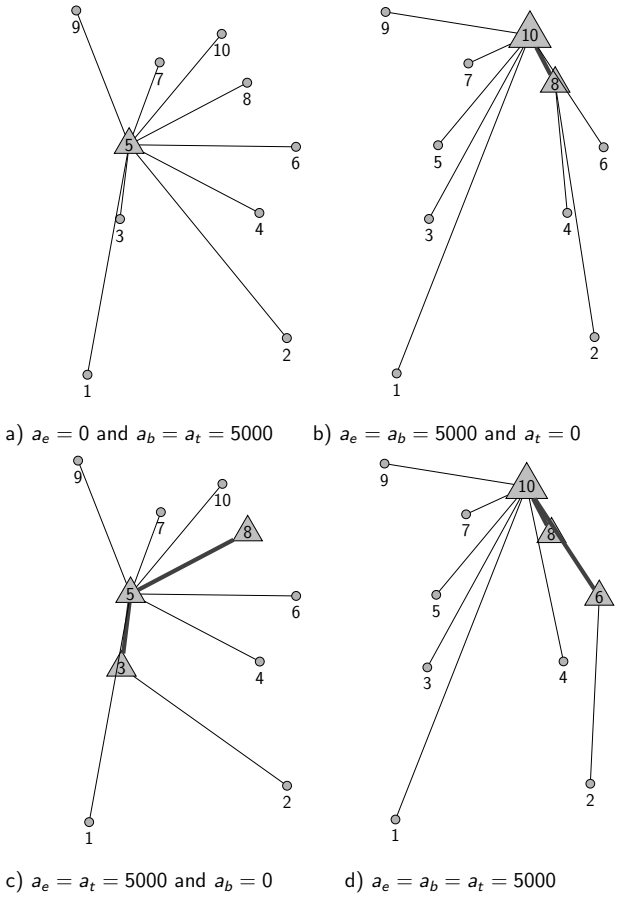


Figure 4.3: System configurations for a 10 nodes instance with loose capacities ($\beta = 1.0$), high economies of scale ($\alpha = 0.2$) and high arc installation costs ($\vartheta = 4$) for different congestion factors.

Besides the congestion parameters other parameters have a profound impact in the analyze of the system in a presence of congestion. Figure 4.4 presents the system configuration considering a 10 nodes instance with a loose capacity ($\beta = 1.0$) for high

Table 4.1: Comparison of capacity loading associated with access, boarding and transfer service for different congestion level.

a_c	a_b	a_t	C^*_{5000}/C^*	Capacity utilization		
				Access	*Board*	*Transf.*
0	5000	5000	23.70	117.00%	0.00%	0.00%
5000	5000	0	3.32	76.00%	40.00%	93.00%
5000	0	5000	1.93	67.00%	64.00%	60.00%
5000	5000	5000	1	47.00%	39.00%	56.00%

level of congestion considering different hub arc installation cost. In general, as the installation costs factor increase, cheaper (shorter) and less number of hub arcs are installed. Observe on this figure that when hub arc installation costs are accounted the hub arcs of the optimal network have almost the same size (installation cost). However, when a medium installation cost is considered the hub network is fully interconnected but a shorter hub arc is installed (the arc (4,6)), while for high installation cost the hub network is partially interconnected. Furthermore, for lower installation cost the load of all service of the optimal network tends to be balanced, i.e. the allocation of non-hub nodes are better distributed between the hubs which results in better loaded of capacities associated with all services provided by the hubs. On the other hand, for high installation cost the system must take into account the trade-off between congestion cost and the cost to install the infrastructure which can overload some services.

Finally, Figure 4.5 presents different configuration considering different level of capacities for a 10 nodes instances with high economies of scale ($\alpha = 0.2$), medium and high arc installation cost ($\vartheta \in \{1, 4\}$) which shows the dependence of the total flow using a given service and the service capacity. Comparing Figure 4.5-a and Figure 4.5-b, when the capacity is tight a different network topology and allocation scheme is necessary to avoid congestion. Observe, that is this case fewer non-hub nodes are allocated to hub 10 which is the hub with more capacity. Comparing Figure 4.5-c and Figure 4.5-d, instances with high arc installation costs and small capacities provide an optimal network such that the loading of flows are balanced resulting in lower congestion costs. However, when dealing with loose capacities the arc installation costs predominate the other costs.

To evaluate the performance of proposed algorithms, a set of computational experiments on AP instances ranging from 10 to 25 nodes were performed. The first set of experiments are performed to compare three OA based algorithms. The second set of experiments aims to compare GBD and OA/GBD versions. The third set of experiments compare the best proposed exact methods and the general purpose solver CPLEX to solve the problem considering an adapted power law function on benchmark-

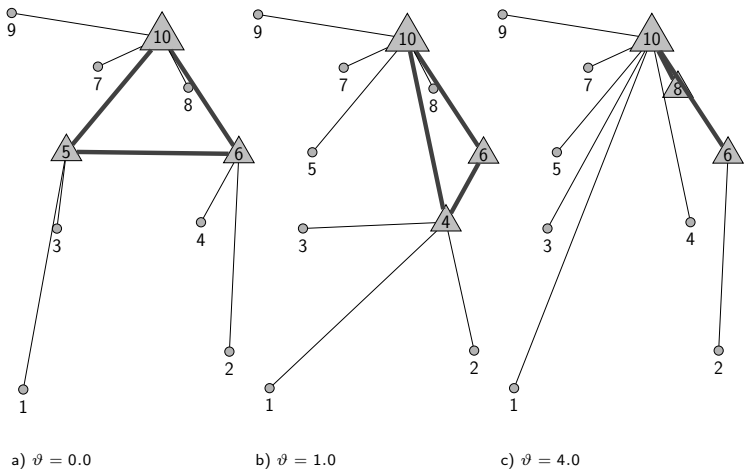


Figure 4.4: System configurations for a 10 nodes instance with loose capacity($\beta = 1.0$), high economies of scale ($\alpha = 0.2$) and aggressive congestion costs ($a_c = a_b = a_t = 5000$) for different arc installation cost parameter ϑ.

ing instances with up to 20 nodes. After that, the performance of the best algorithm found is evaluated to solve instances with up to 25 nodes considering Kleinrock congestion cost function. All tests were run on a GenuineIntel CPU E5530 with 2.4GHz and 94 GB memory computer using the Linux operating system. The OA and GBD based algorithms were implemented in C ++ using the CPLEX 12.5 to solve linear and nonlinear SPs, if needed, and MIP master problems.

Table 4.2 summarizes the computational results comparing three OA approaches: a classical OA implementation (Algorithm 8), an OA-branch-and-cut implementation (Algorithm 9) that adds cuts in the OAMP branch-and-cut tree for every potential incumbent solution (OABC) and an OA-branch-and-cut that adds OA cuts for all potential incumbent solution e for fractional solution in the root node (OABC root cut). The experiments was performed for solving 10 nodes instances considering power law congestion function, where the data are aggregated by economies of scale factor $\alpha \in \{0.2, 0.5, 0.8\}$ and congestion factors $\tilde{a} \in \{5, 5000)\}$ such that the congestion cost parameters $a_c = a_b = a_t$ are equal to $\tilde{a}$. The table of results can be described as following. Columns referring to Time[s] show the average computational time in seconds, columns #cuts show the average number of OA cuts, while columns referring to #BS presents the total number of best solutions found, where each row of the table is associated with a set of six instances. According to the table, OABC root tree approach is on average faster than the other approaches for the most tested instances, i.e. for 17 of 36 instances. Furthermore, the addition of OA cuts for fractional solution results in lower average number of cuts added. Therefore, this OA variants is used in

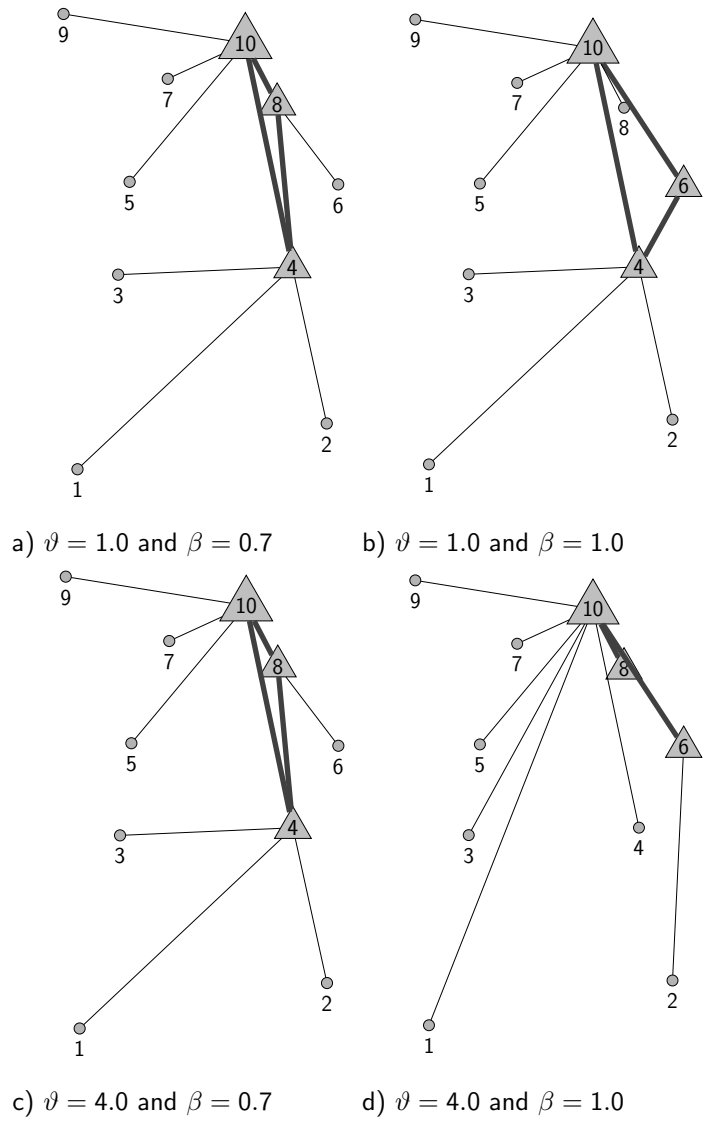


Figure 4.5: System configurations for a 10 nodes instance with high economies of scale ($\alpha = 0.2$), medium and high arc installation cost ($\vartheta \in \{1, 4\}$) for different capacity level parameter β.

subsequent experiments. Only to simplify the notation, this version will be referred as OA.

The following set of experiments is focused on compare the GBD based versions. The first set of experiments compare the classical version (GBD) and the GBD-branch-and-cut version (GBD BC) that add GBD cuts inside the branch-and-cut tree of the MP for each potential incumbent solution. The results using instances with 10 nodes are summarized on Table 4.3, where the data are aggregated considering three economies of scale factor $\alpha \in \{0.2, 0.5, 0.8\}$ and congestion factor $\tilde{a} = a_c = a_b = a_t \in \{5, 5000\}$. For

Table 4.2: Comparison between the classical OA approach, the OA-branch-and-cut algorithm and the OA-branch-and-cut with addition of cut from fractional solution in the root nodes of the MP branch-and-cut tree.

n	ϑ	β	Classic			OABC			OABC root cut		
			Time [s]	# cuts	# BS	Time [s]	# cuts	# BS	Time [s]	# cuts	# BS
100	0.7		11.92	215.0	1	16.74	13407.3	4	7.61	10137.50	2
100	1		21.35	220.0	1	11.41	12357.7	3	27.54	9072.00	2
101	0.7		48.90	193.3	3	39.96	17142.7	2	6.61	11292.50	4
101	1		73.46	276.7	1	9.84	15437.7	1	20.85	13309.00	3
104	0.7		75.18	263.3	3	25.51	14484.3	0	4.93	10246.50	3
104	1		60.59	270.0	1	5.81	13072.7	3	22.61	8243.00	3
			48.57	239.7	10	18.21	14317.0	13	15.03	10383.4	17

The data are aggregated considering three economies of scale factor $\alpha \in \{0.2, 0.5, 0.8\}$ and two congestion factor $a_c = a_b = a_t \in \{5, 5000)\}$

all these experiments, a CPU time limit of 1 hours is used. According this table, GBD BC version presents to be substantially better than the classical one by solving the tested instances on average faster than it. It's important to observe that the average number of added cuts by GBD BC variant is higher than by the classical one, however the need for solving only a single master problem for the branch-and-cut version may explain the faster convergence.

Due to the performance of GBD BC version, a new variant of this method that also adds Pareto-optimal cuts inside the branch-and-cut tree of the master problem is tested. This new variant (GBD PO) is compared with similar versions of hybrid OA/GBD approaches: OA/GBD-branch-and-cut (OA-GBD BC) and an OA/GBD-branch-and-cut approach with Pareto-optimal cuts (OA-GBD PO). Computational results are presented on Table 4.4. According the table, the addition of Pareto-optimal cuts improves GBD and OA/GBD convergence solving the problem faster than when this kind of cuts is not added. Further, proposed approaches based on the integration of OA and GBD methods presents better performance than approaches that are only based on GBD method. The efficiency of the OA-GBD algorithms is confirmed by the lower average CPU times and by fewer average number of cuts added.

The following set of experiments aims to compare the performance of the best exact algorithms found, OABC variant and OA/GBD PO, and the solver CPLEX. Since CPLEX is only able to solve quadratics nonlinear MIP problems, these experiments were performed considering an adapted power law congestion function. Table 4.7, 4.6 and 4.5 summarize the experiment results for 10 and 20 nodes AP instances considering data aggregated by economies of scale factor $\alpha \in \{0.2, 0.5, 0.8\}$. These tables present

Table 4.3: Comparison of the classical GBD variant and GBD-branch-and-cut version considering a 10 node instances.

			Classical GBD		GBD BC	
α	ϑ	β	Time[s]	# cuts	Time[s]	# cuts
0.20	0	0.7	1591.44	240.00	310.13	1261.00
0.50	0	1.0	1800.03	270.83	759.81	2603.67
0.80	1	0.7	1800.90	251.50	132.32	1119.50
0.20	1	1.0	1800.02	273.50	801.21	2720.17
0.50	4	0.7	1800.82	254.17	507.69	1595.17
0.80	4	1.0	1800.03	275.17	819.83	2587.33
Average:			1765.54	260.86	555.16	1981.14

The data are aggregated considering three economies of scale factor $\alpha \in \{0.2, 0.5, 0.8\}$ and congestion factor $\tilde{a} = a_c = a_b = a_t \in \{5, 5000\}$.

Table 4.4: Comparison of the GBD-branch-and-cut variant, GBD-branch-and-cut version with Pareto-optimal cuts, the OA-GBD-branch-and-cut variant and the OA-GBD-branch-and-cut variant with Pareto-optimal cuts considering a 10 node instances.

ϑ	β	GBD BC		GBD PO		OA-GBD BC		OA-GBD PO	
		Time[s]	#cuts	Time[s]	#cuts	Time[s]	#cuts	Time[s]	#cuts
0	0.7	55.68	756.83	16.34	307.17	9.30	245.00	6.38	154.50
0	1	7.79	327.00	7.86	248.00	5.06	206.50	6.24	178.67
1	0.7	886.68	2891.83	151.62	1192.83	324.37	1459.00	60.31	593.83
1	1	372.97	1701.50	79.42	805.33	226.62	1233.33	82.04	679.33
4	0.7	1800.04	4739.33	364.72	1521.17	864.78	1944.33	152.43	816.00
4	1	207.84	1470.33	58.27	770.17	80.93	917.00	39.79	545.17
Average:		555.16	1981.14	113.04	807.44	251.84	1000.86	57.86	494.58

The data are aggregated considering three economies of scale factor $\alpha \in \{0.2, 0.5, 0.8\}$ and congestion factor $\tilde{a} = a_c = a_b = a_t \in \{5, 5000\}$.

the average CPU time and the average optimality gap, when any instance of the set is not solved to optimality. The first table consider the case where there congestion in all the three services provided by the hub. Table 4.7 and 4.6 presents results for the case where traffic congestion and boarding congestion are ignored, respectively. According the tables, OA approach and OA-GBD is faster and presents better optimality gap than CPLEX. Further, both exact methods are very competitive. Observing the table, it's possible note that OA-GBD becomes more competitive with the OA approach for instance with 20 nodes in which the OA MP becomes larger. Further, OA-GBD approach presents better performance for instances where congestion on boarding service or transfer service are ignored, in this case the congestion each method handly only one kind of congestion, i.e. when OA deals with system access service, while GBD deals with only one other kind of service, boarding or transferring. In this case, there are a

balance on the characteristics of both methods, advantage and disadvantage.

Table 4.5: Comparison of the best OA variant, the best OA-GBD variant and CPLEX considering the three kinds of congestion.

N	$\tilde{a}$	ϑ	β	OA		OA-GBD		CPLEX	
				Time[s]	GAP	Time[s]	GAP	Time[s]	GAP
10	5	0	0.7	0.90	–	**0.20**	–	2.59	–
10	5000	0	0.7	**6.30**	–	13.72	–	52.78	–
10	5	1	0.7	0.17	–	**0.06**	–	1.24	–
10	5000	1	0.7	**23.31**	–	108.63	–	153.21	–
10	5	4	0.7	0.16	–	**0.06**	–	0.52	–
10	5000	4	0.7	**63.54**	–	193.46	–	296.23	–
10	5	0	1	0.10	–	**0.06**	–	0.34	–
10	5000	0	1	**6.60**	–	11.77	–	44.84	–
10	5	1	1	0.08	–	**0.04**	–	0.33	–
10	5000	1	1	**34.74**	–	140.99	–	228.50	–
10	5	4	1	0.07	–	**0.04**	–	0.18	–
10	5000	4	1	**39.70**	–	69.83	–	181.72	–
20	5	0	0.7	67.69	–	**29.10**	–	7228.16	–
20	5000	0	0.7	822.85	–	**482.36**	–	28628.52	–
20	5	1	0.7	49.23	–	**28.90**	–	1840.11	–
20	5000	1	0.7	**71089.41**	1.75	84600.05	11.68	84600.01	7.74
20	5	4	0.7	13.11	–	**5.73**	–	1587.80	–
20	5000	4	0.7	**75600.38**	4.33	84600.08	25.42	84600.01	22.42
20	5	0	1	8.97	–	**2.03**	–	1800.35	–
20	5000	0	1	188.80	–	**47.21**	–	12195.10	–
20	5	1	1	**2.26**	–	4.16	–	1351.81	–
20	5000	1	1	**9910.66**	–	69768.91	0.96	83219.41	6.80
20	5	4	1	**1.03**	–	3.72	–	283.50	–
20	5000	4	1	**11697.78**	–	81149.93	0.95	45335.99	–
Average:				7067.82	0.25	13385.88	1.63	14734.72	1.54

The data are aggregated considering three economies of scale factor $\alpha \in \{0.2, 0.5, 0.8\}$

In all experiments $a_c = a_b = a_t = \tilde{a}$.

Table 4.6: Comparison of the best OA variant, the best OA-GBD variants and CPLEX considering only system access and boarding congestion ($a_t = 0$).

n	a_c	a_b	ϑ	β=Tight						β=Loose		
				OA		OA-GBD		CPLEX		OA	OA-GBD	CPLEX
				Time[s]	GAP	Time[s]	GAP	Time[s]	GAP	Time[s]	Times[s]	Times[s]
10	5	5	0	0.78	0.00	**0.23**	0.00	1.19	0.00	0.09	**0.06**	0.36
10	5	5000	0	0.87	0.00	**0.16**	0.00	1.74	0.00	0.09	**0.06**	0.35
10	5000	5	0	2.02	0.00	**1.69**	0.00	16.46	0.00	0.77	**0.56**	4.36
10	5000	5000	0	**3.21**	0.00	6.56	0.00	33.28	0.00	1.78	**1.72**	11.43
10	5	5	1	0.15	0.00	**0.06**	0.00	0.67	0.00	0.08	**0.04**	0.26
10	5	5000	1	0.14	0.00	**0.07**	0.00	0.69	0.00	0.08	**0.04**	0.25
10	5000	5	1	6.00	0.00	**2.68**	0.00	56.91	0.00	2.30	**0.72**	7.61
10	5000	5000	1	**14.04**	0.00	45.00	0.00	116.59	0.00	3.49	**1.90**	11.26
10	5	5	4	0.15	0.00	**0.06**	0.00	0.36	0.00	0.08	**0.04**	0.31
10	5	5000	4	0.15	0.00	**0.06**	0.00	0.36	0.00	0.08	**0.04**	0.31
10	5000	5	4	5.11	0.00	**1.97**	0.00	35.51	0.00	1.65	**1.14**	11.85
10	5000	5000	4	**36.70**	0.00	86.10	0.00	203.02	0.00	1.91	**1.36**	11.99
20	5	5	0	23.56	0.00	**5.51**	0.00	1806.69	0.00	6.78	**1.53**	1333.60
20	5	5000	0	26.78	0.00	**5.85**	0.00	1893.14	0.00	6.75	**1.49**	1348.04
20	5000	5	0	1028.15	0.00	**415.82**	0.00	84600	1.87	122.77	**42.14**	14850.15
20	5000	5000	0	**350.74**	0.00	479.89	0.00	33277.22	0.00	135.50	**57.20**	16727.16
20	5	5	1	22.69	0.00	**7.10**	0.00	2700.42	0.00	**2.25**	4.30	572.76
20	5	5000	1	25.94	0.00	**7.85**	0.00	2810.42	0.00	**2.20**	4.33	566.28
20	5000	5	1	4856.55	0.00	**368.79**	0.00	66063.75	2.19	385.69	**32.44**	42116.41
20	5000	5000	1	**43015.92**	0.00	84600	12.84	83344.71	5.96	694.42	**53.76**	37013.19
20	5	5	4	13.74	0.00	**5.80**	0.00	767.91	0.00	**0.68**	3.70	445.46
20	5	5000	4	13.73	0.00	**5.77**	0.00	777.16	0.00	**0.69**	3.83	445.99
20	5000	5	4	5612.45	0.00	**139.36**	0.00	17156.84	0.00	473.44	**8.98**	21685.24
20	5000	5000	4	84600	5.65	84600	29.96	84600	25.78	926.71	**28.89**	16395.51
		Average:		5819.15	0.26	7116.11	1.78	15844.38	1.49	115.43	10.43	6398.34

The data are aggregated considering three economies of scale factor $\alpha \in \{0.2, 0.5, 0.8\}$

Table 4.7: Comparison of the best OA variant, the best OA-GBD variants and CPLEX considering only system access and transfer congestion ($a_b = 0$).

n	a_c	a_t	ϑ	β=Tight						β=Loose			
				OA		OA-GBD		CPLEX		OA	OA-GBD	CPLEX	
				Time[s]	GAP	Time[s]	GAP	Time[s]	GAP	Time[s]	Times[s]	Times[s]	GAP
10	5	5	0	0.99	–	**0.21**	–	1.75	–	0.07	**0.06**	0.25	–
10	5	5000	0	1.14	–	**0.24**	–	2.14	–	**0.07**	**0.07**	0.25	–
10	5000	5	0	2.13	–	**1.86**	–	10.79	–	0.93	**0.67**	3.18	–
10	5000	5000	0	3.14	–	**3.05**	–	16.26	–	7.41	**6.94**	33.89	–
10	5	5	1	0.16	–	**0.07**	–	0.77	–	0.07	**0.04**	0.21	–
10	5	5000	1	0.17	–	**0.06**	–	0.85	–	0.07	**0.05**	0.21	–
10	5000	5	1	15.49	–	**8.26**	–	84.39	–	1.62	**0.85**	6.03	–
10	5000	5000	1	**14.94**	–	35.55	–	88.07	–	**30.55**	56.40	167.72	–
10	5	5	4	0.16	–	**0.06**	–	0.45	–	0.07	**0.05**	0.26	–
10	5	5000	4	0.15	–	**0.06**	–	0.45	–	0.07	**0.04**	0.26	–
10	5000	5	4	12.38	–	**2.29**	–	58.25	–	1.98	**1.27**	8.81	–
10	5000	5000	4	**27.48**	–	124.78	–	148.41	–	**34.65**	77.87	138.71	–
20	5	5	0	62.41	–	**31.30**	–	13247.36	–	9.29	**2.01**	2498.28	–
20	5	5000	0	**336.36**	–	1000.69	–	17667.50	–	10.95	**2.58**	2568.07	–
20	5000	5	0	1241.50	–	**482.12**	–	75667.96	–	260.55	**42.84**	22820.64	–
20	5000	5000	0	964.46	–	**505.68**	–	76449.56	0.13	183.00	**48.24**	11771.22	–
20	5	5	1	59.23	–	**28.37**	–	2707.48	–	**2.92**	4.21	509.74	–
20	5	5000	1	**345.54**	–	887.60	–	12771.24	–	**2.79**	4.32	515.19	–
20	5000	5	1	9217.82	–	**2399.71**	–	71675.03	0.63	989.41	**32.47**	37617.89	–
20	5000	5000	1	**19733.62**	–	64607.67	1.60	84600	4.61	5355.58	**2740.45**	56186.12	1.93
20	5	5	4	15.80	–	**5.59**	–	1446.38	–	**0.82**	4.03	218.48	–
20	5	5000	4	16.30	–	**5.61**	–	1420.47	–	**0.82**	4.08	219.10	–
20	5000	5	4	2231.88	–	**252.69**	–	23102.19	–	689.42	**31.27**	12462.15	–
20	5000	5000	4	**74797.24**	4.63	84600	8.24	84600	11.18	**2793.43**	4794.12	3291	–
		Average:		4545.85	0.19	6457.67	0.41	19406.99	0.69	432.36	327.29	7527.36	0.08

The data are aggregated considering three economies of scale factor $\alpha \in \{0.2, 0.5, 0.8\}$

Finally, a set of experiments was performed for solving instances with up to 25 nodes for power law and Kleinrock congestion functions. The OA variant was used in this version since your nonlinear SPs are trivial to solve and do not need the CPLEX solve which is not able to solve a problem with the Kleinronck nonlinear function. This set of experiments are presented on Figure 4.8-4.9. According these tables the algorithm is able to solve the most proposed instances of the problem, i.e., more than 90% of problem using power law based congestion function and 77% of problems using Kleinrock based congestion function with a lower optimality gap. When, comparing both congestion functions, the problem using power law function is on average easier to solve than by Kleinrock function.

Table 4.8: Performance of the best OA variant for Power law and Kleinrock function considering only system access and boarding congestion ($a_t = 0$).

n	a_c	a_b	Power law function			Kleinrock function		
			Time[s]	# Opt.	gap	Time[s]	# Opt.	gap
10	5	5	1.64	18	0.00	1.56	18	0.00
10	5	5000	1.56	18	0.00	2.35	18	0.00
10	5000	5	3.35	18	0.00	3.51	18	0.00
10	5000	5000	7.07	18	0.00	13.11	18	0.00
20	5	5	1010.08	18	0.00	13.27	18	0.00
20	5	5000	1081.37	18	0.00	4155.37	18	0.00
20	5000	5	7260.27	17	0.00	2382.18	17	0.00
20	5000	5000	14372.96	15	0.08	13187.87	14	0.42
25	5	5	12422.00	16	0.94	24739.09	11	2.53
25	5	5000	18086.53	15	0.18	47098.08	6	8.87
25	5000	5	28404.24	13	0.50	65003.35	11	15.88
25	5000	5000	36915.50	12	3.39	42225.11	6	9.28
	Aver./sum		9963.88	196	0.42	16568.74	173	3.08

4.5 Conclusion

This paper addresses a congested version of the incomplete hub location problem, where congestions in different service provided for hub-and-spoke system are explored. A nonlinear mixed integer formulation were proposed to model the problem. Furthermore, some exact algorithms based on outer approximation and generalized Benders decomposition were proposed. All of the algorithms are tested and the best algorithms are compared to the general purpose solver CPLEX presenting better performance than this one. Furthermore, the best OA exact algorithm, the OA-branch-and-cut approach, is able to solve instances with up to 25 nodes considering two kind of congestion cost functions.

Table 4.9: Performance of the best OA variant for Power law and Kleinrock function considering only system access and transfer congestion ($a_b = 0$).

n	a_c	a_t	Power law function			Kleinrock function		
			Time[s]	# Opt.	gap	Time[s]	# Opt.	gap
10	5	5	0.25	18	0.00	6.47	18	0.00
10	5	5000	0.28	18	0.00	12.63	18	0.00
10	5000	5	5.76	18	0.00	107.64	18	0.00
10	5000	5000	19.70	18	0.00	17.87	18	0.00
20	5	5	25.08	18	0.00	88.40	17	0.00
20	5	5000	118.79	18	0.00	8198.61	12	0.44
20	5000	5	2438.43	18	0.00	49415.40	15	1.82
20	5000	5000	17304.55	16	0.00	17133.15	15	1.50
25	5	5	13313.47	16	0.77	29923.11	12	1.80
25	5	5000	40681.28	12	2.16	37080.33	3	6.09
25	5000	5	28412.88	14	6.59	77911.70	11	18.90
25	5000	5000	59481.41	6	0.89	42189.39	5	4.79
	Aver./sum		13483.49	190	0.87	21840.39	162	2.95

As possible future research, generalized queueing system can be addressed to compute the congestion costs instead to assume Poisson process which results in Kleinrock delay function.

Chapter 5

Final remarks

This thesis addresses three hub-and-spoke networks design problems focuses on public transportation systems. The first problem, presented in Chapter 2, consists in designing a hub-and-spoke system in which hubs are connected by means of a line minimizing the total weighted travel time. The travel time associated with a given origin and destination pair are assumed to be shortest travel between the travel through a hub networks or a direct connection. This problem accounts the trade-off between the economies of scale to travel in an interhub link which and the time spent to access and leave a given hub line which can make the hub line more and less attractive, respectively. Analyses of the system configurations for differents economies of scale and access/exit times factor shows the impact of these parameters on the network configuration and the use of the hub lines network, where, as more economies of scale and as less access/exit times, higher is the usage of the hub network. Exact algorithms based on Benders decomposition method are proposed to solve the problem. Preliminary experiments shows that the best Benders variant to tackle the proposed instances of the problem is a multiple cuts strategy that one cut for each origin and destination pair; design a specialized algorithm to generate the Benders cuts; and adds Benders cuts inside the branch-and-bound tree. Experiments on benchmark instances show that the Benders variant presents better performance when compared with the solver CPLEX, where instances with up to 100 nodes are solved.

The second problem, presented in Chapter 3, is an extension of the previous one in which consists of locating a set of hub lines considering a budget constraint to install the infrastructure necessary. Exact algorithm based on the best Benders variant proposed for the HLLP are presented (Chapter 2). Computational results shows that the Benders variant and CPLEX present slightly the same performance for instance with up to 20 nodes, however for instances with 25 to 40 nodes the Benders variant can provide lower bound for all instances. For the other hand, CPLEX fails to find a

feasible solution and lower bound for the most of these instances. This lower bound can be used as important tool to evaluate the quality of feasible solutions for the problem. Since this problem is more complex than the variant with single line, three heuristics based on metaheuristics variable neighborhood descent (VND), a greedy randomized adaptive search procedure (GRASP) and an adaptive large neighborhood search (ALNS) is proposed to find near optimal feasible solution for instances with up 75 nodes. Using the Benders decomposition to provide lower bounds, experiments on benchmark instances show that the proposed metaheuristics can find good solutions for the instances. In particular, the GRASP and ALNS can find most best solutions.

The last problem addressed, presented in Chapter 4, consists in design a hub-and-spoke network taking into account the costs to install the necessary infrastructure (hubs and hub arcs), to transport all demand and associated with congestion in systems. One of the main contributions of this Chapter is to consider the congestion effects related to three differents services provided by the hub network: access, boarding and transference. The analyze of the system configurations shows the importance of considering the congestion associated to the three services. Assuming that the hub-and-spoke system can be modeled as a $M/M/1$ network queue, two convex nonlinear function are used to model the congestion costs: the Kleinrock function and an adapted power law function. The problem is modeled by mixed integer nonlinear formulation. Due the complexity of the problem, exacts algorithms based on Outer Approximation (OA) and Generalized Benders decomposition (GBD) are proposed. Computational results show that the addition of cut for each potential incumbent solution improves the convergence of OA and GBD based algorithms. Further, the addition of Pareto-optimal cuts improves the convergence of GBD based algorithms. The best performance algorithms of these preliminary experiments, a branch-and-cut based variant of OA and a method that hybridize the OA and GBD, are compared with the solver CPLEX. The results of computational experiments confirm the efficiency of the proposed algorithm that can solve the most instance with up to 25 nodes and present better performance than CPLEX.

This thesis presents three hub-and-spoke network design problems with high potential for practical application by considering several characteristics of real public transportation systems. Mathematical formulations were proposed to model these problems while exact algorithms, based on decomposition methods, and heuristics methods are proposed to solve them. Analysis of system configurations show how the main parameters impact on the optimal solution and how it is important to consider all of them when designing the network. Further, a set of computational experiments is performed to confirm the performance of proposed algorithms. To improve the convergence of the decomposition methods, different mechanisms were explored. Among the mechanisms

for improving the performance of these methods, the addition of cuts (Benders, OA or GBD) inside the branch-and-bound tree from potential incumbent solutions was very successful for all them. The advantage of this strategy is that these methods converge in a single iteration. Hence, the solution of only a single master problem is needed. Another effective mechanism for strengthening the decomposition methods is the selection of cuts to be added in the master problem. Since subproblems responsible for generating the cuts are, in general, degenerate, then different cuts can be generated. Investing in cut selection for adding stronger cuts results in better convergence. For the Benders decomposition applied to solve the HLLP, an algorithm for selection of cuts is proposed. Further, algorithms based on GBD solve additional linear subproblems to generate non-dominated cuts, i.e. Pareto-optimal cuts.

Possible future research associated to locate hub line network is to model the behavior of users to choose possible routes, instead of assuming that the user always chooses the smallest route among all the available one. Future research related to the problem considering congestion effects is modeling the hub-and-spoke system as a generalized queueing network. It can be done, by considering each service as a queueing system whose distribution of service time and intervals between arrivals are generic. The queueing network can then used to compute congestion costs.

Appendix A

Proof of Proposition 2

Proof. Proof. The subsystem of inequalities (2.47)-(2.50) can be rewritten as:

$$\begin{aligned}
\widehat{\beta}_{r_1} &\leq UB_1 = w_{ij}\Phi_1^i - \theta_{ij} && \text{(A.1)}\\
\widehat{\beta}_{r_1} &\geq LB_1 = \Gamma_{ij} - w_{ij}\Phi_1^j && \text{(A.2)}\\
\widehat{\beta}_{r_1} &\geq \widehat{\beta}_{r_2} - w_{ij}\alpha_{r_1r_2}t_{r_1r_2} && \text{(A.3)}\\
\widehat{\beta}_{r_1} &\leq w_{ij}\alpha_{r_2r_1}t_{r_2r_1} + \widehat{\beta}_{r_2} && \text{(A.4)}\\
\widehat{\beta}_{r_2} &\leq w_{ij}(t_{ir_2} + \tilde{t}^a_{r_2}) - \theta_{ij} && \text{(A.5)}\\
\widehat{\beta}_{r_2} &\geq \Gamma_{ij} - w_{ij}(t_{r_2j} + \tilde{t}^e_{r_2}) && \text{(A.6)}\\
\widehat{\beta}_{r_2} &\geq \widehat{\beta}_{r_3} - w_{ij}\alpha_{r_2r_3}t_{r_2r_3} && \text{(A.7)}\\
\widehat{\beta}_{r_2} &\leq w_{ij}\alpha_{r_3r_2}t_{r_3r_2} + \widehat{\beta}_{r_3} && \text{(A.8)}\\
&\vdots \qquad\qquad \vdots && \text{(A.9)}\\
\widehat{\beta}_{r_p} &\leq w_{ij}(t_{ir_p} + \tilde{t}^a_{r_p}) - \theta_{ij} && \text{(A.10)}\\
\widehat{\beta}_{r_p} &\geq \Gamma_{ij} - w_{ij}(t_{r_pj} + \tilde{t}^e_{r_p}). && \text{(A.11)}
\end{aligned}$$

This subsystem of inequalities can be solved by a procedure based on Fourier-Motzkin elimination (see, Martin, 1999, for details). Since the variable $\widehat{\beta}_{r_1}$ appears only in the first four inequalities, the constraints related to this variable are equivalent to:

$$\max\{LB_1, -w_{ij}\alpha_{r_1r_2}t_{r_1r_2} + \widehat{\beta}_{r_2}\} \leq \widehat{\beta}_{r_1} \leq \min\{UB_1, w_{ij}\alpha_{r_2r_1}t_{r_2r_1} + \widehat{\beta}_{r_2}\}. \qquad \text{(A.12)}$$

Using the above equation, the variable $\widehat{\beta}_{r_1}$ can be eliminated from the original system giving place to the following constraints:

$$\begin{aligned}
\widehat{\beta}_{r_2} &\leq w_{ij}\alpha_{r_1r_2}t_{r_1r_2} + UB_1 = w_{ij}\alpha_{r_1r_2}t_{r_1r_2} + w_{ij}\Phi_1^i - \theta_{ij}\\
\widehat{\beta}_{r_2} &\geq LB_1 - w_{ij}\alpha_{r_2r_1}t_{r_2r_1} = \Gamma_{ij} - w_{ij}\Phi_1^j - w_{ij}\alpha_{r_2r_1}t_{r_2r_1},
\end{aligned}$$

where the other constraints that can be derived from Equation (A.12), i.e., $LB_1 \leq UB_1$ and $-\alpha w_{ij} t_{r_1 r_2} + \widehat{\beta}_{r_2} \leq \alpha w_{ij} t_{r_2 r_1} + \widehat{\beta}_{r_2}$ are redundant.

We can observe that the new constraints may only affect the UB and the LB of the variable $\widehat{\beta}_{r_2}$. Hence, the original bound constraints (A.5) and (A.6) can be substituted by the following bound constraints:

$$\begin{aligned}
\widehat{\beta}_{r_2} \leq & \min\{w_{ij}(t_{ir_2} + \tilde{t}^a_{r_2}) - \theta_{ij}, w_{ij}\alpha_{r_1 r_2} t_{r_1 r_2} + w_{ij}\Phi^i_1 - \theta_{ij}\} \\
= & w_{ij} \min\{(t_{ir_2} + \tilde{t}^a_{r_2}), \alpha_{r_1 r_2} t_{r_1 r_2} + \Phi^i_1\} - \theta_{ij} = w_{ij}\Phi^i_2 - \theta_{ij}. \\
\widehat{\beta}_{r_2} \geq & \max\{\Gamma_{ij} - w_{ij}(t_{r_2 j} + \tilde{t}^e_{r_2}), \Gamma_{ij} - w_{ij}(\alpha_{r_2 r_1} t_{r_2 r_1} + \Phi^j_1)\} \\
= & \Gamma_{ij} + w_{ij} \max\{-(t_{r_2 j} + \tilde{t}^e_{r_2}), -(\alpha_{r_2 r_1} t_{r_2 r_1} + \Phi^j_1)\} \\
= & \Gamma_{ij} - w_{ij} \min\{(t_{r_2 j} + \tilde{t}^e_{r_2}), (\alpha_{r_2 r_1} t_{r_2 r_1} + \Phi^j_1)\} = \Gamma_{ij} - w_{ij}\Phi^j_2
\end{aligned}$$

The value of the other variables in the original system can be found by solving the following subsystem:

$$\begin{aligned}
\widehat{\beta}_{r_2} \leq & \quad UB_2 = w_{ij}\Phi^i_2 - \theta_{ij} \\
\widehat{\beta}_{r_2} \geq & \quad LB_2 = \Gamma_{ij} - w_{ij}\Phi^j_2 \\
\widehat{\beta}_{r_2} \geq & \quad \widehat{\beta}_{r_3} - w_{ij}\alpha_{r_2 r_3} t_{r_2 r_3} \\
\widehat{\beta}_{r_2} \leq & \quad w_{ij}\alpha_{r_3 r_2} t_{r_3 r_2} + \widehat{\beta}_{r_3} \\
\vdots & \quad \vdots \\
\widehat{\beta}_{r_p} \leq & \quad w_{ij}(t_{ir_p} + \tilde{t}^a_{r_p}) - \theta_{ij} \\
\widehat{\beta}_{r_p} \leq & \quad \Gamma_{ij} - w_{ij}(t_{r_p j} + \tilde{t}^e_{r_p}).
\end{aligned}$$

Comparing the new subsystem with the previous one, it is possible to see that the $\widehat{\beta}_{r_2}$ variable can be eliminated in the same way that the variable $\widehat{\beta}_{r_1}$ was eliminated. Therefore, all variables $\widehat{\beta}_{r_l}$, for $l \in 1 \ldots (p-1)$, can be eliminated iteratively by means of the following set of inequalities:

$$\max\{LB_l, -w_{ij}\alpha_{r_l r_{l+1}} t_{r_l r_{l+1}} + \widehat{\beta}_{r_{l+1}}\} \leq \widehat{\beta}_{r_l} \leq \min\{UB_l, w_{ij}\alpha_{r_{l+1} r_l} t_{r_{l+1} r_l} + \widehat{\beta}_{r_{l+1}}\},\ \forall 1 \leq l < p. \tag{A.13}$$

This results in a new upper bound UB_{l+1} and lower bound LB_{l+1} for the variables $\widehat{\beta}_{r_{l+1}}$ given by:

$$UB_{l+1} = w_{ij}\Phi^i_{l+1} - \theta = \min\{w_{ij}(t_{ir_{l+1}} + \tilde{t}^a_{r_{l+1}}), w_{ij}\alpha_{r_l r_{l+1}} t_{r_l r_{l+1}} + w_{ij}\Phi^i_l\} - \theta$$

and

$$LB_{l+1} = \Gamma_{ij} - w_{ij}\Phi^j_{l+1} = \Gamma_{ij} - \min\{w_{ij}(t_{r_{l+1}j} + \tilde{t}^e_{r_{l+1}}), w_{ij}\alpha_{r_l r_{l+1}} t_{r_l r_{l+1}} + w_{ij}\Phi^j_l\}.$$

In the case l equal to $p-1$, the feasibility interval to $\widehat{\beta}_{r_p}$ is given by:

$$\Gamma_{ij} - w_{ij}\Phi^j_p = LB_p \leq \widehat{\beta}_{r_p} \leq UB_p = w_{ij}\Phi^i_p - \theta.$$

□

Appendix B

Proof that Proposition 2 can be used to find $\widehat{\beta}_k$ for $k \in H^1 \setminus H^1_{ij}$

Let Φ^{1i}_s and Φ^{1j}_s be the shortest path from i to s and from s to j, respectively, using only hubs in the line Seg. 1. In the same way, define Φ^{3i}_q and Φ^{3j}_q as the shortest path from i to q and from q to j, respectively, using only hubs in the line Seg. 3. Proposition 3 shows that the values of $\widehat{\beta}_q$ and $\widehat{\beta}_s$ are in their feasibility interval.

Proposition 3. *Let* $\widehat{\beta}_s = w_{ij}(t_{ir_s} + \tilde{t}^a_{r_s})$ *and* $\widehat{\beta}_q = \sum_{l=s+1}^{q} w_{ij}\alpha_{r_{l-1}r_l}t_{r_{l-1}r_l} + w_{ij}(t_{ir_s} + \tilde{t}^a_{r_s})$ *where* s *and* q *are the first hub and last hub, respectively, on the path from* i *to* j*, then*

$$\Gamma_{ij} - \Phi^{1j}_s \leq \widehat{\beta}_s \leq \Phi^{1i}_s - \theta_{ij}$$

and

$$\Gamma_{ij} - \Phi^{3j}_q \leq \widehat{\beta}_q \leq \Phi^{3i}_q - \theta_{ij}.$$

Proof. Proof.

Let Φ^{i*}_s and Φ^{i*}_q be the shortest path on the line from i to s and from i to q, respectively. Therefore, $\Phi^{i*}_s = w_{ij}(t_{ir_s} + f) = \widehat{\beta}_s$ and $\Phi^{i*}_q = \sum_{l=s+1}^{q} w_{ij}\alpha_{r_{l-1}r_l}t_{r_{l-1}r_l} + w_{ij}(t_{ir_s} + f) = \widehat{\beta}_q$.

Since $\theta_{ij} = 0$ and the shortest path from i to s and from i to q in direction of j does not use hubs that are on line Seg. 1 or line Seg. 3, then

$$\widehat{\beta}_s = \Phi^{i*}_s \leq \Phi^{1i}_s = \Phi^{1i}_s - \theta_{ij}, \text{ and}$$

$$\widehat{\beta}_q = \Phi^{i*}_q \leq \Phi^{3i}_q = \Phi^{3i}_q - \theta_{ij}.$$

However, since Γ_{ij} is the value of the shortest path from i to j, this path is always

shorter than a path where j is connected to a hub on line Seg. 1 or Seg. 3:

$$\Gamma_{ij} \leq \Phi^{i*}_s + \Phi^{1j}_s = \widehat{\beta}_s + \Phi^{1j}_s, \text{ and}$$

$$\Gamma_{ij} \leq \Phi^{i*}_q + \Phi^{3j}_q = \widehat{\beta}_q + \Phi^{1j}_q.$$

□

Bibliography

Adulyasak, Y., Cordeau, J.-F., and Jans, R. (2012). Benders decomposition for production routing under demand uncertainty. GERAD Tech Rep. G-2012-57, HEC Montréal, Canada.

Alumur, S. and Kara, B. Y. (2008). Network hub location problems: The state of the art. *European Journal of Operational Research*, 190(1):1 – 21.

Alumur, S. A., Kara, B. Y., and Karasan, O. E. (2009). The design of single allocation incomplete hub networks. *Transportation Research Part B: Methodological*, 43(10):936–951.

Applegate, D., Bixby, R., Chvátal, V., and Cook, W. (2012). Concorde TSP solver.

Aykin, T. (1994). Lagrangian relaxation based approaches to capacitated hub-and-spoke network design problem. *European Journal of Operational Research*, 79:501–523.

Benders, J. F. (1962). Partitioning procedures for solving mixed-variables programming problems. *Numerische Mathematik*, 4(1):238–252.

Birge, J. R. and Louveaux, F. V. (1988). A multicut algorithm for two-stage stochastic linear programs. *European Journal of Operational Research*, 34(3):384–392.

Bruno, G., Ghiani, G., and Improta, G. (1998). A multi-modal approach to the location of a rapid transit line. *European Journal of Operational Research*, 104(2):321–332.

Calık, H., Alumur, S. A., Kara, B. Y., and Karasan, O. E. (2009). A tabu-search based heuristic for the hub covering problem over incomplete hub networks. *Computers & Operations Research*, 36(12):3088–3096.

Camargo, R. S. d., de Miranda Jr., G., and Ferreira, R. P. (2011). A hybrid outer-approximation/Benders decomposition algorithm for the single allocation hub location problem under congestion. *Operations Research Letters*, 39(5):329 – 337.

Camargo, R. S. d., Miranda, G., and Luna, H. (2008). Benders decomposition for the uncapacitated multiple allocation hub location problem. *Computers & Operations Research*, 35(4):1047–1064.

Camargo, R. S. d. and Miranda, G. J. (2012). Single allocation hub location problem under congestion: Network owner and user perspectives. *Expert Systems with Applications*, 39(3):3385–3391.

Camargo, R. S. d., Miranda Jr, G., Ferreira, R., and Luna, H. P. (2009). Multiple allocation hub-and-spoke network design under hub congestion. *Computers & Operations Research*, 36:3097–3106.

Camargo, R. S. d. and Miranda Jr, G. d. (2012). Addressing congestion on single allocation hub-and-spoke networks. *Pesquisa Operacional*, 32(3):465–496.

Campbell, J. F. (1992). Location and allocation for distribution systems with transshipments and transportion economies of scale. *Annals of Operations Research*, 40(1):77–99.

Campbell, J. F. (1994). Integer programming formulations of discrete hub location problems. *European Journal of Operational Research*, 72(2):387–405.

Campbell, J. F., Ernst, A. T., and Krishnamoorthy, M. (2002). Hub location problems. In Drezner, Z. and Hamacher, H. W., editors, *Facility Location: Applications and Theory*, chapter 12, pages 373–407. Springer, 1^a edition.

Campbell, J. F., Ernst, A. T., and Krishnamoorthy, M. (2005a). Hub arc location problems: Part I - Introduction on results. *Management Science*, 51(10):1540–1555.

Campbell, J. F., Ernst, A. T., and Krishnamoorthy, M. (2005b). Hub arc location problems: Part II - Formulations and optimal algorithms. *Management Science*, 51(10):1556–1571.

Campbell, J. F. and O'Kelly, M. E. (2012). Twenty-five years of hub location research. *Transportation Science*, 46(2):153–169.

Church, R. L. and ReVelle, C. S. (1976). Theoretical and computational links between the p-median, location set-covering, and the maximal covering location problem. *Geographical Analysis*, 8(4):406–415.

Codato, G. and Fischetti, M. (2006). Combinatorial Benders' cuts for mixed-integer linear programming. *Operations Research*, 54:756–766.

Contreras, I. and Fernández, E. (2012). General network design: A unified view of combined location and network design problems. *European Journal of Operational Research*, 219(3):680–697.

Contreras, I. and Fernández, E. (2013). Hub location as the minimization of a supermodular set function. *Submitted for publication.*

Contreras, I. and Fernández, E. (2014). Hub location as the minimization of a supermodular set function. *Operations Research*, 62:557–570.

Contreras, I., Fernández, E., and Marín, A. (2009). Tight bounds from a path based formulation for the tree of hub location problem. *Computers & Operations Research*, 36(12):3117–3127.

Contreras, I., Fernández, E., and Marín, A. (2010). The tree of hubs location problem. *European Journal of Operational Research*, 202(2):390–400.

Contreras, I., Tanash, M., and Vidyarthi, N. (2013a). The cycle hub location problem. Technical Report 2013-59, CIRRELT, University of Montreal.

Contreras, I., Tanash, M., and Vidyarthi, N. (2013b). The cycle hub location problem. Tech Rep.CIRRELT-2013-59, Concordia University, Canada.

Current, J. R., Revelle, C. S., and Cohon, J. L. (1987). The median shortest path problem: A multiobjective approach to analyze cost vs. accessibility in the design of transportation networks. *Transportation Science*, 21(3):188–197.

Dufourd, H., Gendreau, M., and Laporte, G. (1996). Locating a transit line using tabu search. *Location Science*, 4(1):1–19.

Duran, M. and Grossmann, I. E. (1986). An outer-approximation algorithm for a class of mixed integer nonlinear programms. *Mathematical Programming*, 36:307–339.

Elhedhli, S. and Hu, F. X. (2005). Hub-and-spoke network design with congestion. *Computers & Operations Research.* To appear.

Elhedhli, S. and Wu, H. (2010). A lagrangean heuristic for hub-and-spoke system design with capacity selection and congestion. *INFORMS Journal on Computing*, 22(2):282–296.

Ernst, A. T. and Krishnamoorthy, M. (1996). Efficient algorithms for the uncapacitated single allocation p-hub median problem. *Location Science*, 4(3):139–154.

Ernst, A. T. and Krishnamoorthy, M. (1999). Solution algorithms for the capacitated single allocation hub location problem. *Annals of Operations Research*, 86:141–159.

Farahani, R. Z., Hekmatfar, M., Arabani, A. B., and Nikbakhsh, E. (2013). Hub location problems: A review of models, classification, solution techniques, and applications. *Comput. Ind. Eng.*, 64(4):1096–1109.

Feo, T. A. and Resende, M. G. (1989). A probabilistic heuristic for a computationally difficult set covering problem. *Operations Research Letters*, 8(2):67–71.

Fischetti, M. and Monaci, M. (2014). Exploiting erraticism in search. *Operations Research*, 62(1):114–122.

Fletcher, R. and Leyffer, S. (1994). Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming*, 66(1-3):327–349.

Fletcher, R. and Leyffer, S. (1996). Solving mixed integer nonlinear programs by outer approximation. *Mathematical Programming*, 66:327–349.

Fortz, B. and Poss, M. (2009). An improved Benders decomposition applied to a multi-layer network design problem. *Operations Research Letters*, 37(5):359 – 364.

García, R., Garzón-Astolfi, A., Marín, A., Mesa, J. A., and Ortega, F. A. (2006). Analysis of the Parameters of Transfers in Rapid Transit Network Design. In Kroon, L. G. and Möhring, R. H., editors, *5th Workshop on Algorithmic Methods and Models for Optimization of Railways (ATMOS'05)*, volume 2, pages 1–15, Dagstuhl, Germany.

Gelareh, S. and Nickel, S. (2011). Hub location problems in transportation networks. *Transportation Research Part E: Logistics and Transportation Review*, 47(6):1092–1111.

Gendreau, M., Laporte, G., and Mesa, J. A. (1995). Locating rapid transit lines. *Journal of Advanced Transportation*, 29(2):145–162.

Geoffrion, A. (1972). Generalized Benders decomposition. *Journal of optimization Theory and Applications*, 10(4):237–260.

Geoffrion, A. M. and Graves, G. W. (1974). Multicommodity distribution system design by Benders decomposition. *Management Ccience*, 20(5):822–844.

Gillen, D. and Levinson, D. M. (1999). Full cost of air travel in the california corridor. *Presented in the 78th Annual meeting of Transportation Research Board*, pages 10–14.

Grossmann, I. E. (2002). Review of nonlinear mixed-integer and disjunctive programming techniques. *Optimization and Engineering*, 3(3):227–252.

Guihaire, V. and Hao, J.-K. (2008). Transit network design and scheduling: A global review. *Transportation Research Part A: Policy and Practice*, 42(10):1251–1273.

Guldmann, J.-M. and Shen, G. (1997). A general mixed integer nonlinear optimization model for hub network design. In *44th North American meeting of the Regional Science Association International*.

Hakimi, S. L., Schmeichel, E. F., and Labbé, M. (1993). On locating path- or tree-shaped facilities on networks. *Networks*, 23(6):543–555.

Hansen, P. and Mladenović, N. (2001). Variable neighborhood search: Principles and applications. *European journal of operational research*, 130(3):449–467.

Kleinrock, L. (1964). *Communication nets; stochastic message flow and delay*. Dover Publications, Incorporated.

Labbé, M., Laporte, G., Martín, I. R., and González, J. J. S. (2004). The ring star problem: Polyhedral analysis and exact algorithm. *Networks*, 43(3):177–189.

Labbé, M., Laporte, G., and Rodríguez-Martín, I. (1998). Path, tree and cycle location. In T.G., C. and G., L., editors, *Fleet management and logistics*, pages 187–204. Kluwer Academic Publisher, Boston.

Labbé, M. and Yaman, H. (2008). Solving the hub location problem in a star-star network. *Networks*, 51(1):19–33.

Labbé, M., Yaman, H., and Gourdin, E. (2005). A branch and cut algorithm for hub location problems with single assignment. *Mathematical Programming*, 102(2):371–405.

Laporte, G., Marín, Á., Mesa, J. A., and Ortega, F. A. (2007). An integrated methodology for the rapid transit network design problem. In *Algorithmic methods for railway optimization*, pages 187–199. Springer.

Laporte, G., Mesa, J. A., and Ortega, F. A. (2002). Locating stations on rapid transit lines. *Computers & Operations Research*, 29(6):741–759.

Lari, I., Ricca, F., and Scozzari, A. (2008). Comparing different metaheuristic approaches for the median path problem with bounded length. *European Journal of Operational Research*, 190(3):587–597.

Lee, C.-H., Ro, H.-B., and Tcha, D.-W. (1993). Topological design of a two-level network with ring-star configuration. *Computers & Operations Research*, 20(6):625 – 637.

Magnanti, T. L. and Wong, R. T. (1981). Accelerating Benders decomposition: Algorithmic enhancement and model selection criteria. *Operations Research*, 29(3):464–483.

Marianov, V. and Serra, D. (2003). Location models for airline hubs behaving as m/d/c queues. *Computers & Operations Research*, 30(7):983–1003.

Marín, Á. (2007). An extension to rapid transit network design problem. *Top*, 15(2):231–241.

Marín, Á. and Jaramillo, P. (2008). Urban rapid transit network capacity expansion. *European Journal of Operational Research*, 191(1):45–60.

Marín, A. and Jaramillo, P. (2009). Urban rapid transit network design: accelerated Benders decomposition. *Annals of Operations Research*, 169(1):35–53.

Martin, R. K. (1999). *Large Scale Linear and Integer Optimization - A Unified Approach*. Kluwer Academic, Boston.

Martins de Sá, E., Contreras, I., Cordeau, J.-F., de Camargo, R. S., and de Miranda, G. (2013a). Hub line location problem. *Submitted for publication*.

Martins de Sá, E., de Camargo, R. S., and de Miranda, G. (2013b). An improved Benders decomposition algorithm for the tree of hubs location problem. *European Journal of Operational Research*, 226(2):185 – 202.

McDaniel, D. and Devine, M. (1977). A modified Benders partitioning algorithm for mixed integer programming. *Management Science*, 24(3):312–319.

Mesa, J. A. and Brian Boffey, T. (1996). A review of extensive facility location in networks. *European Journal of Operational Research*, 95(3):592–603.

Miranda Junior, G. d., Camargo, R. S. d., Pinto, L. R., Conceição, S. V., and Ferreira, R. P. M. (2011). Hub location under hub congestion and demand uncertainty: the Brazilian case study. *Pesquisa Operacional*, 31:319 – 349.

Naoum-Sawaya, J. and Elhedhli, S. (2013). An interior-point Benders based branch-and-cut algorithm for mixed integer programs. *Annals of Operations Research*, 210(1):33–55.

Nickel, S., Schobel, A., and Sonneborn, T. (2001). Hub location problems in urban traffic networks. In Niittymahi and Pursula, editors, *Mathematical Methods and Optimisation in Transportation Systems*, pages 1–12. Kluwer Academic Publisher, Dordrecht.

O'Kelly, M. E. (1986). The location of interacting hub facilities. *Transportation Science*, 20(2):92–106.

O'Kelly, M. E. (1987). A quadratic integer program for the location of interacting hub facilities. *European Journal of Operational Research*, 32(3):393–404.

O'Kelly, M. E. (1998). A geographer's analysis of hub-and-spoke networks. *Journal of Transport Geography*, 3(6):171–186.

O'Kelly, M. E. and Miller, H. J. (1994). The hub network design problem: A review and synthesis. *Journal of Transport Geography*, 2(1):31–40.

Papadakos, N. (2008). Practical enhancements to the magnanti–wong method. *Operations Research Letters*, 36(4):444–449.

Prais, M. and Ribeiro, C. C. (1999). Parameter variation in grasp implementations. In *Extended abstracts of the third metaheuristics international conference*, pages 375–380.

Rei, W., Cordeau, J.-F., Gendreau, M., and Soriano, P. (2009). Accelerating Benders decomposition by local branching. *INFORMS Journal on Computing*, 21(2):333–345.

Ropke, S. and Pisinger, D. (2006). An adaptive large neighborhood search heuristic for the pickup and delivery problem with time windows. *Transportation science*, 40(4):455–472.

Schöbel, A. (2012). Line planning in public transportation: models and methods. *OR spectrum*, 34(3):491–510.

Slater, P. J. (1982). Locating central paths in a graph. *Transportation Science*, 16(1):1–18.

Yaman, H. (2008). Star p-hub median problem with modular arc capacities. *Computers & Operations Research*, 35(9):3009–3019.

Yaman, H. (2009). The hierarchical hub median problem with single assignment. *Transportation Research Part B: Methodological*, 43(6):643–658.